

Manual | EN

TX1200

TwinCAT 2 | PLC Library: TcLON



Table of contents

1 Foreword	15
1.1 Notes on the documentation	15
1.2 For your safety	16
1.3 Notes on information security.....	17
2 Introduction	18
3 Target groups	19
4 LON	20
4.1 Supported SNVTs	20
4.2 How the KL6401 functions	25
5 Configuration with KS2000	27
6 Integration into TwinCAT	30
6.1 KL6401 - Linking to the TwinCAT System Manager	30
6.2 Integration in TwinCAT (CX9020)	35
6.3 Integration into TwinCAT (BC9191)	37
7 Programming	41
7.1 General information.....	56
7.2 POU.....	57
7.2.1 FB_LON_KL6401.....	66
7.2.2 FB_READ_001_SNVT_amp.....	67
7.2.3 FB_READ_002_SNVT_amp_mil	68
7.2.4 FB_READ_003_SNVT_angle	69
7.2.5 FB_READ_004_SNVT_angle_vel.....	70
7.2.6 FB_READ_005_SNVT_btu_kilo.....	71
7.2.7 FB_READ_006_SNVT_btu_mega	71
7.2.8 FB_READ_007_SNVT_char_ascii.....	72
7.2.9 FB_READ_008_SNVT_count	73
7.2.10 FB_READ_009_SNVT_count_inc.....	74
7.2.11 FB_READ_011_SNVT_date_day	74
7.2.12 FB_READ_013_SNVT_elec_kwh.....	75
7.2.13 FB_READ_014_SNVT_elec_whr.....	76
7.2.14 FB_READ_015_SNVT_flow.....	77
7.2.15 FB_READ_016_SNVT_flow_mil.....	78
7.2.16 FB_READ_017_SNVT_length	78
7.2.17 FB_READ_018_SNVT_length_kilo.....	79
7.2.18 FB_READ_019_SNVT_length_micr	80
7.2.19 FB_READ_020_SNVT_length_mil.....	81
7.2.20 FB_READ_021_SNVT_lev_cont.....	81
7.2.21 FB_READ_023_SNVT_mass	82
7.2.22 FB_READ_024_SNVT_mass_kilo.....	83
7.2.23 FB_READ_025_SNVT_mass_mega	84
7.2.24 FB_READ_026_SNVT_mass_mil.....	84
7.2.25 FB_READ_027_SNVT_power	85
7.2.26 FB_READ_028_SNVT_power_kilo.....	86

7.2.27	FB_READ_029_SNVT_ppm.....	87
7.2.28	FB_READ_030_SNVT_press.....	87
7.2.29	FB_READ_031_SNVT_res.....	88
7.2.30	FB_READ_032_SNVT_res_kilo.....	89
7.2.31	FB_READ_033_SNVT_sound_db.....	90
7.2.32	FB_READ_034_SNVT_speed.....	90
7.2.33	FB_READ_035_SNVT_speed_mil.....	91
7.2.34	FB_READ_036_SNVT_str_asc.....	92
7.2.35	FB_READ_037_SNVT_str_int.....	93
7.2.36	FB_READ_038_SNVT_telcom.....	93
7.2.37	FB_READ_039_SNVT_temp.....	94
7.2.38	FB_READ_041_SNVT_vol.....	95
7.2.39	FB_READ_042_SNVT_vol_kilo.....	96
7.2.40	FB_READ_043_SNVT_vol_mil.....	97
7.2.41	FB_READ_044_SNVT_volt.....	97
7.2.42	FB_READ_045_SNVT_volt_dbmv.....	98
7.2.43	FB_READ_046_SNVT_volt_kilo.....	99
7.2.44	FB_READ_047_SNVT_volt_mil.....	100
7.2.45	FB_READ_048_SNVT_amp_f.....	100
7.2.46	FB_READ_049_SNVT_angle_f.....	101
7.2.47	FB_READ_050_SNVT_angle_vel_f.....	102
7.2.48	FB_READ_051_SNVT_count_f.....	103
7.2.49	FB_READ_052_SNVT_count_inc_f.....	103
7.2.50	FB_READ_053_SNVT_flow_f.....	104
7.2.51	FB_READ_054_SNVT_length_f.....	105
7.2.52	FB_READ_055_SNVT_lev_cont_f.....	106
7.2.53	FB_READ_056_SNVT_mass_f.....	106
7.2.54	FB_READ_057_SNVT_power_f.....	107
7.2.55	FB_READ_058_SNVT_ppm_f.....	108
7.2.56	FB_READ_059_SNVT_press_f.....	109
7.2.57	FB_READ_060_SNVT_res_f.....	109
7.2.58	FB_READ_061_SNVT_sound_db_f.....	110
7.2.59	FB_READ_062_SNVT_speed_f.....	111
7.2.60	FB_READ_063_SNVT_temp_f.....	112
7.2.61	FB_READ_064_SNVT_time_f.....	112
7.2.62	FB_READ_065_SNVT_vol_f.....	113
7.2.63	FB_READ_066_SNVT_volt_f.....	114
7.2.64	FB_READ_067_SNVT_btu_f.....	115
7.2.65	FB_READ_068_SNVT_elec_whr_f.....	115
7.2.66	FB_READ_069_SNVT_config_src.....	116
7.2.67	FB_READ_070_SNVT_color.....	117
7.2.68	FB_READ_071_SNVT_grammage.....	118
7.2.69	FB_READ_072_SNVT_grammage_f.....	118
7.2.70	FB_READ_073_SNVT_file_req.....	119
7.2.71	FB_READ_074_SNVT_file_status.....	120
7.2.72	FB_READ_075_SNVT_freq_f.....	121

7.2.73	FB_READ_076_SNVT_freq_hz.....	121
7.2.74	FB_READ_077_SNVT_freq_kilohz.....	122
7.2.75	FB_READ_078_SNVT_freq_milhz.....	123
7.2.76	FB_READ_079_SNVT_lux.....	124
7.2.77	FB_READ_081_SNVT_lev_percent.....	124
7.2.78	FB_READ_082_SNVT_multiplier.....	125
7.2.79	FB_READ_083_SNVT_state.....	126
7.2.80	FB_READ_084_SNVT_time_stamp.....	127
7.2.81	FB_READ_085_SNVT_zerospan.....	128
7.2.82	FB_READ_086_SNVT_magcard.....	128
7.2.83	FB_READ_087_SNVT_elapsed_tm.....	129
7.2.84	FB_READ_088_SNVT_alarm.....	130
7.2.85	FB_READ_089_SNVT_currency.....	131
7.2.86	FB_READ_090_SNVT_file_pos.....	132
7.2.87	FB_READ_091_SNVT_muldiv.....	132
7.2.88	FB_READ_092_SNVT_obj_request.....	133
7.2.89	FB_READ_093_SNVT_obj_status.....	134
7.2.90	FB_READ_094_SNVT_preset.....	135
7.2.91	FB_READ_095_SNVT_switch.....	136
7.2.92	FB_READ_096_SNVT_trans_table.....	136
7.2.93	FB_READ_097_SNVT_override.....	137
7.2.94	FB_READ_098_SNVT_pwr_fact.....	138
7.2.95	FB_READ_099_SNVT_pwr_fact_f.....	139
7.2.96	FB_READ_100_SNVT_density.....	140
7.2.97	FB_READ_101_SNVT_density_f.....	140
7.2.98	FB_READ_102_SNVT_rpm.....	141
7.2.99	FB_READ_103_SNVT_hvac_emerg.....	142
7.2.100	FB_READ_104_SNVT_angle_deg.....	143
7.2.101	FB_READ_105_SNVT_temp_p.....	143
7.2.102	FB_READ_106_SNVT_temp_setpt.....	144
7.2.103	FB_READ_107_SNVT_time_sec.....	145
7.2.104	FB_READ_108_SNVT_hvac_mode.....	146
7.2.105	FB_READ_109_SNVT_occupancy.....	146
7.2.106	FB_READ_110_SNVT_area.....	147
7.2.107	FB_READ_111_SNVT_hvac_overid.....	148
7.2.108	FB_READ_112_SNVT_hvac_status.....	149
7.2.109	FB_READ_113_SNVT_press_p.....	150
7.2.110	FB_READ_114_SNVT_address.....	150
7.2.111	FB_READ_115_SNVT_scene.....	151
7.2.112	FB_READ_116_SNVT_scene_cfg.....	152
7.2.113	FB_READ_117_SNVT_setting.....	153
7.2.114	FB_READ_118_SNVT_evap_state.....	153
7.2.115	FB_READ_119_SNVT_therm_mode.....	154
7.2.116	FB_READ_120_SNVT_defr_mode.....	155
7.2.117	FB_READ_121_SNVT_defr_term.....	156
7.2.118	FB_READ_122_SNVT_defr_state.....	157

7.2.119	FB_READ_123_SNVT_time_min	157
7.2.120	FB_READ_124_SNVT_time_hour	158
7.2.121	FB_READ_125_SNVT_ph	159
7.2.122	FB_READ_126_SNVT_ph_f	160
7.2.123	FB_READ_127_SNVT_chlr_status	161
7.2.124	FB_READ_128_SNVT_tod_event	161
7.2.125	FB_READ_129_SNVT_smo_obscur	162
7.2.126	FB_READ_130_SNVT_fire_test	163
7.2.127	FB_READ_131_SNVT_temp_ror	164
7.2.128	FB_READ_132_SNVT_fire_init	165
7.2.129	FB_READ_133_SNVT_fire_indcte	165
7.2.130	FB_READ_134_SNVT_time_zone	166
7.2.131	FB_READ_135_SNVT_earth_pos	167
7.2.132	FB_READ_136_SNVT_reg_val	168
7.2.133	FB_READ_137_SNVT_reg_val_ts	169
7.2.134	FB_READ_138_SNVT_volt_ac	169
7.2.135	FB_READ_139_SNVT_amp_ac	170
7.2.136	FB_READ_143_SNVT_turbidity	171
7.2.137	FB_READ_144_SNVT_turbidity_f	172
7.2.138	FB_READ_145_SNVT_hvac_type	172
7.2.139	FB_READ_146_SNVT_elec_kwh_l	173
7.2.140	FB_READ_147_SNVT_temp_diff_p	174
7.2.141	FB_READ_148_SNVT_ctrl_req	175
7.2.142	FB_READ_149_SNVT_ctrl_resp	176
7.2.143	FB_READ_150_SNVT_ptz	176
7.2.144	FB_READ_151_SNVT_privacyzone	177
7.2.145	FB_READ_152_SNVT_pos_ctrl	178
7.2.146	FB_READ_153_SNVT_enthalpy	179
7.2.147	FB_READ_154_SNVT_gfci_status	180
7.2.148	FB_READ_155_SNVT_motor_state	180
7.2.149	FB_READ_156_SNVT_pumpset_mn	181
7.2.150	FB_READ_157_SNVT_ex_control	182
7.2.151	FB_READ_158_SNVT_pumpset_sn	183
7.2.152	FB_READ_159_SNVT_pump_sensor	184
7.2.153	FB_READ_160_SNVT_abs_humid	184
7.2.154	FB_READ_161_SNVT_flow_p	185
7.2.155	FB_READ_162_SNVT_dev_c_mode	186
7.2.156	FB_READ_163_SNVT_valve_mode	187
7.2.157	FB_READ_164_SNVT_alarm_2	187
7.2.158	FB_READ_165_SNVT_state_64	188
7.2.159	FB_READ_166_SNVT_nv_type	189
7.2.160	FB_READ_168_SNVT_ent_opmode	190
7.2.161	FB_READ_169_SNVT_ent_state	191
7.2.162	FB_READ_170_SNVT_ent_status	191
7.2.163	FB_READ_171_SNVT_flow_dir	192
7.2.164	FB_READ_172_SNVT_hvac_satsts	193

7.2.165	FB_READ_173_SNVT_dev_status.....	194
7.2.166	FB_READ_174_SNVT_dev_fault	195
7.2.167	FB_READ_175_SNVT_dev_maint	195
7.2.168	FB_READ_176_SNVT_date_event	196
7.2.169	FB_READ_177_SNVT_sched_val.....	197
7.2.170	FB_READ_180_SNVT_sblnd_state.....	198
7.2.171	FB_READ_181_SNVT_rac_ctrl	199
7.2.172	FB_READ_182_SNVT_rac_req.....	199
7.2.173	FB_READ_183_SNVT_count_32	200
7.2.174	FB_READ_184_SNVT_clothes_w_c	201
7.2.175	FB_READ_185_SNVT_clothes_w_m.....	202
7.2.176	FB_READ_186_SNVT_clothes_w_s	203
7.2.177	FB_READ_187_SNVT_clothes_w_a.....	203
7.2.178	FB_READ_188_SNVT_multiplier_s.....	204
7.2.179	FB_READ_189_SNVT_switch_2	205
7.2.180	FB_READ_190_SNVT_color_2	206
7.2.181	FB_READ_191_SNVT_log_status.....	207
7.2.182	FB_READ_192_SNVT_time_stamp_p	207
7.2.183	FB_READ_193_SNVT_log_fx_request	208
7.2.184	FB_READ_194_SNVT_log_fx_status.....	209
7.2.185	FB_READ_195_SNVT_log_request	210
7.2.186	FB_READ_196_SNVT_enthalpy_d	211
7.2.187	FB_READ_197_SNVT_amp_ac_mil.....	211
7.2.188	FB_READ_198_SNVT_time_hour_p.....	212
7.2.189	FB_READ_199_SNVT_lamp_status.....	213
7.2.190	FB_READ_200_SNVT_environment	214
7.2.191	FB_READ_201_SNVT_geo_loc	214
7.2.192	FB_SEND_001_SNVT_amp.....	215
7.2.193	FB_SEND_002_SNVT_amp_mil	217
7.2.194	FB_SEND_003_SNVT_angle	218
7.2.195	FB_SEND_004_SNVT_angle_vel.....	219
7.2.196	FB_SEND_005_SNVT_btu_kilo.....	221
7.2.197	FB_SEND_006_SNVT_btu_mega.....	222
7.2.198	FB_SEND_007_SNVT_char_ascii.....	223
7.2.199	FB_SEND_008_SNVT_count	224
7.2.200	FB_SEND_009_SNVT_count_inc.....	226
7.2.201	FB_SEND_011_SNVT_date_day	227
7.2.202	FB_SEND_013_SNVT_elec_kwh.....	228
7.2.203	FB_SEND_014_SNVT_elec_whr.....	229
7.2.204	FB_SEND_015_SNVT_flow.....	231
7.2.205	FB_SEND_016_SNVT_flow_mil.....	232
7.2.206	FB_SEND_017_SNVT_length	233
7.2.207	FB_SEND_018_SNVT_length_kilo.....	235
7.2.208	FB_SEND_019_SNVT_length_micr	236
7.2.209	FB_SEND_020_SNVT_length_mil.....	237
7.2.210	FB_SEND_021_SNVT_lev_cont.....	239

7.2.211	FB_SEND_023_SNVT_mass	240
7.2.212	FB_SEND_024_SNVT_mass_kilo	241
7.2.213	FB_SEND_025_SNVT_mass_mega	243
7.2.214	FB_SEND_026_SNVT_mass_mil	244
7.2.215	FB_SEND_027_SNVT_power	245
7.2.216	FB_SEND_028_SNVT_power_kilo	247
7.2.217	FB_SEND_029_SNVT_ppm	248
7.2.218	FB_SEND_030_SNVT_press	249
7.2.219	FB_SEND_031_SNVT_res	251
7.2.220	FB_SEND_032_SNVT_res_kilo	252
7.2.221	FB_SEND_033_SNVT_sound_db	253
7.2.222	FB_SEND_034_SNVT_speed	255
7.2.223	FB_SEND_035_SNVT_speed_mil	256
7.2.224	FB_SEND_036_SNVT_str_asc	257
7.2.225	FB_SEND_037_SNVT_str_int	258
7.2.226	FB_SEND_038_SNVT_telcom	260
7.2.227	FB_SEND_039_SNVT_temp	261
7.2.228	FB_SEND_041_SNVT_vol	262
7.2.229	FB_SEND_042_SNVT_vol_kilo	263
7.2.230	FB_SEND_043_SNVT_vol_mil	265
7.2.231	FB_SEND_044_SNVT_volt	266
7.2.232	FB_SEND_045_SNVT_volt_dbmv	267
7.2.233	FB_SEND_046_SNVT_volt_kilo	269
7.2.234	FB_SEND_047_SNVT_volt_mil	270
7.2.235	FB_SEND_048_SNVT_amp_f	271
7.2.236	FB_SEND_049_SNVT_angle_f	273
7.2.237	FB_SEND_050_SNVT_angle_vel_f	274
7.2.238	FB_SEND_051_SNVT_count_f	275
7.2.239	FB_SEND_052_SNVT_count_inc_f	277
7.2.240	FB_SEND_053_SNVT_flow_f	278
7.2.241	FB_SEND_054_SNVT_length_f	279
7.2.242	FB_SEND_055_SNVT_lev_cont_f	281
7.2.243	FB_SEND_056_SNVT_mass_f	282
7.2.244	FB_SEND_057_SNVT_power_f	283
7.2.245	FB_SEND_058_SNVT_ppm_f	285
7.2.246	FB_SEND_059_SNVT_press_f	286
7.2.247	FB_SEND_060_SNVT_res_f	287
7.2.248	FB_SEND_061_SNVT_sound_db_f	289
7.2.249	FB_SEND_062_SNVT_speed_f	290
7.2.250	FB_SEND_063_SNVT_temp_f	291
7.2.251	FB_SEND_064_SNVT_time_f	293
7.2.252	FB_SEND_065_SNVT_vol_f	294
7.2.253	FB_SEND_066_SNVT_volt_f	295
7.2.254	FB_SEND_067_SNVT_btu_f	297
7.2.255	FB_SEND_068_SNVT_elec_whr_f	298
7.2.256	FB_SEND_069_SNVT_config_src	299

7.2.257	FB_SEND_070_SNVT_color	300
7.2.258	FB_SEND_071_SNVT_grammage.....	302
7.2.259	FB_SEND_072_SNVT_grammage_f.....	303
7.2.260	FB_SEND_073_SNVT_file_req	304
7.2.261	FB_SEND_074_SNVT_file_status.....	305
7.2.262	FB_SEND_075_SNVT_freq_f.....	307
7.2.263	FB_SEND_076_SNVT_freq_hz.....	308
7.2.264	FB_SEND_077_SNVT_freq_kilohz.....	309
7.2.265	FB_SEND_078_SNVT_freq_milhz	311
7.2.266	FB_SEND_079_SNVT_lux	312
7.2.267	FB_SEND_081_SNVT_lev_percent	313
7.2.268	FB_SEND_082_SNVT_multiplier.....	315
7.2.269	FB_SEND_083_SNVT_state	316
7.2.270	FB_SEND_084_SNVT_time_stamp	317
7.2.271	FB_SEND_085_SNVT_zerospan	318
7.2.272	FB_SEND_086_SNVT_magcard	320
7.2.273	FB_SEND_087_SNVT_elapsed_tm	321
7.2.274	FB_SEND_088_SNVT_alarm	322
7.2.275	FB_SEND_089_SNVT_currency	323
7.2.276	FB_SEND_090_SNVT_file_pos.....	324
7.2.277	FB_SEND_091_SNVT_muldiv.....	326
7.2.278	FB_SEND_092_SNVT_obj_request	327
7.2.279	FB_SEND_093_SNVT_obj_status.....	328
7.2.280	FB_SEND_094_SNVT_preset.....	329
7.2.281	FB_SEND_095_SNVT_switch	330
7.2.282	FB_SEND_096_SNVT_trans_table	332
7.2.283	FB_SEND_097_SNVT_override	333
7.2.284	FB_SEND_098_SNVT_pwr_fact	334
7.2.285	FB_SEND_099_SNVT_pwr_fact_f	335
7.2.286	FB_SEND_100_SNVT_density.....	337
7.2.287	FB_SEND_101_SNVT_density_f.....	338
7.2.288	FB_SEND_102_SNVT_rpm.....	339
7.2.289	FB_SEND_103_SNVT_hvac_emerg	341
7.2.290	FB_SEND_104_SNVT_angle_deg	342
7.2.291	FB_SEND_105_SNVT_temp_p.....	343
7.2.292	FB_SEND_106_SNVT_temp_setpt	344
7.2.293	FB_SEND_107_SNVT_time_sec.....	346
7.2.294	FB_SEND_108_SNVT_hvac_mode	347
7.2.295	FB_SEND_109_SNVT_occupancy.....	348
7.2.296	FB_SEND_110_SNVT_area.....	350
7.2.297	FB_SEND_111_SNVT_hvac_overid.....	351
7.2.298	FB_SEND_112_SNVT_hvac_status.....	352
7.2.299	FB_SEND_113_SNVT_press_p	353
7.2.300	FB_SEND_114_SNVT_address	355
7.2.301	FB_SEND_115_SNVT_scene	356
7.2.302	FB_SEND_116_SNVT_scene_cfg.....	357

7.2.303	FB_SEND_117_SNVT_setting	358
7.2.304	FB_SEND_118_SNVT_evap_state	360
7.2.305	FB_SEND_119_SNVT_therm_mode.....	361
7.2.306	FB_SEND_120_SNVT_defr_mode.....	362
7.2.307	FB_SEND_121_SNVT_defr_term.....	363
7.2.308	FB_SEND_122_SNVT_defr_state	364
7.2.309	FB_SEND_123_SNVT_time_min	366
7.2.310	FB_SEND_124_SNVT_time_hour.....	367
7.2.311	FB_SEND_125_SNVT_ph.....	368
7.2.312	FB_SEND_126_SNVT_ph_f.....	370
7.2.313	FB_SEND_127_SNVT_chlr_status.....	371
7.2.314	FB_SEND_128_SNVT_tod_event	372
7.2.315	FB_SEND_129_SNVT_smo_obscur	373
7.2.316	FB_SEND_130_SNVT_fire_test.....	375
7.2.317	FB_SEND_131_SNVT_temp_ror.....	376
7.2.318	FB_SEND_132_SNVT_fire_init	377
7.2.319	FB_SEND_133_SNVT_fire_indcte	378
7.2.320	FB_SEND_134_SNVT_time_zone	380
7.2.321	FB_SEND_135_SNVT_earth_pos.....	381
7.2.322	FB_SEND_136_SNVT_reg_val	382
7.2.323	FB_SEND_137_SNVT_reg_val_ts	383
7.2.324	FB_SEND_138_SNVT_volt_ac.....	385
7.2.325	FB_SEND_139_SNVT_amp_ac.....	386
7.2.326	FB_SEND_143_SNVT_turbidity	387
7.2.327	FB_SEND_144_SNVT_turbidity_f.....	389
7.2.328	FB_SEND_145_SNVT_hvac_type.....	390
7.2.329	FB_SEND_146_SNVT_elec_kwh_l.....	391
7.2.330	FB_SEND_147_SNVT_temp_diff_p	392
7.2.331	FB_SEND_148_SNVT_ctrl_req.....	394
7.2.332	FB_SEND_149_SNVT_ctrl_resp	395
7.2.333	FB_SEND_150_SNVT_ptz.....	396
7.2.334	FB_SEND_151_SNVT_privacyzone.....	397
7.2.335	FB_SEND_152_SNVT_pos_ctrl	398
7.2.336	FB_SEND_153_SNVT_enthalpy	400
7.2.337	FB_SEND_154_SNVT_gfci_status.....	401
7.2.338	FB_SEND_155_SNVT_motor_state.....	402
7.2.339	FB_SEND_156_SNVT_pumpset_mn	403
7.2.340	FB_SEND_157_SNVT_ex_control	405
7.2.341	FB_SEND_158_SNVT_pumpset_sn	406
7.2.342	FB_SEND_159_SNVT_pump_sensor	407
7.2.343	FB_SEND_160_SNVT_abs_humid	408
7.2.344	FB_SEND_161_SNVT_flow_p.....	410
7.2.345	FB_SEND_162_SNVT_dev_c_mode	411
7.2.346	FB_SEND_163_SNVT_valve_mode.....	412
7.2.347	FB_SEND_164_SNVT_alarm_2.....	413
7.2.348	FB_SEND_165_SNVT_state_64	415

7.2.349	FB_SEND_166_SNVTV_nv_type	416
7.2.350	FB_SEND_168_SNVTV_ent_opmode	417
7.2.351	FB_SEND_169_SNVTV_ent_state	418
7.2.352	FB_SEND_170_SNVTV_ent_status	419
7.2.353	FB_SEND_171_SNVTV_flow_dir	421
7.2.354	FB_SEND_172_SNVTV_hvac_satsts	422
7.2.355	FB_SEND_173_SNVTV_dev_status	423
7.2.356	FB_SEND_174_SNVTV_dev_fault	424
7.2.357	FB_SEND_175_SNVTV_dev_maint	426
7.2.358	FB_SEND_176_SNVTV_date_event	427
7.2.359	FB_SEND_177_SNVTV_sched_val	428
7.2.360	FB_SEND_180_SNVTV_sblnd_state	429
7.2.361	FB_SEND_181_SNVTV_rac_ctrl	431
7.2.362	FB_SEND_182_SNVTV_rac_req	432
7.2.363	FB_SEND_183_SNVTV_count_32	433
7.2.364	FB_SEND_184_SNVTV_clothes_w_c	434
7.2.365	FB_SEND_185_SNVTV_clothes_w_m	436
7.2.366	FB_SEND_186_SNVTV_clothes_w_s	437
7.2.367	FB_SEND_187_SNVTV_clothes_w_a	438
7.2.368	FB_SEND_188_SNVTV_multiplier_s	439
7.2.369	FB_SEND_189_SNVTV_switch_2	441
7.2.370	FB_SEND_190_SNVTV_color_2	442
7.2.371	FB_SEND_191_SNVTV_log_status	443
7.2.372	FB_SEND_192_SNVTV_time_stamp_p	444
7.2.373	FB_SEND_193_SNVTV_log_fx_request	446
7.2.374	FB_SEND_194_SNVTV_log_fx_status	447
7.2.375	FB_SEND_195_SNVTV_log_request	448
7.2.376	FB_SEND_196_SNVTV_enthalpy_d	449
7.2.377	FB_SEND_197_SNVTV_amp_ac_mil	451
7.2.378	FB_SEND_198_SNVTV_time_hour_p	452
7.2.379	FB_SEND_199_SNVTV_lamp_status	453
7.2.380	FB_SEND_200_SNVTV_environment	454
7.2.381	FB_SEND_201_SNVTV_geo_loc	456
7.3	Data types	457
7.3.1	E_LON_ERROR	463
7.3.2	E_LON_Parameter_Datatypes	480
7.3.3	E_LON_alarm_type_t	488
7.3.4	E_LON_appl_cwc_t	490
7.3.5	E_LON_appl_cwp_t	490
7.3.6	E_LON_appl_cws_t	491
7.3.7	E_LON_appl_rin_t	491
7.3.8	E_LON_boolean_t	492
7.3.9	E_LON_calendar_type_t	492
7.3.10	E_LON_cam_act_t	492
7.3.11	E_LON_cam_func_t	493
7.3.12	E_LON_chiller_t	493

7.3.13	E_LON_color_encoding_t.....	493
7.3.14	E_LON_config_source_t.....	494
7.3.15	E_LON_control_resp_t.....	494
7.3.16	E_LON_currency_t.....	494
7.3.17	E_LON_days_of_week_t.....	497
7.3.18	E_LON_defrost_mode_t.....	497
7.3.19	E_LON_defrost_state_t.....	498
7.3.20	E_LON_defrost_term_t.....	498
7.3.21	E_LON_device_c_mode_t.....	498
7.3.22	E_LON_device_select_t.....	499
7.3.23	E_LON_discrete_levels_t.....	500
7.3.24	E_LON_emerg_t.....	500
7.3.25	E_LON_ent_cmd_t.....	500
7.3.26	E_LON_ent_opmode_cmd_t.....	501
7.3.27	E_LON_evap_t.....	502
7.3.28	E_LON_ex_control_t.....	502
7.3.29	E_LON_file_request_t.....	503
7.3.30	E_LON_file_status_t.....	503
7.3.31	E_LON_fire_indicator_t.....	504
7.3.32	E_LON_fire_initiator_t.....	504
7.3.33	E_LON_fire_test_t.....	505
7.3.34	E_LON_flow_direction_t.....	506
7.3.35	E_LON_gfci_status_t.....	506
7.3.36	E_LON_hvac_hvt_t.....	506
7.3.37	E_LON_hvac_overid_t.....	507
7.3.38	E_LON_hvac_t.....	509
7.3.39	E_LON_learn_mode_t.....	510
7.3.40	E_LON_log_status_t.....	510
7.3.41	E_LON_motor_state_t.....	511
7.3.42	E_LON_nv_type_category_t.....	511
7.3.43	E_LON_object_request_t.....	512
7.3.44	E_LON_occup_t.....	513
7.3.45	E_LON_override_t.....	513
7.3.46	E_LON_pan_dir_t.....	514
7.3.47	E_LON_priority_level_t.....	514
7.3.48	E_LON_privacyzone_t.....	515
7.3.49	E_LON_rail_audio_sensor_type_t.....	515
7.3.50	E_LON_rail_audio_type_t.....	516
7.3.51	E_LON_reg_val_unit_t.....	517
7.3.52	E_LON_sblnd_cmd_source_t.....	519
7.3.53	E_LON_sblnd_error_t.....	520
7.3.54	E_LON_scene_config_t.....	521
7.3.55	E_LON_scene_t.....	522
7.3.56	E_LON_sec_state_t.....	523
7.3.57	E_LON_sec_status_t.....	524
7.3.58	E_LON_setting_t.....	525

7.3.59	E_LON_switch_state_t.....	525
7.3.60	E_LON_telcom_states_t.....	527
7.3.61	E_LON_therm_mode_t.....	528
7.3.62	E_LON_tilt_dir_t.....	529
7.3.63	E_LON_unit_temp_t.....	529
7.3.64	E_LON_valve_mode_t.....	529
7.3.65	E_LON_zoom_t.....	530
7.3.66	AuxiliaryStructure.....	530
7.3.67	ST_LON_Parameter_IN_36B.....	555
7.3.68	ST_LON_Parameter_OUT_36B.....	555
7.3.69	ST_ExplicitMessage.....	556
7.3.70	ST_LON_Communication.....	557
7.3.71	ST_LON_ParameterInterface.....	557
7.3.72	ST_LON_WriteData.....	557
7.3.73	ST_Prm.....	558
7.3.74	ST_KL6401.....	558
7.3.75	ST_LON_AddressTable.....	558
7.3.76	ST_LON_ConfigTable.....	559
7.3.77	ST_LON_DomainTable.....	559
7.3.78	ST_LON_SNVT_alarm.....	560
7.3.79	ST_LON_SNVT_alarm_2.....	560
7.3.80	ST_LON_SNVT_chlr_status.....	561
7.3.81	ST_LON_SNVT_clothes_w_a.....	561
7.3.82	ST_LON_SNVT_clothes_w_c.....	563
7.3.83	ST_LON_SNVT_clothes_w_m.....	563
7.3.84	ST_LON_SNVT_clothes_w_s.....	563
7.3.85	ST_LON_SNVT_color.....	564
7.3.86	ST_LON_SNVT_color_2.....	564
7.3.87	ST_LON_SNVT_ctrl_req.....	564
7.3.88	ST_LON_SNVT_ctrl_resp.....	564
7.3.89	ST_LON_SNVT_currency.....	565
7.3.90	ST_LON_SNVT_date_event.....	565
7.3.91	ST_LON_SNVT_dev_fault.....	565
7.3.92	ST_LON_SNVT_dev_maint.....	566
7.3.93	ST_LON_SNVT_dev_status.....	566
7.3.94	ST_LON_SNVT_earth_pos.....	566
7.3.95	ST_LON_SNVT_elapsed_tm.....	566
7.3.96	ST_LON_SNVT_ent_status.....	567
7.3.97	ST_LON_SNVT_environment.....	568
7.3.98	ST_LON_SNVT_ex_control.....	569
7.3.99	ST_LON_SNVT_file_pos.....	569
7.3.100	ST_LON_SNVT_file_req.....	569
7.3.101	ST_LON_SNVT_file_status.....	570
7.3.102	ST_LON_SNVT_geo_loc.....	570
7.3.103	ST_LON_SNVT_hvac_overid.....	570
7.3.104	ST_LON_SNVT_hvac_satsts.....	570

7.3.105	ST_LON_SNVT_hvac_status	571
7.3.106	ST_LON_SNVT_lamp_status	571
7.3.107	ST_LON_SNVT_log_fx_request	572
7.3.108	ST_LON_SNVT_log_fx_status	572
7.3.109	ST_LON_SNVT_log_status	572
7.3.110	ST_LON_SNVT_muldiv	573
7.3.111	ST_LON_SNVT_nv_type	573
7.3.112	ST_LON_SNVT_obj_request	574
7.3.113	ST_LON_SNVT_obj_status	574
7.3.114	ST_LON_SNVT_pos_ctrl	575
7.3.115	ST_LON_SNVT_preset	576
7.3.116	ST_LON_SNVT_privacyzone	576
7.3.117	ST_LON_SNVT_ptz	576
7.3.118	ST_LON_SNVT_pump_sensor	577
7.3.119	ST_LON_SNVT_pumpset_mn	577
7.3.120	ST_LON_SNVT_pumpset_sn	578
7.3.121	ST_LON_SNVT_rac_ctrl	579
7.3.122	ST_LON_SNVT_rac_req	579
7.3.123	ST_LON_SNVT_reg_val	579
7.3.124	ST_LON_SNVT_reg_val_ts	580
7.3.125	ST_LON_SNVT_sblnd_state	580
7.3.126	ST_LON_SNVT_scene	580
7.3.127	ST_LON_SNVT_scene_cfg	581
7.3.128	ST_LON_SNVT_setting	581
7.3.129	ST_LON_SNVT_str_int	581
7.3.130	ST_LON_SNVT_switch	582
7.3.131	ST_LON_SNVT_switch_2	582
7.3.132	ST_LON_SNVT_temp_setpt	582
7.3.133	ST_LON_SNVT_time_zone	583
7.3.134	ST_LON_SNVT_tod_event	583
7.3.135	ST_LON_SNVT_trans_table	583
7.3.136	ST_LON_SNVT_zerospan	584
7.3.137	str_AddressTable	584
7.4	Globale_Variablen_LON	585
7.5	Error codes	585
8	Appendix	604
8.1	Automatic sending	604
8.2	dwErrorKL	604
8.3	Examples	605
8.4	SNVT-variables (OFF)	605
8.5	Support and Service	607

1 Foreword

1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.

It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.

It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.

We reserve the right to revise and change the documentation at any time and without prior announcement. No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.

Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

with corresponding applications or registrations in various other countries.



EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.

Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.

1.2 For your safety

Safety regulations

Read the following explanations for your safety.

Always observe and follow product-specific safety instructions, which you may find at the appropriate places in this document.

Exclusion of liability

All the components are supplied in particular hardware and software configurations which are appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation, and drive technology who are familiar with the applicable national standards.

Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

Personal injury warnings

⚠ DANGER

Hazard with high risk of death or serious injury.

⚠ WARNING

Hazard with medium risk of death or serious injury.

⚠ CAUTION

There is a low-risk hazard that could result in medium or minor injury.

Warning of damage to property or environment

NOTICE

The environment, equipment, or data may be damaged.

Information on handling the product



This information includes, for example: recommendations for action, assistance or further information on the product.

1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our <https://www.beckhoff.com/secguide>.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at <https://www.beckhoff.com/secinfo>.

2 Introduction

The LON library is an extensive TwinCAT PLC library for data exchange with LON devices. The communication with these devices is realized via SNVTs (**S**tandard **N**etwork **V**ariable **T**ypes). These SNVTs are defined in the LONMark (see also [LONMARK](#) and [LONMARK Resources Files](#)). For each SNVT there is a function block for sending and another function block for receiving.

All function blocks from the library must be called in the same task.

This library should only be used in conjunction with a [KL6401](#) (LON master terminal).

The SNVT must be configured with the [KS2000](#) [[▶ 27](#)] in the terminal.

Linking the SNVTs is done with a LON configuration tool (e.g. LON Maker Echelon). This tool is not supplied by Beckhoff.

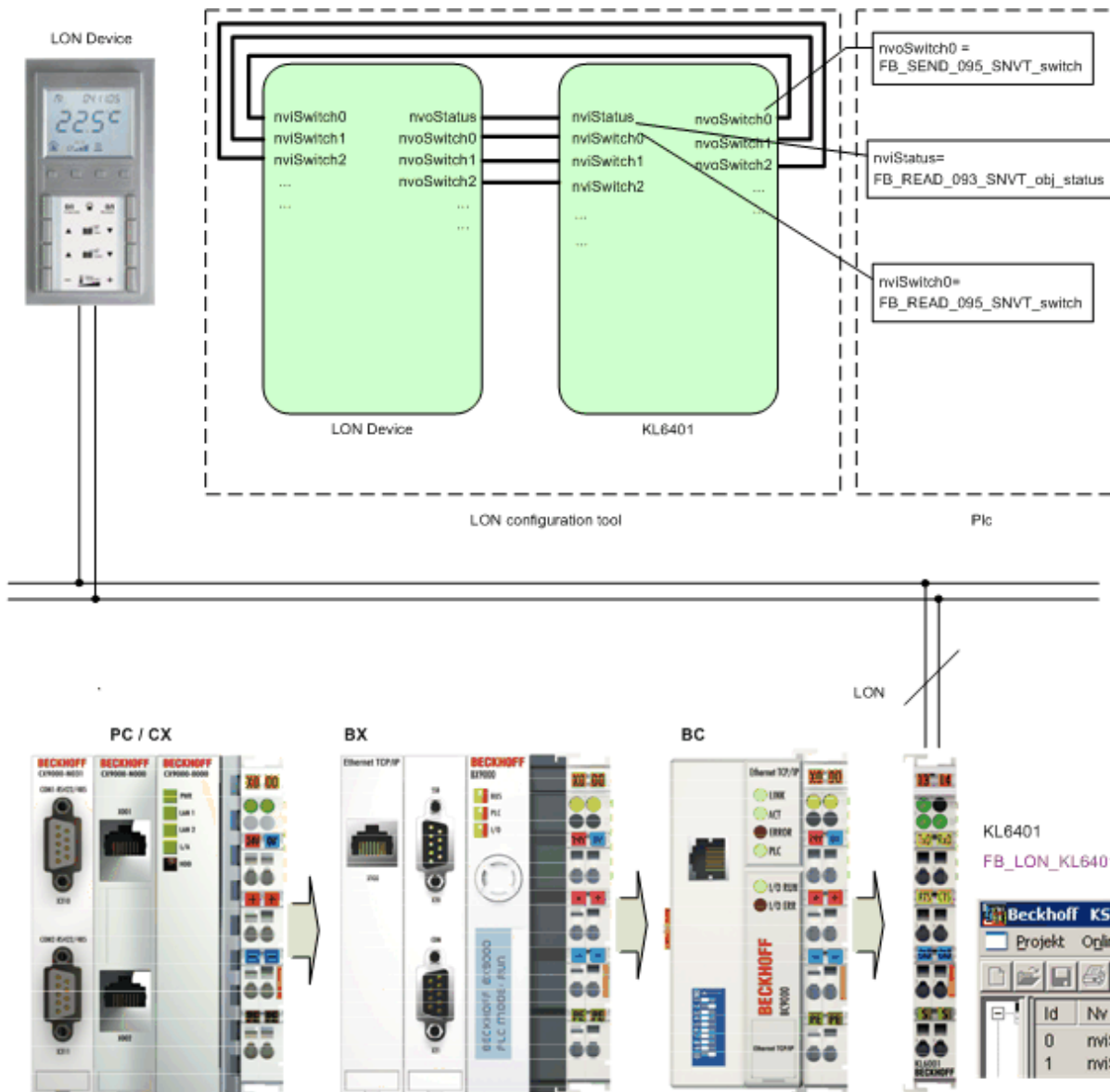
3 Target groups

The user of this library requires basic knowledge of the following.

- TwinCAT PLC-Control
- TwinCAT System Manager
- KS2000
- PCs and networks
- Structure and properties of the Beckhoff Embedded PC and its Bus Terminal system
- Technology of LON devices / LON configuration tools
- Relevant safety regulations for building technical equipment

This software library is intended for building automation system partners of Beckhoff Automation GmbH & Co. KG. The system partners operate in the field of building automation and are concerned with the installation, commissioning, expansion, maintenance and service of measurement, control and regulating systems for the technical equipment of buildings.

4 LON



Each LON device provides network variables (SNVTs) for communication with other devices. The required variables are selected based on the device documentation or the Xif file (machine-readable representation of all used SNVTs) and entered in the KL6401 via the KS2000 [▶ 27]. The KS2000 can now be used to create another Xif file. Both Xif files are read and combined with a LON configuration tool (not provided by Beckhoff). For further information please refer to the respective tool.

For each SNVT entered in the KL6401, a suitable function block has to be programmed on the PLC side. These function blocks can then be used to access the SNVTs of the devices.

4.1 Supported SNVTs

General information about SNVTs can be found on the internet under www.lonmark.org.

The KL6401 supports the following SNVTs (LonMark data types).

SNVT ID	SNVT NAME	Read	Send	Additional information
1	SNVT_amp	Read [▶ 67]	Send [▶ 215]	
2	SNVT_amp_mil	Read [▶ 68]	Send [▶ 217]	

SNVT ID	SNVT NAME	Read	Send	Additional information
3	SNVT_angle	Read [▶ 69]	Send [▶ 218]	
4	SNVT_angle_vel	Read [▶ 70]	Send [▶ 219]	
5	SNVT_btu_kilo	Read [▶ 71]	Send [▶ 221]	
6	SNVT_btu_mega	Read [▶ 71]	Send [▶ 222]	
7	SNVT_char_ascii	Read [▶ 72]	Send [▶ 223]	
8	SNVT_count	Read [▶ 73]	Send [▶ 224]	
9	SNVT_count_inc	Read [▶ 74]	Send [▶ 226]	
10	SNVT_date_cal			SNVT obsolete, not implemented
11	SNVT_date_day	Read [▶ 74]	Send [▶ 227]	
12	SNVT_date_time			SNVT obsolete, not implemented
13	SNVT_elec_kwh	Read [▶ 75]	Send [▶ 228]	
14	SNVT_elec_whr	Read [▶ 76]	Send [▶ 229]	
15	SNVT_flow	Read [▶ 77]	Send [▶ 231]	
16	SNVT_flow_mil	Read [▶ 78]	Send [▶ 232]	
17	SNVT_length	Read [▶ 78]	Send [▶ 233]	
18	SNVT_length_kilo	Read [▶ 79]	Send [▶ 235]	
19	SNVT_length_micr	Read [▶ 80]	Send [▶ 236]	
20	SNVT_length_mil	Read [▶ 81]	Send [▶ 237]	
21	SNVT_lev_cont	Read [▶ 81]	Send [▶ 239]	
22	SNVT_lev_disc			SNVT obsolete, not implemented
23	SNVT_mass	Read [▶ 82]	Send [▶ 240]	
24	SNVT_mass_kilo	Read [▶ 83]	Send [▶ 241]	
25	SNVT_mass_mega	Read [▶ 84]	Send [▶ 243]	
26	SNVT_mass_mil	Read [▶ 84]	Send [▶ 244]	
27	SNVT_power	Read [▶ 85]	Send [▶ 245]	
28	SNVT_power_kilo	Read [▶ 86]	Send [▶ 247]	
29	SNVT_ppm	Read [▶ 87]	Send [▶ 248]	
30	SNVT_press	Read [▶ 87]	Send [▶ 249]	
31	SNVT_res	Read [▶ 88]	Send [▶ 251]	
32	SNVT_res_kilo	Read [▶ 89]	Send [▶ 252]	
33	SNVT_sound_db	Read [▶ 90]	Send [▶ 253]	
34	SNVT_speed	Read [▶ 90]	Send [▶ 255]	
35	SNVT_speed_mil	Read [▶ 91]	Send [▶ 256]	
36	SNVT_str_asc	Read [▶ 92]	Send [▶ 257]	
37	SNVT_str_int	Read [▶ 93]	Send [▶ 258]	
38	SNVT_telcom	Read [▶ 93]	Send [▶ 260]	
39	SNVT_temp	Read [▶ 94]	Send [▶ 261]	
40	SNVT_time_passed			SNVT obsolete, not implemented
41	SNVT_vol	Read [▶ 95]	Send [▶ 262]	
42	SNVT_vol_kilo	Read [▶ 96]	Send [▶ 263]	
43	SNVT_vol_mil	Read [▶ 97]	Send [▶ 265]	

SNVT ID	SNVT NAME	Read	Send	Additional information
44	SNVT_volt	Read [▶ 97]	Send [▶ 266]	
45	SNVT_volt_dbmv	Read [▶ 98]	Send [▶ 267]	
46	SNVT_volt_kilo	Read [▶ 99]	Send [▶ 269]	
47	SNVT_volt_mil	Read [▶ 100]	Send [▶ 270]	
48	SNVT_amp_f	Read [▶ 100]	Send [▶ 271]	
49	SNVT_angle_f	Read [▶ 101]	Send [▶ 273]	
50	SNVT_angle_vel_f	Read [▶ 102]	Send [▶ 274]	
51	SNVT_count_f	Read [▶ 103]	Send [▶ 275]	
52	SNVT_count_inc_f	Read [▶ 103]	Send [▶ 277]	
53	SNVT_flow_f	Read [▶ 104]	Send [▶ 278]	
54	SNVT_length_f	Read [▶ 105]	Send [▶ 279]	
55	SNVT_lev_cont_f	Read [▶ 106]	Send [▶ 281]	
56	SNVT_mass_f	Read [▶ 106]	Send [▶ 282]	
57	SNVT_power_f	Read [▶ 107]	Send [▶ 283]	
58	SNVT_ppm_f	Read [▶ 108]	Send [▶ 285]	
59	SNVT_press_f	Read [▶ 109]	Send [▶ 286]	
60	SNVT_res_f	Read [▶ 109]	Send [▶ 287]	
61	SNVT_sound_db_f	Read [▶ 110]	Send [▶ 289]	
62	SNVT_speed_f	Read [▶ 111]	Send [▶ 290]	
63	SNVT_temp_f	Read [▶ 112]	Send [▶ 291]	
64	SNVT_time_f	Read [▶ 112]	Send [▶ 293]	
65	SNVT_vol_f	Read [▶ 113]	Send [▶ 294]	
66	SNVT_volt_f	Read [▶ 114]	Send [▶ 295]	
67	SNVT_btu_f	Read [▶ 115]	Send [▶ 297]	
68	SNVT_elec_whr_f	Read [▶ 115]	Send [▶ 298]	
69	SNVT_config_src	Read [▶ 116]	Send [▶ 299]	
70	SNVT_color	Read [▶ 117]	Send [▶ 300]	
71	SNVT_grammage	Read [▶ 118]	Send [▶ 302]	
72	SNVT_grammage_f	Read [▶ 118]	Send [▶ 303]	
73	SNVT_file_req	Read [▶ 119]	Send [▶ 304]	Under construction, please not use.
74	SNVT_file_status	Read [▶ 120]	Send [▶ 305]	
75	SNVT_freq_f	Read [▶ 121]	Send [▶ 307]	
76	SNVT_freq_hz	Read [▶ 121]	Send [▶ 308]	
77	SNVT_freq_kilohz	Read [▶ 122]	Send [▶ 309]	
78	SNVT_freq_milhz	Read [▶ 123]	Send [▶ 311]	
79	SNVT_lux	Read [▶ 124]	Send [▶ 312]	
80	SNVT_ISO_7811			SNVT obsolete, not implemented
81	SNVT_lev_percent	Read [▶ 124]	Send [▶ 313]	
82	SNVT_multiplier	Read [▶ 125]	Send [▶ 315]	
83	SNVT_state	Read [▶ 126]	Send [▶ 316]	
84	SNVT_time_stamp	Read [▶ 127]	Send [▶ 317]	
85	SNVT_zerospan	Read [▶ 128]	Send [▶ 318]	

SNVT ID	SNVT NAME	Read	Send	Additional information
86	SNVT_magcard	Read [▶ 128]	Send [▶ 320]	
87	SNVT_elapsed_tm	Read [▶ 129]	Send [▶ 321]	
88	SNVT_alarm	Read [▶ 130]	Send [▶ 322]	
89	SNVT_currency	Read [▶ 131]	Send [▶ 323]	
90	SNVT_file_pos	Read [▶ 132]	Send [▶ 324]	
91	SNVT_muldiv	Read [▶ 132]	Send [▶ 326]	
92	SNVT_obj_request	Read [▶ 133]	Send [▶ 327]	
93	SNVT_obj_status	Read [▶ 134]	Send [▶ 328]	
94	SNVT_preset	Read [▶ 135]	Send [▶ 329]	
95	SNVT_switch	Read [▶ 136]	Send [▶ 330]	
96	SNVT_trans_table	Read [▶ 136]	Send [▶ 332]	
97	SNVT_override	Read [▶ 137]	Send [▶ 333]	
98	SNVT_pwr_fact	Read [▶ 138]	Send [▶ 334]	
99	SNVT_pwr_fact_f	Read [▶ 139]	Send [▶ 335]	
100	SNVT_density	Read [▶ 140]	Send [▶ 337]	
101	SNVT_density_f	Read [▶ 140]	Send [▶ 338]	
102	SNVT_rpm	Read [▶ 141]	Send [▶ 339]	
103	SNVT_hvac_emerg	Read [▶ 142]	Send [▶ 341]	
104	SNVT_angle_deg	Read [▶ 143]	Send [▶ 342]	
105	SNVT_temp_p	Read [▶ 143]	Send [▶ 343]	
106	SNVT_temp_setpt	Read [▶ 144]	Send [▶ 344]	
107	SNVT_time_sec	Read [▶ 145]	Send [▶ 346]	
108	SNVT_hvac_mode	Read [▶ 146]	Send [▶ 347]	
109	SNVT_occupancy	Read [▶ 146]	Send [▶ 348]	
110	SNVT_area	Read [▶ 147]	Send [▶ 350]	
111	SNVT_hvac_overid	Read [▶ 148]	Send [▶ 351]	
112	SNVT_hvac_status	Read [▶ 149]	Send [▶ 352]	
113	SNVT_press_p	Read [▶ 150]	Send [▶ 353]	
114	SNVT_address	Read [▶ 150]	Send [▶ 355]	
115	SNVT_scene	Read [▶ 151]	Send [▶ 356]	
116	SNVT_scene_cfg	Read [▶ 152]	Send [▶ 357]	
117	SNVT_setting	Read [▶ 153]	Send [▶ 358]	
118	SNVT_evap_state	Read [▶ 153]	Send [▶ 360]	
119	SNVT_therm_mode	Read [▶ 154]	Send [▶ 361]	
120	SNVT_defr_mode	Read [▶ 155]	Send [▶ 362]	
121	SNVT_defr_term	Read [▶ 156]	Send [▶ 363]	
122	SNVT_defr_state	Read [▶ 157]	Send [▶ 364]	
123	SNVT_time_min	Read [▶ 157]	Send [▶ 366]	
124	SNVT_time_hour	Read [▶ 158]	Send [▶ 367]	
125	SNVT_ph	Read [▶ 159]	Send [▶ 368]	
126	SNVT_ph_f	Read [▶ 160]	Send [▶ 370]	
127	SNVT_chlr_status	Read [▶ 161]	Send [▶ 371]	
128	SNVT_tod_event	Read [▶ 161]	Send [▶ 372]	

SNVT ID	SNVT NAME	Read	Send	Additional information
129	SNVT_smo_obscur	Read [▶ 162]	Send [▶ 373]	
130	SNVT_fire_test	Read [▶ 163]	Send [▶ 375]	
131	SNVT_temp_ror	Read [▶ 164]	Send [▶ 376]	
132	SNVT_fire_init	Read [▶ 165]	Send [▶ 377]	
133	SNVT_fire_indcte	Read [▶ 165]	Send [▶ 378]	
134	SNVT_time_zone	Read [▶ 166]	Send [▶ 380]	
135	SNVT_earth_pos	Read [▶ 167]	Send [▶ 381]	
136	SNVT_reg_val	Read [▶ 168]	Send [▶ 382]	
137	SNVT_reg_val_ts	Read [▶ 169]	Send [▶ 383]	
138	SNVT_volt_ac	Read [▶ 169]	Send [▶ 385]	
139	SNVT_amp_ac	Read [▶ 170]	Send [▶ 386]	
143	SNVT_turbidity	Read [▶ 171]	Send [▶ 387]	
144	SNVT_turbidity_f	Read [▶ 172]	Send [▶ 389]	
145	SNVT_hvac_type	Read [▶ 172]	Send [▶ 390]	
146	SNVT_elec_kwh_l	Read [▶ 173]	Send [▶ 391]	
147	SNVT_temp_diff_p	Read [▶ 174]	Send [▶ 392]	
148	SNVT_ctrl_req	Read [▶ 175]	Send [▶ 394]	
149	SNVT_ctrl_resp	Read [▶ 176]	Send [▶ 395]	
150	SNVT_ptz	Read [▶ 176]	Send [▶ 396]	
151	SNVT_privacyzone	Read [▶ 177]	Send [▶ 397]	
152	SNVT_pos_ctrl	Read [▶ 178]	Send [▶ 398]	
153	SNVT_enthalpy	Read [▶ 179]	Send [▶ 400]	
154	SNVT_gfci_status	Read [▶ 180]	Send [▶ 401]	
155	SNVT_motor_state	Read [▶ 180]	Send [▶ 402]	
156	SNVT_pumpset_mn	Read [▶ 181]	Send [▶ 403]	
157	SNVT_ex_control	Read [▶ 182]	Send [▶ 405]	
158	SNVT_pumpset_sn	Read [▶ 183]	Send [▶ 406]	
159	SNVT_pump_sensor	Read [▶ 184]	Send [▶ 407]	
160	SNVT_abs_humid	Read [▶ 184]	Send [▶ 408]	
161	SNVT_flow_p	Read [▶ 185]	Send [▶ 410]	
162	SNVT_dev_c_mode	Read [▶ 186]	Send [▶ 411]	
163	SNVT_valve_mode	Read [▶ 187]	Send [▶ 412]	
164	SNVT_alarm_2	Read [▶ 187]	Send [▶ 413]	
165	SNVT_state_64	Read [▶ 188]	Send [▶ 415]	
166	SNVT_nv_type	Read [▶ 189]	Send [▶ 416]	
168	SNVT_ent_opmode	Read [▶ 190]	Send [▶ 417]	
169	SNVT_ent_state	Read [▶ 191]	Send [▶ 418]	
170	SNVT_ent_status	Read [▶ 191]	Send [▶ 419]	
171	SNVT_flow_dir	Read [▶ 192]	Send [▶ 421]	
172	SNVT_hvac_satsts	Read [▶ 193]	Send [▶ 422]	
173	SNVT_dev_status	Read [▶ 194]	Send [▶ 423]	

SNVT ID	SNVT NAME	Read	Send	Additional information
174	SNVT_dev_fault	Read [▶ 195]	Send [▶ 424]	
175	SNVT_dev_maint	Read [▶ 195]	Send [▶ 426]	
176	SNVT_date_event	Read [▶ 196]	Send [▶ 427]	
177	SNVT_sched_val	Read [▶ 197]	Send [▶ 428]	
178	SNVT_sec_state			SNVT obsolete, not implemented
179	SNVT_sec_status			SNVT obsolete, not implemented
180	SNVT_sblnd_state	Read [▶ 198]	Send [▶ 429]	
181	SNVT_rac_ctrl	Read [▶ 199]	Send [▶ 431]	
182	SNVT_rac_req	Read [▶ 199]	Send [▶ 432]	
183	SNVT_count_32	Read [▶ 200]	Send [▶ 433]	
184	SNVT_clothes_w_c	Read [▶ 201]	Send [▶ 434]	
185	SNVT_clothes_w_m	Read [▶ 202]	Send [▶ 436]	
186	SNVT_clothes_w_s	Read [▶ 203]	Send [▶ 437]	
187	SNVT_clothes_w_a	Read [▶ 203]	Send [▶ 438]	
188	SNVT_multiplier_s	Read [▶ 204]	Send [▶ 439]	
189	SNVT_switch_2	Read [▶ 205]	Send [▶ 441]	
190	SNVT_color_2	Read [▶ 206]	Send [▶ 442]	
191	SNVT_log_status	Read [▶ 207]	Send [▶ 443]	
192	SNVT_time_stamp_p	Read [▶ 207]	Send [▶ 444]	
193	SNVT_log_fx_request	Read [▶ 208]	Send [▶ 446]	
194	SNVT_log_fx_status	Read [▶ 209]	Send [▶ 447]	
195	SNVT_log_request	Read [▶ 210]	Send [▶ 448]	
196	SNVT_enthalpy_d	Read [▶ 211]	Send [▶ 449]	
197	SNVT_amp_ac_mil	Read [▶ 211]	Send [▶ 451]	
198	SNVT_time_hour_p	Read [▶ 212]	Send [▶ 452]	
199	SNVT_lamp_status	Read [▶ 213]	Send [▶ 453]	
200	SNVT_environment	Read [▶ 214]	Send [▶ 454]	
201	SNVT_geo_loc	Read [▶ 214]	Send [▶ 456]	

4.2 How the KL6401 functions

Sending

The KL6401 sends single SNVT variables. This means that an SNVT variable sent to the KL6401 is sent to the LON network individually. Only when this has successfully been sent can the next SNVT variable be transferred to the KL6401.

Receiving

The KL6401 has 2 buffers, the telegram buffer, and the index buffer.

The input variables of a LON telegram that has been received are placed in the telegram buffer. The index number of the incoming telegram is also entered into the index buffer.

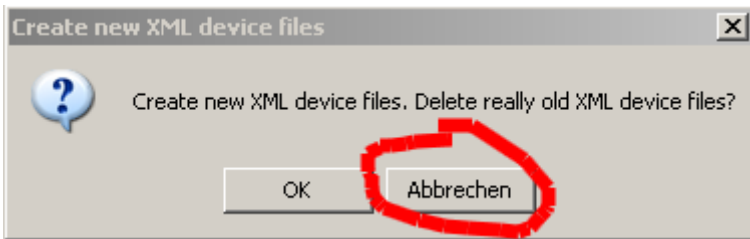
The TwinCAT LON function block evaluates the index buffer, transfers input variables that have been received to the application program and removes them from the index buffer. If the function block does not read the data out of the telegram buffer quickly enough, it can save up to 62 entries (i.e. the maximum number of SNVT variables).

If a telegram that has already been received (which the function block has not yet transferred to the application program, and which therefore still has an entry in the index buffer) is received from the KL6401 again, then a new entry is not made in the index buffer, but the input variables in the telegram buffer are updated.

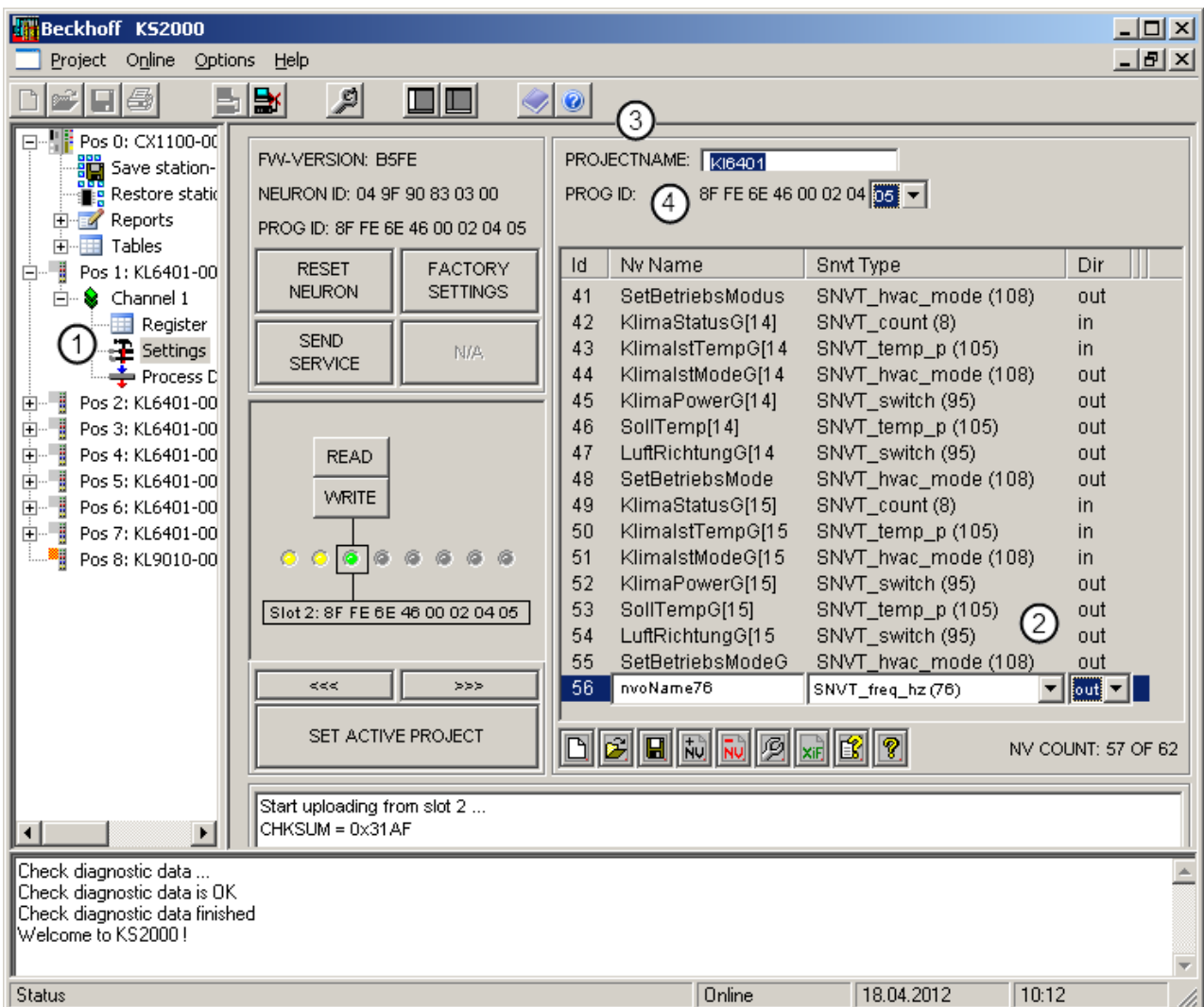
5 Configuration with KS2000

Commissioning procedure

1. Install the configuration software KS2000, version 4.3.0.39 or higher.
2. Plug the KL6401 into your bus coupler, and terminate the K-bus with a KL9010.
3. Log on. Should the dialog box "Create new XML Device Files" be open, cancel at this point.



The terminal can now be configured under "Settings" ① .



Enter the SNVTs in the table ② . Use to add a variable and to remove a selected variable.

The column "Id" cannot be edited. It contains the NV index. This index is also required as input variable "wNVIndex" for the PLC function blocks [► 57].

In column "Nv Name" any text can be entered. These descriptions are required in the LON configuration tool (not provided by Beckhoff) for identifying the respective variables.

Select the required SNVT in column "SNVT Type".

Column "Dir" (Direction) indicates whether the variable is sent (out) or received (in).


③ Set any project name (click on "PROJECTNAME").

Then edit the program ID (Prog ID) ④. Do this by clicking Program ID, and selecting a Program ID. This program ID may only be issued once in your LON project. A maximum of 256 (0-255) program IDs can be used. This means that you can use 256 different configurations in one project.



LON terminals with the same configuration also have the same program ID.



Use the button  to create the Xif file. This file is required in the LON configuration tool (not provided by Beckhoff).



Use the button  to save the configuration in a BLC file. This file can be used to load the configuration if a terminal is replaced or to use the same configuration in another terminal.

The data can now be written to the terminal. Use the buttons [>>>] or [<<<] to select the required slot and then the button "WRITE" to write the data to the terminal. The slot may not be activated (green) during this process.



The terminal is supplied with slot 0 active (factory setting). This slot contains a fixed configuration and cannot be modified. To transfer other values a different slot has to be configured and activated. Only one slot at a time is active.

The project must be activated after the download has successfully been completed. Use the button "SET ACTIVE PROJECT". The KL6401 must now be disconnected from the power supply. After switching on the KL6401 (switching the power supply on again) the desired configuration is active.

Description of the editing functions



Deletes the table



Opens an existing SNVT configuration file (*.BLC) for the KL6401



Saves the current configuration as a BLC file



Edits an SNVT variable



Inserts an SNVT variable



Deletes an SNVT variable



Creates an XIF file for a LON configuration tool (not provided by Beckhoff)



Opens the Help



Information

6 Integration into TwinCAT

6.1 KL6401 - Linking to the TwinCAT System Manager

How do I link the KL6401 to the System Manager?

Click onto the variable *wParameter* of the input structure with the right mouse button and choose "Change Link". (see figure 1)

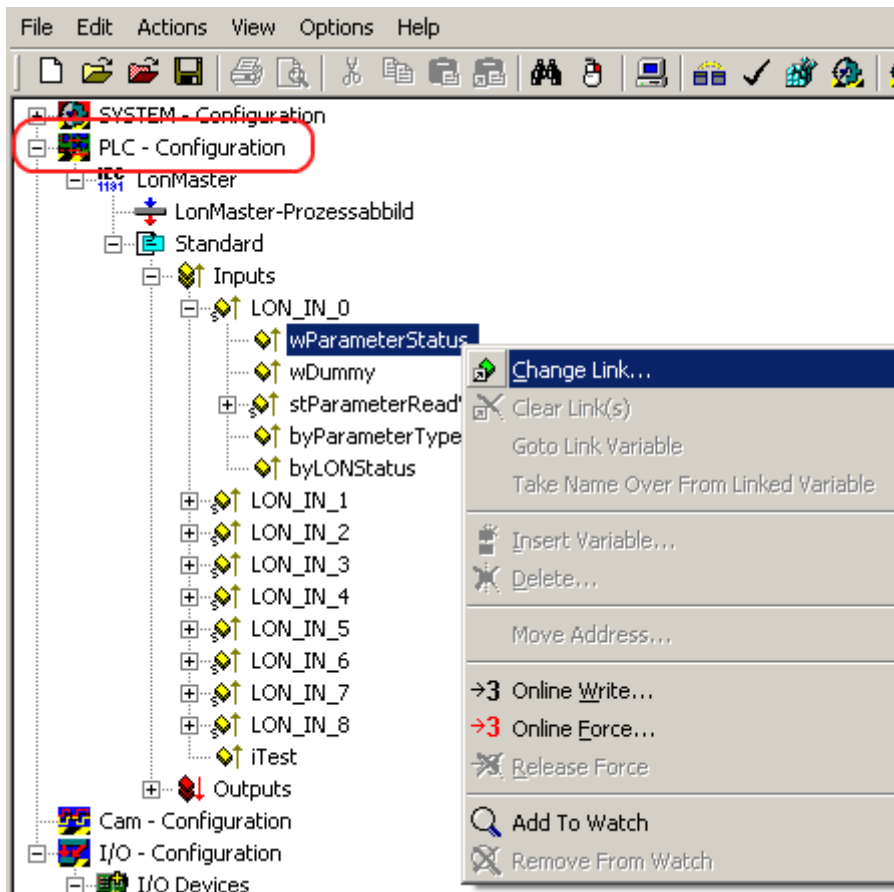


Figure 1

Choose the desired terminal in the "I/O - Configuration", click on "Parameter Status" and confirm with "OK". (see figure 2)

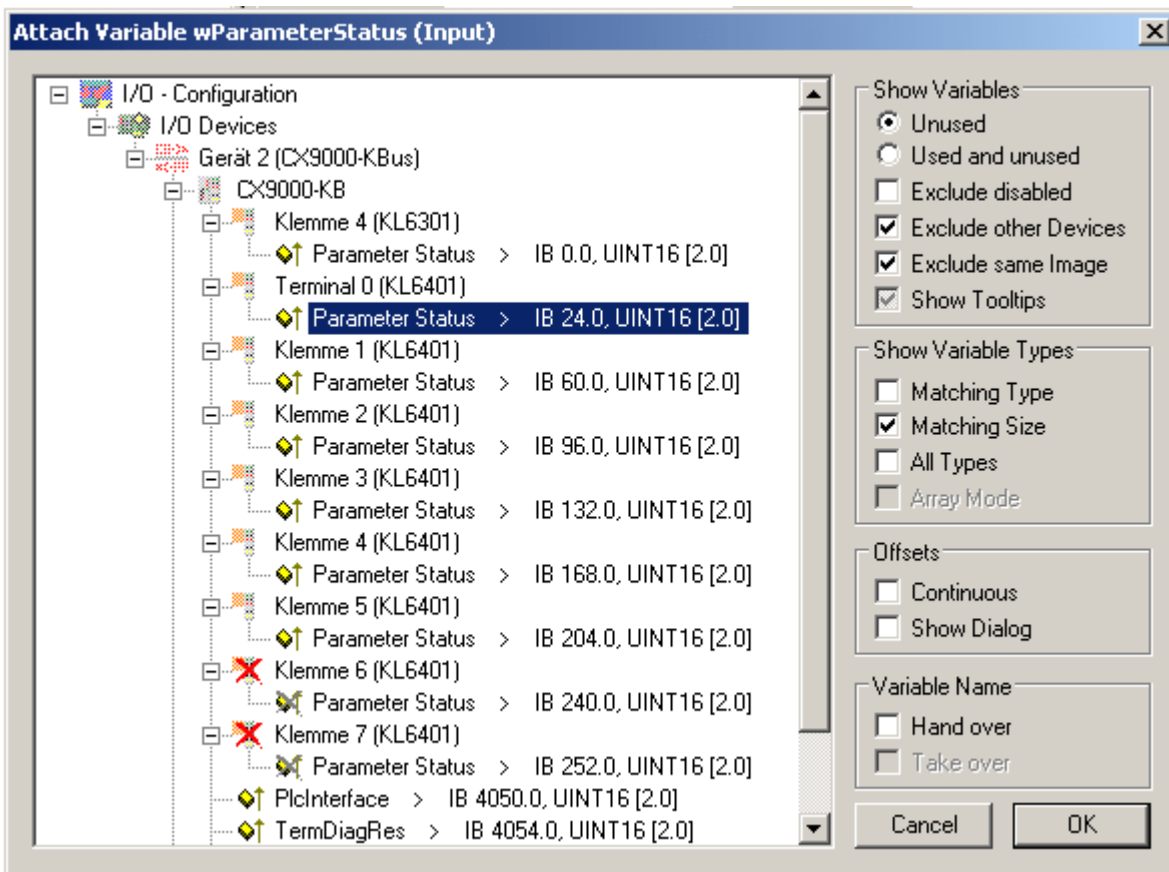


Fig. 1: TcLON_VerknuepfungenSMBild2

Figure 2

In the next step click on the input structure *stParameterReadValue* with the right mouse button and choose "Change Link". (see figure 3) *wDummy* is not linked.

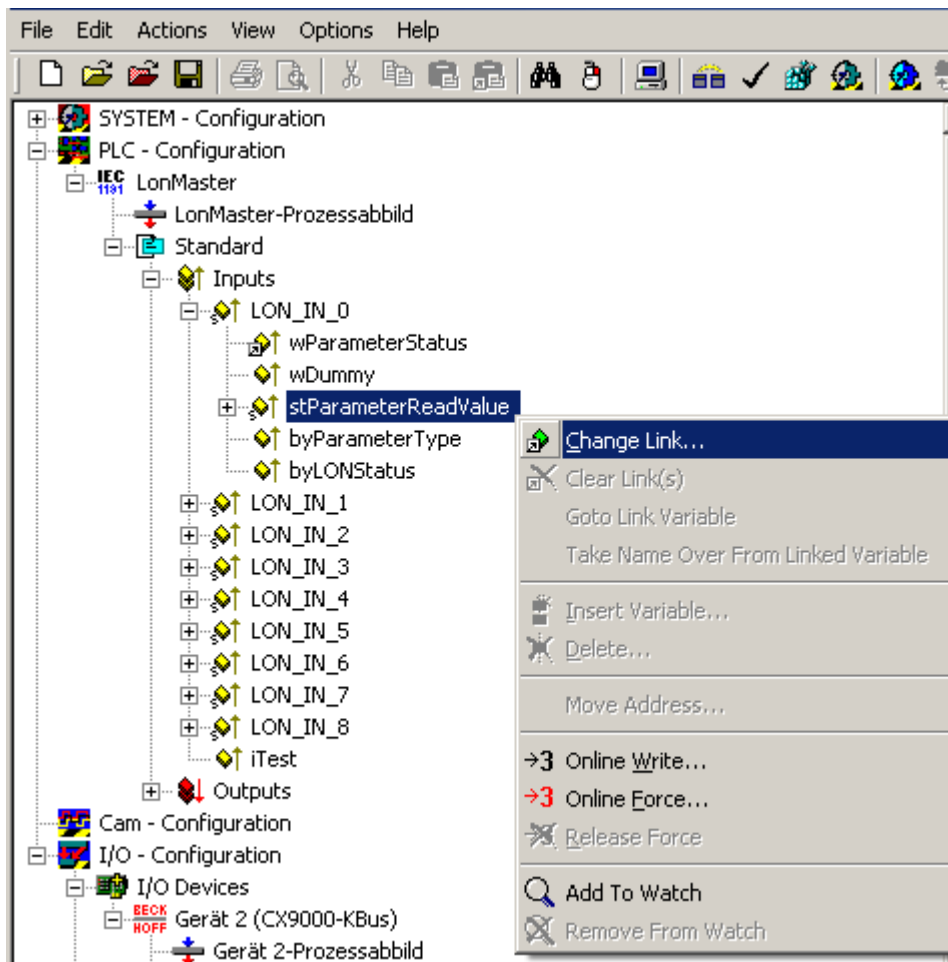


Fig. 2: TcLON_VerknuepfungenSMBild3

Figure 3

Choose "All Types" and "Continuous" and mark "Parameter Input Data 1" to "Parameter Input Data 8" with the left mouse button + left >SHIFT< button. (see figure 4) Then press the "OK" button.

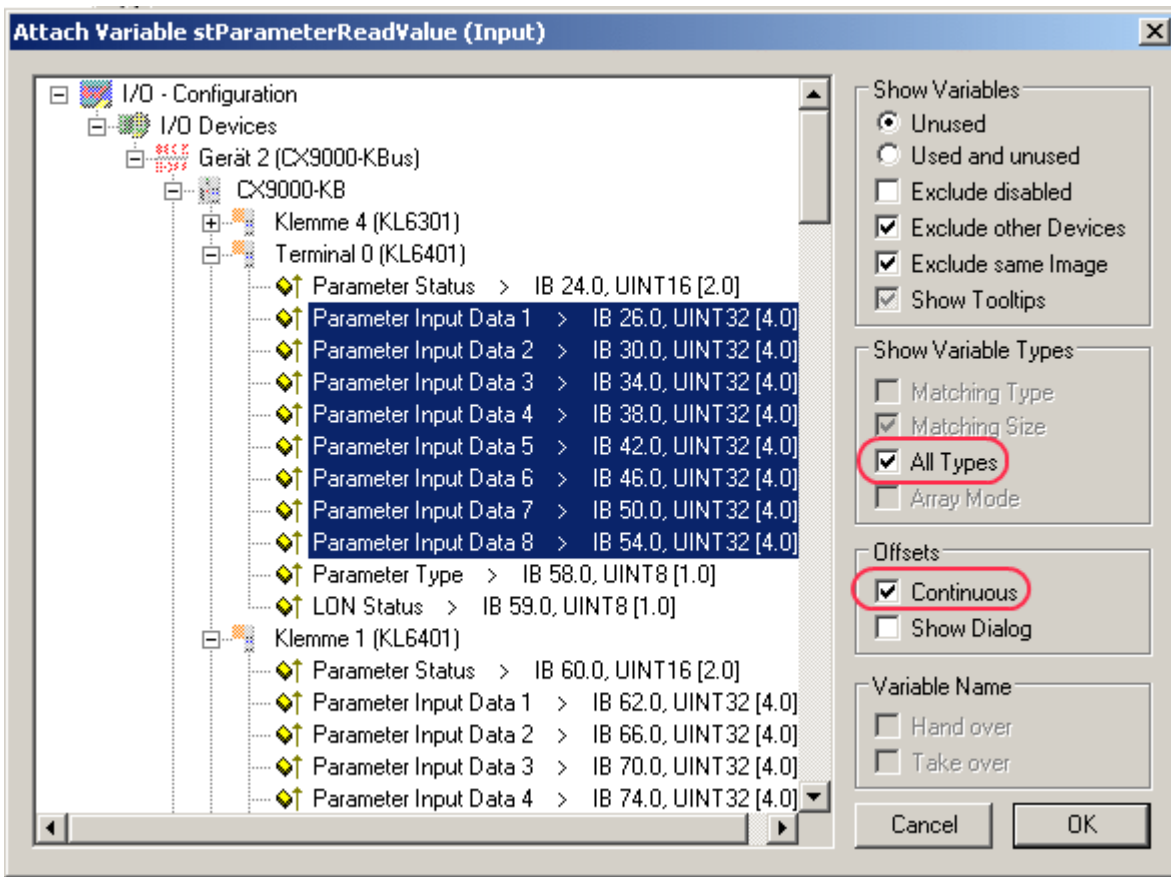


Fig. 3: TcLON_VerknuepfungenSMBild4

Figure 4

Link the variables *byParameterType* and *byLONStatus*. (see figure 5 and 6)

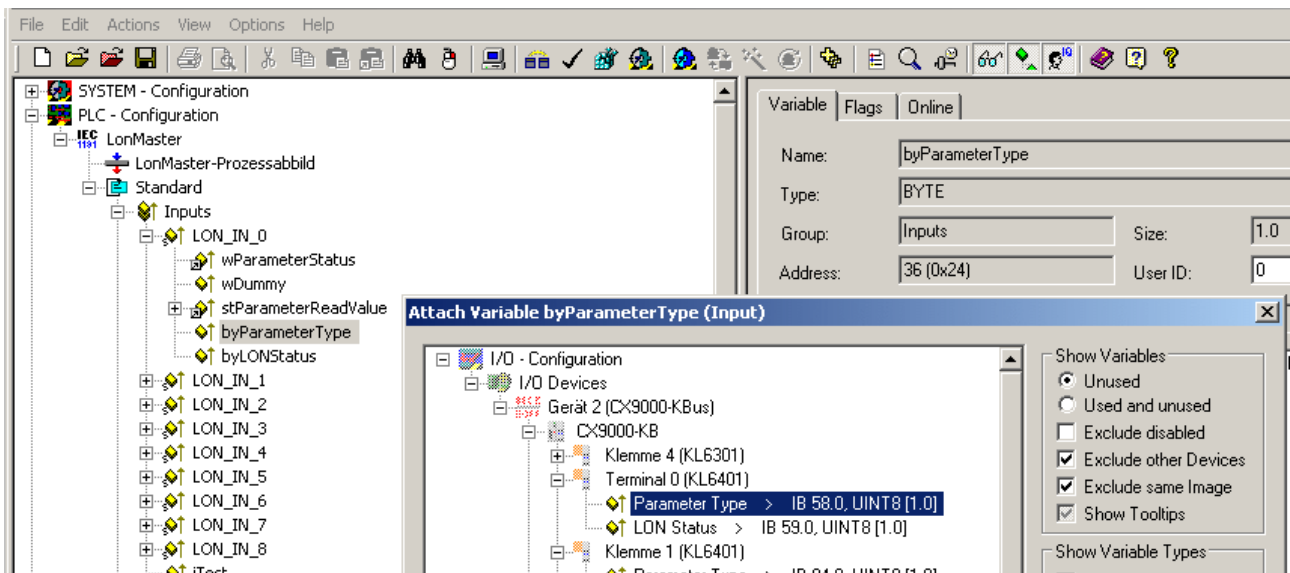


Fig. 4: TcLON_VerknuepfungenSMBild5

Figure 5

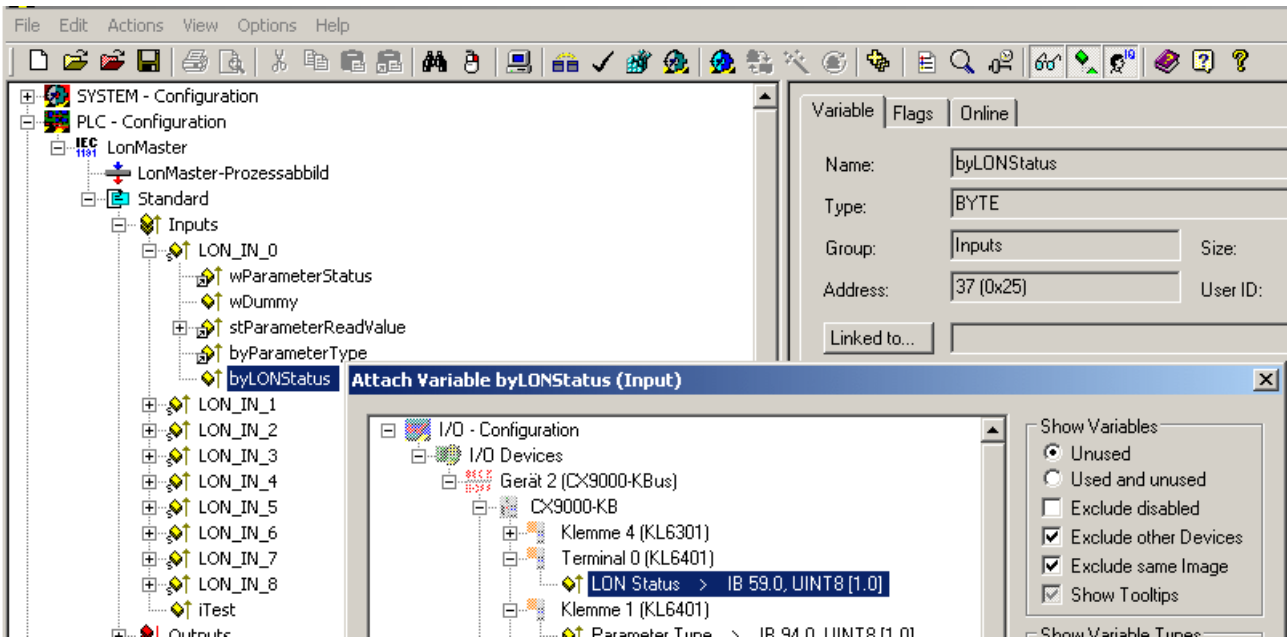


Fig. 5: TcLON_VerkneufungenSMBild6

Figure 6

You can now check the connection. To do this, go to the KL6401 and open it (I/O - Configuration / I/O Devices). All terminal data should be marked by a small arrow (see Figure 7). If that is the case, then proceed in the same way with the outputs.

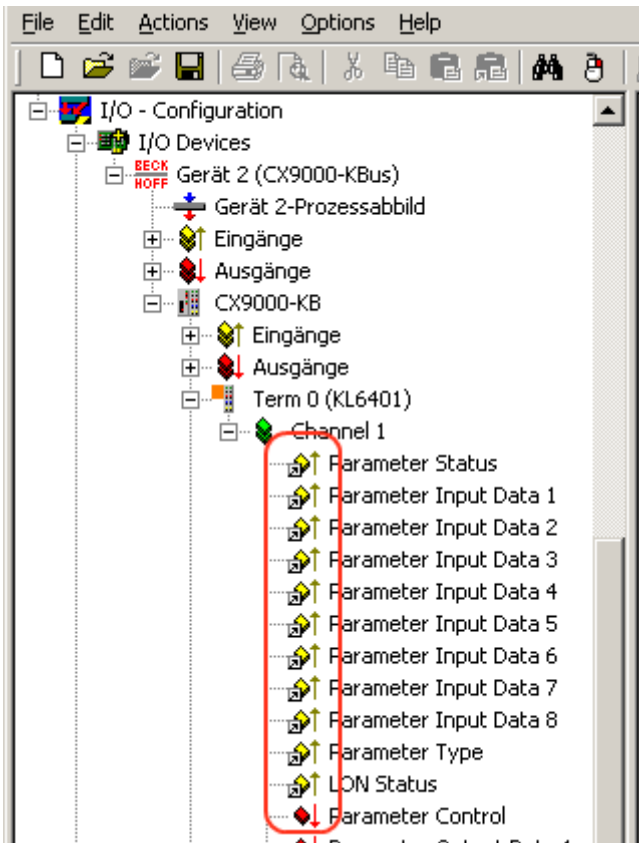


Figure 7

6.2 Integration in TwinCAT (CX9020)

This example explains how to write a simple PLC program for LON in TwinCAT and how to link it with the hardware. The task is to change the state of a switching output with a button.

<https://infosys.beckhoff.com/content/1033/tcplcliblon/Resources/11997701515/.zip> <https://infosys.beckhoff.com/content/1033/tcplcliblon/Resources/11997701515/.zip>

Hardware

Setting up the components

The following hardware is required:

- 1x Embedded PC [CX9020](#)
- 1x digital 2-channel input terminal KL1002 (for the set and reset functions)
- 1x LON terminal [KL6401](#)
- 1x end terminal KL9010

Set up the hardware and the LON components as described in the respective documents.

This example assumes that a Set button was connected to the first KL1002 input and a Reset button to the second.

Software

Creation of the PLC program

Create a new PLC project for PC-based systems (ARM) and add the *TcLON.lib* library.

Next, generate the following global variables:

```
VAR_GLOBAL
  bSet      AT %I*      : BOOL;
  bReset    AT %I*      : BOOL;
  stParameter_IN  AT %I* : ST_LON_Parameter_IN_36B;
  stParameter_OUT AT %Q* : ST_LON_Parameter_OUT_36B;
  stLON_Com  : ST_LON_Communication;
END_VAR
```

bSwitch: Input variable for the Set button.

bReset: Input variable for the Reset button.

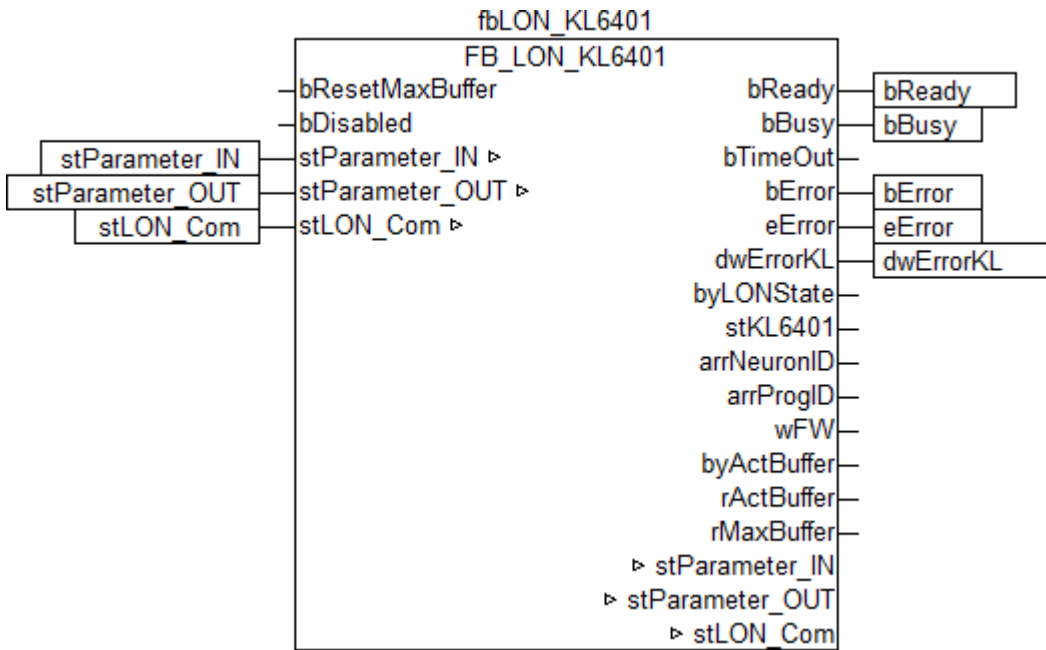
stParameter_IN : [Input variable \[► 555\]](#) for the LON terminal.

stParameter_OUT : [Output variable \[► 555\]](#) for the LON terminal.

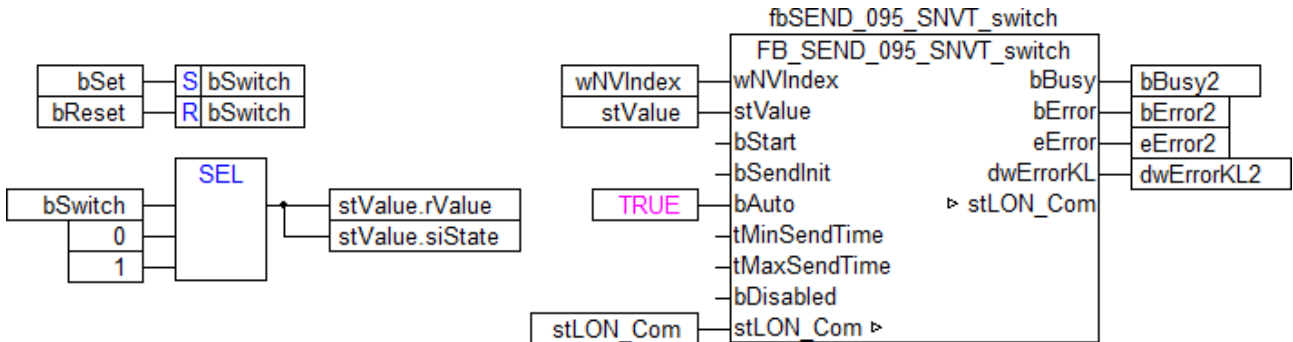
stLON_Com : Required for [communication \[► 557\]](#) with LON.

All function blocks with LON must be called in the same task.

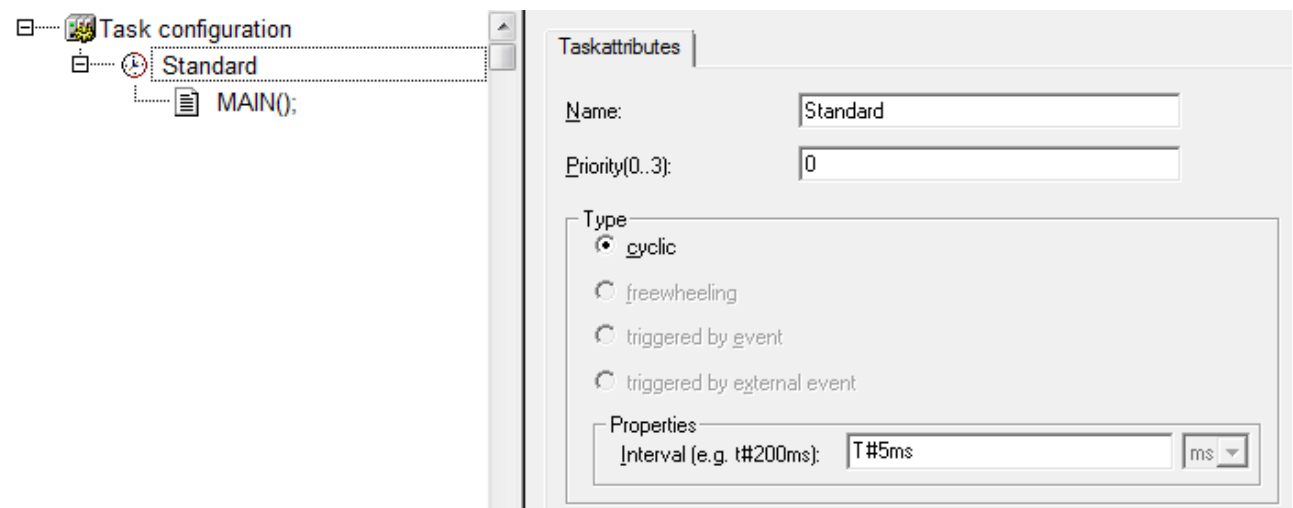
Therefore, create a MAIN program (CFC) in which the [FB_LON_KL6401\(\) \[► 66\]](#) and [EIB_SEND_095_SNVt_switch\(\) \[► 330\]](#) function blocks are called up. Make sure to link the communication block with *stParameter_IN*, *stParameter_OUT* and *stLON_Com*.



Link the local variable `bSwitch` with the global variables `bSet` and `bReset`, then with the input of the selection. Link the local variable `stValue` with the output of the selection, then with the input `stValue` of the send block.



Go to the task configuration and give the task a shorter interval time. More detailed information can be found in the `FB_LON_KL6401()` [▶ 66] block description.

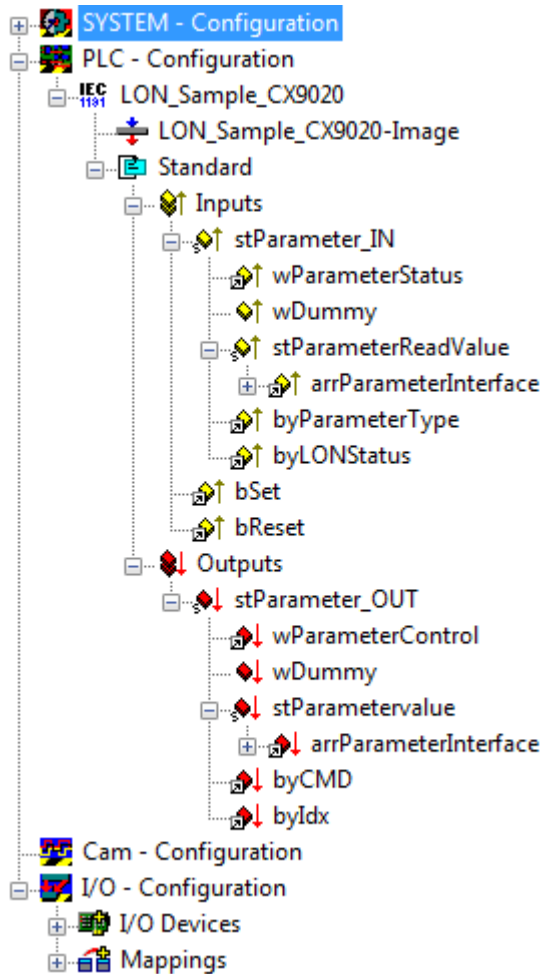


Load the project to the CX as boot project and save it.

Configuration in the System Manager

Create a new TwinCAT System Manager project, select the CX as the target system, and search for the associated hardware.

Add the PLC program created above under PLC configuration.



Now link the global variables of the PLC program with the Bus Terminal inputs and outputs, create the allocations, and activate the configuration. Then start the device in run mode. Your CX is now ready for use.

The switching output can be set or reset by pressing the button.

6.3 Integration into TwinCAT (BC9191)

This example explains how to write a simple PLC program for LON in TwinCAT and how to link it with the hardware. The task is to change the state of a switching output with a button.

<https://infosys.beckhoff.com/content/1033/tcplcliblon/Resources/11997702923/.zip> <https://infosys.beckhoff.com/content/1033/tcplcliblon/Resources/11997702923/>

Hardware

Setting up the components

The following hardware is required:

- 1x Bus Terminal Controller [BC9191](#)
- 1x potential feed terminal 24V DC
- 1x digital 2-channel input terminal KL1002 (for the set and reset functions)
- 1x LON terminal [KL6401](#)
- 1x end terminal KL9010

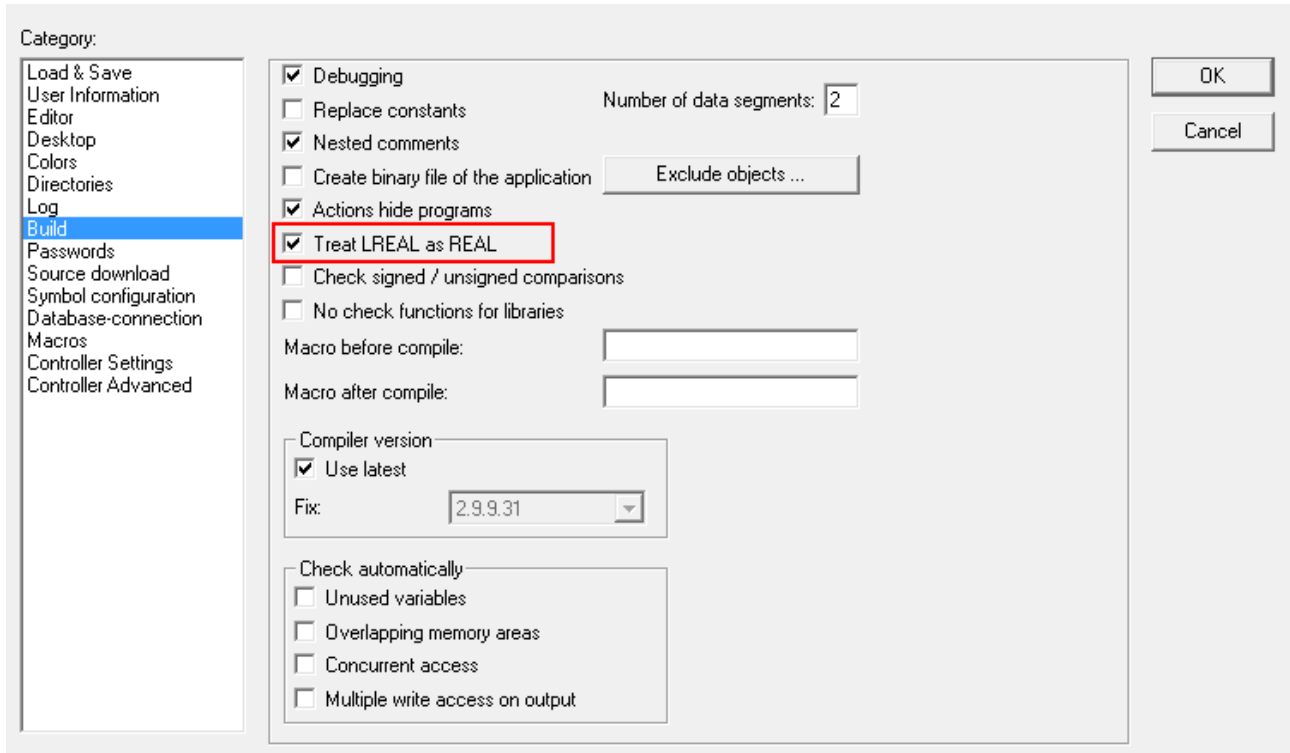
Set up the hardware and the LON components as described in the respective documents.

This example assumes that a Set button was connected to the first KL1002 input and a Reset button to the second.

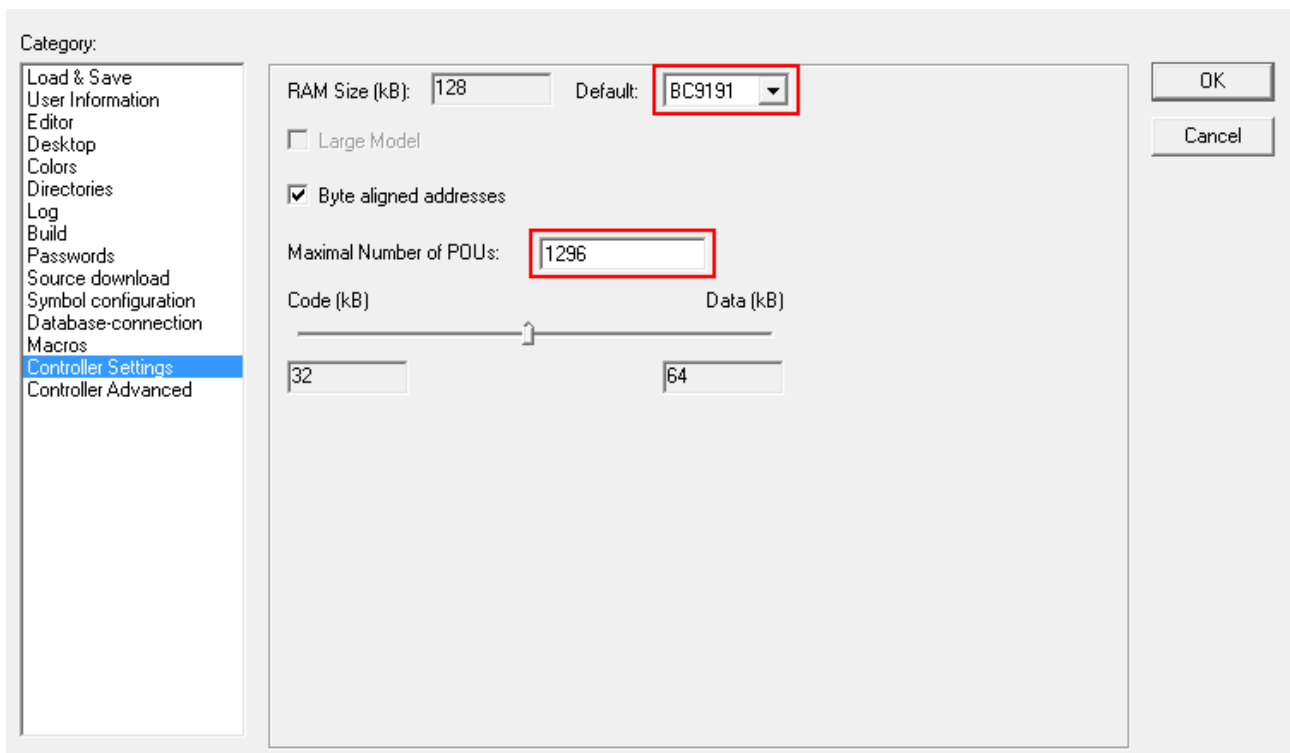
Software

Creation of the PLC program

Create a new PLC project for BC-based systems (BCxx50 via AMS) and add the libraries *TcLON.lbx* and *TcSystemBCxx50.lbx*. Then navigate to *Project*→*Options...* →*Build* and select *TreatLREAL as REAL*.



Under *Controller settings* then change the controller to BC9191 and increase the maximum number of POU's to 1296.



Next, generate the following global variables:

```
VAR_GLOBAL
  bSet      AT %I*      : BOOL;
  bReset    AT %I*      : BOOL;
  stParameter_IN  AT %I* : ST_LON_Parameter_IN_36B;
  stParameter_OUT AT %Q* : ST_LON_Parameter_OUT_36B;
  stLON_Com          : ST_LON_Communication;
END_VAR
```

bSwitch: Input variable for the Set button.

bReset: Input variable for the Reset button.

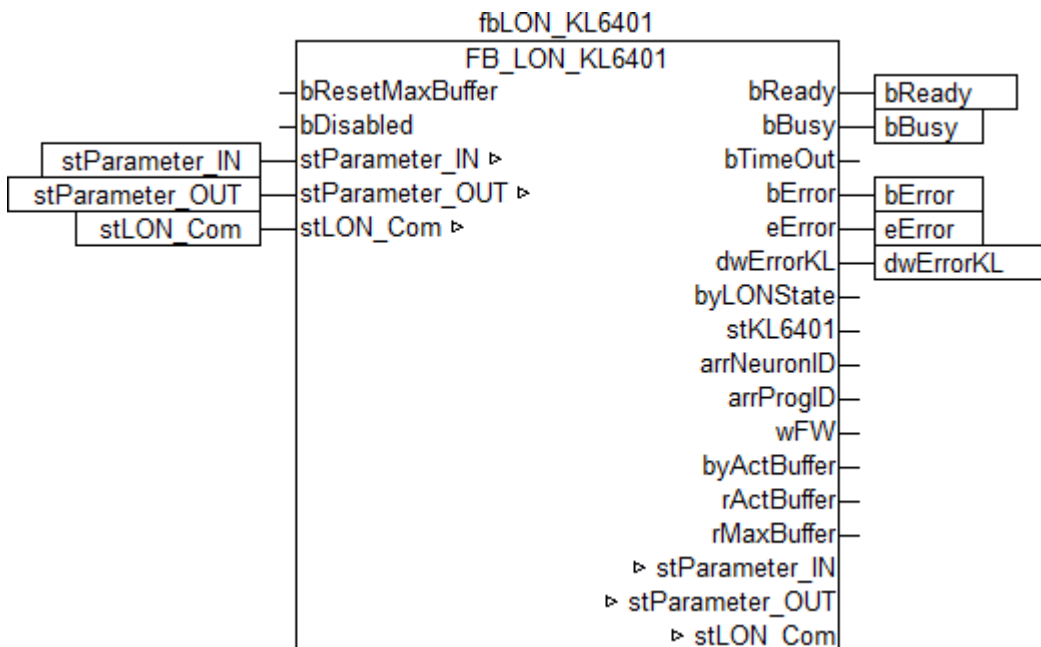
stParameter_IN : Input variable [▶ 555] for the LON terminal.

stParameter_OUT : Output variable [▶ 555] for the LON terminal.

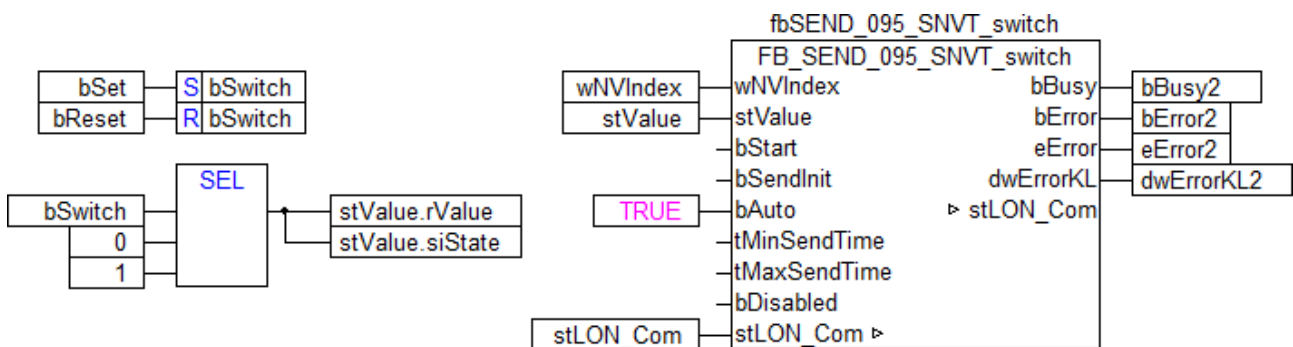
stLON_Com : Required for communication [▶ 557] with LON.

Since BC devices can only process one task, communication with LON cannot run separately.

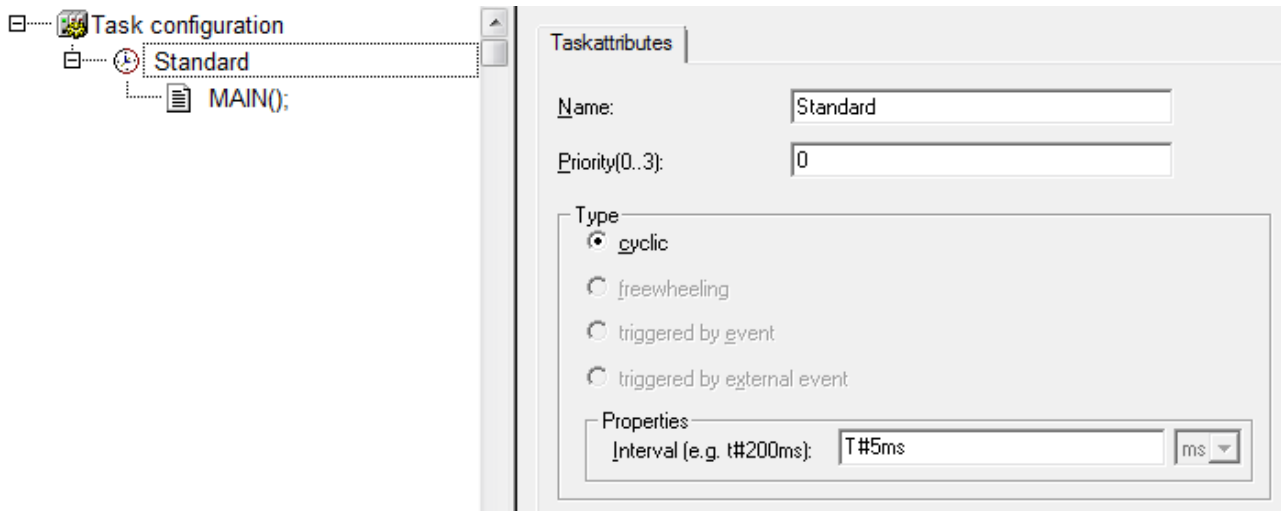
Therefore, create a MAIN program (CFC) in which the FB_LON_KL6401() [▶ 66] and FB_SEND_095_SNVT_switch() [▶ 330] function blocks are called. Make sure to link the communication block with *stParameter_IN*, *stParameter_OUT* and *stLON_Com*.



Link the local variable *bSwitch* with the global variables *bSet* and *bReset*, then with the selected input. Link the local variable *stValue* with the selected output, then with the input *stValue* of the send block.



Go to the task configuration and give the task a lower interval time. More detailed information can be found in the FB_LON_KL6401() [▶ 66] block description.

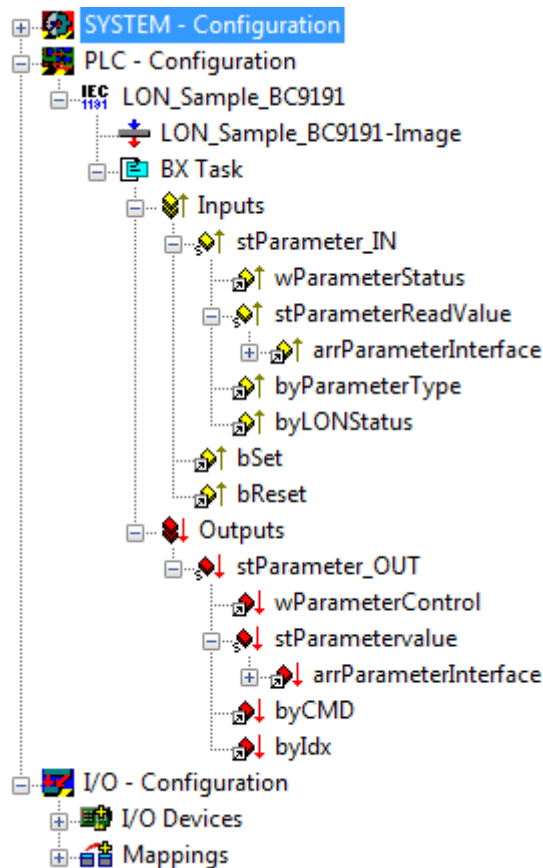


Now load the project as a boot project to the BC and save it.

Configuration in the System Manager

Create a new TwinCAT System Manager project, select the BC as the target system, and search for the associated hardware.

Add the PLC program created above under PLC configuration.



Now link the global variables of the PLC program with the Bus Terminal inputs and outputs, create the allocations, and activate the configuration. Then start the device in run mode. Your BC is now ready for use.

The switching output can be set or reset by pressing the button.

7 Programming

Content
General information [▶ 56]
KL6401 - Linking to the TwinCAT System Manager [▶ 30]

POUs

POUs	Description
FB_LON_KL6401 [▶ 66]	Send / receiver block

POUs/Read

POUs	Description
FB_READ_001_SNVT_amp [▶ 67]	Electric current (Amperes)
FB_READ_002_SNVT_amp_mil [▶ 68]	Electric current (milliAmperes)
FB_READ_003_SNVT_angle [▶ 69]	Angular distance (radians)
FB_READ_004_SNVT_angle_vel [▶ 70]	Angular velocity (radians/second)
FB_READ_005_SNVT_btu_kilo [▶ 71]	Thermal energy (kilo-Btus)
FB_READ_006_SNVT_btu_mega [▶ 71]	Thermal energy (mega-Btus)
FB_READ_007_SNVT_char_ascii [▶ 72]	ASCII character (8-bit ASCII character)
FB_READ_008_SNVT_count [▶ 73]	Absolute count (units)
FB_READ_009_SNVT_count_inc [▶ 74]	Increment count (units (delta))
FB_READ_011_SNVT_date_day [▶ 74]	Day of week (day names)
FB_READ_013_SNVT_elec_kwh [▶ 75]	Electric energy (kiloWatt-hours)
FB_READ_014_SNVT_elec_whr [▶ 76]	Electric energy (Watt-hours)
FB_READ_015_SNVT_flow [▶ 77]	Flow volume (liters/second)
FB_READ_016_SNVT_flow_mil [▶ 78]	Flow volume (milliliters/second)
FB_READ_017_SNVT_length [▶ 78]	Length (meters)
FB_READ_018_SNVT_length_kilo [▶ 79]	Length (kilometers)
FB_READ_019_SNVT_length_micr [▶ 80]	Length(micrometers (microns))
FB_READ_020_SNVT_length_mil [▶ 81]	Length (millimeters)
FB_READ_021_SNVT_lev_cont [▶ 81]	Continuous level (% of full level)
FB_READ_023_SNVT_mass [▶ 82]	Mass (grams)
FB_READ_024_SNVT_mass_kilo [▶ 83]	Mass (kilograms)
FB_READ_025_SNVT_mass_mega [▶ 84]	Mass (metric tons)
FB_READ_026_SNVT_mass_mil [▶ 84]	Mass (milligrams)
FB_READ_027_SNVT_power [▶ 85]	Power (Watts)
FB_READ_028_SNVT_power_kilo [▶ 86]	Power (kiloWatts)
FB_READ_029_SNVT_ppm [▶ 87]	Concentration (ppm)
FB_READ_030_SNVT_press [▶ 87]	Pressure (gauge) (kiloPascals)
FB_READ_031_SNVT_res [▶ 88]	Electric resistance (Ohms)
FB_READ_032_SNVT_res_kilo [▶ 89]	Electric resistance (kiloOhms)
FB_READ_033_SNVT_sound_db [▶ 90]	Sound level (dB)
FB_READ_034_SNVT_speed [▶ 90]	Linear velocity (meters/second)
FB_READ_035_SNVT_speed_mil [▶ 91]	Linear velocity (meters/second)

POUs	Description
FB_READ_036_SNVT_str_asc [► 92]	Character string
FB_READ_037_SNVT_str_int [► 93]	Wide character string with locale code
FB_READ_038_SNVT_telcom [► 93]	Telecomm states
FB_READ_039_SNVT_temp [► 94]	Temperature (degrees Celsius)
FB_READ_041_SNVT_vol [► 95]	Volume (liters)
FB_READ_042_SNVT_vol_kilo [► 96]	Volume (kiloliters)
FB_READ_043_SNVT_vol_mil [► 97]	Volume (milliliters)
FB_READ_044_SNVT_volt [► 97]	Electric voltage (Volts)
FB_READ_045_SNVT_volt_dbmv [► 98]	Electric voltage (dB microVolts)
FB_READ_046_SNVT_volt_kilo [► 99]	Electric voltage (kiloVolts)
FB_READ_047_SNVT_volt_mil [► 100]	Electric voltage (milliVolts)
FB_READ_048_SNVT_amp_f [► 100]	Electric current (Amperes)
FB_READ_049_SNVT_angle_f [► 101]	Angular distance (radians)
FB_READ_050_SNVT_angle_vel_f [► 102]	Angular velocity (radians/second)
FB_READ_051_SNVT_count_f [► 103]	Absolute count (units)
FB_READ_052_SNVT_count_inc_f [► 103]	Increment count (units (delta))
FB_READ_053_SNVT_flow_f [► 104]	Flow volume (liters/second)
FB_READ_054_SNVT_length_f [► 105]	Length (meters)
FB_READ_055_SNVT_lev_cont_f [► 106]	Continuous level
FB_READ_056_SNVT_mass_f [► 106]	Mass (grams)
FB_READ_057_SNVT_power_f [► 107]	Power (Watts)
FB_READ_058_SNVT_ppm_f [► 108]	Concentration (ppm)
FB_READ_059_SNVT_press_f [► 109]	Pressure (gauge) (Pascals)
FB_READ_060_SNVT_res_f [► 109]	Electric resistance (Ohms)
FB_READ_061_SNVT_sound_db_f [► 110]	Sound level (dBspl)
FB_READ_062_SNVT_speed_f [► 111]	Linear velocity (meters/second)
FB_READ_063_SNVT_temp_f [► 112]	Temperature (degrees Celsius)
FB_READ_064_SNVT_time_f [► 112]	Elapsed time (seconds)
FB_READ_065_SNVT_vol_f [► 113]	Volume (liters)
FB_READ_066_SNVT_volt_f [► 114]	Electric voltage (Volts)
FB_READ_067_SNVT_btu_f [► 115]	Thermal energy (Btus)
FB_READ_068_SNVT_elec_whr_f [► 115]	Electric energy (Watt-hours)
FB_READ_069_SNVT_config_src [► 116]	Configuration source
FB_READ_070_SNVT_color [► 117]	CIELAB color (L*,a*,b)
FB_READ_071_SNVT_grammage [► 118]	Grammage (grams/sq meter)
FB_READ_072_SNVT_grammage_f [► 118]	Grammage (grams/sq meter)
FB_READ_073_SNVT_file_req [► 119]	File request
FB_READ_074_SNVT_file_status [► 120]	File status
FB_READ_075_SNVT_freq_f [► 121]	Frequency (Hertz)
FB_READ_076_SNVT_freq_hz [► 121]	Frequency (Hertz)
FB_READ_077_SNVT_freq_kilohz [► 122]	Frequency (kiloHertz)
FB_READ_078_SNVT_freq_milhz [► 123]	Frequency (Hertz)
FB_READ_079_SNVT_lux [► 124]	Illumination (lux)

POUs	Description
FB_READ_081_SNVT_lev_percent [► 124]	Percentage level (% of full level)
FB_READ_082_SNVT_multiplier [► 125]	Multiplier
FB_READ_083_SNVT_state [► 126]	State vector
FB_READ_084_SNVT_time_stamp [► 127]	Time stamp
FB_READ_085_SNVT_zerospan [► 128]	Zero and span
FB_READ_086_SNVT_magcard [► 128]	ISO 7811 (40 hexadecimal digits)
FB_READ_087_SNVT_elapsed_tm [► 129]	Elapsed time
FB_READ_088_SNVT_alarm [► 130]	Alarm status
FB_READ_089_SNVT_currency [► 131]	Currency
FB_READ_090_SNVT_file_pos [► 132]	File position
FB_READ_091_SNVT_muldiv [► 132]	Multiply/Divide
FB_READ_092_SNVT_obj_request [► 133]	Object request
FB_READ_093_SNVT_obj_status [► 134]	Object status
FB_READ_094_SNVT_preset [► 135]	Preset
FB_READ_095_SNVT_switch [► 136]	Switch
FB_READ_096_SNVT_trans_table [► 136]	Translation table
FB_READ_097_SNVT_override [► 137]	Override code
FB_READ_098_SNVT_pwr_fact [► 138]	Power factor
FB_READ_099_SNVT_pwr_fact_f [► 139]	Power factor (multiplier)
FB_READ_100_SNVT_density [► 140]	Density (kilograms/cubic meter)
FB_READ_101_SNVT_density_f [► 140]	Density (kilograms/cubic meter)
FB_READ_102_SNVT_rpm [► 141]	Angular velocity (revolutions/minute (RPM))
FB_READ_103_SNVT_hvac_emerg [► 142]	HVAC emergency mode
FB_READ_104_SNVT_angle_deg [► 143]	Angular distance(degrees)
FB_READ_105_SNVT_temp_p [► 143]	Temperature (degrees Celsius)
FB_READ_106_SNVT_temp_setpt [► 144]	Temperature
FB_READ_107_SNVT_time_sec [► 145]	Elapsed time (seconds)
FB_READ_108_SNVT_hvac_mode [► 146]	HVAC mode
FB_READ_109_SNVT_occupancy [► 146]	Occupancy
FB_READ_110_SNVT_area [► 147]	Area (square meters)
FB_READ_111_SNVT_hvac_overid [► 148]	HVAC override
FB_READ_112_SNVT_hvac_status [► 149]	HVAC status
FB_READ_113_SNVT_press_p [► 150]	Pressure
FB_READ_114_SNVT_address [► 150]	Neuron address
FB_READ_115_SNVT_scene [► 151]	Scene control
FB_READ_116_SNVT_scene_cfg [► 152]	Scene configuration
FB_READ_117_SNVT_setting [► 153]	Setting control
FB_READ_118_SNVT_evap_state [► 153]	Evaporator state
FB_READ_119_SNVT_therm_mode [► 154]	Thermostat mode
FB_READ_120_SNVT_defr_mode [► 155]	Defrost mode
FB_READ_121_SNVT_defr_term [► 156]	Defrost termination
FB_READ_122_SNVT_defr_state [► 157]	Defrost state
FB_READ_123_SNVT_time_min [► 157]	Elapsed time

POUs	Description
FB_READ_124_SNVT_time_hour [▶ 158]	Elapsed time
FB_READ_125_SNVT_ph [▶ 159]	Acidity (pH)
FB_READ_126_SNVT_ph_f [▶ 160]	Acidity (pH)
FB_READ_127_SNVT_chlr_status [▶ 161]	Chiller status
FB_READ_128_SNVT_tod_event [▶ 161]	Time of day event
FB_READ_129_SNVT_smo_obscur [▶ 162]	Smoke obscuration
FB_READ_130_SNVT_fire_test [▶ 163]	Fire test request
FB_READ_131_SNVT_temp_ror [▶ 164]	Temperature rate of change/rise
FB_READ_132_SNVT_fire_init [▶ 165]	Fire initiator type
FB_READ_133_SNVT_fire_indcte [▶ 165]	Fire indicator type
FB_READ_134_SNVT_time_zone [▶ 166]	Time zone descriptor
FB_READ_135_SNVT_earth_pos [▶ 167]	Earth position
FB_READ_136_SNVT_reg_val [▶ 168]	Register value
FB_READ_137_SNVT_reg_val_ts [▶ 169]	Register value
FB_READ_138_SNVT_volt_ac [▶ 169]	Voltage in alternating current
FB_READ_139_SNVT_amp_ac [▶ 170]	Amperage in alternating current
FB_READ_143_SNVT_turbidity [▶ 171]	Turbidity
FB_READ_144_SNVT_turbidity_f [▶ 172]	Turbidity
FB_READ_145_SNVT_hvac_type [▶ 172]	HVAC unit type
FB_READ_146_SNVT_elec_kwh_l [▶ 173]	Electric energy
FB_READ_147_SNVT_temp_diff_p [▶ 174]	Temp difference
FB_READ_148_SNVT_ctrl_req [▶ 175]	Control request
FB_READ_149_SNVT_ctrl_resp [▶ 176]	Control response
FB_READ_150_SNVT_ptz [▶ 176]	Camera PTZ
FB_READ_151_SNVT_privacyzone [▶ 177]	Privacy zone
FB_READ_152_SNVT_pos_ctrl [▶ 178]	Position control
FB_READ_153_SNVT_enthalpy [▶ 179]	Enthalpy (kiloJoules/kg)
FB_READ_154_SNVT_gfci_status [▶ 180]	GFCI status type
FB_READ_155_SNVT_motor_state [▶ 180]	Motor state
FB_READ_156_SNVT_pumpset_mn [▶ 181]	Pumpset
FB_READ_157_SNVT_ex_control [▶ 182]	Exclusive control
FB_READ_158_SNVT_pumpset_sn [▶ 183]	Pumpset sensor
FB_READ_159_SNVT_pump_sensor [▶ 184]	Pump sensor
FB_READ_160_SNVT_abs_humid [▶ 184]	Absolute humidity
FB_READ_161_SNVT_flow_p [▶ 185]	Flow volume
FB_READ_162_SNVT_dev_c_mode [▶ 186]	Device control mode
FB_READ_163_SNVT_valve_mode [▶ 187]	Valve mode
FB_READ_164_SNVT_alarm_2 [▶ 187]	Alarm status 2
FB_READ_165_SNVT_state_64 [▶ 188]	State vector
FB_READ_166_SNVT_nv_type [▶ 189]	Network variable type
FB_READ_168_SNVT_ent_opmode [▶ 190]	Entry operation mode
FB_READ_169_SNVT_ent_state [▶ 191]	Entry state
FB_READ_170_SNVT_ent_status [▶ 191]	Entry status

POUs	Description
FB_READ_171_SNVT_flow_dir [▶ 192]	Flow direction
FB_READ_172_SNVT_hvac_satsts [▶ 193]	HVAC saturation status
FB_READ_173_SNVT_dev_status [▶ 194]	Device status
FB_READ_174_SNVT_dev_fault [▶ 195]	Device fault states
FB_READ_175_SNVT_dev_maint [▶ 195]	Device maintenance
FB_READ_176_SNVT_date_event [▶ 196]	Date event
FB_READ_177_SNVT_sched_val [▶ 197]	Scheduler value
FB_READ_180_SNVT_sblnd_state [▶ 198]	Sunblind State
FB_READ_181_SNVT_rac_ctrl [▶ 199]	Rail-Audio Controller Control
FB_READ_182_SNVT_rac_req [▶ 199]	Rail-Audio Controller Request
FB_READ_183_SNVT_count_32 [▶ 200]	Absolute count
FB_READ_184_SNVT_clothes_w_c [▶ 201]	Clothes Washer Command
FB_READ_185_SNVT_clothes_w_m [▶ 202]	Clothes Washer-Management Status
FB_READ_186_SNVT_clothes_w_s [▶ 203]	Clothes Washer Status
FB_READ_187_SNVT_clothes_w_a [▶ 203]	Clothes Washer Alarm
FB_READ_188_SNVT_multiplier_s [▶ 204]	Value multiplier
FB_READ_189_SNVT_switch_2 [▶ 205]	Switch with scene and setting control.
FB_READ_190_SNVT_color_2 [▶ 206]	Color.
FB_READ_191_SNVT_log_status [▶ 207]	Log status (hundredths of second)
FB_READ_192_SNVT_time_stamp_p [▶ 207]	Precision timestamp.
FB_READ_193_SNVT_log_fx_request [▶ 208]	Log file transfer request.
FB_READ_194_SNVT_log_fx_status [▶ 209]	Log file transfer status.
FB_READ_195_SNVT_log_request [▶ 210]	Log status request.
FB_READ_196_SNVT_enthalpy_d [▶ 211]	Enthalpy difference (kJ/kg) .
FB_READ_197_SNVT_amp_ac_mil [▶ 211]	Electrical current (milliAmperes) .
FB_READ_198_SNVT_time_hour_p [▶ 212]	Time hour (hours) .
FB_READ_199_SNVT_lamp_status [▶ 213]	Lamp Status.
FB_READ_200_SNVT_environment [▶ 214]	Environment.
FB_READ_201_SNVT_geo_loc [▶ 214]	Geographic Location.

POUs/Send

POUs	Description
FB_SEND_001_SNVT_amp [▶ 215]	Electric current (Amperes)
FB_SEND_002_SNVT_amp_mil [▶ 217]	Electric current (milliAmperes)
FB_SEND_003_SNVT_angle [▶ 218]	Angular distance (radians)
FB_SEND_004_SNVT_angle_vel [▶ 219]	Angular velocity (radians/second) ⁹
FB_SEND_005_SNVT_btu_kilo [▶ 221]	Thermal energy (kilo-Btus)
FB_SEND_006_SNVT_btu_mega [▶ 222]	Thermal energy (mega-Btus)
FB_SEND_007_SNVT_char_ascii [▶ 223]	ASCII character (8-bit ASCII character)
FB_SEND_008_SNVT_count [▶ 224]	Absolute count (units)
FB_SEND_009_SNVT_count_inc [▶ 226]	Increment count (units (delta))
FB_SEND_011_SNVT_date_day [▶ 227]	Day of week (day names)
FB_SEND_013_SNVT_elec_kwh [▶ 228]	Electric energy (kiloWatt-hours)

POUs	Description
FB_SEND_014_SNVT_elec_whr [▶ 229]	Electric energy (Watt-hours)
FB_SEND_015_SNVT_flow [▶ 231]	Flow volume (liters/second)
FB_SEND_016_SNVT_flow_mil [▶ 232]	Flow volume (milliliters/second)
FB_SEND_017_SNVT_length [▶ 233]	Length (meters)
FB_SEND_018_SNVT_length_kilo [▶ 235]	Length (kilometers)
FB_SEND_019_SNVT_length_micr [▶ 236]	Length (micrometers (microns))
FB_SEND_020_SNVT_length_mil [▶ 237]	Length (millimeters)
FB_SEND_021_SNVT_lev_cont [▶ 239]	Continuous level (% of full level)
FB_SEND_023_SNVT_mass [▶ 240]	Mass (grams)
FB_SEND_024_SNVT_mass_kilo [▶ 241]	Mass (kilograms)
FB_SEND_025_SNVT_mass_mega [▶ 243]	Mass (metric tons)
FB_SEND_026_SNVT_mass_mil [▶ 244]	Mass (milligrams)
FB_SEND_027_SNVT_power [▶ 245]	Power (Watts)
FB_SEND_028_SNVT_power_kilo [▶ 247]	Power (kiloWatts)
FB_SEND_029_SNVT_ppm [▶ 248]	Concentration (ppm)
FB_SEND_030_SNVT_press [▶ 249]	Pressure (gauge) (kiloPascals)
FB_SEND_031_SNVT_res [▶ 251]	Electric resistance (Ohms)
FB_SEND_032_SNVT_res_kilo [▶ 252]	Electric resistance (kiloOhms)
FB_SEND_033_SNVT_sound_db [▶ 253]	Sound level (dB)
FB_SEND_034_SNVT_speed [▶ 255]	Linear velocity (meters/second)
FB_SEND_035_SNVT_speed_mil [▶ 256]	Linear velocity (meters/second)
FB_SEND_036_SNVT_str_asc [▶ 257]	Character string (30 characters max)
FB_SEND_037_SNVT_str_int [▶ 258]	Wide character string
FB_SEND_038_SNVT_telcom [▶ 260]	Telecomm states (telecomm state names)
FB_SEND_039_SNVT_temp [▶ 261]	Temperature (degrees Celsius)
FB_SEND_041_SNVT_vol [▶ 262]	Volume (liters)
FB_SEND_042_SNVT_vol_kilo [▶ 263]	Volume (kiloliters)
FB_SEND_043_SNVT_vol_mil [▶ 265]	Volume (milliliters)
FB_SEND_044_SNVT_volt [▶ 266]	Electric voltage (Volts)
FB_SEND_045_SNVT_volt_dbmv [▶ 267]	Electric voltage (dB microVolts)
FB_SEND_046_SNVT_volt_kilo [▶ 269]	Electric voltage (kiloVolts)
FB_SEND_047_SNVT_volt_mil [▶ 270]	Electric voltage (milliVolts)
FB_SEND_048_SNVT_amp_f [▶ 271]	Electric current (Amperes)
FB_SEND_049_SNVT_angle_f [▶ 273]	Angular distance (radians)
FB_SEND_050_SNVT_angle_vel_f [▶ 274]	Angular velocity (radians/second)
FB_SEND_051_SNVT_count_f [▶ 275]	Absolute count (units)
FB_SEND_052_SNVT_count_inc_f [▶ 277]	Increment count (units (delta))
FB_SEND_053_SNVT_flow_f [▶ 278]	Flow volume (liters/second)
FB_SEND_054_SNVT_length_f [▶ 279]	Length (meters)
FB_SEND_055_SNVT_lev_cont_f [▶ 281]	Continuous level (% of full scale)
FB_SEND_056_SNVT_mass_f [▶ 282]	Mass (grams)
FB_SEND_057_SNVT_power_f [▶ 283]	Power (Watts)
FB_SEND_058_SNVT_ppm_f [▶ 285]	Concentration (ppm)

POUs	Description
FB_SEND_059_SNVT_press_f [▶ 286]	Pressure (gauge) (Pascals)
FB_SEND_060_SNVT_res_f [▶ 287]	Electric resistance (Ohms)
FB_SEND_061_SNVT_sound_db_f [▶ 289]	Sound level (dBspl)
FB_SEND_062_SNVT_speed_f [▶ 290]	Linear velocity (meters/second)
FB_SEND_063_SNVT_temp_f [▶ 291]	Temperature (degrees Celsius)
FB_SEND_064_SNVT_time_f [▶ 293]	Elapsed time (seconds)
FB_SEND_065_SNVT_vol_f [▶ 294]	Volume (liters)
FB_SEND_066_SNVT_volt_f [▶ 295]	Electric voltage (Volts)
FB_SEND_067_SNVT_btu_f [▶ 297]	Thermal energy (Btus)
FB_SEND_068_SNVT_elec_whr_f [▶ 298]	Electric energy (Watt-hours)
FB_SEND_069_SNVT_config_src [▶ 299]	Configuration source (configuration source names)
FB_SEND_070_SNVT_color [▶ 300]	CIELAB color (L*,a*,b)
FB_SEND_071_SNVT_grammage [▶ 302]	Grammage (grams/sq meter)
FB_SEND_072_SNVT_grammage_f [▶ 303]	Grammage (grams/sq meter)
FB_SEND_073_SNVT_file_req [▶ 304]	File request
FB_SEND_074_SNVT_file_status [▶ 305]	File status
FB_SEND_075_SNVT_freq_f [▶ 307]	Frequency (Hertz)
FB_SEND_076_SNVT_freq_hz [▶ 308]	Frequency (Hertz)
FB_SEND_077_SNVT_freq_kilohz [▶ 309]	Frequency (kiloHertz)
FB_SEND_078_SNVT_freq_milhz [▶ 311]	Frequency (Hertz)
FB_SEND_079_SNVT_lux [▶ 312]	Illumination (lux)
FB_SEND_081_SNVT_lev_percent [▶ 313]	Percentage level (% of full level)
FB_SEND_082_SNVT_multiplier [▶ 315]	Multiplier (16-bit unsigned value)
FB_SEND_083_SNVT_state [▶ 316]	State vector (16 individual bit values)
FB_SEND_084_SNVT_time_stamp [▶ 317]	Time stamp (year, month, day, hour, minute, second)
FB_SEND_085_SNVT_zerospan [▶ 318]	Zero and span (Zero, span)
FB_SEND_086_SNVT_magcard [▶ 320]	ISO 7811 (40 hexadecimal digits)
FB_SEND_087_SNVT_elapsed_tm [▶ 321]	Elapsed time (day, hour, minute, second, millisecond)
FB_SEND_088_SNVT_alarm [▶ 322]	Alarm status
FB_SEND_089_SNVT_currency [▶ 323]	Currency (unit, magnitude, value)
FB_SEND_090_SNVT_file_pos [▶ 324]	File position (pointer, length)
FB_SEND_091_SNVT_muldiv [▶ 326]	Multiply/Divide (multiplier, divisor)
FB_SEND_092_SNVT_obj_request [▶ 327]	Object request (ID, request)
FB_SEND_093_SNVT_obj_status [▶ 328]	Object status (ID, status flags)
FB_SEND_094_SNVT_preset [▶ 329]	Preset (mode, data, time)
FB_SEND_095_SNVT_switch [▶ 330]	Switch (value, state)
FB_SEND_096_SNVT_trans_table [▶ 332]	Translation table (points, interpolation)
FB_SEND_097_SNVT_override [▶ 333]	Override code (override code names)
FB_SEND_098_SNVT_pwr_fact [▶ 334]	Power factor (multiplier)
FB_SEND_099_SNVT_pwr_fact_f [▶ 335]	Power factor (multiplier) .
FB_SEND_100_SNVT_density [▶ 337]	Density (kilograms/cubic meter)
FB_SEND_101_SNVT_density_f [▶ 338]	Density (kilograms/cubic meter) .

POUs	Description
FB_SEND_102_SNVT_rpm [▶ 339]	Angular velocity (revolutions/minute (RPM))
FB_SEND_103_SNVT_hvac_emerg [▶ 341]	HVAC emergency mode (emergency mode names)
FB_SEND_104_SNVT_angle_deg [▶ 342]	Angular distance (degrees)
FB_SEND_105_SNVT_temp_p [▶ 343]	Temperature (degrees Celsius)
FB_SEND_106_SNVT_temp_setpt [▶ 344]	Temperature (6 temperature values)
FB_SEND_107_SNVT_time_sec [▶ 346]	Elapsed time (seconds)
FB_SEND_108_SNVT_hvac_mode [▶ 347]	HVAC mode (HVAC mode names)
FB_SEND_109_SNVT_occupancy [▶ 348]	Occupancy (occupancy code names)
FB_SEND_110_SNVT_area [▶ 350]	Area (square meters)
FB_SEND_111_SNVT_hvac_overid [▶ 351]	HVAC override (state, pct, flow)
FB_SEND_112_SNVT_hvac_status [▶ 352]	HVAC status (mode, 5 percents, flag)
FB_SEND_113_SNVT_press_p [▶ 353]	Pressure (gauge) (Pascals)
FB_SEND_114_SNVT_address [▶ 355]	Neuron address (16-bit address value)
FB_SEND_115_SNVT_scene [▶ 356]	Scene control (function, scene number)
FB_SEND_116_SNVT_scene_cfg [▶ 357]	Scene configuration
FB_SEND_117_SNVT_setting [▶ 358]	Setting control (function, setting, rotation)
FB_SEND_118_SNVT_evap_state [▶ 360]	Evaporator state (evaporator state names)
FB_SEND_119_SNVT_therm_mode [▶ 361]	Thermostat mode (thermostat mode names)
FB_SEND_120_SNVT_defr_mode [▶ 362]	Defrost mode (defrost mode names)
FB_SEND_121_SNVT_defr_term [▶ 363]	Defrost termination (defrost termination names)
FB_SEND_122_SNVT_defr_state [▶ 364]	Defrost state (defrost state names)
FB_SEND_123_SNVT_time_min [▶ 366]	Elapsed time (minutes)
FB_SEND_124_SNVT_time_hour [▶ 367]	Elapsed time (hours)
FB_SEND_125_SNVT_ph [▶ 368]	Acidity (pH) . Ratio of concentration of ions
FB_SEND_126_SNVT_ph_f [▶ 370]	Acidity (pH) . Ratio of concentration of ions
FB_SEND_127_SNVT_chlr_status [▶ 371]	Chiller status (run mode, op mode, state bits)
FB_SEND_128_SNVT_tod_event [▶ 372]	Time of day event (current, next, time)
FB_SEND_129_SNVT_smo_obscur [▶ 373]	Smoke obscuration (percent obscuration)
FB_SEND_130_SNVT_fire_test [▶ 375]	Fire test request (fire test names)
FB_SEND_131_SNVT_temp_ror [▶ 376]	Temperature rate of change/rise (degrees Celsius/minute)
FB_SEND_132_SNVT_fire_init [▶ 377]	Fire initiator type (fire initiator type names)
FB_SEND_133_SNVT_fire_indcte [▶ 378]	Fire indicator type (fire indicator type names)
FB_SEND_134_SNVT_time_zone [▶ 380]	Time zone descriptor (offset, type, startDST, endDST)
FB_SEND_135_SNVT_earth_pos [▶ 381]	Earth position
FB_SEND_136_SNVT_reg_val [▶ 382]	Register value
FB_SEND_137_SNVT_reg_val_ts [▶ 383]	Register value
FB_SEND_138_SNVT_volt_ac [▶ 385]	Voltage in alternating current (volts AC)
FB_SEND_139_SNVT_amp_ac [▶ 386]	Amperage in alternating current (amperes AC)
FB_SEND_143_SNVT_turbidity [▶ 387]	Turbidity (nephelometric turbidity units)
FB_SEND_144_SNVT_turbidity_f [▶ 389]	Turbidity (nephelometric turbidity units)
FB_SEND_145_SNVT_hvac_type [▶ 390]	HVAC unit type (HVAC unit type names)
FB_SEND_146_SNVT_elec_kwh_l [▶ 391]	Electric energy (kiloWatt-hours) .

POUs	Description
FB_SEND_147_SNVTempDiffP [▶ 392]	Temp difference (degrees Celsius) .
FB_SEND_148_SNVCtrlReq [▶ 394]	Control request (receiver ID, sender ID, sender priority) .
FB_SEND_149_SNVCtrlResp [▶ 395]	Control response (status, sender, controller ID)
FB_SEND_150_SNVPTZ [▶ 396]	Camera PTZ (pan, pan speed, tilt, tilt speed, zoom, zoom speed) .
FB_SEND_151_SNVPrivacyZone [▶ 397]	Privacy zone (action, zone number, camera ID)
FB_SEND_152_SNVPosCtrl [▶ 398]	Position control
FB_SEND_153_SNVEnthalpy [▶ 400]	Enthalpy (kiloJoules/kg)
FB_SEND_154_SNVGFCIStatus [▶ 401]	GFCI status type
FB_SEND_155_SNVMotorState [▶ 402]	Motor state (motor state names)
FB_SEND_156_SNVpumpset_mn [▶ 403]	Pumpset
FB_SEND_157_SNVex_control [▶ 405]	Exclusive control (status, address)
FB_SEND_158_SNVpumpset_sn [▶ 406]	Pumpset sensor
FB_SEND_159_SNVpump_sensor [▶ 407]	Pump sensor (speed, temperature, status)
FB_SEND_160_SNVabs_humid [▶ 408]	Absolute humidity (gram/kilogram)
FB_SEND_161_SNVflow_p [▶ 410]	Flow volume (cubic meters/hour)
FB_SEND_162_SNVdev_c_mode [▶ 411]	Device control mode (device control mode names)
FB_SEND_163_SNVvalve_mode [▶ 412]	Valve mode (valve mode names)
FB_SEND_164_SNValarm_2 [▶ 413]	Alarm status 2
FB_SEND_165_SNVstate_64 [▶ 415]	State vector (64 individual bit values)
FB_SEND_166_SNVnv_type [▶ 416]	Network variable type
FB_SEND_168_SNVent_opmode [▶ 417]	Entry operation mode
FB_SEND_169_SNVent_state [▶ 418]	Entry state
FB_SEND_170_SNVent_status [▶ 419]	Entry status
FB_SEND_171_SNVflow_dir [▶ 421]	Flow direction (flow direction names)
FB_SEND_172_SNVhvac_satsts [▶ 422]	HVAC saturation status
FB_SEND_173_SNVdev_status [▶ 423]	Device status
FB_SEND_174_SNVdev_fault [▶ 424]	Device fault states
FB_SEND_175_SNVdev_maint [▶ 426]	Device maintenance
FB_SEND_176_SNVdate_event [▶ 427]	Date event
FB_SEND_177_SNVsched_val [▶ 428]	Scheduler value
FB_SEND_180_SNVsblnd_state [▶ 429]	Sunblind State
FB_SEND_181_SNVrac_ctrl [▶ 431]	Rail-Audio Controller Control
FB_SEND_182_SNVrac_req [▶ 432]	Rail-Audio Controller Request
FB_SEND_183_SNVcount_32 [▶ 433]	Absolute count. A 32-bit counter
FB_SEND_184_SNVclothes_w_c [▶ 434]	Clothes Washer Command
FB_SEND_185_SNVclothes_w_m [▶ 436]	Clothes Washer-Management Status
FB_SEND_186_SNVclothes_w_s [▶ 437]	Clothes Washer Status
FB_SEND_187_SNVclothes_w_a [▶ 438]	Clothes Washer Alarm
FB_SEND_188_SNVmultiplier_s [▶ 439]	Multiplier. Value multiplier
FB_SEND_189_SNVswitch_2 [▶ 441]	Switch with scene and setting control
FB_SEND_190_SNVcolor_2 [▶ 442]	Color.
FB_SEND_191_SNVlog_status [▶ 443]	Log status (hundredths of second)

POUs	Description
FB_SEND_192_SNVT_time_stamp_p [▶ 444]	Precision timestamp. (seconds)
FB_SEND_193_SNVT_log_fx_request [▶ 446]	Log file transfer request.
FB_SEND_194_SNVT_log_fx_status [▶ 447]	Log file transfer status.
FB_SEND_195_SNVT_log_request [▶ 448]	Log status request.
FB_SEND_196_SNVT_enthalpy_d [▶ 449]	Enthalpy difference (kJ/kg)
FB_SEND_197_SNVT_amp_ac_mil [▶ 451]	Electrical current (milliAmperes)
FB_SEND_198_SNVT_time_hour_p [▶ 452]	Time hour (hours)
FB_SEND_199_SNVT_lamp_status [▶ 453]	Lamp Status
FB_SEND_200_SNVT_environment [▶ 454]	Environment
FB_SEND_201_SNVT_geo_loc [▶ 456]	Geographic Location

Data types/Enums

Data types	Description
E_LON_alarm_type_t [▶ 488]	Used by: SNVT_alarm / SNVT_alarm_2
E_LON_appl_cwc_t [▶ 490]	Used by: SNVT_clothes_w_c / SNVT_clothes_w_s
E_LON_appl_cwp_t [▶ 490]	Used by: SNVT_clothes_w_c
E_LON_appl_cws_t [▶ 491]	Used by: SNVT_clothes_w_c / SNVT_clothes_w_s
E_LON_appl_rin_t [▶ 491]	Used by: SNVT_clothes_w_c
E_LON_boolean_t [▶ 492]	Used by: SCPTautoAnswer / SCPTcoolingResetEnable / SCPTdefrostHold / SCPTdefrostInternalSchedule / SCPTheatingResetEnable / SCPThighLimit1Enable / SCPThighLimit2Enable / SCPTlowLimit1Enable / SCPTlowLimit2Enable / SCPTscheduleInternal / SNVT_clothes_w_c / SNVT_pump_sensor / SNVT_pumpset_mn / SNVT_pumpset_sn
E_LON_calendar_type_t [▶ 492]	Used by: SNVT_time_zone
E_LON_cam_act_t [▶ 492]	Used by: SNVT_pos_ctrl
E_LON_cam_func_t [▶ 493]	Used by: SNVT_pos_ctrl
E_LON_chiller_t [▶ 493]	Used by: SNVT_chlr_status
E_LON_color_encoding_t [▶ 493]	Used by: SNVT_color_2
E_LON_config_source_t [▶ 494]	Used by: SNVT_config_src
E_LON_control_resp_t [▶ 494]	Used by: SNVT_ctrl_resp
E_LON_currency_t [▶ 494]	Used by: SNVT_currency
E_LON_days_of_week_t [▶ 497]	Used by: SCPTtimePeriod / SNVT_date_day / SNVT_time_zone
E_LON_defrost_mode_t [▶ 497]	Used by: SNVT_defr_mode
E_LON_defrost_state_t [▶ 498]	Used by: SNVT_defr_state
E_LON_defrost_term_t [▶ 498]	Used by: SNVT_defr_term
E_LON_device_c_mode_t [▶ 498]	Used by: SNVT_dev_c_mode
E_LON_device_select_t [▶ 499]	Used by: SNVT_dev_fault / SNVT_dev_maint / SNVT_dev_status
E_LON_discrete_levels_t [▶ 500]	Used by: SNVT_clothes_w_c / SNVT_lev_disc
E_LON_emerg_t [▶ 500]	Used by: SNVT_hvac_emerg
E_LON_ent_cmd_t [▶ 500]	Used by: SNVT_ent_state

Data types	Description
E_LON_ent_opmode_cmd_t [▶ 501]	Used by: SNVT_ent_opmode / SNVT_ent_status
E_LON_evap_t [▶ 502]	Used by: SNVT_evap_state
E_LON_ex_control_t [▶ 502]	Used by: SNVT_ex_control
E_LON_file_request_t [▶ 503]	Used by: SNVT_file_req
E_LON_file_status_t [▶ 503]	Used by: SNVT_file_status
E_LON_fire_indicator_t [▶ 504]	Used by: SNVT_fire_indcte
E_LON_fire_initiator_t [▶ 504]	Used by: SNVT_fire_init
E_LON_fire_test_t [▶ 505]	Used by: SNVT_fire_test
E_LON_flow_direction_t [▶ 506]	Used by: SNVT_flow_dir
E_LON_gfci_status_t [▶ 506]	Used by: SNVT_gfci_status
E_LON_hvac_hvt_t [▶ 506]	Used by: SNVT_hvac_type
E_LON_hvac_overid_t [▶ 507]	Used by: SNVT_hvac_overid
E_LON_hvac_t [▶ 509]	Used by: SNVT_chlr_status / SNVT_hvac_mode / SNVT_hvac_status
E_LON_learn_mode_t [▶ 510]	Used by: SNVT_preset
E_LON_log_status_t [▶ 510]	Used by: SCPTlogRecord / SNVT_log_status
E_LON_motor_state_t [▶ 511]	Used by: SNVT_motor_state / SNVT_pumpset_mn
E_LON_nv_type_category_t [▶ 511]	Used by: SNVT_nv_type
E_LON_object_request_t [▶ 512]	Used by: SNVT_obj_request
E_LON_occup_t [▶ 513]	Used by: SNVT_occupancy / SNVT_tod_event
E_LON_override_t [▶ 513]	Used by: SNVT_override
E_LON_pan_dir_t [▶ 514]	Used by: SNVT_ptz
E_LON_priority_level_t [▶ 514]	Used by: SNVT_alarm / SNVT_alarm_2 / SNVT_pumpset_mn
E_LON_privacyzone_t [▶ 515]	Used by: SNVT_privacyzone
E_LON_rail_audio_sensor_type_t [▶ 515]	Used by: SNVT_rac_ctrl / SNVT_rac_req
E_LON_rail_audio_type_t [▶ 516]	Used by: SNVT_rac_ctrl / SNVT_rac_req
E_LON_reg_val_unit_t [▶ 517]	Used by: SNVT_reg_val / SNVT_reg_val_ts
E_LON_sblnd_cmd_source_t [▶ 519]	Used by: SNVT_sblnd_state
E_LON_sblnd_error_t [▶ 520]	Used by: SNVT_sblnd_state
E_LON_scene_config_t [▶ 521]	Used by: SNVT_scene_cfg
E_LON_scene_t [▶ 522]	Used by: SNVT_scene
E_LON_sec_state_t [▶ 523]	Used by: SNVT_sec_state
E_LON_sec_status_t [▶ 524]	Used by: SNVT_sec_status
E_LON_setting_t [▶ 525]	Used by: SNVT_setting
E_LON_switch_state_t [▶ 525]	Used by: SNVT_switch_2
E_LON_telcom_states_t [▶ 527]	Used by: SNVT_telcom
E_LON_therm_mode_t [▶ 528]	Used by: SNVT_therm_mode
E_LON_tilt_dir_t [▶ 529]	Used by: SNVT_ptz
E_LON_unit_temp_t [▶ 529]	Used by: SNVT_pump_sensor
E_LON_valve_mode_t [▶ 529]	Used by: SNVT_valve_mode
E_LON_zoom_t [▶ 530]	Used by: SNVT_ptz

Data types/Hardware Types

Data types	Description
ST_LON_Parameter_IN_36B [▶ 555]	Process Image of the inputs
ST_LON_Parameter_OUT_36B [▶ 555]	Process Image of the outputs

Data types/LON_TYPES

Data types	Description
E_LON_Parameter_Datatypes [▶ 480]	Enums SNVT types

Data types/Structure/AuxiliaryStructure/SNVT_chlr_status

Data types	Description
ST_LON_chlr_state [▶ 533]	Used by: SNVT_chlr_status

Data types/Structure/AuxiliaryStructure/SNVT_clothes_w_c

Data types	Description
ST_LON_action [▶ 533]	Used by: SNVT_clothes_w_c
ST_LON_dry [▶ 533]	Used by: SNVT_clothes_w_c
ST_LON_duration [▶ 534]	Used by: SNVT_clothes_w_c
ST_LON_function [▶ 534]	Used by: SNVT_clothes_w_c
ST_LON_rinse [▶ 534]	Used by: SNVT_clothes_w_c
ST_LON_spin [▶ 534]	Used by: SNVT_clothes_w_c
ST_LON_wash [▶ 535]	Used by: SNVT_clothes_w_c

Data types/Structure/AuxiliaryStructure/SNVT_clothes_w_s

Data types	Description
ST_LON_alarm [▶ 535]	Used by: SNVT_clothes_w_s

Data types/Structure/AuxiliaryStructure/SNVT_color_2

Data types	Description
ST_LON_CIE1931_lumen [▶ 537]	Used by: SNVT_color_2
ST_LON_CIE1931_percent [▶ 537]	Used by: SNVT_color_2
ST_LON_color_value [▶ 537]	Used by: SNVT_color_2
ST_LON_RGB [▶ 538]	Used by: SNVT_color_2

Data types/Structure/AuxiliaryStructure/SNVT_ctrl_resp

Data types	Description
ST_LON_range [▶ 538]	Used by: SNVT_ctrl_resp
ST_LON_sender [▶ 538]	Used by: SNVT_ctrl_resp

Data types/Structure/AuxiliaryStructure/SNVT_dev_fault

Data types	Description
ST_LON_Dev_type1 [▶ 539]	Used by: SNVT_dev_fault
ST_LON_pump_ctrl1 [▶ 539]	Used by: SNVT_dev_fault
ST_LON_valve_pos1 [▶ 540]	Used by: SNVT_dev_fault

Data types/Structure/AuxiliaryStructure/SNVT_dev_maint

Data types	Description
ST_LON_Dev_type2 [▶ 541]	Used by: SNVT_dev_maint
ST_LON_pump_ctrl2 [▶ 541]	Used by: SNVT_dev_maint
ST_LON_valve_pos2 [▶ 541]	Used by: SNVT_dev_maint

Data types/Structure/AuxiliaryStructure/SNVT_dev_status

Data types	Description
ST_LON_Dev_type3 [▶ 542]	Used by: SNVT_dev_status
ST_LON_pump_ctrl3 [▶ 542]	Used by: SNVT_dev_status
ST_LON_valve_pos3 [▶ 543]	Used by: SNVT_dev_status

Data types/Structure/AuxiliaryStructure/SNVT_ex_control

Data types	Description
ST_LON_Control_device_addr [▶ 544]	Used by: SNVT_ex_control

Data types/Structure/AuxiliaryStructure/SNVT_file_req

Data types	Description
ST_LON_addrt [▶ 545]	Used by: SNVT_file_req
ST_LON_dest_address [▶ 545]	Used by: SNVT_file_req
ST_LON_gp [▶ 545]	Used by: SNVT_file_req
ST_LON_sn [▶ 545]	Used by: SNVT_file_req

Data types/Structure/AuxiliaryStructure/SNVT_file_status

Data types	Description
ST_LON_address [▶ 546]	Used by: FB_Write_Address_Table / FB_Read_Address_Table
ST_LON_adr [▶ 546]	Used by: SNVT_file_status
ST_LON_descriptor [▶ 547]	Used by: SNVT_file_status

Data types/Structure/AuxiliaryStructure/SNVT_lamp_status

Data types	Description
ST_LON_Alarm_actual [▶ 547]	Used by: SNVT_lamp_status
ST_LON_alarm_previous [▶ 549]	Used by: SNVT_lamp_status

Data types/Structure/AuxiliaryStructure/SNVT_pos_ctrl

Data types	Description
ST_LON_abspos [▶ 550]	Used by: SNVT_pos_ctrl
ST_LON_Value [▶ 551]	Used by: SNVT_pos_ctrl

Data types/Structure/AuxiliaryStructure/SNVT_rac_ctrl

Data types	Description
ST_LON_addr_dest [▶ 551]	Used by: SNVT_rac_req / SNVT_rac_ctrl
ST_LON_addr_init [▶ 551]	Used by: SNVT_rac_req / SNVT_rac_ctrl

Data types	Description
ST_LON_addr_talk [▶ 552]	Used by: SNVT_rac_ctrl
ST_LON_p2m [▶ 552]	Used by: SNVT_rac_req / SNVT_rac_ctrl
ST_LON_p2p [▶ 552]	Used by: SNVT_rac_req / SNVT_rac_ctrl

Data types/Structure/AuxiliaryStructure/SNVT_rac_req

Data types	Description
ST_LON_rac_req_addr_dest [▶ 553]	
ST_LON_rac_req_addr_init [▶ 553]	

Data types/Structure/AuxiliaryStructure/SNVT_switch_2

Data types	Description
ST_LON_setting [▶ 553]	Used by: SNVT_switch_2

Data types/Structure/AuxiliaryStructure/SNVT_time_zone

Data types	Description
ST_LON_end_DST [▶ 554]	Used by: SNVT_time_zone
ST_LON_M_end_DST [▶ 554]	Used by: SNVT_time_zone
ST_LON_M_start_DST [▶ 555]	Used by: SNVT_time_zone
ST_LON_start_DST [▶ 555]	Used by: SNVT_time_zone

Data types/Structure

Data types	Description
ST_KL6401 [▶ 558]	Structure for configuration
ST_LON_AddressTable [▶ 558]	Used by: FB_Write_Address_Table / FB_Read_Address_Table
ST_LON_ConfigTable [▶ 559]	Used by: FB_Write_Config_Table / FB_Read_Config_Table
ST_LON_DomainTable [▶ 559]	Used by: FB_Write_Domain_Table / FB_Read_Domain_Table
ST_LON_SNVT_alarm [▶ 560]	Used by: SNVT_alarm
ST_LON_SNVT_alarm_2 [▶ 560]	Used by: SNVT_alarm_2
ST_LON_SNVT_chlr_status [▶ 561]	Used by: SNVT_chlr_status
ST_LON_SNVT_clothes_w_a [▶ 561]	Used by: SNVT_clothes_w_a
ST_LON_SNVT_clothes_w_c [▶ 563]	Used by: SNVT_clothes_w_c
ST_LON_SNVT_clothes_w_m [▶ 563]	Used by: SNVT_clothes_w_m
ST_LON_SNVT_clothes_w_s [▶ 563]	Used by: SNVT_clothes_w_s
ST_LON_SNVT_color [▶ 564]	Used by: SNVT_color
ST_LON_SNVT_color_2 [▶ 564]	Used by: SNVT_color_2
ST_LON_SNVT_ctrl_req [▶ 564]	Used by: SNVT_ctrl_req
ST_LON_SNVT_ctrl_resp [▶ 564]	Used by: SNVT_ctrl_resp
ST_LON_SNVT_currency [▶ 565]	Used by: SNVT_currency
ST_LON_SNVT_date_event [▶ 565]	Used by: SNVT_date_event
ST_LON_SNVT_dev_fault [▶ 565]	Used by: SNVT_dev_fault
ST_LON_SNVT_dev_maint [▶ 566]	Used by: SNVT_dev_maint

Data types	Description
ST_LON_SNVT_dev_status [▶ 566]	Used by: SNVT_dev_status
ST_LON_SNVT_earth_pos [▶ 566]	Used by: SNVT_earth_pos
ST_LON_SNVT_elapsed_tm [▶ 566]	Used by: SNVT_elapsed_tm
ST_LON_SNVT_ent_status [▶ 567]	Used by: SNVT_ent_status
ST_LON_SNVT_environment [▶ 568]	Used by: SNVT_environment
ST_LON_SNVT_ex_control [▶ 569]	Used by: SNVT_ex_control
ST_LON_SNVT_file_pos [▶ 569]	Used by: SNVT_file_pos
ST_LON_SNVT_file_req [▶ 569]	Used by: SNVT_file_req
ST_LON_SNVT_file_status [▶ 570]	Used by: SNVT_file_status
ST_LON_SNVT_geo_loc [▶ 570]	Used by: SNVT_geo_loc
ST_LON_SNVT_hvac_overid [▶ 570]	Used by: SNVT_hvac_overid
ST_LON_SNVT_hvac_satsts [▶ 570]	Used by: SNVT_hvac_satsts
ST_LON_SNVT_hvac_status [▶ 571]	Used by: SNVT_hvac_status
ST_LON_SNVT_lamp_status [▶ 571]	Used by: SNVT_lamp_status
ST_LON_SNVT_log_fx_request [▶ 572]	Used by: SNVT_log_fx_request
ST_LON_SNVT_log_fx_status [▶ 572]	Used by: SNVT_log_fx_status
ST_LON_SNVT_log_status [▶ 572]	Used by: SNVT_log_status
ST_LON_SNVT_muldiv [▶ 573]	Used by: SNVT_muldiv
ST_LON_SNVT_nv_type [▶ 573]	Used by: SNVT_nv_type
ST_LON_SNVT_obj_request [▶ 574]	Used by: SNVT_obj_request
ST_LON_SNVT_obj_status [▶ 574]	Used by: SNVT_obj_status
ST_LON_SNVT_pos_ctrl [▶ 575]	Used by: SNVT_pos_ctrl
ST_LON_SNVT_preset [▶ 576]	Used by: SNVT_preset
ST_LON_SNVT_privacyzone [▶ 576]	Used by: SNVT_privacyzone
ST_LON_SNVT_ptz [▶ 576]	Used by: SNVT_ptz
ST_LON_SNVT_pump_sensor [▶ 577]	Used by: SNVT_pump_sensor
ST_LON_SNVT_pumpset_mn [▶ 577]	Used by: SNVT_pumpset_mn
ST_LON_SNVT_pumpset_sn [▶ 578]	Used by: SNVT_pumpset_sn
ST_LON_SNVT_rac_ctrl [▶ 579]	Used by: SNVT_rac_ctrl
ST_LON_SNVT_rac_req [▶ 579]	Used by: SNVT_rac_req
ST_LON_SNVT_reg_val [▶ 579]	Used by: SNVT_rac_val
ST_LON_SNVT_reg_val_ts [▶ 580]	Used by: SNVT_rac_val_ts
ST_LON_SNVT_sblnd_state [▶ 580]	Used by: SNVT_sblnd_state
ST_LON_SNVT_scene [▶ 580]	Used by: SNVT_scene
ST_LON_SNVT_scene_cfg [▶ 581]	Used by: SNVT_scene_cfg
ST_LON_SNVT_setting [▶ 581]	Used by: SNVT_setting
ST_LON_SNVT_str_int [▶ 581]	Used by: SNVT_str_int
ST_LON_SNVT_switch [▶ 582]	Used by: SNVT_switch
ST_LON_SNVT_switch_2 [▶ 582]	Used by: SNVT_switch_2
ST_LON_SNVT_temp_setpt [▶ 582]	Used by: SNVT_temp_setpt
ST_LON_SNVT_time_zone [▶ 583]	Used by: SNVT_time_zone
ST_LON_SNVT_tod_event [▶ 583]	Used by: SNVT_tod_event
ST_LON_SNVT_trans_table [▶ 583]	Used by: SNVT_trans_table

Data types	Description
ST_LON_SNVT_zerospan [▶ 584]	Used by: SNVT_zerospan
str_AddressTable [▶ 584]	adress table

Data types

Data types	Description
E_LON_ERROR [▶ 463]	Error messages
ST_ExplicitMessage [▶ 556]	Explicit message
ST_LON_Communication [▶ 557]	Connection between "FB_LON_KL6401" and the read / send function blocks.
ST_LON_ParameterInterface [▶ 557]	LON parameter interface
ST_LON_WriteData [▶ 557]	Structure of transmit buffer
ST_Prm [▶ 558]	Structure for configuration

Resources

Resources	Description
Globale Variablen_LON [▶ 585]	Default value for all send function blocks.

7.1 General information

● Installation

i Beginning with TwinCAT 2.11 Build 2229 (R3 and x64 Engineering), the libraries "TcLON.lib/.lb6/.lbox" will be installed automatically.

● Name of the library

i This library replaces the "TcKL6401.lib/.lb6/.lbox". Only the name of the libraries has changed. The modules are still compatible.

Hardware documentation in Beckhoff Information System: [KL6401 - LON Bus Terminal](#)

Further libraries are required

For PC systems (x86) and Embedded-PCs (CXxxxx):

- Standard.lib
- TcBase.lib
- TcSystem.lib
- TcUtilities.lib

For Bus Terminal Controller of BCxx00 series:

- Standard.lb6
- TcPlcUtilitiesBC.lb6
- PlcHelperBC.lb6
- PlcSystemBC.lb6

For Bus Terminal Controller of BCxx50, BCxx20 and BC9191 series:

- Standard.lbx
- TcBaseBCxx50.lbx
- TcSystemBCxx50.lbx

For Bus Terminal Controller of BXxx00 series:

- Standard.lbx

- TcBaseBX.lbx
- TcSystemBX.lbx

● Memory usage

I By linking the library PLC program memory is already consumed. Depending on the application program the remaining memory can not be sufficient.

7.2 POU's

POUs	Description
FB_LON_KL6401 [▶ 66]	Send / receiver block

POUs/Read

POUs	Description
FB_READ_001_SNVT_amp [▶ 67]	Electric current (Amperes)
FB_READ_002_SNVT_amp_mil [▶ 68]	Electric current (milliAmperes)
FB_READ_003_SNVT_angle [▶ 69]	Angular distance (radians)
FB_READ_004_SNVT_angle_vel [▶ 70]	Angular velocity (radians/second)
FB_READ_005_SNVT_btu_kilo [▶ 71]	Thermal energy (kilo-Btus)
FB_READ_006_SNVT_btu_mega [▶ 71]	Thermal energy (mega-Btus)
FB_READ_007_SNVT_char_ascii [▶ 72]	ASCII character (8-bit ASCII character)
FB_READ_008_SNVT_count [▶ 73]	Absolute count (units)
FB_READ_009_SNVT_count_inc [▶ 74]	Increment count (units (delta))
FB_READ_011_SNVT_date_day [▶ 74]	Day of week (day names)
FB_READ_013_SNVT_elec_kwh [▶ 75]	Electric energy (kiloWatt-hours)
FB_READ_014_SNVT_elec_whr [▶ 76]	Electric energy (Watt-hours)
FB_READ_015_SNVT_flow [▶ 77]	Flow volume (liters/second)
FB_READ_016_SNVT_flow_mil [▶ 78]	Flow volume (milliliters/second)
FB_READ_017_SNVT_length [▶ 78]	Length (meters)
FB_READ_018_SNVT_length_kilo [▶ 79]	Length (kilometers)
FB_READ_019_SNVT_length_micr [▶ 80]	Length(micrometers (microns))
FB_READ_020_SNVT_length_mil [▶ 81]	Length (millimeters)
FB_READ_021_SNVT_lev_cont [▶ 81]	Continuous level (% of full level)
FB_READ_023_SNVT_mass [▶ 82]	Mass (grams)
FB_READ_024_SNVT_mass_kilo [▶ 83]	Mass (kilograms)
FB_READ_025_SNVT_mass_mega [▶ 84]	Mass (metric tons)
FB_READ_026_SNVT_mass_mil [▶ 84]	Mass (milligrams)
FB_READ_027_SNVT_power [▶ 85]	Power (Watts)
FB_READ_028_SNVT_power_kilo [▶ 86]	Power (kiloWatts)
FB_READ_029_SNVT_ppm [▶ 87]	Concentration (ppm)
FB_READ_030_SNVT_press [▶ 87]	Pressure (gauge) (kiloPascals)
FB_READ_031_SNVT_res [▶ 88]	Electric resistance (Ohms)
FB_READ_032_SNVT_res_kilo [▶ 89]	Electric resistance (kiloOhms)
FB_READ_033_SNVT_sound_db [▶ 90]	Sound level (dB)
FB_READ_034_SNVT_speed [▶ 90]	Linear velocity (meters/second)
FB_READ_035_SNVT_speed_mil [▶ 91]	Linear velocity (meters/second)

POUs	Description
FB_READ_036_SNVT_str_asc [► 92]	Character string
FB_READ_037_SNVT_str_int [► 93]	Wide character string with locale code
FB_READ_038_SNVT_telcom [► 93]	Telecomm states
FB_READ_039_SNVT_temp [► 94]	Temperature (degrees Celsius)
FB_READ_041_SNVT_vol [► 95]	Volume (liters)
FB_READ_042_SNVT_vol_kilo [► 96]	Volume (kiloliters)
FB_READ_043_SNVT_vol_mil [► 97]	Volume (milliliters)
FB_READ_044_SNVT_volt [► 97]	Electric voltage (Volts)
FB_READ_045_SNVT_volt_dbmv [► 98]	Electric voltage (dB microVolts)
FB_READ_046_SNVT_volt_kilo [► 99]	Electric voltage (kiloVolts)
FB_READ_047_SNVT_volt_mil [► 100]	Electric voltage (milliVolts)
FB_READ_048_SNVT_amp_f [► 100]	Electric current (Amperes)
FB_READ_049_SNVT_angle_f [► 101]	Angular distance (radians)
FB_READ_050_SNVT_angle_vel_f [► 102]	Angular velocity (radians/second)
FB_READ_051_SNVT_count_f [► 103]	Absolute count (units)
FB_READ_052_SNVT_count_inc_f [► 103]	Increment count (units (delta))
FB_READ_053_SNVT_flow_f [► 104]	Flow volume (liters/second)
FB_READ_054_SNVT_length_f [► 105]	Length (meters)
FB_READ_055_SNVT_lev_cont_f [► 106]	Continuous level
FB_READ_056_SNVT_mass_f [► 106]	Mass (grams)
FB_READ_057_SNVT_power_f [► 107]	Power (Watts)
FB_READ_058_SNVT_ppm_f [► 108]	Concentration (ppm)
FB_READ_059_SNVT_press_f [► 109]	Pressure (gauge) (Pascals)
FB_READ_060_SNVT_res_f [► 109]	Electric resistance (Ohms)
FB_READ_061_SNVT_sound_db_f [► 110]	Sound level (dBspl)
FB_READ_062_SNVT_speed_f [► 111]	Linear velocity (meters/second)
FB_READ_063_SNVT_temp_f [► 112]	Temperature (degrees Celsius)
FB_READ_064_SNVT_time_f [► 112]	Elapsed time (seconds)
FB_READ_065_SNVT_vol_f [► 113]	Volume (liters)
FB_READ_066_SNVT_volt_f [► 114]	Electric voltage (Volts)
FB_READ_067_SNVT_btu_f [► 115]	Thermal energy (Btus)
FB_READ_068_SNVT_elec_whr_f [► 115]	Electric energy (Watt-hours)
FB_READ_069_SNVT_config_src [► 116]	Configuration source
FB_READ_070_SNVT_color [► 117]	CIELAB color (L*,a*,b)
FB_READ_071_SNVT_grammage [► 118]	Grammage (grams/sq meter)
FB_READ_072_SNVT_grammage_f [► 118]	Grammage (grams/sq meter)
FB_READ_073_SNVT_file_req [► 119]	File request
FB_READ_074_SNVT_file_status [► 120]	File status
FB_READ_075_SNVT_freq_f [► 121]	Frequency (Hertz)
FB_READ_076_SNVT_freq_hz [► 121]	Frequency (Hertz)
FB_READ_077_SNVT_freq_kilohz [► 122]	Frequency (kiloHertz)
FB_READ_078_SNVT_freq_milhz [► 123]	Frequency (Hertz)
FB_READ_079_SNVT_lux [► 124]	Illumination (lux)

POUs	Description
FB_READ_081_SNVT_lev_percent [▶ 124]	Percentage level (% of full level)
FB_READ_082_SNVT_multiplier [▶ 125]	Multiplier
FB_READ_083_SNVT_state [▶ 126]	State vector
FB_READ_084_SNVT_time_stamp [▶ 127]	Time stamp
FB_READ_085_SNVT_zerospan [▶ 128]	Zero and span
FB_READ_086_SNVT_magcard [▶ 128]	ISO 7811 (40 hexadecimal digits)
FB_READ_087_SNVT_elapsed_tm [▶ 129]	Elapsed time
FB_READ_088_SNVT_alarm [▶ 130]	Alarm status
FB_READ_089_SNVT_currency [▶ 131]	Currency
FB_READ_090_SNVT_file_pos [▶ 132]	File position
FB_READ_091_SNVT_muldiv [▶ 132]	Multiply/Divide
FB_READ_092_SNVT_obj_request [▶ 133]	Object request
FB_READ_093_SNVT_obj_status [▶ 134]	Object status
FB_READ_094_SNVT_preset [▶ 135]	Preset
FB_READ_095_SNVT_switch [▶ 136]	Switch
FB_READ_096_SNVT_trans_table [▶ 136]	Translation table
FB_READ_097_SNVT_override [▶ 137]	Override code
FB_READ_098_SNVT_pwr_fact [▶ 138]	Power factor
FB_READ_099_SNVT_pwr_fact_f [▶ 139]	Power factor (multiplier)
FB_READ_100_SNVT_density [▶ 140]	Density (kilograms/cubic meter)
FB_READ_101_SNVT_density_f [▶ 140]	Density (kilograms/cubic meter)
FB_READ_102_SNVT_rpm [▶ 141]	Angular velocity (revolutions/minute (RPM))
FB_READ_103_SNVT_hvac_emerg [▶ 142]	HVAC emergency mode
FB_READ_104_SNVT_angle_deg [▶ 143]	Angular distance(degrees)
FB_READ_105_SNVT_temp_p [▶ 143]	Temperature (degrees Celsius)
FB_READ_106_SNVT_temp_setpt [▶ 144]	Temperature
FB_READ_107_SNVT_time_sec [▶ 145]	Elapsed time (seconds)
FB_READ_108_SNVT_hvac_mode [▶ 146]	HVAC mode
FB_READ_109_SNVT_occupancy [▶ 146]	Occupancy
FB_READ_110_SNVT_area [▶ 147]	Area (square meters)
FB_READ_111_SNVT_hvac_overid [▶ 148]	HVAC override
FB_READ_112_SNVT_hvac_status [▶ 149]	HVAC status
FB_READ_113_SNVT_press_p [▶ 150]	Pressure
FB_READ_114_SNVT_address [▶ 150]	Neuron address
FB_READ_115_SNVT_scene [▶ 151]	Scene control
FB_READ_116_SNVT_scene_cfg [▶ 152]	Scene configuration
FB_READ_117_SNVT_setting [▶ 153]	Setting control
FB_READ_118_SNVT_evap_state [▶ 153]	Evaporator state
FB_READ_119_SNVT_therm_mode [▶ 154]	Thermostat mode
FB_READ_120_SNVT_defr_mode [▶ 155]	Defrost mode
FB_READ_121_SNVT_defr_term [▶ 156]	Defrost termination
FB_READ_122_SNVT_defr_state [▶ 157]	Defrost state
FB_READ_123_SNVT_time_min [▶ 157]	Elapsed time

POUs	Description
FB_READ_124_SNVT_time_hour [▶ 158]	Elapsed time
FB_READ_125_SNVT_ph [▶ 159]	Acidity (pH)
FB_READ_126_SNVT_ph_f [▶ 160]	Acidity (pH)
FB_READ_127_SNVT_chlr_status [▶ 161]	Chiller status
FB_READ_128_SNVT_tod_event [▶ 161]	Time of day event
FB_READ_129_SNVT_smo_obscur [▶ 162]	Smoke obscuration
FB_READ_130_SNVT_fire_test [▶ 163]	Fire test request
FB_READ_131_SNVT_temp_ror [▶ 164]	Temperature rate of change/rise
FB_READ_132_SNVT_fire_init [▶ 165]	Fire initiator type
FB_READ_133_SNVT_fire_indcte [▶ 165]	Fire indicator type
FB_READ_134_SNVT_time_zone [▶ 166]	Time zone descriptor
FB_READ_135_SNVT_earth_pos [▶ 167]	Earth position
FB_READ_136_SNVT_reg_val [▶ 168]	Register value
FB_READ_137_SNVT_reg_val_ts [▶ 169]	Register value
FB_READ_138_SNVT_volt_ac [▶ 169]	Voltage in alternating current
FB_READ_139_SNVT_amp_ac [▶ 170]	Amperage in alternating current
FB_READ_143_SNVT_turbidity [▶ 171]	Turbidity
FB_READ_144_SNVT_turbidity_f [▶ 172]	Turbidity
FB_READ_145_SNVT_hvac_type [▶ 172]	HVAC unit type
FB_READ_146_SNVT_elec_kwh_l [▶ 173]	Electric energy
FB_READ_147_SNVT_temp_diff_p [▶ 174]	Temp difference
FB_READ_148_SNVT_ctrl_req [▶ 175]	Control request
FB_READ_149_SNVT_ctrl_resp [▶ 176]	Control response
FB_READ_150_SNVT_ptz [▶ 176]	Camera PTZ
FB_READ_151_SNVT_privacyzone [▶ 177]	Privacy zone
FB_READ_152_SNVT_pos_ctrl [▶ 178]	Position control
FB_READ_153_SNVT_enthalpy [▶ 179]	Enthalpy (kiloJoules/kg)
FB_READ_154_SNVT_gfci_status [▶ 180]	GFCI status type
FB_READ_155_SNVT_motor_state [▶ 180]	Motor state
FB_READ_156_SNVT_pumpset_mn [▶ 181]	Pumpset
FB_READ_157_SNVT_ex_control [▶ 182]	Exclusive control
FB_READ_158_SNVT_pumpset_sn [▶ 183]	Pumpset sensor
FB_READ_159_SNVT_pump_sensor [▶ 184]	Pump sensor
FB_READ_160_SNVT_abs_humid [▶ 184]	Absolute humidity
FB_READ_161_SNVT_flow_p [▶ 185]	Flow volume
FB_READ_162_SNVT_dev_c_mode [▶ 186]	Device control mode
FB_READ_163_SNVT_valve_mode [▶ 187]	Valve mode
FB_READ_164_SNVT_alarm_2 [▶ 187]	Alarm status 2
FB_READ_165_SNVT_state_64 [▶ 188]	State vector
FB_READ_166_SNVT_nv_type [▶ 189]	Network variable type
FB_READ_168_SNVT_ent_opmode [▶ 190]	Entry operation mode
FB_READ_169_SNVT_ent_state [▶ 191]	Entry state
FB_READ_170_SNVT_ent_status [▶ 191]	Entry status

POUs	Description
FB_READ_171_SNVT_flow_dir [▶ 192]	Flow direction
FB_READ_172_SNVT_hvac_satsts [▶ 193]	HVAC saturation status
FB_READ_173_SNVT_dev_status [▶ 194]	Device status
FB_READ_174_SNVT_dev_fault [▶ 195]	Device fault states
FB_READ_175_SNVT_dev_maint [▶ 195]	Device maintenance
FB_READ_176_SNVT_date_event [▶ 196]	Date event
FB_READ_177_SNVT_sched_val [▶ 197]	Scheduler value
FB_READ_180_SNVT_sblnd_state [▶ 198]	Sunblind State
FB_READ_181_SNVT_rac_ctrl [▶ 199]	Rail-Audio Controller Control
FB_READ_182_SNVT_rac_req [▶ 199]	Rail-Audio Controller Request
FB_READ_183_SNVT_count_32 [▶ 200]	Absolute count
FB_READ_184_SNVT_clothes_w_c [▶ 201]	Clothes Washer Command
FB_READ_185_SNVT_clothes_w_m [▶ 202]	Clothes Washer-Management Status
FB_READ_186_SNVT_clothes_w_s [▶ 203]	Clothes Washer Status
FB_READ_187_SNVT_clothes_w_a [▶ 203]	Clothes Washer Alarm
FB_READ_188_SNVT_multiplier_s [▶ 204]	Value multiplier
FB_READ_189_SNVT_switch_2 [▶ 205]	Switch with scene and setting control.
FB_READ_190_SNVT_color_2 [▶ 206]	Color.
FB_READ_191_SNVT_log_status [▶ 207]	Log status (hundredths of second)
FB_READ_192_SNVT_time_stamp_p [▶ 207]	Precision timestamp.
FB_READ_193_SNVT_log_fx_request [▶ 208]	Log file transfer request.
FB_READ_194_SNVT_log_fx_status [▶ 209]	Log file transfer status.
FB_READ_195_SNVT_log_request [▶ 210]	Log status request.
FB_READ_196_SNVT_enthalpy_d [▶ 211]	Enthalpy difference (kJ/kg) .
FB_READ_197_SNVT_amp_ac_mil [▶ 211]	Electrical current (milliAmperes) .
FB_READ_198_SNVT_time_hour_p [▶ 212]	Time hour (hours) .
FB_READ_199_SNVT_lamp_status [▶ 213]	Lamp Status.
FB_READ_200_SNVT_environment [▶ 214]	Environment.
FB_READ_201_SNVT_geo_loc [▶ 214]	Geographic Location.

POUs/Send

POUs	Description
FB_SEND_001_SNVT_amp [▶ 215]	Electric current (Amperes)
FB_SEND_002_SNVT_amp_mil [▶ 217]	Electric current (milliAmperes)
FB_SEND_003_SNVT_angle [▶ 218]	Angular distance (radians)
FB_SEND_004_SNVT_angle_vel [▶ 219]	Angular velocity (radians/second) ⁹
FB_SEND_005_SNVT_btu_kilo [▶ 221]	Thermal energy (kilo-Btus)
FB_SEND_006_SNVT_btu_mega [▶ 222]	Thermal energy (mega-Btus)
FB_SEND_007_SNVT_char_ascii [▶ 223]	ASCII character (8-bit ASCII character)
FB_SEND_008_SNVT_count [▶ 224]	Absolute count (units)
FB_SEND_009_SNVT_count_inc [▶ 226]	Increment count (units (delta))
FB_SEND_011_SNVT_date_day [▶ 227]	Day of week (day names)
FB_SEND_013_SNVT_elec_kwh [▶ 228]	Electric energy (kiloWatt-hours)

POUs	Description
FB_SEND_014_SNVT_elec_whr [▶ 229]	Electric energy (Watt-hours)
FB_SEND_015_SNVT_flow [▶ 231]	Flow volume (liters/second)
FB_SEND_016_SNVT_flow_mil [▶ 232]	Flow volume (milliliters/second)
FB_SEND_017_SNVT_length [▶ 233]	Length (meters)
FB_SEND_018_SNVT_length_kilo [▶ 235]	Length (kilometers)
FB_SEND_019_SNVT_length_micr [▶ 236]	Length (micrometers (microns))
FB_SEND_020_SNVT_length_mil [▶ 237]	Length (millimeters)
FB_SEND_021_SNVT_lev_cont [▶ 239]	Continuous level (% of full level)
FB_SEND_023_SNVT_mass [▶ 240]	Mass (grams)
FB_SEND_024_SNVT_mass_kilo [▶ 241]	Mass (kilograms)
FB_SEND_025_SNVT_mass_mega [▶ 243]	Mass (metric tons)
FB_SEND_026_SNVT_mass_mil [▶ 244]	Mass (milligrams)
FB_SEND_027_SNVT_power [▶ 245]	Power (Watts)
FB_SEND_028_SNVT_power_kilo [▶ 247]	Power (kiloWatts)
FB_SEND_029_SNVT_ppm [▶ 248]	Concentration (ppm)
FB_SEND_030_SNVT_press [▶ 249]	Pressure (gauge) (kiloPascals)
FB_SEND_031_SNVT_res [▶ 251]	Electric resistance (Ohms)
FB_SEND_032_SNVT_res_kilo [▶ 252]	Electric resistance (kiloOhms)
FB_SEND_033_SNVT_sound_db [▶ 253]	Sound level (dB)
FB_SEND_034_SNVT_speed [▶ 255]	Linear velocity (meters/second)
FB_SEND_035_SNVT_speed_mil [▶ 256]	Linear velocity (meters/second)
FB_SEND_036_SNVT_str_asc [▶ 257]	Character string (30 characters max)
FB_SEND_037_SNVT_str_int [▶ 258]	Wide character string
FB_SEND_038_SNVT_telcom [▶ 260]	Telecomm states (telecomm state names)
FB_SEND_039_SNVT_temp [▶ 261]	Temperature (degrees Celsius)
FB_SEND_041_SNVT_vol [▶ 262]	Volume (liters)
FB_SEND_042_SNVT_vol_kilo [▶ 263]	Volume (kiloliters)
FB_SEND_043_SNVT_vol_mil [▶ 265]	Volume (milliliters)
FB_SEND_044_SNVT_volt [▶ 266]	Electric voltage (Volts)
FB_SEND_045_SNVT_volt_dbmv [▶ 267]	Electric voltage (dB microVolts)
FB_SEND_046_SNVT_volt_kilo [▶ 269]	Electric voltage (kiloVolts)
FB_SEND_047_SNVT_volt_mil [▶ 270]	Electric voltage (milliVolts)
FB_SEND_048_SNVT_amp_f [▶ 271]	Electric current (Amperes)
FB_SEND_049_SNVT_angle_f [▶ 273]	Angular distance (radians)
FB_SEND_050_SNVT_angle_vel_f [▶ 274]	Angular velocity (radians/second)
FB_SEND_051_SNVT_count_f [▶ 275]	Absolute count (units)
FB_SEND_052_SNVT_count_inc_f [▶ 277]	Increment count (units (delta))
FB_SEND_053_SNVT_flow_f [▶ 278]	Flow volume (liters/second)
FB_SEND_054_SNVT_length_f [▶ 279]	Length (meters)
FB_SEND_055_SNVT_lev_cont_f [▶ 281]	Continuous level (% of full scale)
FB_SEND_056_SNVT_mass_f [▶ 282]	Mass (grams)
FB_SEND_057_SNVT_power_f [▶ 283]	Power (Watts)
FB_SEND_058_SNVT_ppm_f [▶ 285]	Concentration (ppm)

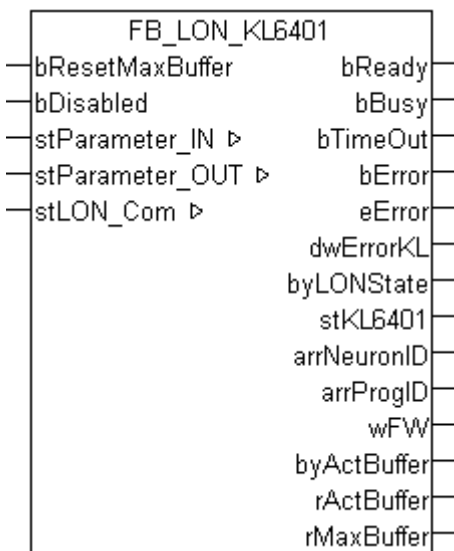
POUs	Description
FB_SEND_059_SNVT_press_f [▶ 286]	Pressure (gauge) (Pascals)
FB_SEND_060_SNVT_res_f [▶ 287]	Electric resistance (Ohms)
FB_SEND_061_SNVT_sound_db_f [▶ 289]	Sound level (dBspl)
FB_SEND_062_SNVT_speed_f [▶ 290]	Linear velocity (meters/second)
FB_SEND_063_SNVT_temp_f [▶ 291]	Temperature (degrees Celsius)
FB_SEND_064_SNVT_time_f [▶ 293]	Elapsed time (seconds)
FB_SEND_065_SNVT_vol_f [▶ 294]	Volume (liters)
FB_SEND_066_SNVT_volt_f [▶ 295]	Electric voltage (Volts)
FB_SEND_067_SNVT_btu_f [▶ 297]	Thermal energy (Btus)
FB_SEND_068_SNVT_elec_whr_f [▶ 298]	Electric energy (Watt-hours)
FB_SEND_069_SNVT_config_src [▶ 299]	Configuration source (configuration source names)
FB_SEND_070_SNVT_color [▶ 300]	CIELAB color (L*,a*,b)
FB_SEND_071_SNVT_grammage [▶ 302]	Grammage (grams/sq meter)
FB_SEND_072_SNVT_grammage_f [▶ 303]	Grammage (grams/sq meter)
FB_SEND_073_SNVT_file_req [▶ 304]	File request
FB_SEND_074_SNVT_file_status [▶ 305]	File status
FB_SEND_075_SNVT_freq_f [▶ 307]	Frequency (Hertz)
FB_SEND_076_SNVT_freq_hz [▶ 308]	Frequency (Hertz)
FB_SEND_077_SNVT_freq_kilohz [▶ 309]	Frequency (kiloHertz)
FB_SEND_078_SNVT_freq_milhz [▶ 311]	Frequency (Hertz)
FB_SEND_079_SNVT_lux [▶ 312]	Illumination (lux)
FB_SEND_081_SNVT_lev_percent [▶ 313]	Percentage level (% of full level)
FB_SEND_082_SNVT_multiplier [▶ 315]	Multiplier (16-bit unsigned value)
FB_SEND_083_SNVT_state [▶ 316]	State vector (16 individual bit values)
FB_SEND_084_SNVT_time_stamp [▶ 317]	Time stamp (year, month, day, hour, minute, second)
FB_SEND_085_SNVT_zerospan [▶ 318]	Zero and span (Zero, span)
FB_SEND_086_SNVT_magcard [▶ 320]	ISO 7811 (40 hexadecimal digits)
FB_SEND_087_SNVT_elapsed_tm [▶ 321]	Elapsed time (day, hour, minute, second, millisecond)
FB_SEND_088_SNVT_alarm [▶ 322]	Alarm status
FB_SEND_089_SNVT_currency [▶ 323]	Currency (unit, magnitude, value)
FB_SEND_090_SNVT_file_pos [▶ 324]	File position (pointer, length)
FB_SEND_091_SNVT_muldiv [▶ 326]	Multiply/Divide (multiplier, divisor)
FB_SEND_092_SNVT_obj_request [▶ 327]	Object request (ID, request)
FB_SEND_093_SNVT_obj_status [▶ 328]	Object status (ID, status flags)
FB_SEND_094_SNVT_preset [▶ 329]	Preset (mode, data, time)
FB_SEND_095_SNVT_switch [▶ 330]	Switch (value, state)
FB_SEND_096_SNVT_trans_table [▶ 332]	Translation table (points, interpolation)
FB_SEND_097_SNVT_override [▶ 333]	Override code (override code names)
FB_SEND_098_SNVT_pwr_fact [▶ 334]	Power factor (multiplier)
FB_SEND_099_SNVT_pwr_fact_f [▶ 335]	Power factor (multiplier) .
FB_SEND_100_SNVT_density [▶ 337]	Density (kilograms/cubic meter)
FB_SEND_101_SNVT_density_f [▶ 338]	Density (kilograms/cubic meter) .

POUs	Description
FB_SEND_102_SNVT_rpm [▶ 339]	Angular velocity (revolutions/minute (RPM))
FB_SEND_103_SNVT_hvac_emerg [▶ 341]	HVAC emergency mode (emergency mode names)
FB_SEND_104_SNVT_angle_deg [▶ 342]	Angular distance (degrees)
FB_SEND_105_SNVT_temp_p [▶ 343]	Temperature (degrees Celsius)
FB_SEND_106_SNVT_temp_setpt [▶ 344]	Temperature (6 temperature values)
FB_SEND_107_SNVT_time_sec [▶ 346]	Elapsed time (seconds)
FB_SEND_108_SNVT_hvac_mode [▶ 347]	HVAC mode (HVAC mode names)
FB_SEND_109_SNVT_occupancy [▶ 348]	Occupancy (occupancy code names)
FB_SEND_110_SNVT_area [▶ 350]	Area (square meters)
FB_SEND_111_SNVT_hvac_overid [▶ 351]	HVAC override (state, pct, flow)
FB_SEND_112_SNVT_hvac_status [▶ 352]	HVAC status (mode, 5 percents, flag)
FB_SEND_113_SNVT_press_p [▶ 353]	Pressure (gauge) (Pascals)
FB_SEND_114_SNVT_address [▶ 355]	Neuron address (16-bit address value)
FB_SEND_115_SNVT_scene [▶ 356]	Scene control (function, scene number)
FB_SEND_116_SNVT_scene_cfg [▶ 357]	Scene configuration
FB_SEND_117_SNVT_setting [▶ 358]	Setting control (function, setting, rotation)
FB_SEND_118_SNVT_evap_state [▶ 360]	Evaporator state (evaporator state names)
FB_SEND_119_SNVT_therm_mode [▶ 361]	Thermostat mode (thermostat mode names)
FB_SEND_120_SNVT_defr_mode [▶ 362]	Defrost mode (defrost mode names)
FB_SEND_121_SNVT_defr_term [▶ 363]	Defrost termination (defrost termination names)
FB_SEND_122_SNVT_defr_state [▶ 364]	Defrost state (defrost state names)
FB_SEND_123_SNVT_time_min [▶ 366]	Elapsed time (minutes)
FB_SEND_124_SNVT_time_hour [▶ 367]	Elapsed time (hours)
FB_SEND_125_SNVT_ph [▶ 368]	Acidity (pH) . Ratio of concentration of ions
FB_SEND_126_SNVT_ph_f [▶ 370]	Acidity (pH) . Ratio of concentration of ions
FB_SEND_127_SNVT_chlr_status [▶ 371]	Chiller status (run mode, op mode, state bits)
FB_SEND_128_SNVT_tod_event [▶ 372]	Time of day event (current, next, time)
FB_SEND_129_SNVT_smo_obscur [▶ 373]	Smoke obscuration (percent obscuration)
FB_SEND_130_SNVT_fire_test [▶ 375]	Fire test request (fire test names)
FB_SEND_131_SNVT_temp_ror [▶ 376]	Temperature rate of change/rise (degrees Celsius/minute)
FB_SEND_132_SNVT_fire_init [▶ 377]	Fire initiator type (fire initiator type names)
FB_SEND_133_SNVT_fire_indcte [▶ 378]	Fire indicator type (fire indicator type names)
FB_SEND_134_SNVT_time_zone [▶ 380]	Time zone descriptor (offset, type, startDST, endDST)
FB_SEND_135_SNVT_earth_pos [▶ 381]	Earth position
FB_SEND_136_SNVT_reg_val [▶ 382]	Register value
FB_SEND_137_SNVT_reg_val_ts [▶ 383]	Register value
FB_SEND_138_SNVT_volt_ac [▶ 385]	Voltage in alternating current (volts AC)
FB_SEND_139_SNVT_amp_ac [▶ 386]	Amperage in alternating current (amperes AC)
FB_SEND_143_SNVT_turbidity [▶ 387]	Turbidity (nephelometric turbidity units)
FB_SEND_144_SNVT_turbidity_f [▶ 389]	Turbidity (nephelometric turbidity units)
FB_SEND_145_SNVT_hvac_type [▶ 390]	HVAC unit type (HVAC unit type names)
FB_SEND_146_SNVT_elec_kwh_l [▶ 391]	Electric energy (kiloWatt-hours) .

POUs	Description
FB_SEND_147_SNVT_temp_diff_p [▶ 392]	Temp difference (degrees Celsius) .
FB_SEND_148_SNVT_ctrl_req [▶ 394]	Control request (receiver ID, sender ID, sender priority) .
FB_SEND_149_SNVT_ctrl_resp [▶ 395]	Control response (status, sender, controller ID)
FB_SEND_150_SNVT_ptz [▶ 396]	Camera PTZ (pan, pan speed, tilt, tilt speed, zoom, zoom speed) .
FB_SEND_151_SNVT_privacyzone [▶ 397]	Privacy zone (action, zone number, camera ID)
FB_SEND_152_SNVT_pos_ctrl [▶ 398]	Position control
FB_SEND_153_SNVT_enthalpy [▶ 400]	Enthalpy (kiloJoules/kg)
FB_SEND_154_SNVT_gfci_status [▶ 401]	GFCI status type
FB_SEND_155_SNVT_motor_state [▶ 402]	Motor state (motor state names)
FB_SEND_156_SNVT_pumpset_mn [▶ 403]	Pumpset
FB_SEND_157_SNVT_ex_control [▶ 405]	Exclusive control (status, address)
FB_SEND_158_SNVT_pumpset_sn [▶ 406]	Pumpset sensor
FB_SEND_159_SNVT_pump_sensor [▶ 407]	Pump sensor (speed, temperature, status)
FB_SEND_160_SNVT_abs_humid [▶ 408]	Absolute humidity (gram/kilogram)
FB_SEND_161_SNVT_flow_p [▶ 410]	Flow volume (cubic meters/hour)
FB_SEND_162_SNVT_dev_c_mode [▶ 411]	Device control mode (device control mode names)
FB_SEND_163_SNVT_valve_mode [▶ 412]	Valve mode (valve mode names)
FB_SEND_164_SNVT_alarm_2 [▶ 413]	Alarm status 2
FB_SEND_165_SNVT_state_64 [▶ 415]	State vector (64 individual bit values)
FB_SEND_166_SNVT_nv_type [▶ 416]	Network variable type
FB_SEND_168_SNVT_ent_opmode [▶ 417]	Entry operation mode
FB_SEND_169_SNVT_ent_state [▶ 418]	Entry state
FB_SEND_170_SNVT_ent_status [▶ 419]	Entry status
FB_SEND_171_SNVT_flow_dir [▶ 421]	Flow direction (flow direction names)
FB_SEND_172_SNVT_hvac_satsts [▶ 422]	HVAC saturation status
FB_SEND_173_SNVT_dev_status [▶ 423]	Device status
FB_SEND_174_SNVT_dev_fault [▶ 424]	Device fault states
FB_SEND_175_SNVT_dev_maint [▶ 426]	Device maintenance
FB_SEND_176_SNVT_date_event [▶ 427]	Date event
FB_SEND_177_SNVT_sched_val [▶ 428]	Scheduler value
FB_SEND_180_SNVT_sblnd_state [▶ 429]	Sunblind State
FB_SEND_181_SNVT_rac_ctrl [▶ 431]	Rail-Audio Controller Control
FB_SEND_182_SNVT_rac_req [▶ 432]	Rail-Audio Controller Request
FB_SEND_183_SNVT_count_32 [▶ 433]	Absolute count. A 32-bit counter
FB_SEND_184_SNVT_clothes_w_c [▶ 434]	Clothes Washer Command
FB_SEND_185_SNVT_clothes_w_m [▶ 436]	Clothes Washer-Management Status
FB_SEND_186_SNVT_clothes_w_s [▶ 437]	Clothes Washer Status
FB_SEND_187_SNVT_clothes_w_a [▶ 438]	Clothes Washer Alarm
FB_SEND_188_SNVT_multiplier_s [▶ 439]	Multiplier. Value multiplier
FB_SEND_189_SNVT_switch_2 [▶ 441]	Switch with scene and setting control
FB_SEND_190_SNVT_color_2 [▶ 442]	Color.
FB_SEND_191_SNVT_log_status [▶ 443]	Log status (hundredths of second)

POUs	Description
FB_SEND_192_SNVT_time_stamp_p [▶ 444]	Precision timestamp. (seconds)
FB_SEND_193_SNVT_log_fx_request [▶ 446]	Log file transfer request.
FB_SEND_194_SNVT_log_fx_status [▶ 447]	Log file transfer status.
FB_SEND_195_SNVT_log_request [▶ 448]	Log status request.
FB_SEND_196_SNVT_enthalpy_d [▶ 449]	Enthalpy difference (kJ/kg)
FB_SEND_197_SNVT_amp_ac_mil [▶ 451]	Electrical current (milliAmperes)
FB_SEND_198_SNVT_time_hour_p [▶ 452]	Time hour (hours)
FB_SEND_199_SNVT_lamp_status [▶ 453]	Lamp Status
FB_SEND_200_SNVT_environment [▶ 454]	Environment
FB_SEND_201_SNVT_geo_loc [▶ 456]	Geographic Location

7.2.1 FB_LON_KL6401



This function block is used for sending / receiving LON SNVTs via Bus Terminal KL6401. An instance of this function block is required for each terminal. One instance can send or receive up to 62 SNVTs.

The FB must be bound to the send/receive blocks via the VAR_IN_OUT structure stLON_Com [▶ 557].



Restrictions

- Only one call per instance
- Call must be made once per PLC cycle
- Instance must be called in the same PLC task as the send and receive blocks assigned to it

VAR_INPUT

```

bResetMaxBuffer : BOOL;
bDisabled       : BOOL := FALSE;
  
```

bResetMaxBuffer: Deletes the value *rMaxBuffer* for the maximum utilization of the send buffer.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bReady          : BOOL;
bBusy           : BOOL;
bTimeOut        : BOOL;
bError          : BOOL;
eError          : E_LON_ERROR;
  
```

```
dwErrorKL      : DWORD;
byLONState    : BYTE;
stKL6401      : ST_KL6401;
arrNeuronID   : ARRAY [0..5] OF BYTE;
arrProgID     : ARRAY [0..7] OF BYTE;
wFW           : WORD;
byActBuffer   : BYTE;
rActBuffer    : REAL;
rMaxBuffer    : REAL;
```

bReady: Initialization is ready.

bBusy: If the sending of data is in process, this output will be TRUE.

bTimeout: Timeout while Init.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

byLONState: LON state.

stKL6401: Structure for configuration (for future applications) (see [ST_KL6401 \[▶ 558\]](#)).

arrNeuronID: Displaying the Neuron-ID of the terminal (will not be supported until FW 4C).

arrProgID: Displaying the Program-ID of the terminal (will not be supported until FW 4C).

wFW: Displaying the Firmware-version of the terminal (will not be supported until FW 4C).

byActBuffer: Number of orders in the send buffer.

rActBuffer: Current utilization of the send buffer in percent.

rMaxBuffer: Maximum utilization of the send buffer in percent. The value can be deleted using the input variable *bResetMaxBuffer*.

VAR_IN_OUT

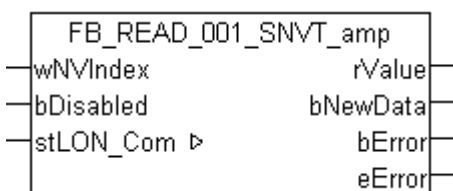
```
stParameter_IN : ST_LON_Parameter_IN_36B;
stParameter_OUT : ST_LON_Parameter_OUT_36B;
stLON_Com      : ST_LON_Communication;
```

stParameter_IN: Input variable of the terminal (see [ST_LON_Parameter_IN_36B \[▶ 555\]](#)).

stParameter_OUT: Output variable of the terminal (see [ST_LON_Parameter_OUT_36B \[▶ 555\]](#)).

stLON_Com: This structure connects the function block with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)). The send and receive is done by this function block. The send/receive function blocks have only the task to prepare/evaluate the data.

7.2.2 FB_READ_001_SNVT_amp



This function-block receives the following LON-input-variable (nvi):

SNVT name: SNVT_amp.

SNVT number: 001.

SNVT description: Electric current (ampere).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: -3276.8 / Max: 3276.7.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

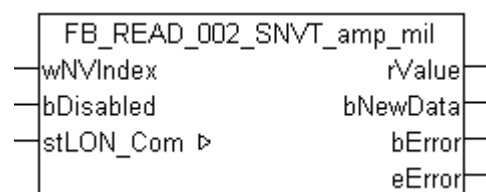
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.3 FB_READ_002_SNVT_amp_mil



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_amp_mil.

SNVT number: 002.

SNVT description: Electric current (milliampere).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue      : REAL;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

rValue: Min: -3276.8 / Max: 3276.7.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

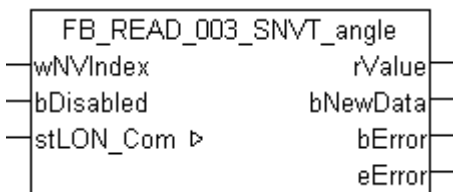
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [[▶ 66](#)] with the send/receive function blocks (see [ST_LON_Communication](#) [[▶ 557](#)]).

7.2.4 FB_READ_003_SNVT_angle



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_angle.

SNVT number: 003.

SNVT description: Angular distance (radian).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue      : REAL;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

rValue: Min: 0 / Max: 65.535.

bNewData: Is TRUE for one cycle once new data were received.

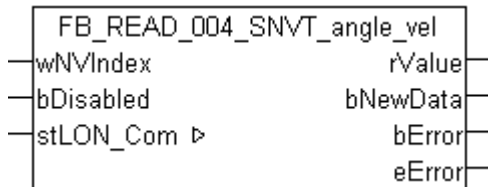
bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [► 66] with the send/receive function blocks (see [ST_LON_Communication](#) [► 557]).

7.2.5 FB_READ_004_SNVT_angle_vel

This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_angle_vel.

SNVT number: 004.

SNVT description: Angular velocity (radian/second).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: -3276.8 / Max: 3276.7.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

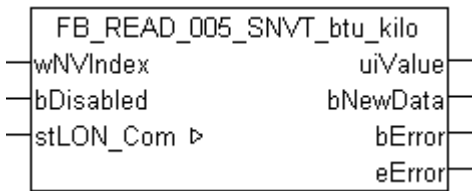
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [► 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [► 66] with the send/receive function blocks (see [ST_LON_Communication](#) [► 557]).

7.2.6 FB_READ_005_SNVT_btu_kilo



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_btu_kilo.

SNVT number: 005.

SNVT description: Thermal energy (kilo BTU).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
uiValue       : UINT;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

uiValue: Min: 0 / Max: 65535.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

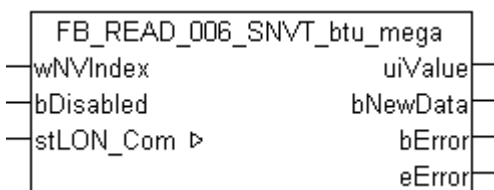
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.7 FB_READ_006_SNVT_btu_mega



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_btu_mega.

SNVT number: 006.

SNVT description: Thermal energy (mega BTU).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
uiValue       : UINT;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

uiValue: Min: 0 / Max: 65535.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

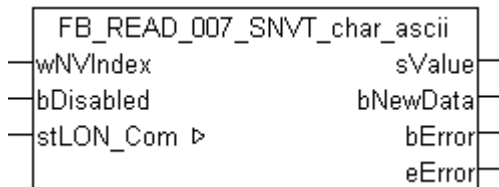
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.8 FB_READ_007_SNVT_char_ascii



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_char_ascii.

SNVT number: 007.

SNVT description: ASCII character (8-bit ASCII character).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
sValue        : STRING(1);
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```


sValue: ASCII character (8-bit ASCII character).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

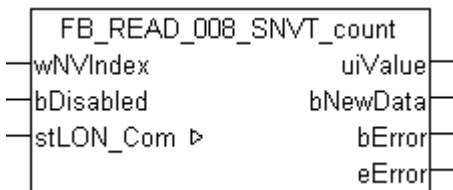
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.9 FB_READ_008_SNVT_count



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_count.

SNVT number: 008.

SNVT description: Absolute count (unit).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
uiValue : UINT;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

uiValue: Min: 0 / Max: 65535.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

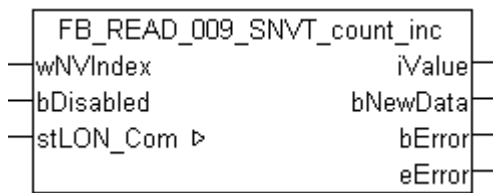
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.10 FB_READ_009_SNVT_count_inc



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_count_inc.

SNVT number: 009.

SNVT description: Increment count (unit (delta)).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
iValue       : INT;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

iValue: Min: -32768 / Max: 32767.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

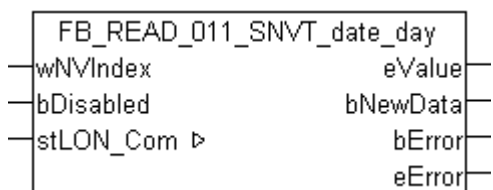
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.11 FB_READ_011_SNVT_date_day



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_date_day.

SNVT number: 011.

SNVT description: Day of week (day names).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
eValue       : E_LON_days_of_week_t;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

eValue: Enum to be received, (see [E_LON_days_of_week_t \[▶ 497\]](#)).

bNewData: Becomes TRUE for 1 cycle when the function block has received data.

bError: The output becomes TRUE as soon as an error occurs. This error is described via the *eError* variable.

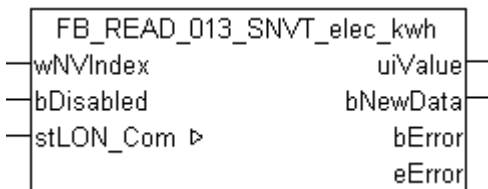
eError: The output outputs an error code in the event of an error (see [E_LON_ERROR \[▶ 463\]](#)). *bError* goes TRUE at the same time.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.12 FB_READ_013_SNVT_elec_kwh



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_elec_kwh.

SNVT number: 013.

SNVT description: Electric energy (kilowatt hour).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
uiValue      : UINT;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

uiValue: Min: 0 / Max: 65535.

bNewData: Is TRUE for one cycle once new data were received.

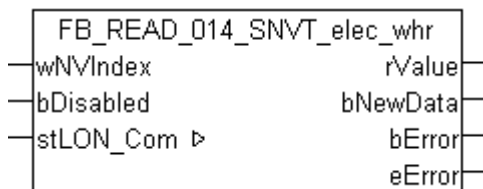
bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.13 FB_READ_014_SNVT_elec_whr

This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_elec_whr.

SNVT number: 014.

SNVT description: Electric energy (watt-hour).

VAR_INPUT

```
wNVIndex     : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 6553.5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

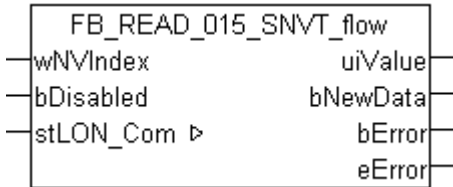
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [[▶ 66](#)] with the send/receive function blocks (see [ST_LON_Communication](#) [[▶ 557](#)]).

7.2.14 FB_READ_015_SNVT_flow



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_flow.

SNVT number: 015.

SNVT description: Flow volume (liter/second).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
uiValue : UINT;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

uiValue: Min: 0 / Max: 65535.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

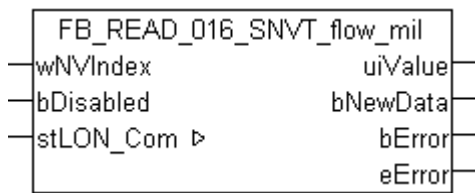
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [[▶ 463](#)]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [[▶ 66](#)] with the send/receive function blocks (see [ST_LON_Communication](#) [[▶ 557](#)]).

7.2.15 FB_READ_016_SNVT_flow_mil



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_flow_mil.

SNVT number: 016.

SNVT description: Flow volume (milliliter/second).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
uiValue       : UINT;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

uiValue: Min: 0 / Max: 65535.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

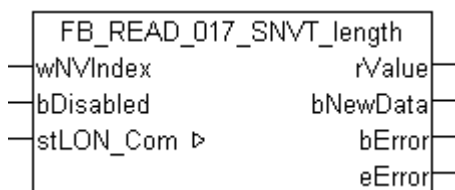
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.16 FB_READ_017_SNVT_length



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_length.

SNVT number: 017.

SNVT description: Length (meter).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue        : REAL;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

rValue: Min: 0 / Max: 6553.5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

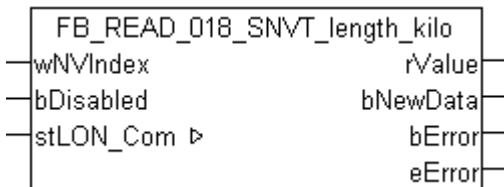
eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.17 FB_READ_018_SNVT_length_kilo



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_length_kilo.

SNVT number: 018.

SNVT description: Length (kilometer).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue        : REAL;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

rValue: Min: 0 / Max: 6553.5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

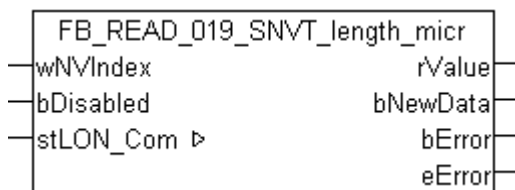
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.18 FB_READ_019_SNVT_length_micr



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_length_micr.

SNVT number: 019.

SNVT description: Length (micrometer).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

rValue: Min: 0 / Max: 6553.5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

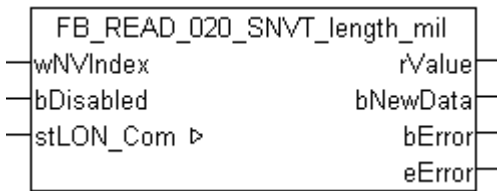
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.19 FB_READ_020_SNVT_length_mil



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_length_mil.

SNVT number: 020.

SNVT description: Length (millimeter).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 6553.5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

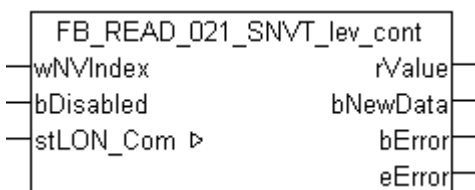
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.20 FB_READ_021_SNVT_lev_cont



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_lev_cont.

SNVT number: 021.

SNVT description: Continuous level (% of full level).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 100.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

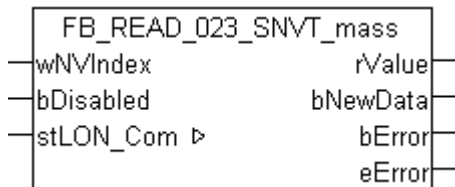
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.21 FB_READ_023_SNVT_mass



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_mass.

SNVT number: 023.

SNVT description: Mass (gram).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 6553.5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

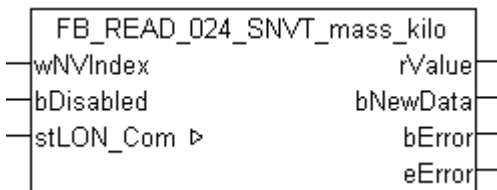
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.22 FB_READ_024_SNVT_mass_kilo



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_mass_kilo.

SNVT number: 024.

SNVT description: Mass (kilogram).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

rValue: Min: 0 / Max: 6553.5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

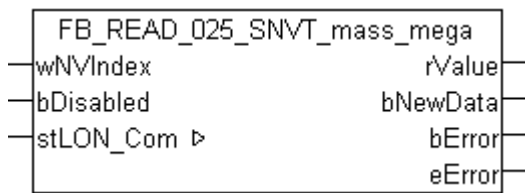
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.23 FB_READ_025_SNVT_mass_mega



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_mass_mega.

SNVT number: 025.

SNVT description: Mass (metric ton).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

rValue: Min: 0 / Max: 6553.5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

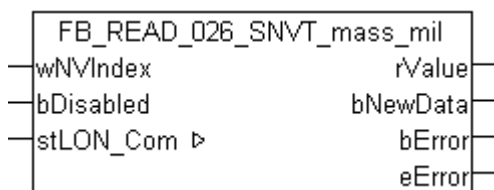
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.24 FB_READ_026_SNVT_mass_mil



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_mass_mil.

SNVT number: 026.

SNVT description: Mass (milligram).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 6553.5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

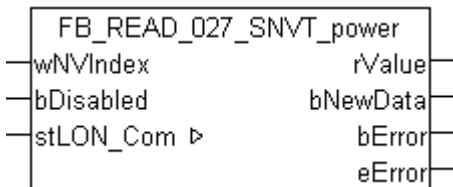
eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.25 FB_READ_027_SNVT_power



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_power

SNVT number: 027

SNVT description: Power (watt).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 6553.5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

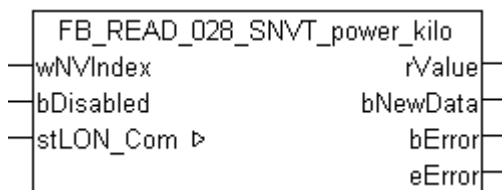
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.26 FB_READ_028_SNVT_power_kilo



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_power_kilo.

SNVT number: 028.

SNVT description: Power (kilowatt).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

rValue: Min: 0 / Max: 6553.5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

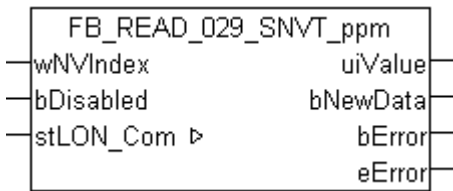
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.27 FB_READ_029_SNVT_ppm



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_ppm.

SNVT number: 029.

SNVT description: Concentration (ppm).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
uiValue       : UINT;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

uiValue: Min: 0 / Max: 65535.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

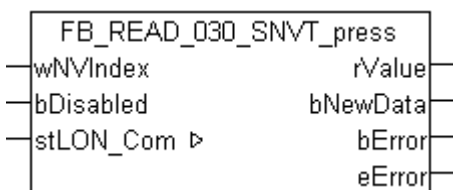
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.28 FB_READ_030_SNVT_press



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_press.

SNVT number: 030.

SNVT description: Pressure (gauge) (kilopascal).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: -3276.8 / Max: 3276.7.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

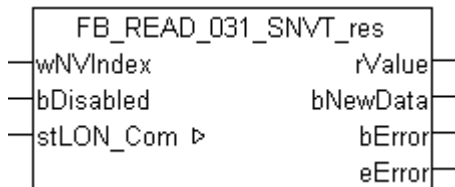
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.29 FB_READ_031_SNVT_res



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_res.

SNVT number: 031.

SNVT description: Electric resistance (ohm).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```


rValue: Min: 0 / Max: 6553.5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

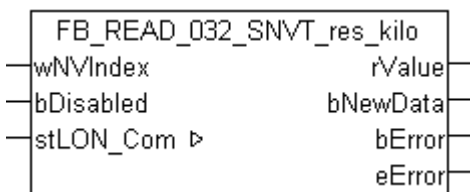
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.30 FB_READ_032_SNVT_res_kilo



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_res_kilo.

SNVT number: 032.

SNVT description: Electric resistance (kiloohm).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

rValue: Min: 0 / Max: 6553.5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

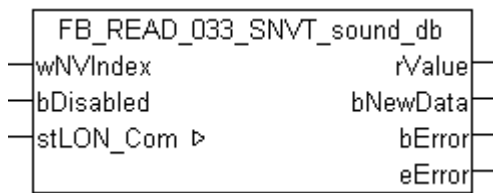
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.31 FB_READ_033_SNVT_sound_db



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_sound_db.

SNVT number: 033.

SNVT description: Sound level (dB).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: -327.68 / Max: 327.67.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

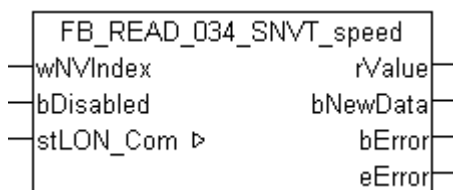
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.32 FB_READ_034_SNVT_speed



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_speed.

SNVT number: 034.

SNVT description: Linear velocity (meter/second).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 6553.5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

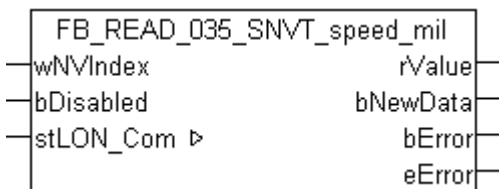
eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.33 FB_READ_035_SNVT_speed_mil



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_speed_mil.

SNVT number: 035.

SNVT description: Linear velocity (meter/second).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 65.535.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

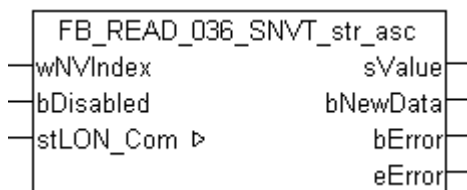
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.34 FB_READ_036_SNVT_str_asc



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_str_asc.

SNVT number: 036.

SNVT description: Character string (30 characters max) (ASCII character string).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
sValue : STRING(31);
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

sValue: STRING(31).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

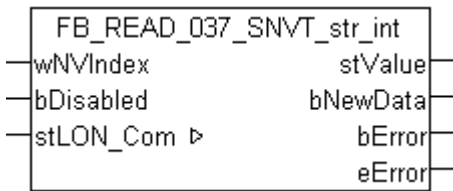
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.35 FB_READ_037_SNVT_str_int



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_str_int.

SNVT number: 037.

SNVT description: Wide character string with locale code (15 characters max) (Wide character string).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_str_int;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure of the data received (see [ST_LON_SNVT_str_int \[▶ 581\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

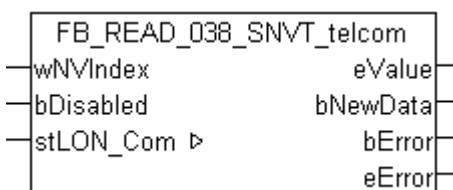
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.36 FB_READ_038_SNVT_telcom



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_telcom.

SNVT number: 038.

SNVT description: Telephone states (telephone state names).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
eValue       : E_LON_telcom_states_t;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

eValue: Enum, to be received (see [E_LON_telcom_states_t](#) [► 527]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

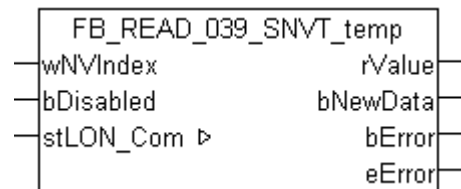
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [► 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [► 66] with the send/receive function blocks (see [ST_LON_Communication](#) [► 557]).

7.2.37 FB_READ_039_SNVTemp



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_temp.

SNVT number: 039.

SNVT description: Temperature (degree celsius).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue      : REAL;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

rValue: Min: 0 / Max: 6553.5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

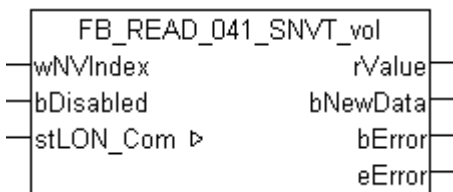
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.38 FB_READ_041_SNVT_vol



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_vol.

SNVT number: 041.

SNVT description: Volume (liter).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue      : REAL;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

rValue: Min: 0 / Max: 6553.5.

bNewData: Is TRUE for one cycle once new data were received.

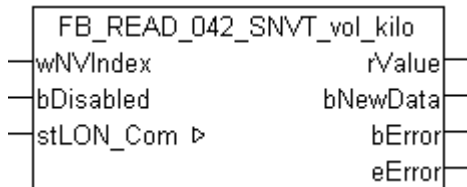
bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.39 FB_READ_042_SNVT_vol_kilo

This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_vol_kilo.

SNVT number: 042.

SNVT description: Volume (kiloliter).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

rValue: Min: 0 / Max: 6553.5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

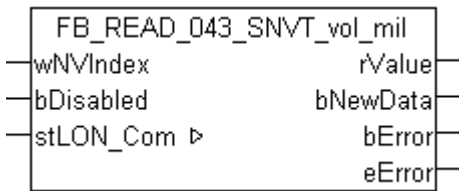
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.40 FB_READ_043_SNVT_vol_mil



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_vol_mil.

SNVT number: 043.

SNVT description: Volume (milliliter).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 6553.5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

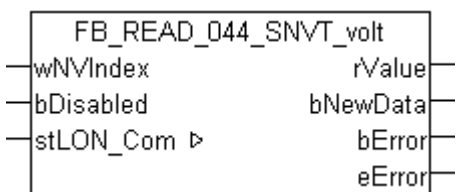
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.41 FB_READ_044_SNVT_volt



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_volt.

SNVT number: 044.

SNVT description: Electric voltage (volt).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: -3276.8 / Max: 3276.7.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

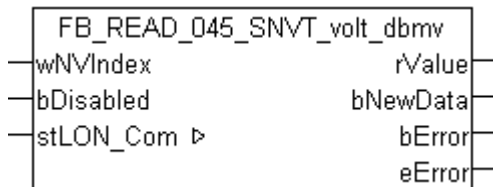
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.42 FB_READ_045_SNVT_volt_dbmv



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_volt_dbmv.

SNVT number: 045.

SNVT description: Electric voltage (dB microvolt).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: -327.68 / Max: 327.67.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

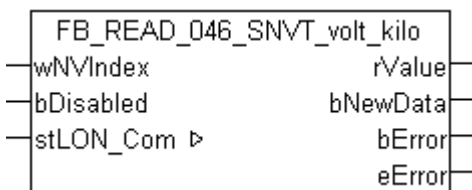
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

`stLON_Com : ST_LON_Communication;`

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.43 FB_READ_046_SNVT_volt_kilo



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_volt_kilo.

SNVT number: 046.

SNVT description: Electric voltage (kilovolt).

VAR_INPUT

`wNVIndex : WORD;`
`bDisabled : BOOL := FALSE;`

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

`rValue : REAL;`
`bNewData : BOOL;`
`bError : BOOL;`
`eError : E_LON_ERROR;`

rValue: Min: -3276.8 / Max: 3276.7.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

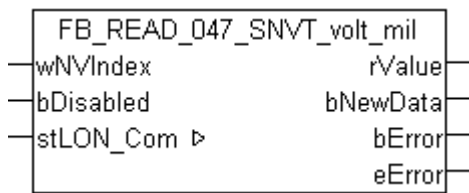
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

`stLON_Com : ST_LON_Communication;`

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.44 FB_READ_047_SNVT_volt_mil



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_volt_mil.

SNVT number: 047.

SNVT description: Electric voltage (millivolt).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: -3276.8 / Max: 3276.7.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

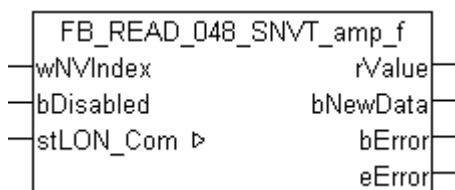
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.45 FB_READ_048_SNVT_amp_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_amp_f.

SNVT number: 048.

SNVT description: Electric current (ampere).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

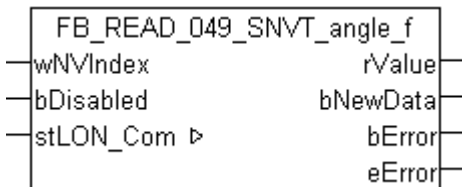
eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.46 FB_READ_049_SNVT_angle_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_angle_f.

SNVT number: 049.

SNVT description: Angular distance (radian).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

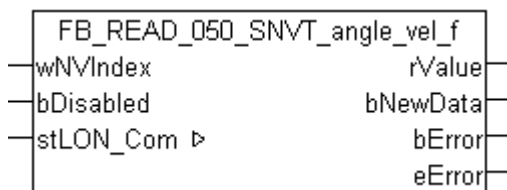
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.47 FB_READ_050_SNVT_angle_vel_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_angle_vel_f.

SNVT number: 050.

SNVT description: Angular velocity (radian/second).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

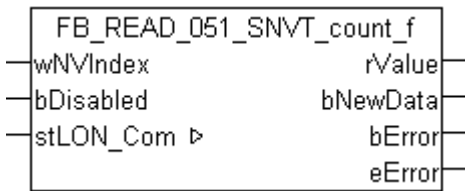
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.48 FB_READ_051_SNVT_count_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_count_f.

SNVT number: 051.

SNVT description: Absolute count (unit).

VAR_INPUT

```

wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
  
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

rValue        : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
  
```

rValue: Min: 0 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

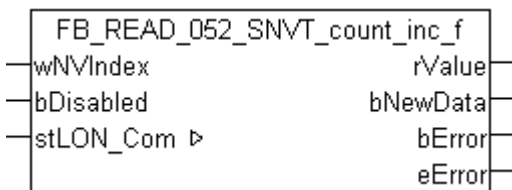
VAR_IN_OUT

```

stLON_Com     : ST_LON_Communication;
  
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.49 FB_READ_052_SNVT_count_inc_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_count_inc_f.

SNVT number: 052.

SNVT description: Increment count (unit (delta)).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

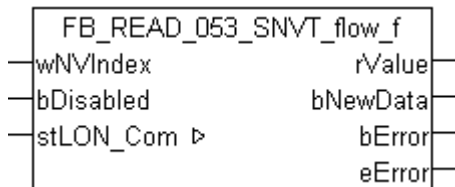
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.50 FB_READ_053_SNVT_flow_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_flow_f.

SNVT number: 053.

SNVT description: Flow volume (liter/second).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```


rValue: Min: -3.40E+38 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

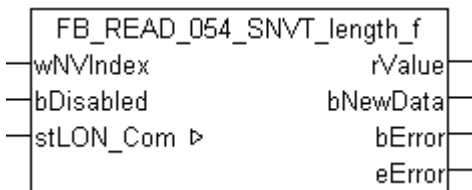
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.51 FB_READ_054_SNVT_length_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_length_f.

SNVT number: 054.

SNVT description: Length (meter).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

rValue: Min: 0 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

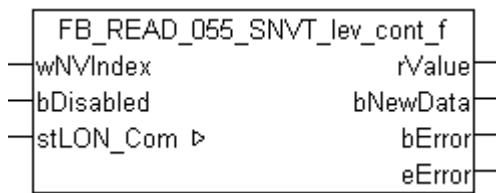
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.52 FB_READ_055_SNVT_lev_cont_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_lev_cont_f.

SNVT number: 055.

SNVT description: Continuous level (% of full scale).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 100.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

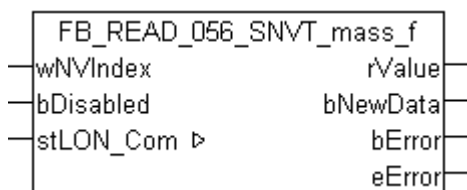
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.53 FB_READ_056_SNVT_mass_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_mass_f.

SNVT number: 056.

SNVT description: Mass (gram).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

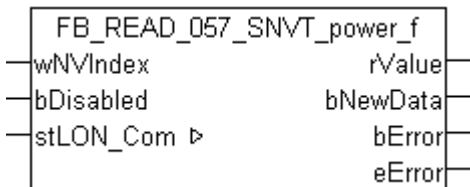
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.54 FB_READ_057_SNVT_power_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_power_f.

SNVT number: 057.

SNVT description: Power (Watts).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

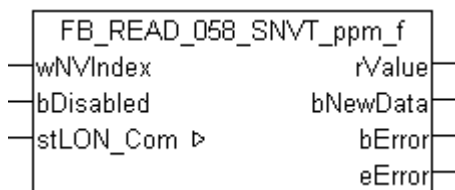
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.55 FB_READ_058_SNVT_ppm_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_ppm_f.

SNVT number: 058.

SNVT description: Concentration (ppm).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

rValue: Min: 0 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

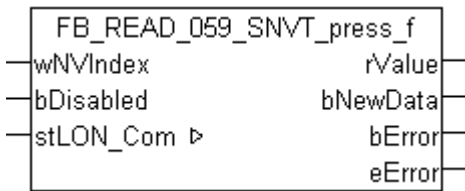
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.56 FB_READ_059_SNVT_press_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_press_f.

SNVT number: 059.

SNVT description: Pressure (gauge) (pascal).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

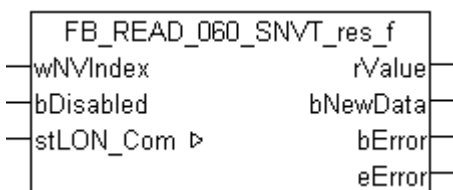
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.57 FB_READ_060_SNVT_res_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_res_f.

SNVT number: 060.

SNVT description: Electric resistance (ohms).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

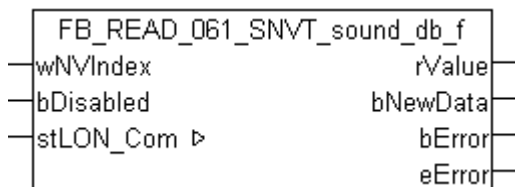
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.58 FB_READ_061_SNVT_sound_db_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_sound_db_f.

SNVT number: 061.

SNVT description: Sound level (dB spl).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

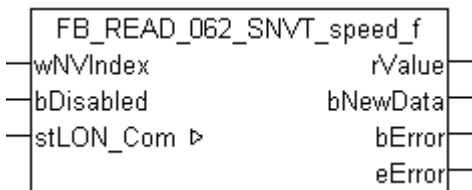
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.59 FB_READ_062_SNVT_speed_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_speed_f.

SNVT number: 062.

SNVT description: Linear velocity (meter/second).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

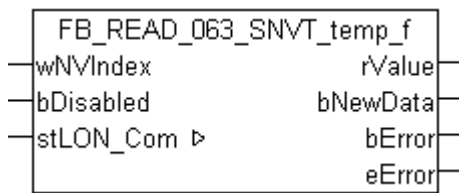
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.60 FB_READ_063_SNVT_temp_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_temp_f.

SNVT number: 063.

SNVT description: Temperature (degree celsius).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

rValue: Min: -273,17 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

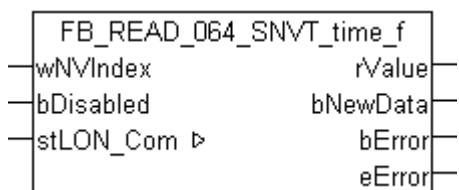
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.61 FB_READ_064_SNVT_time_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_time_f.

SNVT number: 064.

SNVT description: Elapsed time (seconds).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

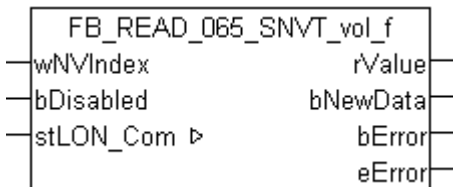
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.62 FB_READ_065_SNVT_vol_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_vol_f.

SNVT number: 065.

SNVT description: Volume (liter).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

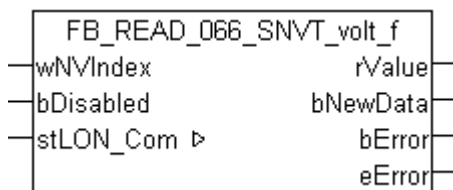
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.63 FB_READ_066_SNVT_volt_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_volt_f.

SNVT number: 066.

SNVT description: Electric voltage (volt).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

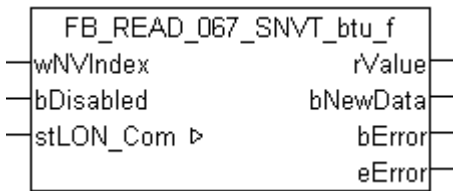
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.64 FB_READ_067_SNVT_btu_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_btu_f.

SNVT number: 067.

SNVT description: Thermal energy (BTU).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

rValue: Min: 0 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

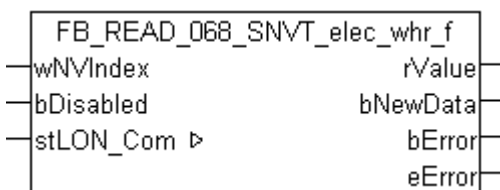
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.65 FB_READ_068_SNVT_elec_whr_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_elec_whr_f.

SNVT number: 068.

SNVT description: Electric energy (watt-hour).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

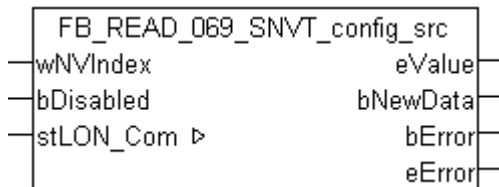
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.66 FB_READ_069_SNVT_config_src



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_config_src.

SNVT number: 069.

SNVT description: Configuration source (configuration source names).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
eValue       : E_LON_config_source_t;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

eValue: Enum, to be received ([E_LON_config_source_t](#) [▶ 494]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

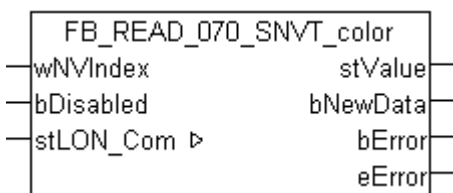
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.67 FB_READ_070_SNVT_color



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_color.

SNVT number: 070.

SNVT description: CIE LAB color (L*,a*,b).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue : ST_LON_SNVT_color;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_color](#) [▶ 564]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

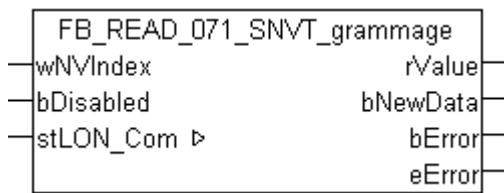
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.68 FB_READ_071_SNVT_grammage



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_grammage.

SNVT number: 071.

SNVT description: Grammage (gram/sq meter).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 6553.5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

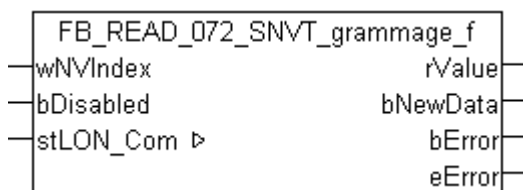
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.69 FB_READ_072_SNVT_grammage_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_grammage_f

SNVT number: 072

SNVT description: Grammage (gram/sq meter)

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

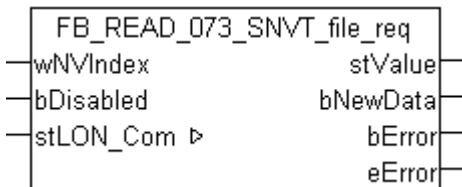
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.70 FB_READ_073_SNVT_file_req



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_file_req.

SNVT number: 073.

SNVT description: File request.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_file_req;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT file req \[▶ 569\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

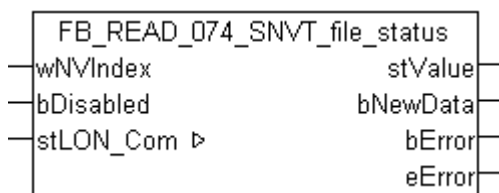
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.71 FB_READ_074_SNVT_file_status



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_file_status.

SNVT number: 074.

SNVT description: File status.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_file_status;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT file status \[▶ 570\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

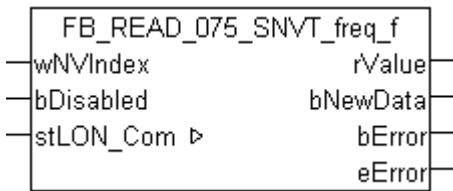
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.72 FB_READ_075_SNVT_freq_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_freq_f

SNVT number: 075

SNVT description: Frequency (hertz).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

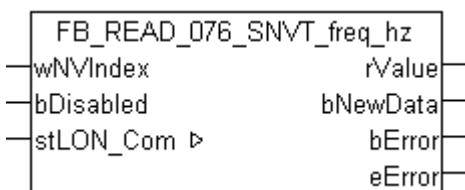
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.73 FB_READ_076_SNVT_freq_hz



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_freq_hz.

SNVT number: 076.

SNVT description: Frequency (hertz).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 6553.5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

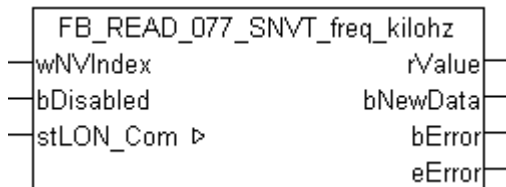
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.74 FB_READ_077_SNVT_freq_kilohz



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_freq_kilohz.

SNVT number: 077.

SNVT description: Frequency (kilohertz).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 6553.5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

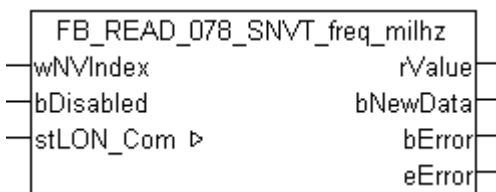
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.75 FB_READ_078_SNVT_freq_milhz



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_freq_milhz.

SNVT number: 078.

SNVT description: Frequency (hertz).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

rValue: Min: 0 / Max: 6.5535.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

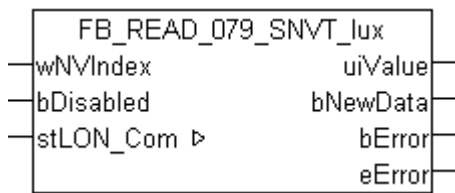
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.76 FB_READ_079_SNVT_lux



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_lux.

SNVT number: 079.

SNVT description: Illumination (lux).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
uiValue       : UINT;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

uiValue: Min: 0 / Max: 65535.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

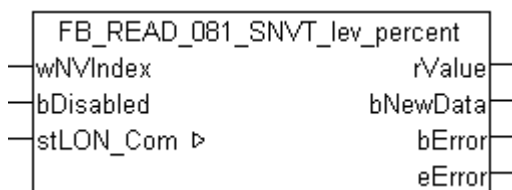
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.77 FB_READ_081_SNVT_lev_percent



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_lev_percent.

SNVT number: 081.

SNVT description: Percentage level in 0.005 % steps. SNVT_switch should be used instead of SNVT_lev_percent, with the exception of network variables that are used to communicate a percentage value and that require the additional resolution provided by SNVT_lev_percent; or for network variable members of functional profiles that are designed primarily for interfacing with SNVT_lev_percent members of other profiles.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue        : REAL;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

rValue: Min: -163.84 / Max: 163.835.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

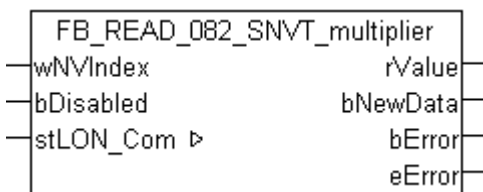
eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.78 FB_READ_082_SNVT_multiplier



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_multiplier.

SNVT number: 082.

SNVT description: Multiplier (16-bit unsigned value).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue      : REAL;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

rValue: Min: 0 / Max: 32.7675.

bNewData: Is TRUE for one cycle once new data were received.

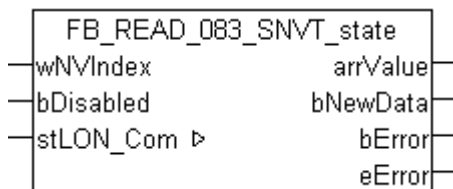
bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.79 FB_READ_083_SNVT_state

This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_state.

SNVT number: 083.

SNVT description: State vector (16 individual bit values). Each state is a boolean single bit value.

VAR_INPUT

```
wNVIndex     : WORD;
bDisabled    : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
arrValue     : ARRAY [0..15] OF BOOL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

arrValue: 0-15 Bit.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

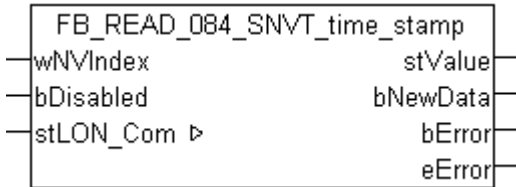
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.80 FB_READ_084_SNVT_time_stamp



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_time_stamp.

SNVT number: 084.

SNVT description: Time stamp (year, month, day, hour, minute, second).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue : Timestruct;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

stValue: Structure, to be received (see [Timestruct](#)). The structure variables *wDayOfWeek* and *wMilliseconds* are not valid and therefore always null.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

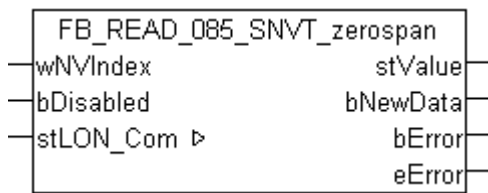
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.81 FB_READ_085_SNVT_zerospanspan



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_zerospanspan.

SNVT number: 085.

SNVT description: Zero and span (Zero, span). Linear transformation parameters: multiply by the span-factor, then add the zero-term.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_zerospanspan;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_zerospanspan \[▶ 584\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

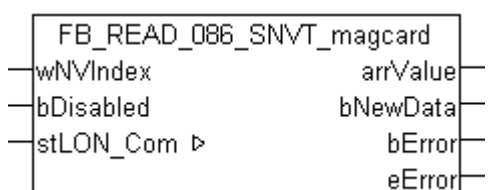
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.82 FB_READ_086_SNVT_magcard



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_magcard

SNVT number: 086

SNVT description: ISO 7811 (40 hexadecimal digits). This data item contains data according to the ISO 7811 standard for card stripes.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
arrValue      : ARRAY [0..40] OF BYTE;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

arrValue: 1-40 BYTE.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

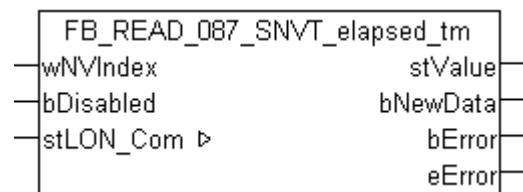
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.83 FB_READ_087_SNVT_elapsed_tm



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_elapsed_tm.

SNVT number: 087.

SNVT description: Elapsed time (day, hour, minute, second, millisecond).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue      : ST_LON_SNVT_elapsed_tm;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_elapsed_tm](#) [▶ 566]).

bNewData: Is TRUE for one cycle once new data were received.

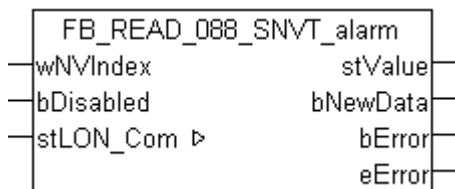
bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL64010](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.84 FB_READ_088_SNVT_alarm

This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_alarm.

SNVT number: 088.

SNVT description: Alarm status.

VAR_INPUT

```
wNVIndex     : WORD;
bDisabled    : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue      : ST_LON_SNVT_alarm;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_alarm](#) [▶ 560]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

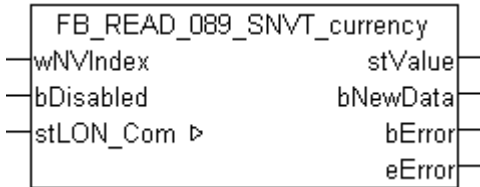
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.85 FB_READ_089_SNVT_currency



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_currency

SNVT number: 089

SNVT description: Currency (unit, magnitude, value).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue : ST_LON_SNVT_currency;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_currency](#) [▶ 565]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

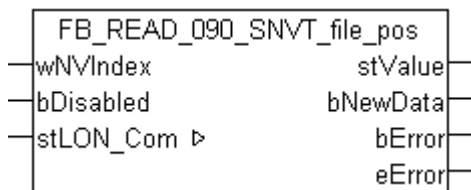
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.86 FB_READ_090_SNVT_file_pos



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_file_pos.

SNVT number: 090.

SNVT description: File position (pointer, length).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_file_pos;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_file_pos \[▶ 569\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

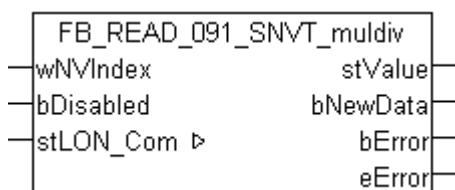
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.87 FB_READ_091_SNVT_muldiv



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_muldiv.

SNVT number: 091.

SNVT description: Gain factor/damping factor (multiplier, divisor).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_muldiv;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_muldiv \[▶ 573\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

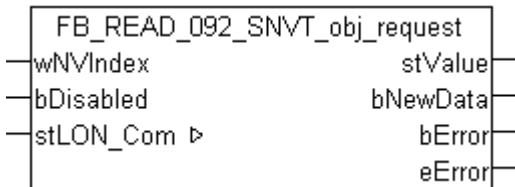
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.88 FB_READ_092_SNVT_obj_request



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_obj_request.

SNVT number: 092.

SNVT description: Object request (ID, request).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue      : ST_LON_SNVT_obj_request;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_obj_request \[► 574\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

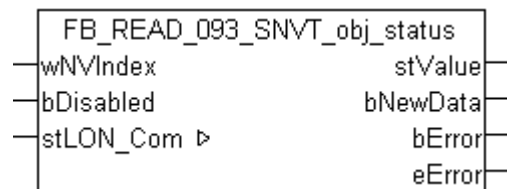
bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[► 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[► 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[► 557\]](#)).

7.2.89 FB_READ_093_SNVT_obj_status

This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_obj_status.

SNVT number: 093.

SNVT description: Object status (ID, status flags (4 byte)).

VAR_INPUT

```
wNVIndex     : WORD;
bDisabled    : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue      : ST_LON_SNVT_obj_status;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_obj_status \[► 574\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

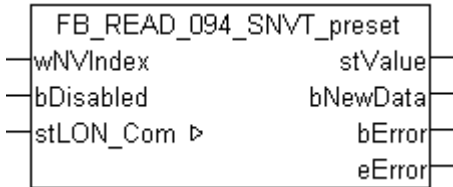
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[► 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.90 FB_READ_094_SNVT_preset



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_preset.

SNVT number: 094.

SNVT description: Preset (mode, data, time).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue : ST_LON_SNVT_preset;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_preset](#) [▶ 576]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

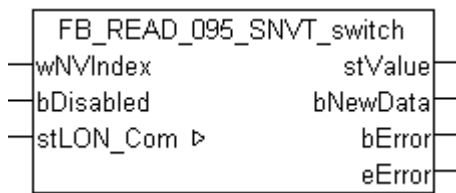
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.91 FB_READ_095_SNVT_switch



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_switch.

SNVT number: 095.

SNVT description: Switch (value, state).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_switch;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_switch \[▶ 582\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

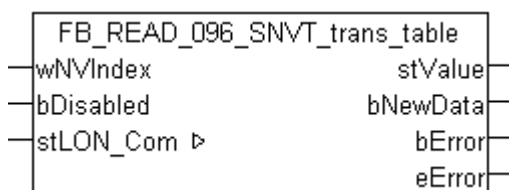
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.92 FB_READ_096_SNVT_trans_table



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_trans_table.

SNVT number: 096.

SNVT description: Translation table (points, interpolation).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_trans_table;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_trans_table](#) [▶ 583]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

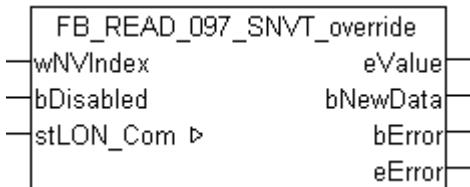
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.93 FB_READ_097_SNVT_override



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_override.

SNVT number: 097.

SNVT description: Override code (override code names).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
eValue      : E_LON_override_t;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

eValue: Enum, to be received (see [E_LON_override_t](#) [▶ 513]).

bNewData: Is TRUE for one cycle once new data were received.

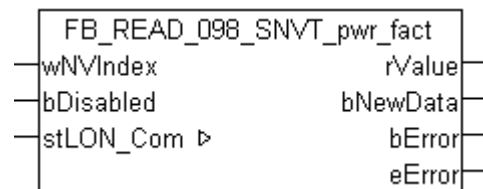
bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com   : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL64010](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.94 FB_READ_098_SNVT_pwr_fact

This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_pwr_fact.

SNVT number: 098.

SNVT description: Power factor (multiplier).

VAR_INPUT

```
wNVIndex    : WORD;
bDisabled    : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue      : REAL;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

rValue: Min: -1 / Max: 1.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

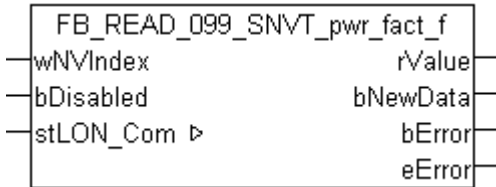
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.95 FB_READ_099_SNVT_pwr_fact_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_pwr_fact_f.

SNVT number: 099.

SNVT description: Power factor (multiplier).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

rValue: Min: -1 / Max: 1.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

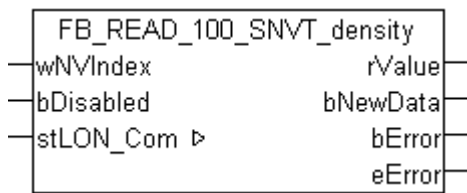
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.96 FB_READ_100_SNVT_density



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_density.

SNVT number: 100.

SNVT description: Density (kilogram/cubic meter).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 32767.5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

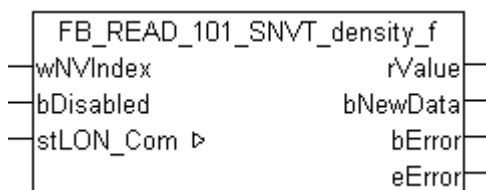
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.97 FB_READ_101_SNVT_density_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_density_f.

SNVT number: 101.

SNVT description: Density (kilogram/cubic meter).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

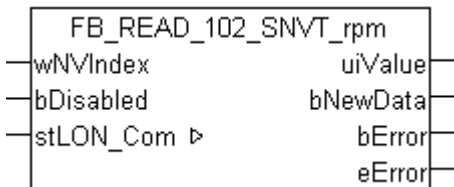
eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.98 FB_READ_102_SNVT_rpm



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_rpm.

SNVT number: 102.

SNVT description: Angular velocity (revolutions/minute (RPM)).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
uiValue       : UINT;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

uiValue: Min: 0 / Max: 65535.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

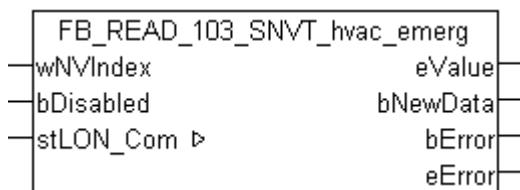
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.99 FB_READ_103_SNVT_hvac_emerg



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_hvac_emerg.

SNVT number: 103.

SNVT description: HVAC emergency mode (emergency mode names).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
eValue : E_LON_emerg_t;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

eValue: Enum, to be received (see [E_LON_emerg_t](#) [▶ 500]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

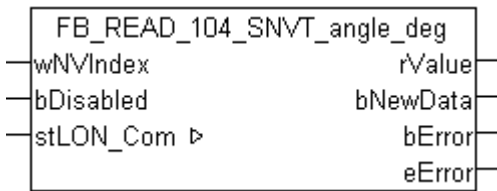
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.100 FB_READ_104_SNVT_angle_deg



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_angle_deg.

SNVT number: 104.

SNVT description: Angular distance (degree).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: -359.98 / Max: 360.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

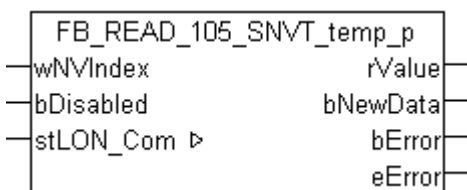
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.101 FB_READ_105_SNVT_temp_p



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_temp_p.

SNVT number: 105.

SNVT description: Temperature (degree celsius).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: -273.17 / Max: 327.67.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

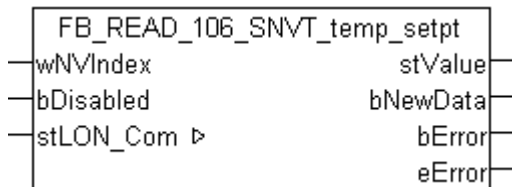
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.102 FB_READ_106_SNVT_temp_setpt



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_temp_setpt.

SNVT number: 106.

SNVT description: Temperature (6 temperature values).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_temp_setpt;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```


stValue: Structure, to be received (see [ST_LON_SNVT temp_setpt \[▶ 582\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

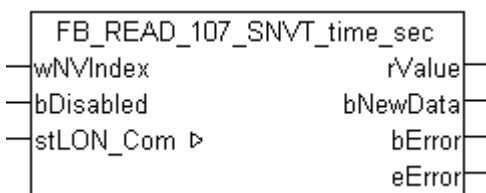
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.103 FB_READ_107_SNVT_time_sec



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_time_sec.

SNVT number: 107.

SNVT description: Elapsed time (seconds).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

rValue: Min: 0 / Max: 6553.5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

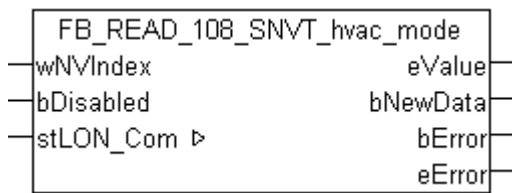
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.104 FB_READ_108_SNVT_hvac_mode



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_hvac_mode.

SNVT number: 108.

SNVT description: HVAC mode (HVAC mode names).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
eValue       : E_LON_hvac_t;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

eValue: Enum, to be received (see [E_LON_hvac_t](#) [▶ 509]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

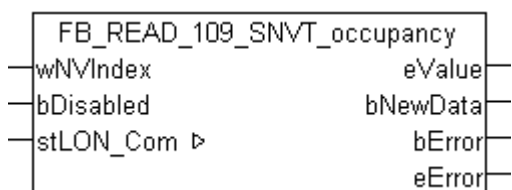
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.105 FB_READ_109_SNVT_occupancy



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_occupancy.

SNVT number: 109.

SNVT description: Occupancy (occupancy code names).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
eValue       : E_LON_occup_t;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

eValue: Enum, to be received (see [E_LON_occup_t \[▶ 513\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

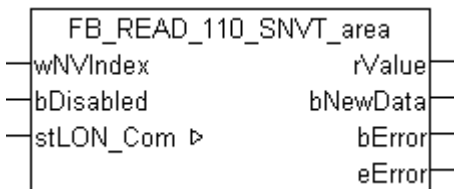
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.106 FB_READ_110_SNV area



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_area.

SNVT number: 110.

SNVT description: Area (square meter).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue      : REAL;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

rValue: Min: 0 / Max: 13.107.

bNewData: Is TRUE for one cycle once new data were received.

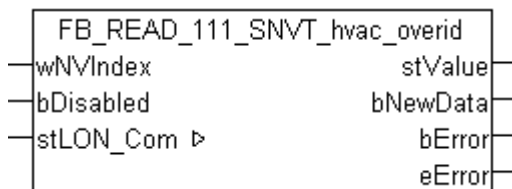
bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.107 FB_READ_111_SNVT_hvac_overid

This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_hvac_overid.

SNVT number: 111.

SNVT description: HVAC override (state, pct, flow).

VAR_INPUT

```
wNVIndex     : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue      : ST_LON_SNVT_hvac_overid;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_hvac_overid \[▶ 570\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

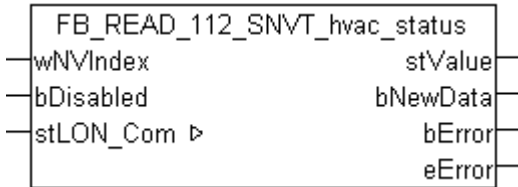
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.108 FB_READ_112_SNVT_hvac_status



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_hvac_status.

SNVT number: 112.

SNVT description: HVAC status (mode, 5 percents, flag).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue : ST_LON_SNVT_hvac_status;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_hvac_status](#) [▶ 571]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

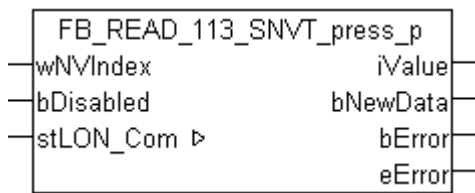
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.109 FB_READ_113_SNVT_press_p



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_press_p.

SNVT number: 113.

SNVT description: Pressure (gauge) (pascal).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
iValue       : INT;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

iValue: Min: -32768 / Max: 32767.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

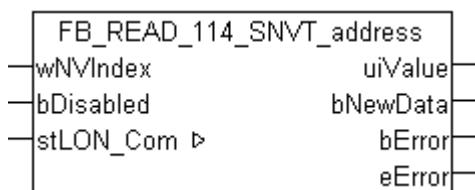
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.110 FB_READ_114_SNVT_address



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_address.

SNVT number: 114.

SNVT description: Neuron address (16-bit address value).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
uiValue       : UINT;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

uiValue: Min: 16384 / Max: 64767.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

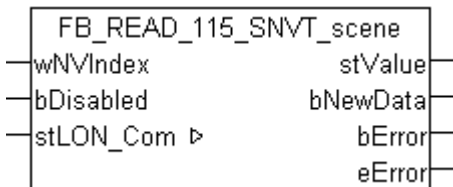
eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.111 FB_READ_115_SNVT_scene



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_scene.

SNVT number: 115.

SNVT description: Scene control (function, scene number).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_scene;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_scene](#) [▶ 580]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

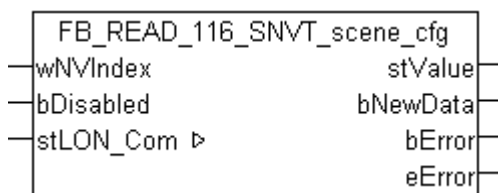
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.112 FB_READ_116_SNVT_scene_cfg



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_scene_cfg.

SNVT number: 116.

SNVT description: Scene configuration (function, scene number, setting, rotation, fade, delay, priority).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_scene_cfg;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_scene_cfg](#) [▶ 581]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

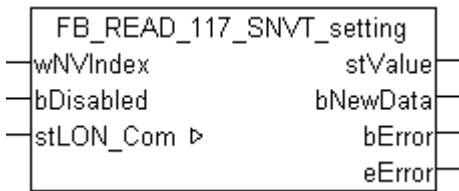
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.113 FB_READ_117_SNVT_setting



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_setting.

SNVT number: 117.

SNVT description: Setting control (function, setting, rotation).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_setting;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_setting \[▶ 581\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

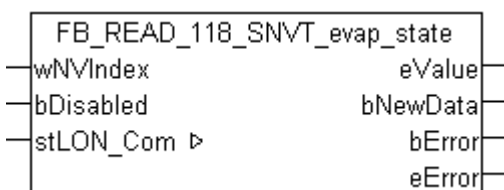
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.114 FB_READ_118_SNVT_evap_state



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_evap_state.

SNVT number: 118.

SNVT description: Evaporator state (evaporator state names).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
eValue        : E_LON_evap_t;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

eValue: Enum, to be received (see [E_LON_evap_t \[► 502\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

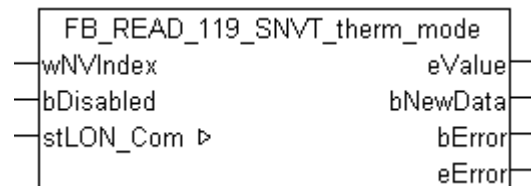
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[► 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[► 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[► 557\]](#)).

7.2.115 FB_READ_119_SNVT_therm_mode



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_therm_mode.

SNVT number: 119.

SNVT description: Thermostat mode (thermostat mode names).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
eValue      : E_LON_therm_mode_t;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

eValue: Enum, to be received (see [E_LON_therm_mode_t \[▶ 528\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

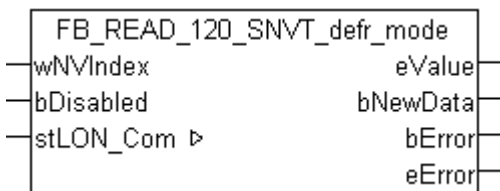
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.116 FB_READ_120_SNVT_defr_mode



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_defr_mode.

SNVT number: 120.

SNVT description: Defrost mode (defrost mode names).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
eValue      : E_LON_defrost_mode_t;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

eValue: Enum, to be received (see [E_LON_defrost_mode_t \[▶ 497\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

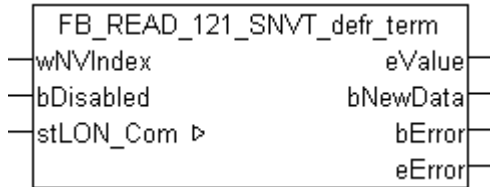
bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.117 FB_READ_121_SNVT_defr_term

This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_defr_term.

SNVT number: 121.

SNVT description: Defrost termination (defrost termination names).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
eValue        : E_LON_defrost_term_t;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

eValue: Enum, to be received (see [E_LON_defrost_term_t](#) [▶ 498]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

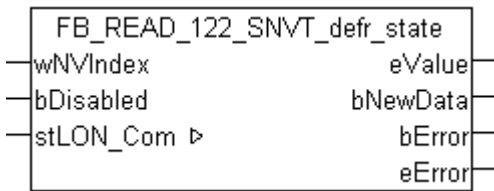
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.118 FB_READ_122_SNVT_defr_state



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_defr_state.

SNVT number: 122.

SNVT description: Defrost state (defrost state names).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
eValue       : E_LON_defrost_state_t;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

eValue: Enum, to be received (see E_LON_telcom_states_t).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

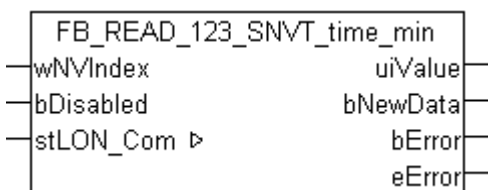
```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

Also see about this

- E_LON_defrost_state_t [▶ 498]

7.2.119 FB_READ_123_SNVT_time_min



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_time_min.

SNVT number: 123.

SNVT description: Elapsed time (minute).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
uiValue       : UINT;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

uiValue: Min: 0 / Max: 65535.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

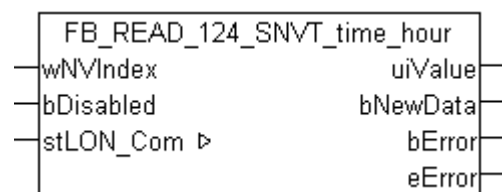
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.120 FB_READ_124_SNVT_time_hour



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_time_hour.

SNVT number: 124.

SNVT description: Elapsed time (hour).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
uiValue      : UINT;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

uiValue: Min: 0 / Max: 65535.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

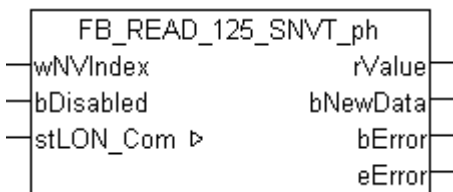
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [[▶ 66](#)] with the send/receive function blocks (see [ST_LON_Communication](#) [[▶ 557](#)]).

7.2.121 FB_READ_125_SNVT_ph



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_ph.

SNVT number: 125.

SNVT description: Acidity (pH). Ratio of concentration of ions.

VAR_INPUT

```
wNVIndex     : WORD;
bDisabled    : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: -32.768 / Max: 32.767.

bNewData: Is TRUE for one cycle once new data were received.

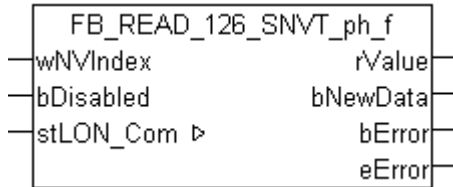
bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.122 FB_READ_126_SNVT_ph_f

This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_ph_f.

SNVT number: 126.

SNVT description: Acidity (pH). Ratio of concentration of ions.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

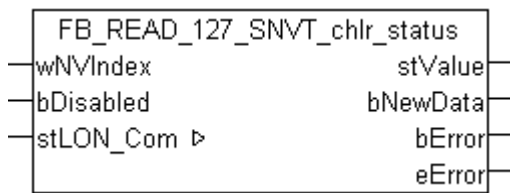
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.123 FB_READ_127_SNVT_chlr_status



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_chlr_status.

SNVT number: 127.

SNVT description: Chiller status (run mode, op mode, state bits).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_chlr_status;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_chlr_status \[▶ 561\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

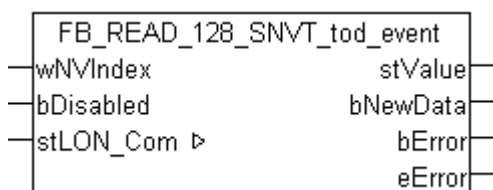
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.124 FB_READ_128_SNVT_tod_event



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_tod_event.

SNVT number: 128.

SNVT description: Time of day event (current, next, time). Occupancy scheduling event.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_tod_event;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_tod_event](#) [▶ 583]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

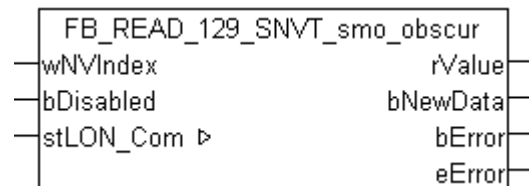
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.125 FB_READ_129_SNVT_smo_obscur



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_smo_obscur.

SNVT number: 129.

SNVT description: Smoke obscuration (percent obscuration).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue      : REAL;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

rValue: Min: 0 / Max: 5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

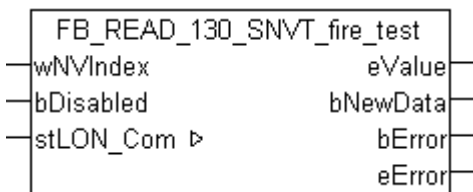
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.126 FB_READ_130_SNVT_fire_test



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_fire_test.

SNVT number: 130.

SNVT description: Fire test request (fire test names).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
eValue        : E_LON_fire_test_t;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

eValue: Enum, to be received (see [E_LON_fire_test_t \[▶ 505\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

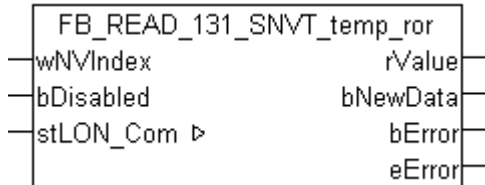
bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.127 FB_READ_131_SNVT_temp_ror

This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_temp_ror.

SNVT number: 131.

SNVT description: Temperature rate of change/rise (degree celsius/minute).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: -16384 / Max: 16383.5.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

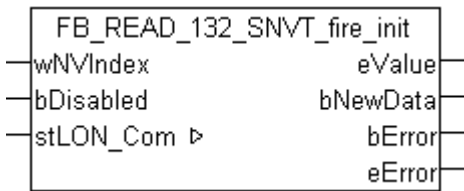
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.128 FB_READ_132_SNVT_fire_init



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_fire_init.

SNVT number: 132.

SNVT description: Fire initiator type (fire initiator type names).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
eValue       : E_LON_fire_initiator_t;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

eValue: Enum, to be received (see [E_LON_fire_initiator_t \[▶ 504\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

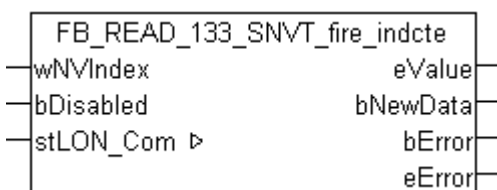
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.129 FB_READ_133_SNVT_fire_indcte



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_fire_indcte.

SNVT number: 133.

SNVT description: Fire indicator type (fire indicator type names).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
eValue        : E_LON_fire_indicator_t;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

eValue: Enum, to be received (see [E_LON_fire_indicator_t](#) [► 504]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

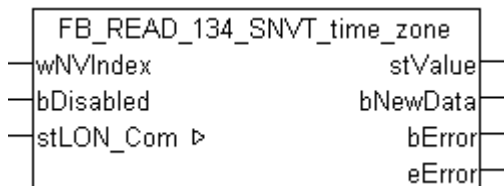
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [► 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [► 66] with the send/receive function blocks (see [ST_LON_Communication](#) [► 557]).

7.2.130 FB_READ_134_SNVT_time_zone



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_time_zone.

SNVT number: 134.

SNVT description: Time zone descriptor (offset, type, startDST, endDST).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue      : ST_LON_SNVT_time_zone;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_time_zone \[▶ 583\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

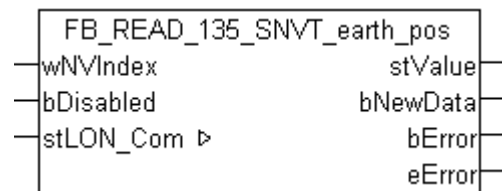
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.131 FB_READ_135_SNVT_earth_pos



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_earth_pos.

SNVT number: 135.

SNVT description: Earth position (lat & long direction, latitude deg & min, longitude deg & min, height).

VAR_INPUT

```
wNVIndex     : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue      : ST_LON_SNVT_earth_pos;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_earth_pos \[▶ 566\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

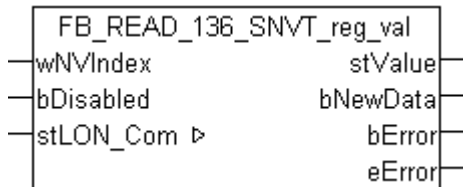
bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.132 FB_READ_136_SNVT_reg_val

This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_reg_val.

SNVT number: 136.

SNVT description: Register value (raw value, unit code, number of decimals).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_reg_val;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_reg_val](#) [▶ 579]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

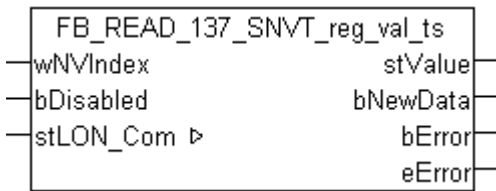
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.133 FB_READ_137_SNVT_reg_val_ts



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_reg_val_ts.

SNVT number: 137.

SNVT description: Register value (raw value, unit code, number of decimals, status, state, timestamp).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_reg_val_ts;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_reg_val_ts \[▶ 580\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

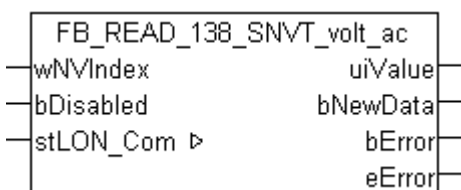
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.134 FB_READ_138_SNVT_volt_ac



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_volt_ac.

SNVT number: 138.

SNVT description: Voltage in alternating current (volt AC).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
uiValue       : UINT;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

uiValue: Min: 0 / Max: 65535.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

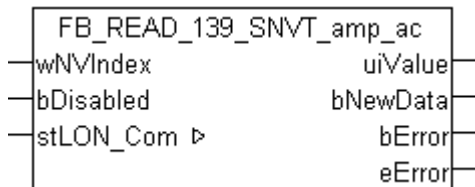
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.135 FB_READ_139_SNVT_amp_ac



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_amp_ac.

SNVT number: 139.

SNVT description: Amperage in alternating current (ampere AC).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
uiValue       : UINT;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

uiValue: Min: 0 / Max: 65535.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

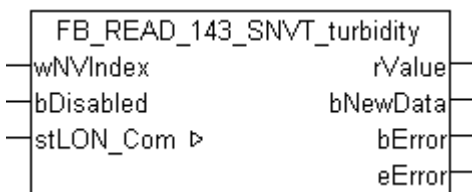
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.136 FB_READ_143_SNVT_turbidity



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_turbidity.

SNVT number: 143.

SNVT description: Turbidity (nephelometric turbidity unit).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

rValue: Min: 0 / Max: 65.535.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

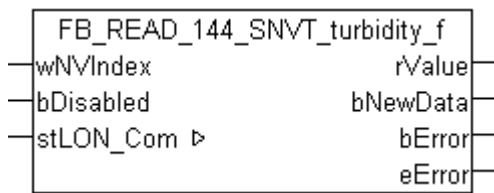
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.137 FB_READ_144_SNVT_turbidity_f



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_turbidity_f.

SNVT number: 144.

SNVT description: Turbidity (nephelometric turbidity unit).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 3.40E+38.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

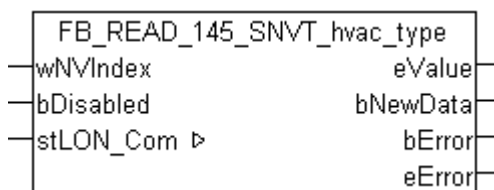
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.138 FB_READ_145_SNVT_hvac_type



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_hvac_type.

SNVT number: 145.

SNVT description: HVAC unit type (HVAC unit type names).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
eValue        : E_LON_hvac_hvt_t;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

eValue: Enum, to be received (see [E_LON_hvac_hvt_t](#) [▶ 506]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

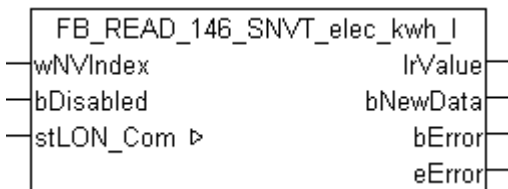
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.139 FB_READ_146_SNVT_elec_kwh_I



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_elec_kwh_I.

SNVT number: 146.

SNVT description: Electric energy (kilowatt hour).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

lrValue      : LREAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;

```

lrValue: Min: -214748364.8 / Max: 214748364.7.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

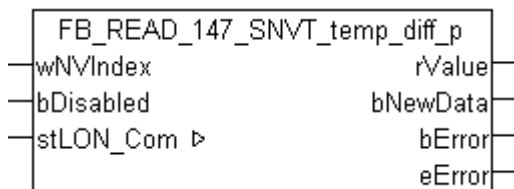
VAR_IN_OUT

```

stLON_Com    : ST_LON_Communication;

```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.140 FB_READ_147_SNVT_temp_diff_p

This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_temp_diff_p.

SNVT number: 147.

SNVT description: Temp difference (degree celsius).

VAR_INPUT

```

wNVIndex     : WORD;
bDisabled    : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;

```

rValue: Min: -327.68 / Max: 327.67.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

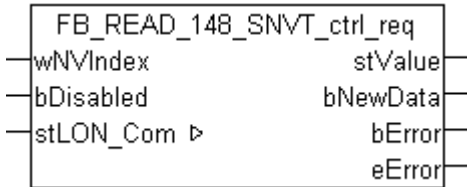
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.141 FB_READ_148_SNVT_ctrl_req



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_ctrl_req.

SNVT number: 148.

SNVT description: Control request (receiver ID, sender ID, sender priority).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue : ST_LON_SNVT_ctrl_req;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_ctrl_req](#) [▶ 564]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

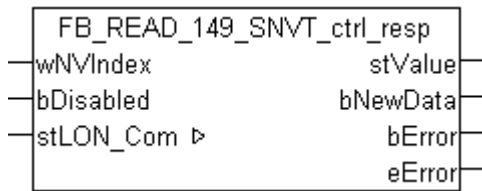
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.142 FB_READ_149_SNVT_ctrl_resp



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_ctrl_resp.

SNVT number: 149.

SNVT description: Control response (status, sender, controller ID).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_ctrl_resp;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_ctrl_resp \[▶ 564\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

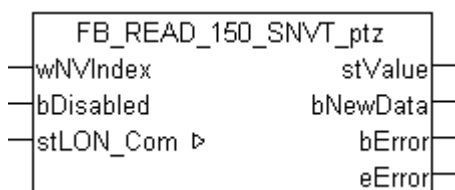
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.143 FB_READ_150_SNVT_ptz



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_ptz.

SNVT number: 150.

SNVT description: Camera PTZ (pan, pan speed, tilt, tilt speed, zoom, zoom speed).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_ptz;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_ptz \[▶ 576\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

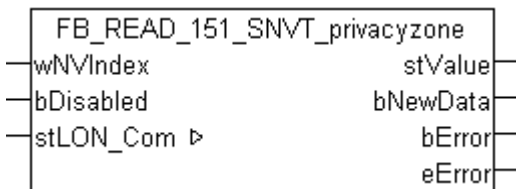
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.144 FB_READ_151_SNVT_privacyzone



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_privacyzone.

SNVT number: 151.

SNVT description: Privacy zone (action, zone number, camera ID).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue      : ST_LON_SNVT_privacyzone;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_privacyzone \[▶ 576\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

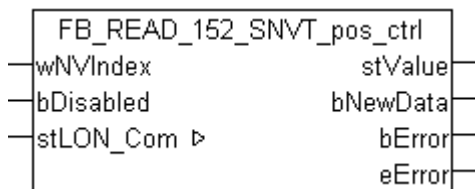
bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.145 FB_READ_152_SNVT_pos_ctrl

This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_pos_ctrl.

SNVT number: 152.

SNVT description: Position control (receiver, controller ID, controller priority, function, action, value).

VAR_INPUT

```
wNVIndex     : WORD;
bDisabled    : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue      : ST_LON_SNVT_pos_ctrl;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_pos_ctrl \[▶ 575\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

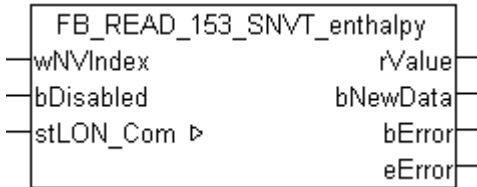
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.146 FB_READ_153_SNVT_enthalpy



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_enthalpy.

SNVT number: 153.

SNVT description: Enthalpy (kJ/kg).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

rValue: Min: -327.68 / Max: 327.67.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

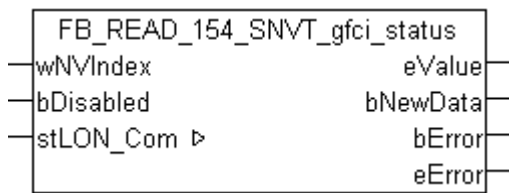
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.147 FB_READ_154_SNVT_gfci_status



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_gfci_status.

SNVT number: 154.

SNVT description: GFCl status type (GFCl status type names).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
eValue       : E_LON_gfci_status_t;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

eValue: Enum, to be received (see [E_LON_gfci_status_t \[▶ 506\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

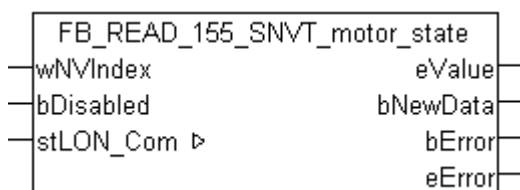
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.148 FB_READ_155_SNVT_motor_state



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_motor_state.

SNVT number: 155.

SNVT description: Motor state (motor state names).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
eValue       : E_LON_motor_state_t;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

eValue: Enum, to be received (see [E_LON_motor_state_t](#) [▶ 511]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

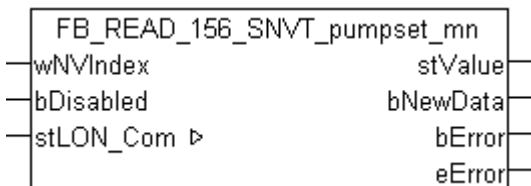
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.149 FB_READ_156_SNVT_pumpset_mn



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_pumpset_mn.

SNVT number: 156.

SNVT description: Pumpset (main, booster, priority, ready, emerg, main enabled, booster enabled, maint).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue      : ST_LON_SNVT_pumpset_mn;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_pumpset_mn](#) [▶ 577]).

bNewData: Is TRUE for one cycle once new data were received.

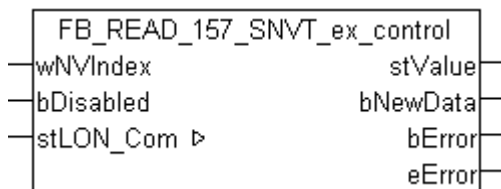
bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL64010](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.150 FB_READ_157_SNVT_ex_control

This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_ex_control.

SNVT number: 157.

SNVT description: Exclusive control (status, address).

VAR_INPUT

```
wNVIndex     : WORD;
bDisabled    : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue      : ST_LON_SNVT_ex_control;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_ex_control](#) [▶ 569]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

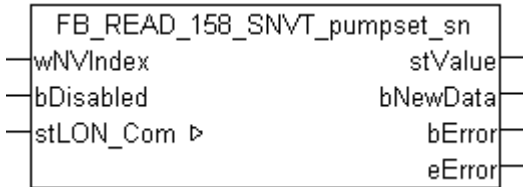
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.151 FB_READ_158_SNVT_pumpset_sn



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_pumpset_sn.

SNVT number: 158.

SNVT description: Pumpset sensor (dilution, exhaust, pressure, vacuum, ...).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue : ST_LON_SNVT_pumpset_sn;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_pumpset_sn](#) [▶ 578]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

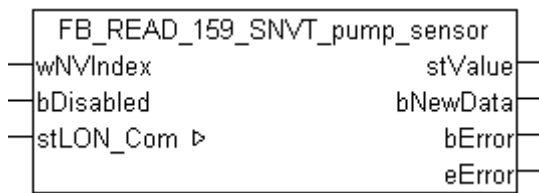
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.152 FB_READ_159_SNVT_pump_sensor



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_pump_sensor.

SNVT number: 159.

SNVT description: Pump sensor (speed, temperature, status).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_pump_sensor;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_pump_sensor \[▶ 577\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

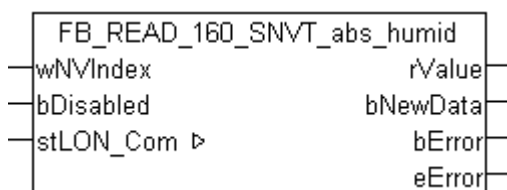
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.153 FB_READ_160_SNVT_abs_humid



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_abs_humid.

SNVT number: 160.

SNVT description: Absolute humidity (gram/kilogram).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 655.35.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

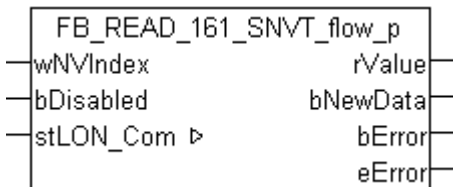
eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.154 FB_READ_161_SNVT_flow_p



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_flow_p.

SNVT number: 161.

SNVT description: Flow volume (cubic meter/hour).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: 0 / Max: 655.35.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

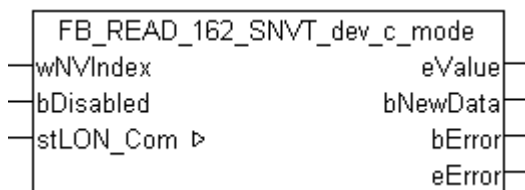
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.155 FB_READ_162_SNVT_dev_c_mode



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_dev_c_mode.

SNVT number: 162.

SNVT description: Device control mode (device control mode names).

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
eValue : E_LON_device_c_mode_t;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

eValue: Enum, to be received (see [E_LON_device_c_mode_t](#) [▶ 498]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

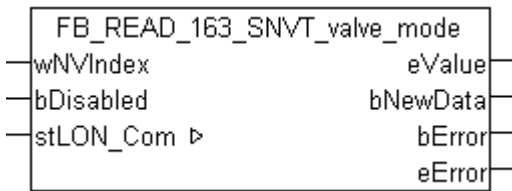
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.156 FB_READ_163_SNVT_valve_mode



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_valve_mode.

SNVT number: 163.

SNVT description: Valve mode (valve mode names).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
eValue        : E_LON_valve_mode_t;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

eValue: Enum, to be received (see [E_LON_valve_mode_t](#) [▶ 529]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

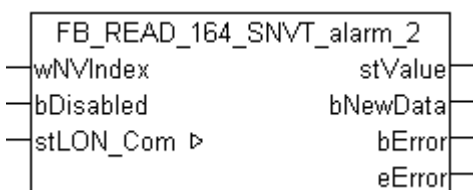
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.157 FB_READ_164_SNVT_alarm_2



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_alarm_2.

SNVT number: 164.

SNVT description: Alarm status 2. Used to report alarm status for a functional block or device. Replaces SNVT_alarm.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_alarm_2;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_alarm_2](#) [▶ 560]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

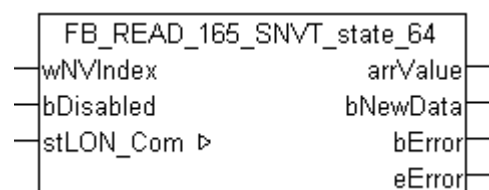
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.158 FB_READ_165_SNVT_state_64



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_state_64.

SNVT number: 165.

SNVT description: State vector (64 individual bit values). Each state is a boolean single-bit value.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
arrValue   : ARRAY [0..63] OF BOOL;
bNewData   : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
```

arrValue: 0-63 Bit.

bNewData: Is TRUE for one cycle once new data were received.

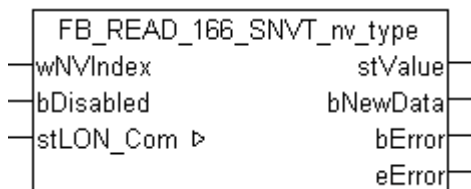
bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.159 FB_READ_166_SNVT_nv_type

This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_nv_type.

SNVT number: 166.

SNVT description: Network variable type. Type description for changeable network variables.

VAR_INPUT

```
wNVIndex   : WORD;
bDisabled   : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue     : ST_LON_SNVT_nv_type;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_nv_type](#) [▶ 573]).

bNewData: Is TRUE for one cycle once new data were received.

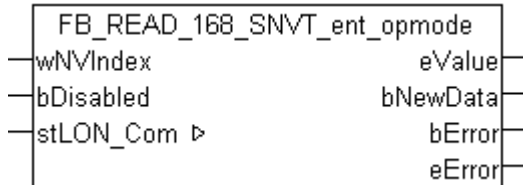
bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.160 FB_READ_168_SNVT_ent_opmode

This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_ent_opmode.

SNVT number: 168.

SNVT description: Entry operation mode (entry operation mode names) used to send operation-mode information to an entry object, e.g., a door, lock, sluice, or something which allows/prohibits entry to an area.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
eValue       : E_LON_ent_opmode_cmd_t;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

eValue: Enum, to be received (see [E_LON_ent_opmode_cmd_t](#) [▶ 501]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

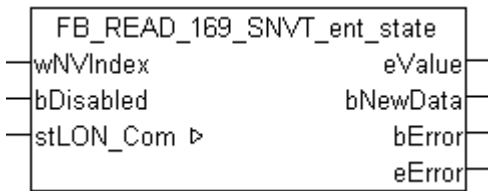
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.161 FB_READ_169_SNVT_ent_state



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_ent_state.

SNVT number: 169.

SNVT description: Entry state (entry state names). Desired state for an entry object, e.g., a door, lock, sluice, or something that controls entry of an area.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
eValue       : E_LON_ent_cmd_t;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

eValue: Enum, to be received (see [E_LON_ent_cmd_t \[▶ 500\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

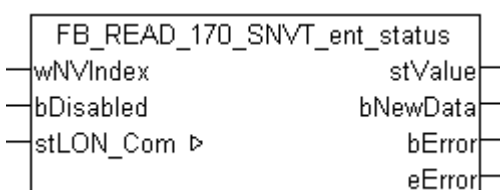
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.162 FB_READ_170_SNVT_ent_status



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_ent_status.

SNVT number: 170.

SNVT description: Entry status. Status information from an entry object, e.g., a door, lock, sluice, or something that allows/prohibits entry into an area.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVNT_ent_status;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVNT_ent_status \[▶ 567\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

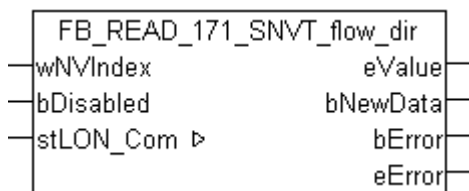
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL64010 \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.163 FB_READ_171_SNVNT_flow_dir



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_flow_dir.

SNVT number: 171.

SNVT description: Flow direction (flow direction names). Direction of allowable flow, or direction of present flow.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
eValue      : E_LON_flow_direction_t;
bNewData   : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
```

eValue: Enum, to be received (see [E_LON_flow_direction_t](#) [▶ 506]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

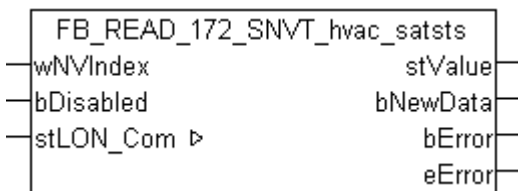
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com   : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.164 FB_READ_172_SNVT_hvac_satsts



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_hvac_satsts.

SNVT number: 172.

SNVT description: HVAC saturation status.

A value of 0 in a field indicates that the resource associated with that field has not saturated or reached an end stop before attaining the required setpoint.

A value of 1 indicates that the resource associated with that field has saturated or reached an end stop without attaining the required setpoint.

VAR_INPUT

```
wNVIndex    : WORD;
bDisabled   : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue     : ST_LON_SNVT_hvac_satsts;
bNewData   : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_hvac_satsts](#) [▶ 570]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

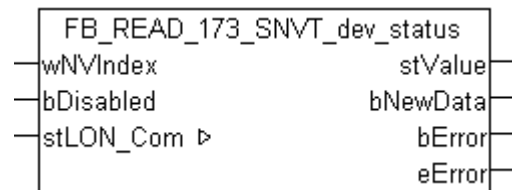
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.165 FB_READ_173_SNVT_dev_status



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_dev_status.

SNVT number: 173.

SNVT description: Status of the device.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_dev_status;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

stValue: stValue, to be received (see [ST_LON_SNVT_dev_status](#) [▶ 566]).

If *stValue.eDevice_select* = eLON_DV_PUMP_CTRL (0), then the structure *stValue.stDev_type.stPump_ctrl* is valid.

If *stValue.eDevice_select* = eLON_DV_VALVE_POS (1), then the structure *stValue.stDev_type.stValvePos* is valid.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

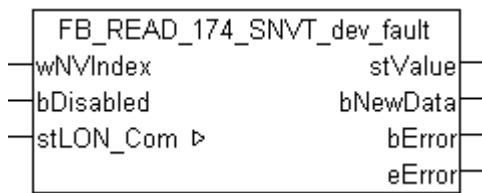
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.166 FB_READ_174_SNVT_dev_fault



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_dev_fault.

SNVT number: 174.

SNVT description: Device fault states. Fault information for the device.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_dev_fault;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_dev_fault \[► 565\]](#)).

If *stValue.eDevice_select* = eLON_DV_PUMP_CTRL (0), then the structure *stValue.stDev_type.stPump_ctrl* is valid.

If *stValue.eDevice_select* = eLON_DV_VALVE_POS (1), then the structure *stValue.stDev_type.stValvePos* is valid.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

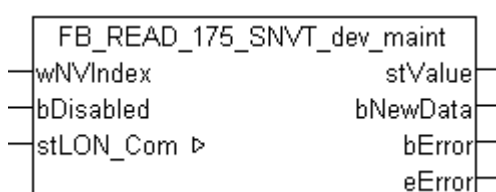
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[► 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[► 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[► 557\]](#)).

7.2.167 FB_READ_175_SNVT_dev_maint



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_dev_maint.

SNVT number: 175.

SNVT description: Device maintenance. Device-maintenance states.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_dev_maint;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_dev_maint](#) [▶ 566]).

If *stValue.eDevice_select* = eLON_DV_PUMP_CTRL (0), then the structure *stValue.stDev_type.stPump_ctrl* is valid.

If *stValue.eDevice_select* = eLON_DV_VALVE_POS (1), then the structure *stValue.stDev_type.stValvePos* is valid.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

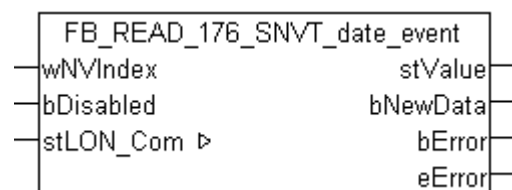
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.168 FB_READ_176_SNVT_date_event



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_date_event.

SNVT number: 176.

SNVT description: State of an event.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue      : ST_LON_SNVT_date_event;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_date_event \[▶ 565\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

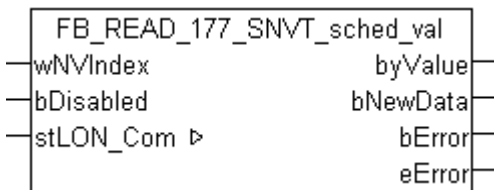
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.169 FB_READ_177_SNVT_sched_val



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_sched_val.

SNVT number: 177.

SNVT description: Scheduler value. Index from a scheduler that selects an entry in a SCPT value definition array or is a direct value output.

VAR_INPUT

```
wNVIndex     : WORD;
bDisabled    : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
byValue      : BYTE;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

byValue: Min: 0 / Max: 255.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

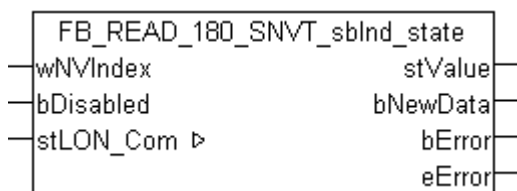
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.170 FB_READ_180_SNVT_sblnd_state



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_sblnd_state.

SNVT number: 180.

SNVT description: Sunblind State.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_sblnd_state;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_sblnd_state](#) [▶ 580]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

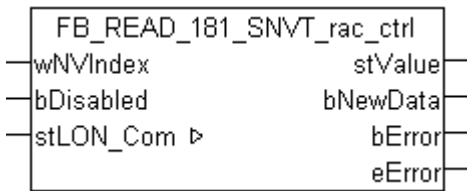
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.171 FB_READ_181_SNVT_rac_ctrl



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_rac_ctrl.

SNVT number: 181.

SNVT description: Rail-Audio Controller Control. Invokes audio control for a given source.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_rac_ctrl;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_rac_ctrl \[▶ 579\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

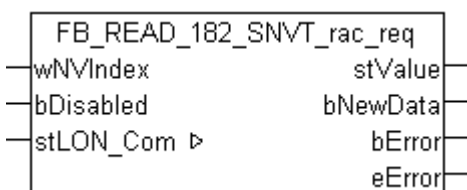
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.172 FB_READ_182_SNVT_rac_req



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_rac_req.

SNVT number: 182.

SNVT description: Rail-Audio Controller Request. Requests audio control for a given source.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_rac_req;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_rac_req](#) [▶ 579]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

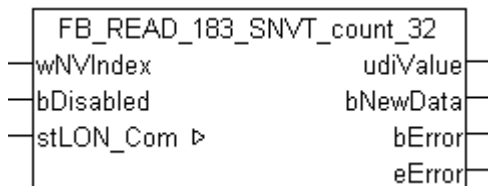
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.173 FB_READ_183_SNVT_count_32



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_count_32.

SNVT number: 183.

SNVT description: Absolute count. A 32-bit counter.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
udiValue : UDINT;
bNewData : BOOL;
bError   : BOOL;
eError   : E_LON_ERROR;
```

udiValue: Min: 0 / Max: 4294967294.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

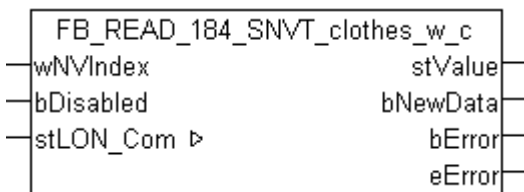
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.174 FB_READ_184_SNVT_clothes_w_c



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_clothes_w_c.

SNVT number: 184.

SNVT description: Clothes Washer Command. Used to program and start a clothes washer.

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue : ST_LON_SNVT_clothes_w_c;
bNewData : BOOL;
bError   : BOOL;
eError   : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_clothes_w_c \[▶ 563\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

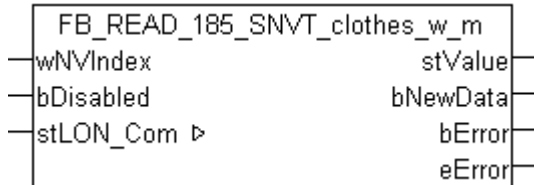
bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.175 FB_READ_185_SNVT_clothes_w_m

This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_clothes_w_m.

SNVT number: 185.

SNVT description: Clothes Washer-Management Status. Provides status of door/lid and drain.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_clothes_w_m;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_clothes_w_m](#) [▶ 563]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

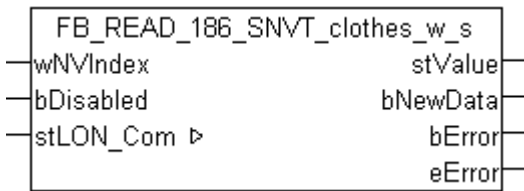
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.176 FB_READ_186_SNVT_clothes_w_s



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_clothes_w_s.

SNVT number: 186.

SNVT description: Clothes Washer Status. Used to provide present status from a clothes washer, including command and alarm information.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_clothes_w_s;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_clothes_w_s \[▶ 563\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

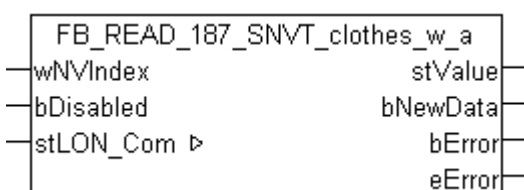
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.177 FB_READ_187_SNVT_clothes_w_a



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_clothes_w_a.

SNVT number: 187.

SNVT description: Clothes Washer Alarm. Used to provide alarm status for a clothes washer.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_clothes_w_a;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_clothes_w_a](#) [▶ 561]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

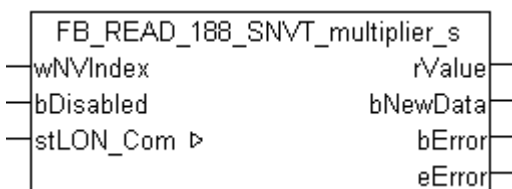
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.178 FB_READ_188_SNVT_multiplier_s



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_multiplier_s.

SNVT number: 188.

SNVT description: Multiplier. Value multiplier.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue      : REAL;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

rValue: Min: 0 / Max: 2.54.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

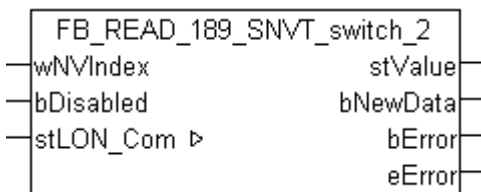
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.179 FB_READ_189_SNVT_switch_2



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_switch_2.

SNVT number: 189.

SNVT description: Switch with scene and setting control. An enhanced version of SNVT_switch with scene and setting controls similar to SNVT_scene and SNVT_setting.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_switch_2;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_switch_2 \[▶ 582\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

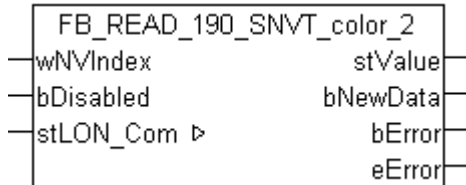
bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.180 FB_READ_190_SNVT_color_2

This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_color_2.

SNVT number: 190.

SNVT description: Color.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_color_2;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_color_2](#) [▶ 564]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

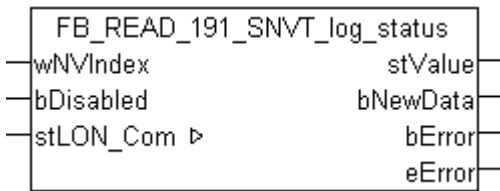
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.181 FB_READ_191_SNVT_log_status



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_log_status.

SNVT number: 191.

SNVT description: Log status (hundredths of second). Reports the current status of a data log. Updated based on the cpLogNotificationThreshold value. Reports status only; alarms reported via Node Object nvoAlarm2 output. Required if the Node Object does not include an nvoLogStat output.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_log_status;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_log_status \[▶ 572\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

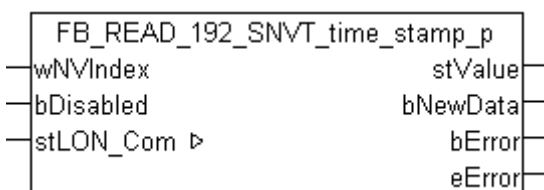
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.182 FB_READ_192_SNVT_time_stamp_p



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_time_stamp_p.

SNVT number: 192.

SNVT description: Precision timestamp (seconds). Timestamp with hundredths of a second resolution.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : Timestruct;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

stValue: Structure, to be received (see [Timestruct](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

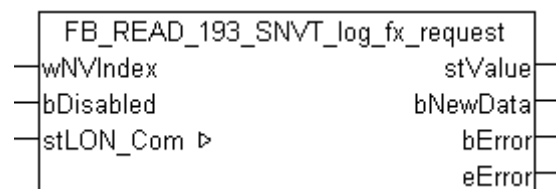
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.183 FB_READ_193_SNVT_log_fx_request



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_log_fx_request.

SNVT number: 193.

SNVT description: Log file transfer request. Requests a data log to be transferred via FTP. Must be followed by a standard FTP request to get the data log file. Required on devices implementing the Data Logger functional profile that support data log transfer via FTP.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue      : ST_LON_SNVT_log_fx_request;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_log_fx_request \[▶ 572\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

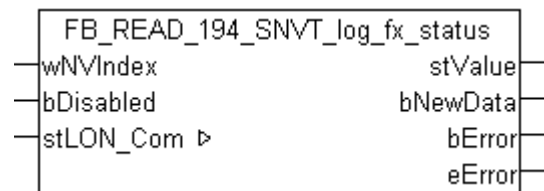
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.184 FB_READ_194_SNVT_log_fx_status



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_log_fx_status.

SNVT number: 194.

SNVT description: Log file transfer status. Reports the status of a data log file transfer using FTP. Required on devices implementing the Data Logger functional profile that support data log transfer via FTP.

VAR_INPUT

```
wNVIndex     : WORD;
bDisabled    : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue      : ST_LON_SNVT_log_fx_status;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_log_fx_status \[▶ 572\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

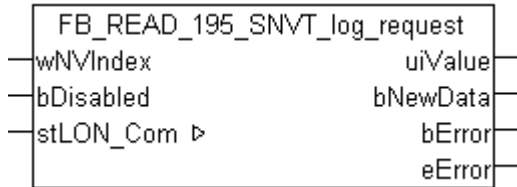
bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [[▶ 66](#)] with the send/receive function blocks (see [ST_LON_Communication](#) [[▶ 557](#)]).

7.2.185 FB_READ_195_SNVT_log_request

This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_log_request.

SNVT number: 195.

SNVT description: Log status request. Requests the current status of a data log. Status is reported by a SNVT_log_status output.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
uiValue       : UINT;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

uiValue: Min: 0 / Max: 65535.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

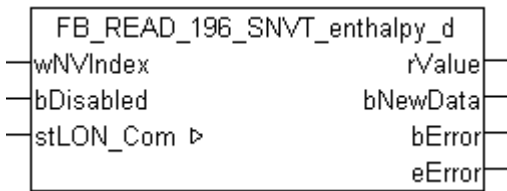
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [[▶ 463](#)]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [[▶ 66](#)] with the send/receive function blocks (see [ST_LON_Communication](#) [[▶ 557](#)]).

7.2.186 FB_READ_196_SNVT_enthalpy_d



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_enthalpy_d.

SNVT number: 196.

SNVT description: Enthalpy difference (kJ/kg).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

rValue: Min: -327.68 / Max: 327.66.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

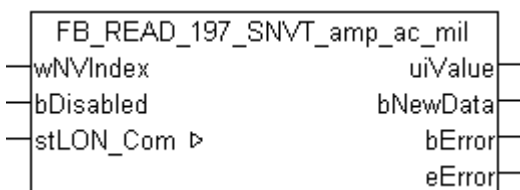
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.187 FB_READ_197_SNVT_amp_ac_mil



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_amp_ac_mil.

SNVT number: 197.

SNVT description: Electrical current (milliampere).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
uiValue       : UINT;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

uiValue: Min: 0 / Max: 65535.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

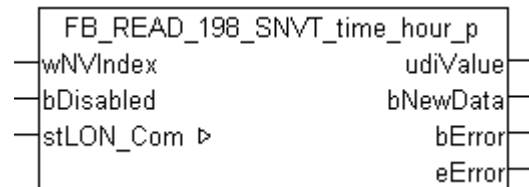
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.188 FB_READ_198_SNVT_time_hour_p



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_time_hour_p.

SNVT number: 198.

SNVT description: Time hour (hour).

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
udiValue       : UDINT;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

udiValue: Min: 0 / Max: 4294967294.

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

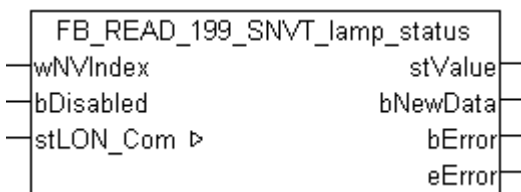
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.189 FB_READ_199_SNVT_lamp_status



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_lamp_status.

SNVT number: 199.

SNVT description: Lamp status.

VAR_INPUT

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue : ST_LON_SNVT_lamp_status;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_lamp_status \[▶ 571\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

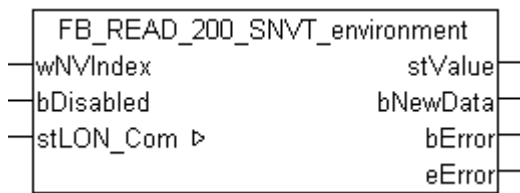
eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.190 FB_READ_200_SNVT_environment



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_environment.

SNVT number: 200.

SNVT description: Environment.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_environment;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_environment](#) [▶ 568]).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

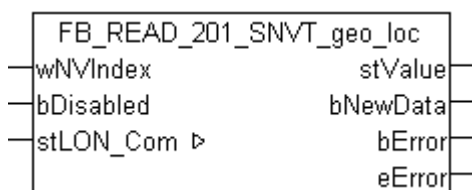
eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.191 FB_READ_201_SNVT_geo_loc



This function-block receives the following LON-input-variables (nvi):

SNVT name: SNVT_geo_loc.

SNVT number: 201.

SNVT description: Geographic location.

VAR_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
stValue       : ST_LON_SNVT_geo_loc;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

stValue: Structure, to be received (see [ST_LON_SNVT_geo_loc \[▶ 570\]](#)).

bNewData: Is TRUE for one cycle once new data were received.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.192 FB_SEND_001_SNVT_amp



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_amp.

SNVT number: 001.

SNVT description: Electric current (ampere).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
```

```
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled    : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -3276.8 / Max: 3276.7.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

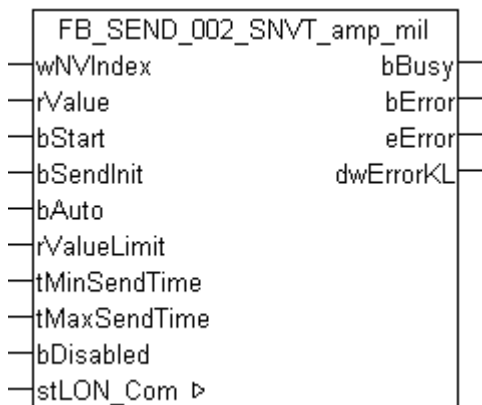
dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.193 FB_SEND_002_SNVT_amp_mil



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_amp_mil.

SNVT number: 002.

SNVT description: Electric current (milliampere).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -3276.8 / Max: 3276.7.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

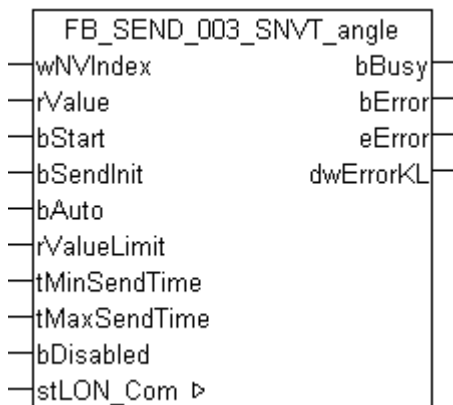
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.194 FB_SEND_003_SNVT_angle



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_angle.

SNVT number: 003.

SNVT description: Angular distance (radian).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 65.535.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

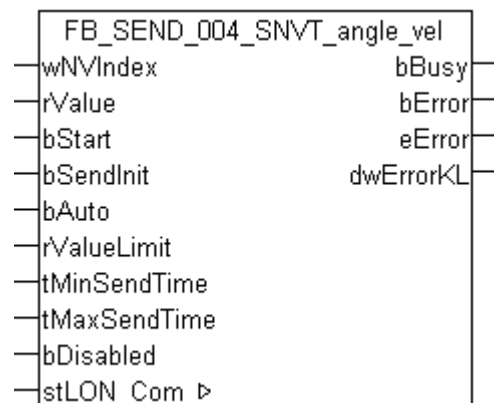
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.195 FB_SEND_004_SNVT_angle_vel



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_angle_vel.

SNVT number: 004.

SNVT description: Angular velocity (radian/second).

VAR_INPUT

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -3276.8 / Max: 3276.7.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [► 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.196 FB_SEND_005_SNVT_btu_kilo



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_btu_kilo.

SNVT number: 005.

SNVT description: Thermal energy (kilo BTU).

VAR_INPUT

```
wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

uiValue: Min: 0 / Max: 65535.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

uiValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*uiValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

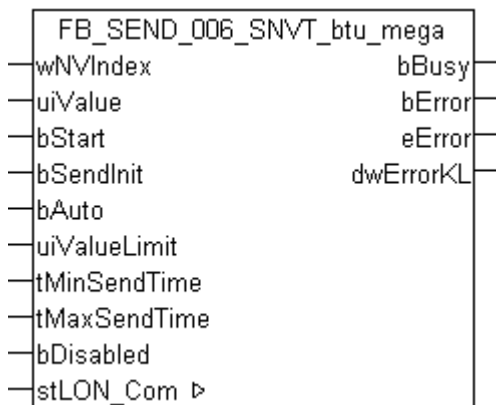
dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.197 FB_SEND_006_SNVT_btu_mega



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_btu_mega.

SNVT number: 006.

SNVT description: Thermal energy (mega BTU).

VAR_INPUT

```
wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

uiValue: Min: 0 / Max: 65535.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

uiValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*uiValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

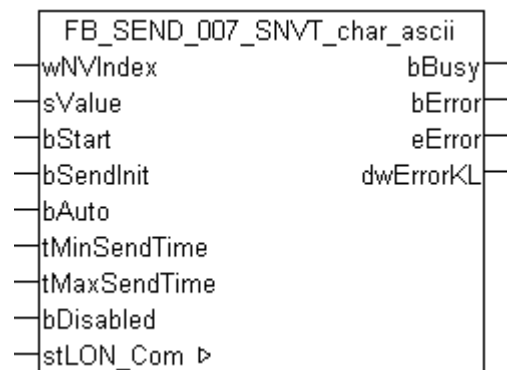
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.198 FB_SEND_007_SNVT_char_ascii



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_char_ascii.

SNVT number: 007.

SNVT description: ASCII character (8-bit ASCII character).

VAR_INPUT

```
wNVIndex : WORD;
sValue   : STRING(1);
bStart   : BOOL;
```

```

bSendInit    : BOOL := bSendInitDefault;
bAuto        : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled    : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

sValue: STRING(1).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

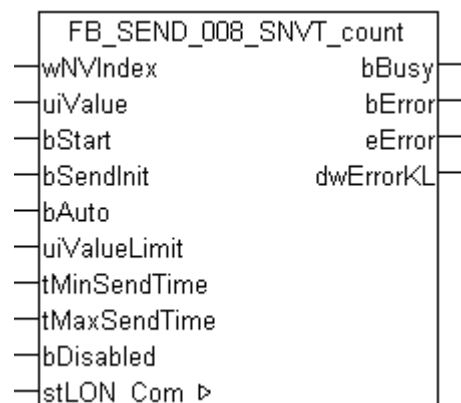
```

stLON_Com    : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.199 FB_SEND_008_SNVT_count



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_count.

SNVT number: 008.

SNVT description: Absolute count (unit).

VAR_INPUT

```
wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

uiValue: Min: 0 / Max: 65535.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

uiValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*uiValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

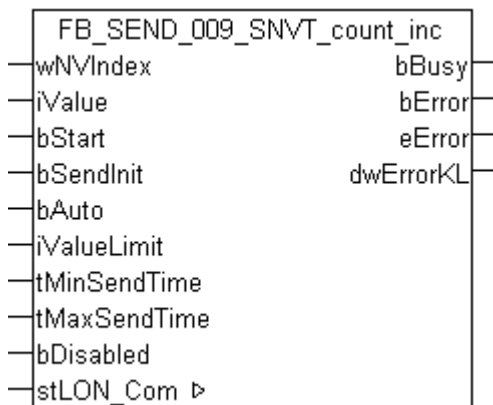
dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.200 FB_SEND_009_SNVT_count_inc



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_count_inc.

SNVT number: 009.

SNVT description: Increment count (unit (delta)).

VAR_INPUT

```
wNVIndex      : WORD;
iValue        : INT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
iValueLimit   : INT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

iValue: Min: -32768 / Max: 32767.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

iValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*iValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.201 FB_SEND_011_SNVT_date_day



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_date_day.

SNVT number: 011.

SNVT description: Day of week (day names).

VAR_INPUT

```
wNVIndex : WORD;
eValue : E_LON_days_of_week_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

eValue: Enum day of week (see [E_LON_days_of_week_t](#) [▶ 497]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

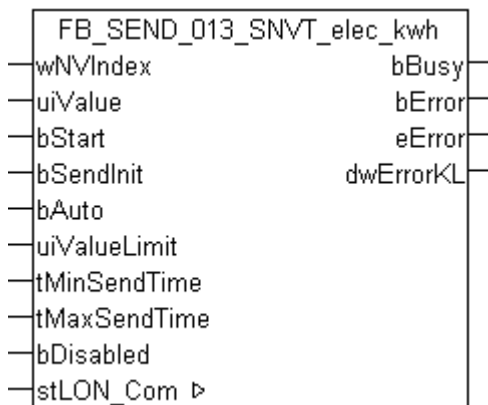
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com  : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.202 FB_SEND_013_SNVT_elec_kwh



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_elec_kwh.

SNVT number: 013.

SNVT description: Electric energy (kilowatt hour).

VAR_INPUT

```
wNVIndex   : WORD;
uiValue    : UINT;
bStart     : BOOL;
bSendInit  : BOOL := bSendInitDefault;
bAuto      : BOOL := bAutoDefault;
uiValueLimit : UINT := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled  : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

uiValue: Min: 0 / Max: 65535.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

uiValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*uiValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

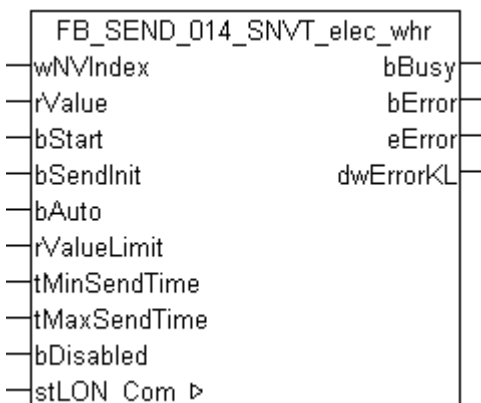
dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.203 FB_SEND_014_SNVT_elec_whr



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_elec_whr.

SNVT number: 014.

SNVT description: Electric energy (watt-hour).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrororKL   : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

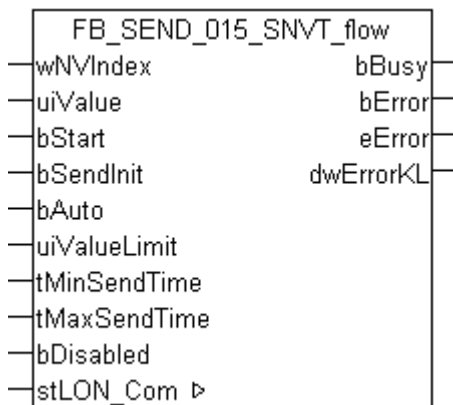
dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.204 FB_SEND_015_SNVT_flow



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_flow.

SNVT number: 015.

SNVT description: Flow volume (liter/second).

VAR_INPUT

```
wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

uiValue: Min: 0 / Max: 65535.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

uiValueLimit: Parameter for [automatic sending \[▶ 604\]](#). The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time, even if the minimum value change (*uiValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.205 FB_SEND_016_SNVT_flow_mil



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_flow_mil.

SNVT number: 016.

SNVT description: Flow volume (milliliter / second).

VAR_INPUT

```
wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto        : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

uiValue: Min: 0 / Max: 65535.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

uiValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*uiValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

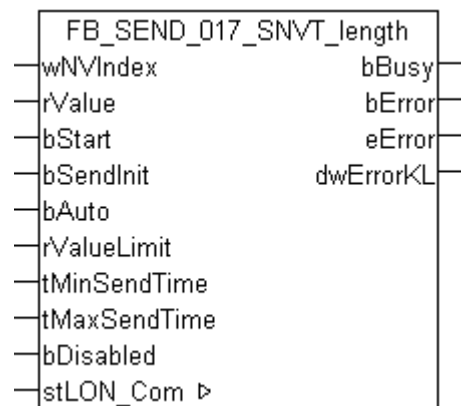
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.206 FB_SEND_017_SNVT_length



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_length.

SNVT number: 017.

SNVT description: Length (meter).

VAR_INPUT

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 6553.5

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [► 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

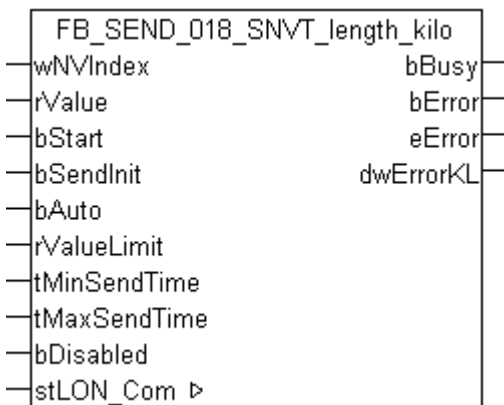
```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.207 FB_SEND_018_SNVT_length_kilo



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_length_kilo.

SNVT number: 018.

SNVT description: Length (kilometer).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 6553.5

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

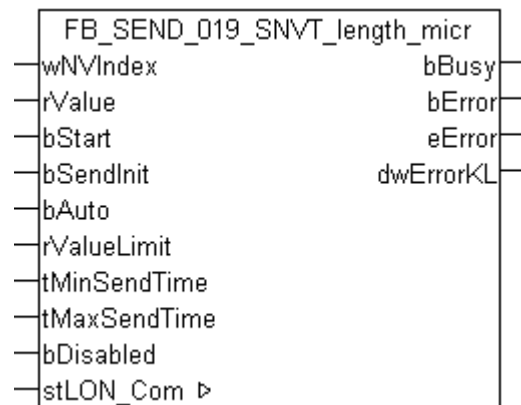
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.208 FB_SEND_019_SNVT_length_micr



This function block sends the following LON output variable (nvo):

SNVT name: SNVT_length_micr.

SNVT number: 019.

Description: Length (micrometer).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

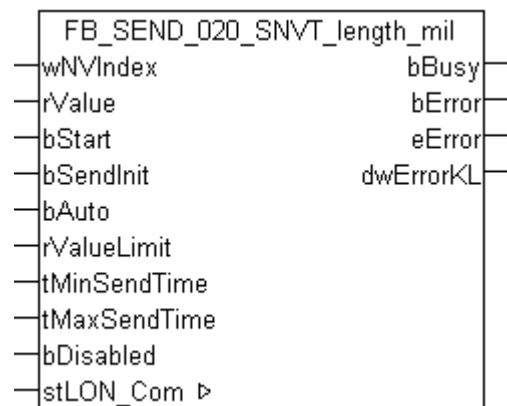
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.209 FB_SEND_020_SNVT_length_mil



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_length_mil.

SNVT number: 020.

SNVT description: Length (millimeter).

VAR_INPUT

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [► 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.210 FB_SEND_021_SNVT_lev_cont



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_lev_cont.

SNVT number: 021.

SNVT description: Continuous level (% of full level).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 100.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending \[▶ 604\]](#). The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

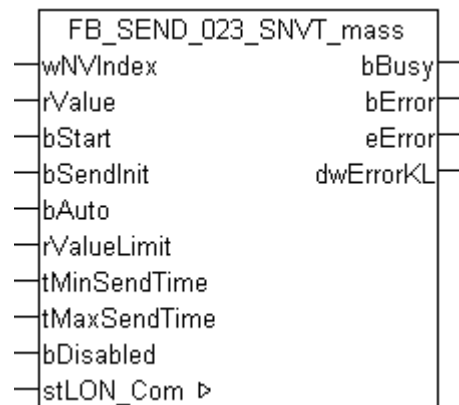
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.211 FB_SEND_023_SNVT_mass



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_mass.

SNVT number: 023.

SNVT description: pMass (gram).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.212 FB_SEND_024_SNVT_mass_kilo



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_mass_kilo.

SNVT number: 024.

SNVT description: Mass (kilogram).

VAR_INPUT

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [► 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

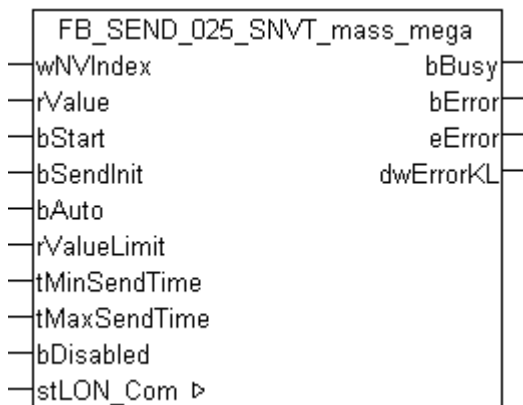
```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.213 FB_SEND_025_SNVT_mass_mega



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_mass_mega.

SNVT number: 025.

SNVT description: Mass (metric tons).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto        : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending \[▶ 604\]](#). The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.214 FB_SEND_026_SNVT_mass_mil



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_mass_mil.

SNVT number: 026.

SNVT description: Mass (milligram).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.215 FB_SEND_027_SNVT_power



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_power.

SNVT number: 027.

SNVT description: Power (Watts).

VAR_INPUT

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [► 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.216 FB_SEND_028_SNVT_power_kilo



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_power_kilo.

SNVT number: 028.

SNVT description: Power (kilowatt).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending \[▶ 604\]](#). The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

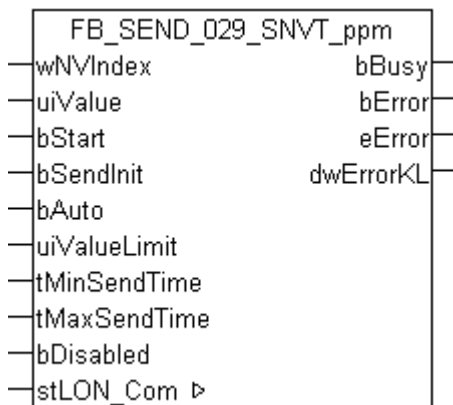
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.217 FB_SEND_029_SNVT_ppm



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_ppm.

SNVT number: 029.

SNVT description: Concentration (ppm).

VAR_INPUT

```
wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto        : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

uiValue: Min: 0 / Max: 65535.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

uiValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*uiValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.218 FB_SEND_030_SNVT_press



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_press.

SNVT number: 030.

SNVT description: Pressure (gauge) (kilopascal).

VAR_INPUT

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -3276.8 / Max: 3276.7.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [► 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

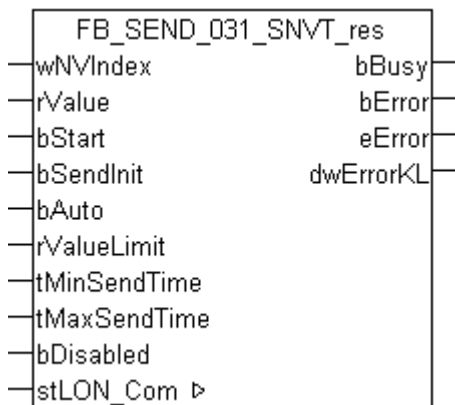
```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.219 FB_SEND_031_SNVT_res



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_res.

SNVT number: 031.

SNVT description: Electric resistance (Ohms).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending \[▶ 604\]](#). The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.220 FB_SEND_032_SNVT_res_kilo



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_res_kilo.

SNVT number: 032.

SNVT description: Electric resistance (kiloohm).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

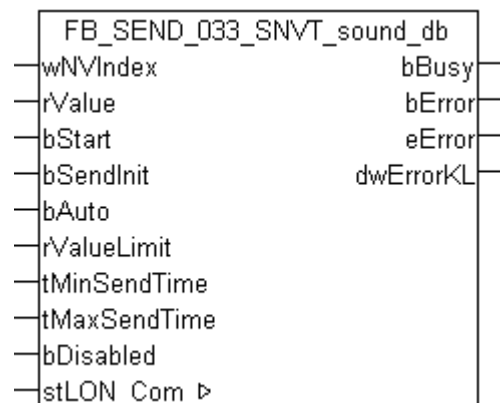
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.221 FB_SEND_033_SNVT_sound_db



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_sound_db.

SNVT number: 033.

SNVT description: Sound level (dB).

VAR_INPUT

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -327.68 / Max: 327.67.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [► 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

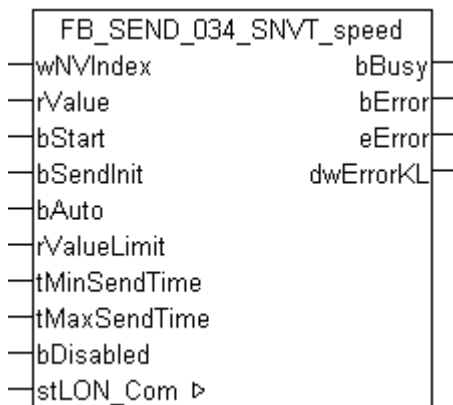
```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.222 FB_SEND_034_SNVT_speed



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_speed.

SNVT number: 034.

SNVT description: Linear velocity (meter/second).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending \[▶ 604\]](#). The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.223 FB_SEND_035_SNVT_speed_mil



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_speed_mil.

SNVT number: 035.

SNVT description: Linear velocity (meter/second).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto        : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 65.535.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

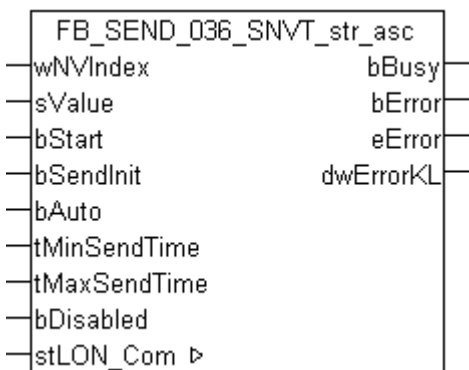
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.224 FB_SEND_036_SNVT_str_asc



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_str_asc.

SNVT number: 036.

SNVT description: Character string (30 characters max) (ASCII character string).

VAR_INPUT

```
wNVIndex   : WORD;
sValue     : STRING(31);
bStart     : BOOL;
```

```

bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

sValue: STRING(31).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy       : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
dwErrorKL   : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

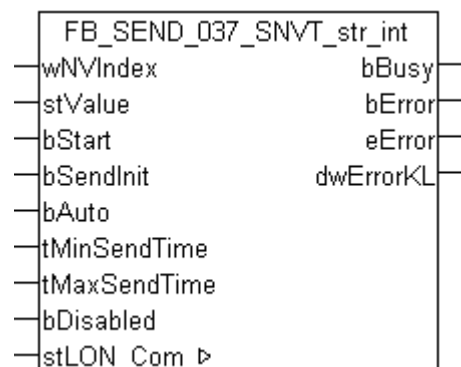
```

stLON_Com   : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.225 FB_SEND_037_SNVT_str_int



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_str_int.

SNVT number: 037.

SNVT description: Wide character string with locale code (15 characters max) (Wide character string).

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_str_int;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_str_int](#) [[▶ 581](#)]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [[▶ 604](#)] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [[▶ 604](#)]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [[▶ 604](#)]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [[▶ 463](#)]). Simultaneously *bError* is TRUE.

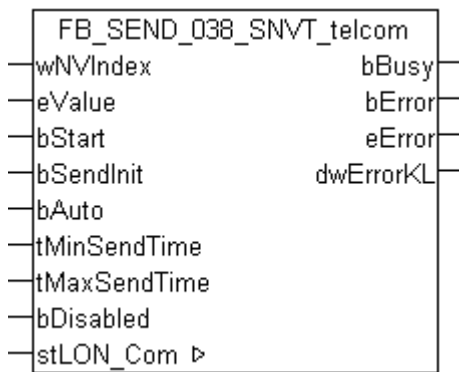
dwErrorKL: [Error identifier](#) [[▶ 604](#)] of the function block [FB_LON_KL6401\(\)](#) [[▶ 66](#)]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [[▶ 66](#)] with the send/receive function blocks (see [ST_LON_Communication](#) [[▶ 557](#)]).

7.2.226 FB_SEND_038_SNVT_telcom



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_telcom.

SNVT number: 038.

SNVT description: Telecomm states (telecomm state names).

VAR_INPUT

```
wNVIndex      : WORD;
eValue        : E_LON_telcom_states_t;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

eValue: Enum of the data to be sent ([E_LON_telcom_states_t](#) [► 527]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [► 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block `FB_LON_KL6401()` [▶ 66]. In this case the variable `eError` is set to "eKL6401_Error". Simultaneously `bError` is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects `FB_LON_KL6401()` [▶ 66] with the send/receive function blocks (see `ST_LON_Communication` [▶ 557]).

7.2.227 FB_SEND_039_SNVTemp



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_temp.

SNVT number: 039.

SNVT description: Temperature (degree celsius).

VAR_INPUT

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -274 / Max: 6279.5.

bStart: A positive edge starts sending (regardless of `bAuto`).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time `tMaxSendTime` runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

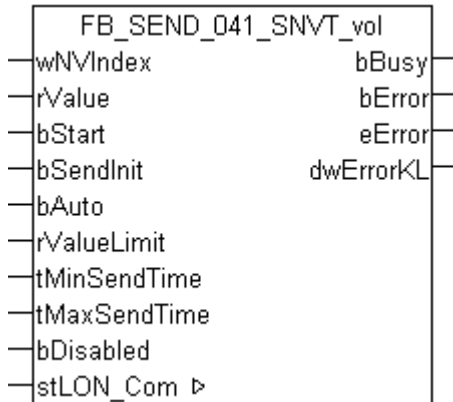
dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.228 FB_SEND_041_SNVT_vol



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_vol.

SNVT number: 041.

SNVT description: Volume (liter).

VAR_INPUT

```
wNVIndex    : WORD;
rValue      : REAL;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.229 FB_SEND_042_SNVT_vol_kilo



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_vol_kilo.

SNVT number: 042.

SNVT description: Volume (kiloliter).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrororKL   : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.230 FB_SEND_043_SNVT_vol_mil



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_vol_mil.

SNVT number: 043.

SNVT description: Volume (milliliter).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending \[▶ 604\]](#). The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

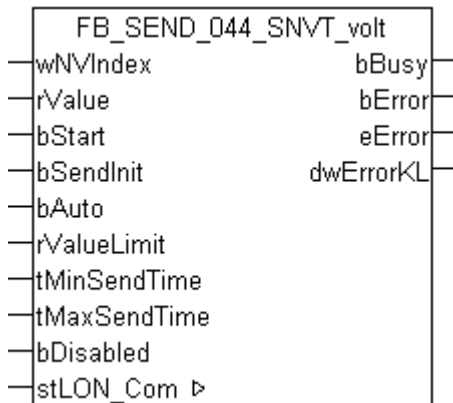
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.231 FB_SEND_044_SNVT_volt



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_volt.

SNVT number: 044.

SNVT description: Electric voltage (volt).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -3276.8 / Max: 3276.7.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

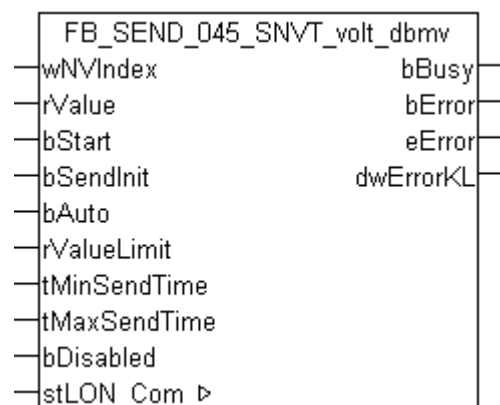
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.232 FB_SEND_045_SNVT_volt_dbmv



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_volt_dbmv.

SNVT number: 045.

SNVT description: Electric voltage (dB microvolt).

VAR_INPUT

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -327.68 / Max: 327.67.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [► 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.233 FB_SEND_046_SNVT_volt_kilo



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_volt_kilo.

SNVT number: 046.

SNVT description: Electric voltage (kilovolt).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -3276.8 / Max: 3276.7.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.234 FB_SEND_047_SNVT_volt_mil



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_volt_mil.

SNVT number: 047.

SNVT description: Electric voltage (millivolt).

VAR_INPUT

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto        : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -3276.8 / Max: 3276.7.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.235 FB_SEND_048_SNVT_amp_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_amp_f.

SNVT number: 048.

SNVT description: Electric current (ampere).

VAR_INPUT

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [► 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.236 FB_SEND_049_SNVT_angle_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_angle_f.

SNVT number: 049.

SNVT description: Angular distance (radian).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

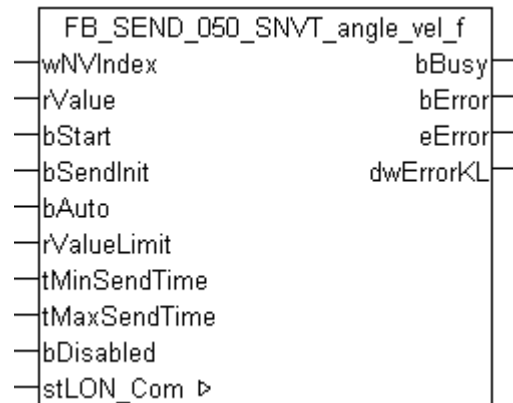
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.237 FB_SEND_050_SNVT_angle_vel_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_angle_vel_f.

SNVT number: 050.

SNVT description: Angular velocity (radian/second).

VAR_INPUT

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.238 FB_SEND_051_SNVT_count_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_count_f.

SNVT number: 051.

SNVT description: Absolute count (unit).

VAR_INPUT

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [► 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.239 FB_SEND_052_SNVT_count_inc_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_count_inc_f.

SNVT number: 052.

SNVT description: Increment count (unit (delta)).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending \[▶ 604\]](#). The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

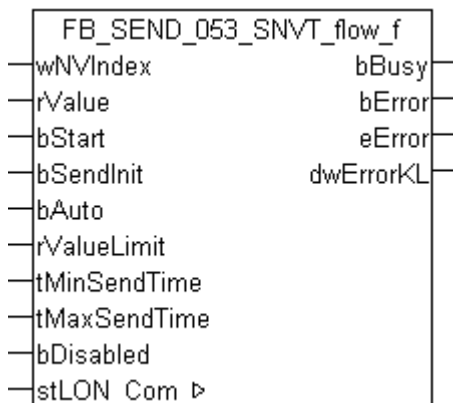
dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.240 FB_SEND_053_SNVT_flow_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_flow_f.

SNVT number: 053.

SNVT description: Flow volume (liter/second).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.241 FB_SEND_054_SNVT_length_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_length_f.

SNVT number: 054.

SNVT description: Length (meter).

VAR_INPUT

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [► 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

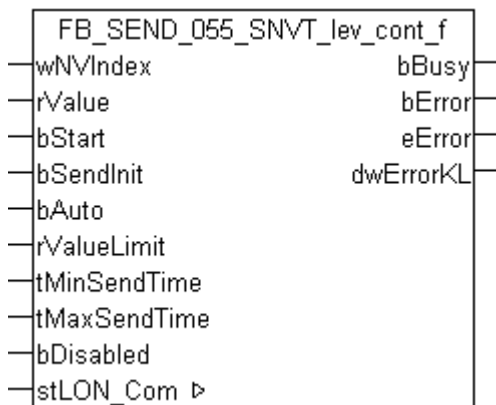
```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.242 FB_SEND_055_SNVT_lev_cont_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_lev_cont_f.

SNVT number: 055.

SNVT description: Continuous level (% of full scale).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 100.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.243 FB_SEND_056_SNVT_mass_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_mass_f.

SNVT number: 056.

SNVT description: Mass (gram).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.244 FB_SEND_057_SNVT_power_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_power_f.

SNVT number: 057.

SNVT description: Power (Watts).

VAR_INPUT

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [► 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.245 FB_SEND_058_SNVT_ppm_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_ppm_f.

SNVT number: 058.

SNVT description: Concentration (ppm).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending \[▶ 604\]](#). The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.246 FB_SEND_059_SNVT_press_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_press_f.

SNVT number: 059.

SNVT description: Pressure (gauge) (pascal).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.247 FB_SEND_060_SNVT_res_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_res_f.

SNVT number: 060.

SNVT description: Electric resistance (Ohms).

VAR_INPUT

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [► 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

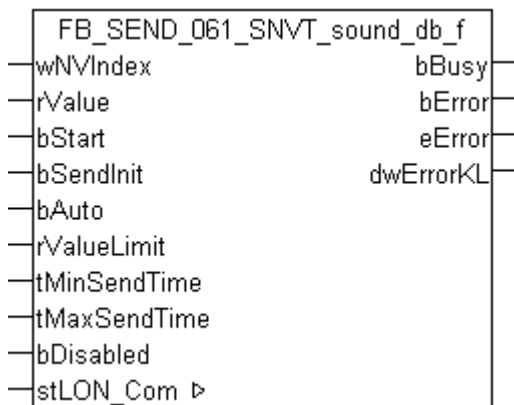
```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.248 FB_SEND_061_SNVT_sound_db_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_sound_db_f.

SNVT number: 061.

SNVT description: Sound level (dBspl).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending \[▶ 604\]](#). The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

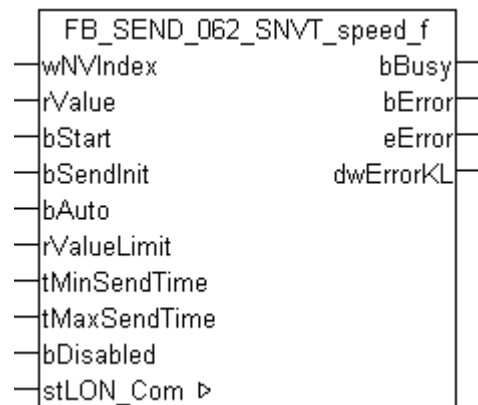
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.249 FB_SEND_062_SNVT_speed_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_speed_f.

SNVT number: 062.

SNVT description: Linear velocity (meter/second).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto        : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

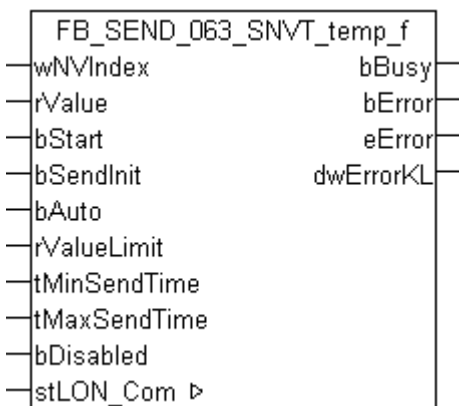
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.250 FB_SEND_063_SNVT_temp_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_temp_f.

SNVT number: 063.

SNVT description: Temperature (degree celsius).

VAR_INPUT

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -273,17 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [► 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.251 FB_SEND_064_SNVT_time_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_time_f.

SNVT number: 064.

SNVT description: Elapsed time (seconds).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending \[▶ 604\]](#). The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

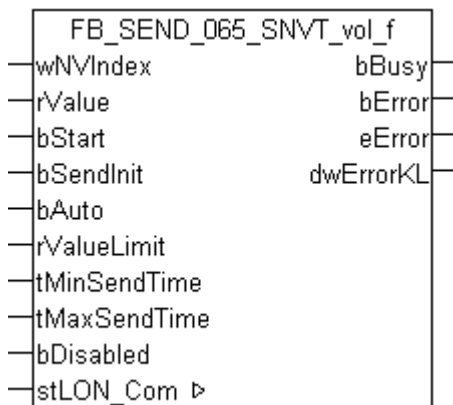
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.252 FB_SEND_065_SNVT_vol_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_vol_f.

SNVT number: 065.

SNVT description: Volume (liter).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

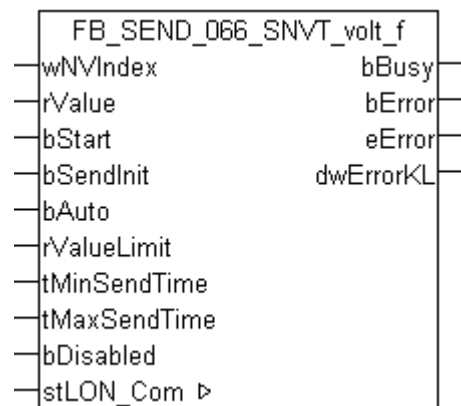
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.253 FB_SEND_066_SNVT_volt_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_volt_f.

SNVT number: 066.

SNVT description: Electric voltage (volt).

VAR_INPUT

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [► 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

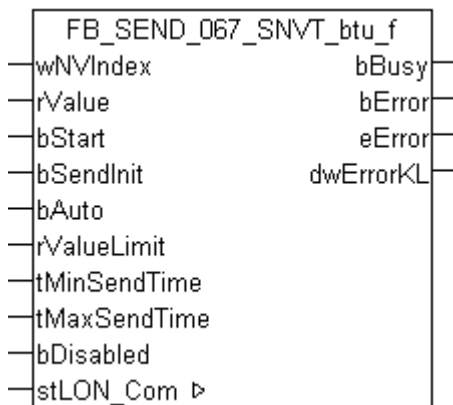
```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.254 FB_SEND_067_SNVT_btu_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_btu_f.

SNVT number: 067.

SNVT description: Thermal energy (BTU).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending \[▶ 604\]](#). The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

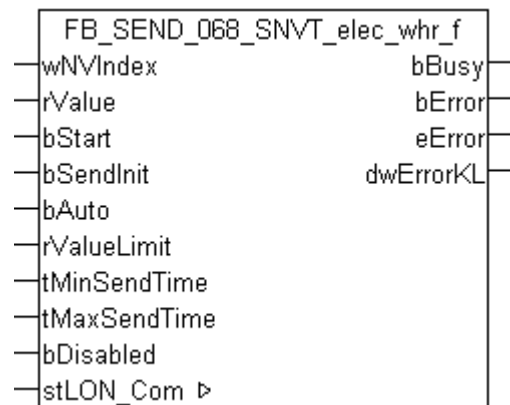
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.255 FB_SEND_068_SNVT_elec_whr_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_elec_whr_f.

SNVT number: 068.

SNVT description: Electric energy (watt-hour).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.256 FB_SEND_069_SNVT_config_src



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_config_src.

SNVT number: 069.

SNVT description: Configuration source (configuration source names).

VAR_INPUT

```
wNVIndex   : WORD;
eValue     : E_LON_config_source_t;
bStart     : BOOL;
```

```

bSendInit      : BOOL := bSendInitDefault;
bAuto          : BOOL := bAutoDefault;
tMinSendTime   : TIME := tMinSendTimeDefault;
tMaxSendTime   : TIME := tMaxSendTimeDefault;
bDisabled      : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

eValue: Enum of the data to be sent (see [E_LON_config_source_t](#) [▶ 494]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy          : BOOL;
bError         : BOOL;
eError         : E_LON_ERROR;
dwErrorKL      : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```

stLON_Com      : ST_LON_Communication;

```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.257 FB_SEND_070_SNVT_color



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_color.

SNVT number: 070.

SNVT description: CIELAB color (L*,a*,b).

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_color;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_color](#) [[▶ 564](#)]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [[▶ 604](#)] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [[▶ 604](#)]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [[▶ 604](#)]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [[▶ 463](#)]). Simultaneously *bError* is TRUE.

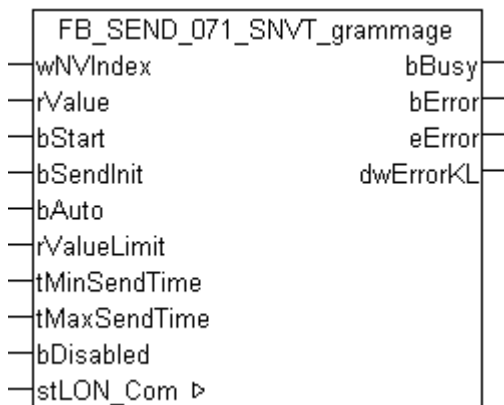
dwErrorKL: Error identifier [[▶ 604](#)] of the function block [FB_LON_KL6401\(\)](#) [[▶ 66](#)]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [[▶ 66](#)] with the send/receive function blocks (see [ST_LON_Communication](#) [[▶ 557](#)]).

7.2.258 FB_SEND_071_SNVT_grammage



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_grammage.

SNVT number: 071.

SNVT description: Grammage (gram/sq meter).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[► 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending \[► 604\]](#). The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending \[► 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[► 604\]](#). The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

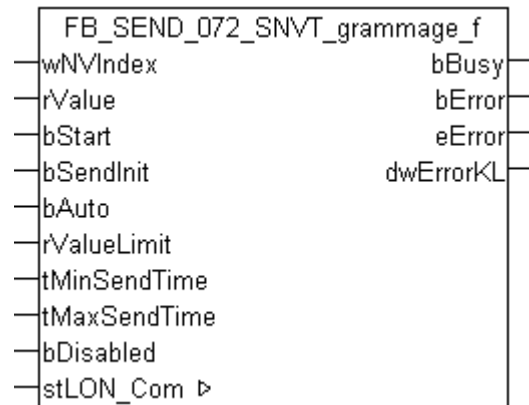
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.259 FB_SEND_072_SNVT_grammage_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_grammage_f.

SNVT number: 072.

SNVT description: Grammage (gram/sq meter).

VAR_INPUT

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

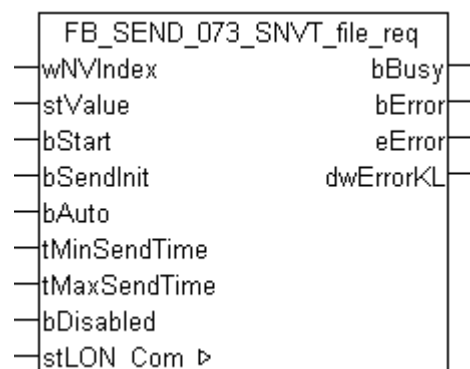
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.260 FB_SEND_073_SNVT_file_req



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_file_req.

SNVT number: 073.

SNVT description: File request.

VAR_INPUT

```
wNVIndex   : WORD;
stValue    : ST_LON_SNVT_file_req;
bStart     : BOOL;
```



```
bSendInit : BOOL := bSendInitDefault;
bAuto     : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled  : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT file req \[▶ 569\]](#)).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

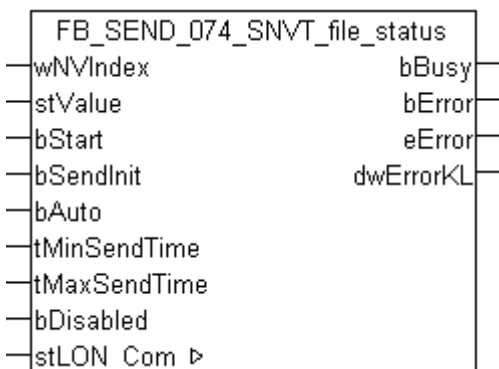
dwErrorKL: [Error identifier \[▶ 604\]](#) of the function block [FB_LON_KL6401\(\) \[▶ 66\]](#). In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.261 FB_SEND_074_SNVT_file_status



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_file_status.

SNVT number: 074.

SNVT description: File status.

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_file_status;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_file_status](#) [► 570]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [► 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier](#) [► 604] of the function block [FB_LON_KL6401\(\)](#) [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [► 66] with the send/receive function blocks (see [ST_LON_Communication](#) [► 557]).

7.2.262 FB_SEND_075_SNVT_freq_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_freq_f.

SNVT number: 075.

SNVT description: Frequency (hertz).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending \[▶ 604\]](#). The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.263 FB_SEND_076_SNVT_freq_hz



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_freq_hz.

SNVT number: 076.

SNVT description: Frequency (hertz).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

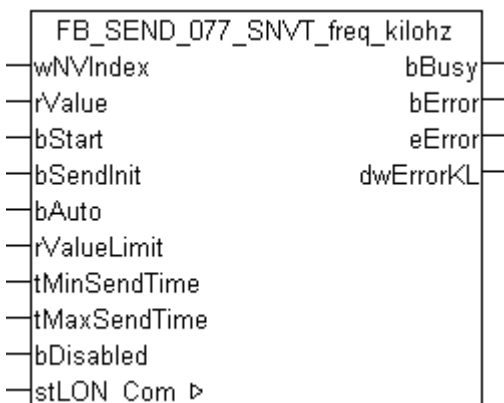
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.264 FB_SEND_077_SNVT_freq_kilohz



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_freq_kilohz.

SNVT number: 077.

SNVT description: Frequency (kilohertz).

VAR_INPUT

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [► 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.265 FB_SEND_078_SNVT_freq_milhz



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_freq_milhz.

SNVT number: 078.

SNVT description: Frequency (hertz).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 6.5535.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending \[▶ 604\]](#). The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

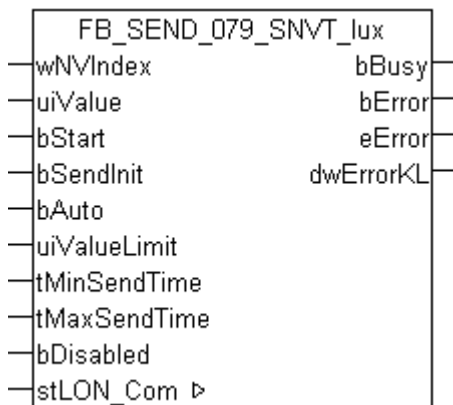
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.266 FB_SEND_079_SNVT_lux



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_lux.

SNVT number: 079.

SNVT description: Illumination (lux).

VAR_INPUT

```
wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

uiValue: Min: 0 / Max: 65535.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

uiValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*uiValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

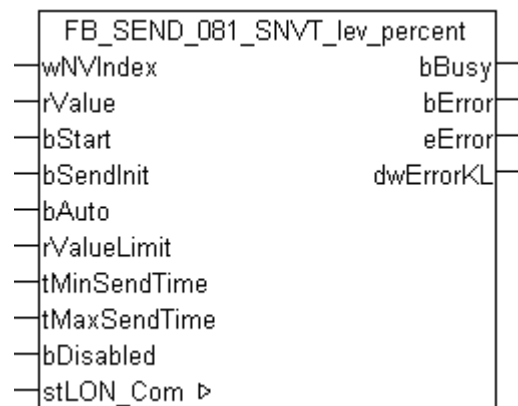
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.267 FB_SEND_081_SNVT_lev_percent



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_lev_percent.

SNVT number: 081.

SNVT description: Percentage level in 0.005 % steps. SNVT_switch should be used instead of SNVT_lev_percent, except for network variables that are used to communicate a percentage value and that require the additional resolution provided by SNVT_lev_percent; or for network variable members of functional profiles that are designed primarily for interfacing with SNVT_lev_percent members of other profiles.

VAR_INPUT

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -163.84 / Max: 163.835.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [► 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.268 FB_SEND_082_SNVT_multiplier



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_multiplier.

SNVT number: 082.

SNVT description: Multiplier (16-bit unsigned value).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 32.7675.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

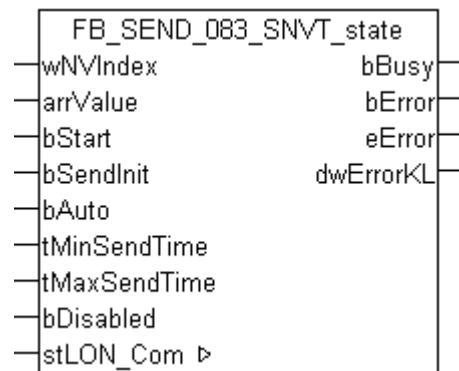
dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.269 FB_SEND_083_SNVT_state



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_state.

SNVT number: 083.

SNVT description: State vector (16 individual bit values). Each state is a boolean single bit value. SNVT_state_64 is preferred.

VAR_INPUT

```
wNVIndex      : WORD;
arrValue      : ARRAY [0..15] OF BOOL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

arrValue: 0-15 Bit.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

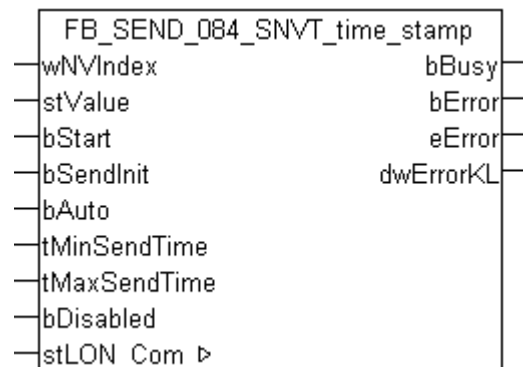
dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.270 FB_SEND_084_SNVT_time_stamp



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_time_stamp.

SNVT number: 084.

SNVT description: Time stamp (year, month, day, hour, minute, second).

VAR_INPUT

```
wNVIndex    : WORD;
stValue     : TIMESTRUCT;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [TIMESTRUCT](#)). The structure variables *wDayOfWeek* and *wMilliseconds* are not valid here and will not be transferred.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [[▶ 604](#)] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for automatic sending. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [[▶ 463](#)]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [[▶ 604](#)] of the function block [FB_LON_KL6401\(\)](#) [[▶ 66](#)]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [[▶ 66](#)] with the send/receive function blocks (see [ST_LON_Communication](#) [[▶ 557](#)]).

7.2.271 FB_SEND_085_SNVT_zerospa



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_zerospa.

SNVT number: 085.

SNVT description: Zero and span. Linear transformation parameters: multiply by the span-factor, then add the zero-term.

VAR_INPUT

```

wNVIndex      : WORD;
stValue       : ST_LON_SNVT_zerospan;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_zerospan](#) [► 584]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [► 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block [FB_LON_KL6401\(\)](#) [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [► 66] with the send/receive function blocks (see [ST_LON_Communication](#) [► 557]).

7.2.272 FB_SEND_086_SNVT_magcard



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_magcard.

SNVT number: 086.

SNVT description: ISO 7811 (40 hexadecimal digits). This data item contains data according to the ISO 7811 standard for card stripes.

VAR_INPUT

```

wNVIndex      : WORD;
arrValue      : ARRAY [0..40] OF BYTE;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

arrValue: 1-40 Byte.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.273 FB_SEND_087_SNVT_elapsed_tm



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_elapsed_tm.

SNVT number: 087.

SNVT description: Elapsed time (day, hour, minute, second, millisecond).

VAR_INPUT

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_elapsed_tm;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_elapsed_tm](#) [▶ 566]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier \[▶ 604\]](#) of the function block [FB_LON_KL6401\(\) \[▶ 66\]](#). In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

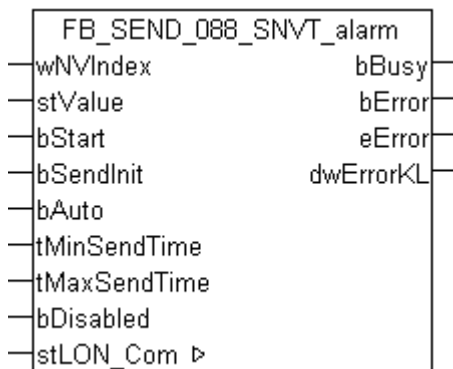
VAR_IN_OUT

```

stLON_Com  : ST_LON_Communication;

```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.274 FB_SEND_088_SNVT_alarm

This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_alarm.

SNVT number: 088.

SNVT description: Alarm status.

VAR_INPUT

```

wNVIndex   : WORD;
stValue    : ST_LON_SNVT_alarm;
bStart     : BOOL;
bSendInit  : BOOL := bSendInitDefault;
bAuto      : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled  : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_alarm \[▶ 560\]](#)).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

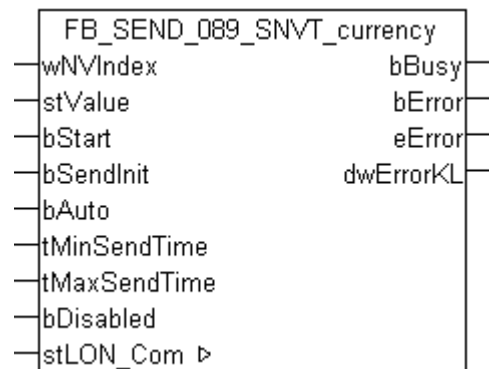
dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.275 FB_SEND_089_SNVT_currency



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_currency.

SNVT number: 089.

SNVT description: Currency (unit, magnitude, value).

VAR_INPUT

```
wNVIndex : WORD;
stValue  : ST_LON_SNVT_currency;
bStart   : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto    : BOOL := bAutoDefault;
```

```
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled    : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_currency](#) [▶ 565]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

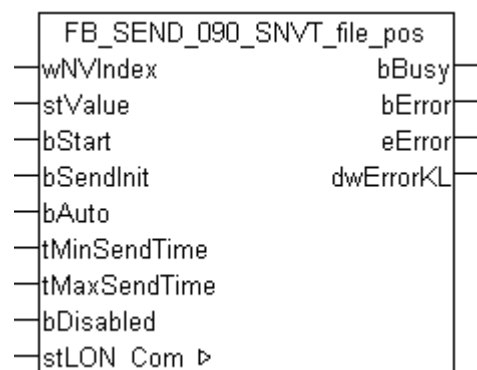
dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.276 FB_SEND_090_SNVT_file_pos



This function block sends the following LON output variable (nvo):

SNVT name: SNVT_file_pos.

SNVT number: 090.

Description: File position (pointer, length).

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_file_pos;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_file_pos \[► 569\]](#)).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[► 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending \[► 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[► 604\]](#). The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[► 463\]](#)). Simultaneously *bError* is TRUE.

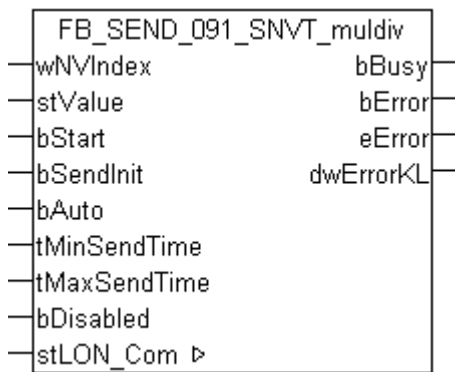
dwErrorKL: [Error identifier \[► 604\]](#) of the function block [FB_LON_KL6401\(\) \[► 66\]](#). In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[► 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[► 557\]](#)).

7.2.277 FB_SEND_091_SNVT_muldiv



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_muldiv.

SNVT number: 091.

SNVT description: Multiply/Divide (multiplier, divisor). Gain factor.

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_muldiv;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_muldiv \[► 573\]](#)).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[► 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending \[► 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[► 604\]](#). The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[► 463\]](#)). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block `FB_LON_KL6401()` [▶ 66]. In this case the variable `eError` is set to "eKL6401_Error". Simultaneously `bError` is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects `FB_LON_KL6401()` [▶ 66] with the send/receive function blocks (see `ST_LON_Communication` [▶ 557]).

7.2.278 FB_SEND_092_SNVT_obj_request



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_obj_request.

SNVT number: 092.

SNVT description: Object request (ID, request).

VAR_INPUT

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_obj_request;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see `ST_LON_SNVT_obj_request` [▶ 574]).

bStart: A positive edge starts sending (regardless of `bAuto`).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time `tMaxSendTime` runs out (polling).

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier \[▶ 604\]](#) of the function block [FB_LON_KL6401\(\) \[▶ 66\]](#). In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```

stLON_Com  : ST_LON_Communication;

```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.279 FB_SEND_093_SNVT_obj_status

This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_obj_status.

SNVT number: 093.

SNVT description: Object status (ID, status flags).

VAR_INPUT

```

wNVIndex    : WORD;
stValue     : ST_LON_SNVT_obj_status;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_obj_status \[▶ 574\]](#)).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.280 FB_SEND_094_SNVT_preset



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_preset.

SNVT number: 094.

SNVT description: Preset (mode, data, time).

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_preset;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
```

```
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled    : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_preset](#) [▶ 576]).

bStart: Event triggered sending

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

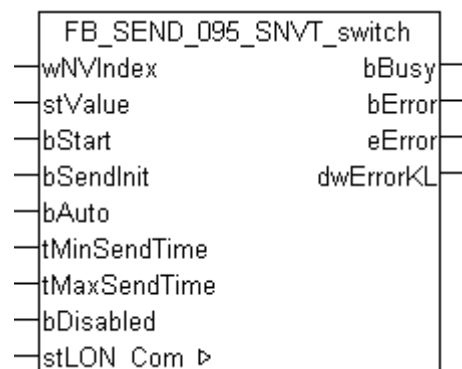
dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.281 FB_SEND_095_SNVT_switch



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_switch.

SNVT number: 095.

SNVT description: Switch (value, state).

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_switch;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_switch](#) [► 582]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [► 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [► 463]). Simultaneously *bError* is TRUE.

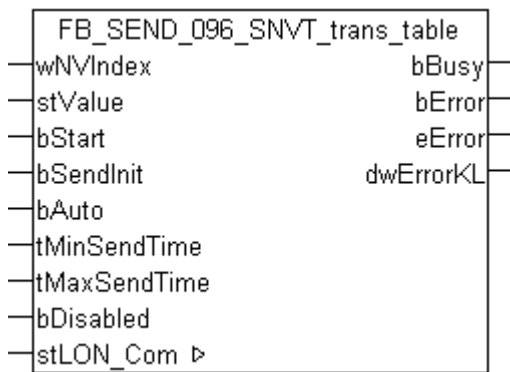
dwErrorKL: [Error identifier](#) [► 604] of the function block [FB_LON_KL6401\(\)](#) [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [► 66] with the send/receive function blocks (see [ST_LON_Communication](#) [► 557]).

7.2.282 FB_SEND_096_SNVT_trans_table



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_trans_table.

SNVT number: 096.

SNVT description: Translation table (points, interpolation).

VAR_INPUT

```

wNVIndex      : WORD;
stValue       : ST_LON_SNVT_trans_table;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
  
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_trans_table](#) [► 583]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [► 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
  
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [► 463]). Simultaneously *bError* is TRUE.

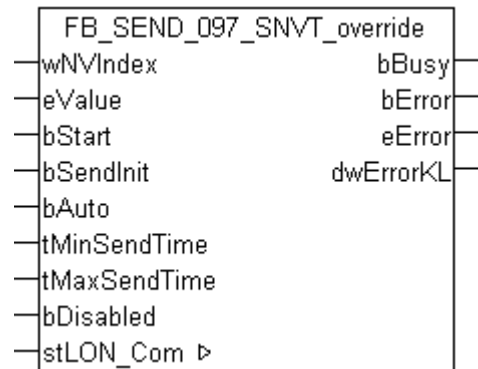
dwErrorKL: Error identifier [▶ 604] of the function block `FB_LON_KL6401()` [▶ 66]. In this case the variable `eError` is set to "eKL6401_Error". Simultaneously `bError` is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects `FB_LON_KL6401()` [▶ 66] with the send/receive function blocks (see `ST_LON_Communication` [▶ 557]).

7.2.283 FB_SEND_097_SNVT_override



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_override.

SNVT number: 097.

SNVT description: Override code (override code names).

VAR_INPUT

```
wNVIndex : WORD;
eValue : E_LON_override_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

eValue: Enum of the data to be sent (see `E_LON_override_t` [▶ 513]).

bStart: A positive edge starts sending (regardless of `bAuto`).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time `tMaxSendTime` runs out (polling).

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier \[▶ 604\]](#) of the function block [FB_LON_KL6401\(\) \[▶ 66\]](#). In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```

stLON_Com  : ST_LON_Communication;

```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.284 FB_SEND_098_SNVT_pwr_fact

This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_pwr_fact.

SNVT number: 098.

SNVT description: Power factor (multiplier).

VAR_INPUT

```

wNVIndex    : WORD;
rValue      : REAL;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
rValueLimit : REAL := 0.1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -1 / Max: 1.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.285 FB_SEND_099_SNVT_pwr_fact_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_pwr_fact_f.

SNVT number: 099.

SNVT description: Power factor (multiplier).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 0.1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -1 / Max: 1.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

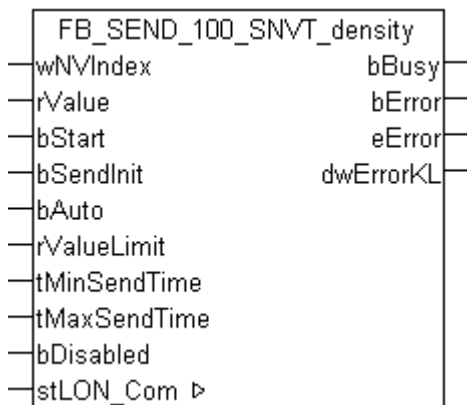
dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.286 FB_SEND_100_SNVT_density



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_density.

SNVT number: 100.

SNVT description: Density (kilogram/cubic meter).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 32767.5.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

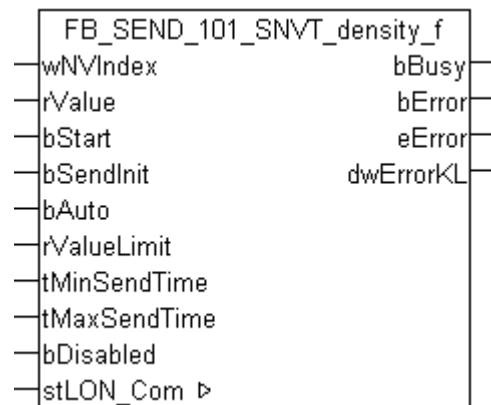
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.287 FB_SEND_101_SNVT_density_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_density_f.

SNVT number: 101.

SNVT description: Density (kilogram/cubic meter).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

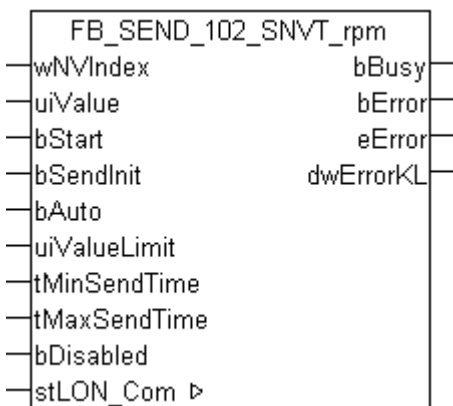
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.288 FB_SEND_102_SNVT_rpm



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_rpm.

SNVT number: 102.

SNVT description: Angular velocity (revolutions/minute (RPM)).

VAR_INPUT

```

wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

uiValue: Min: 0 / Max: 65535.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

uiValueLimit: Parameter for automatic sending [► 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time, even if the minimum value change (*uiValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

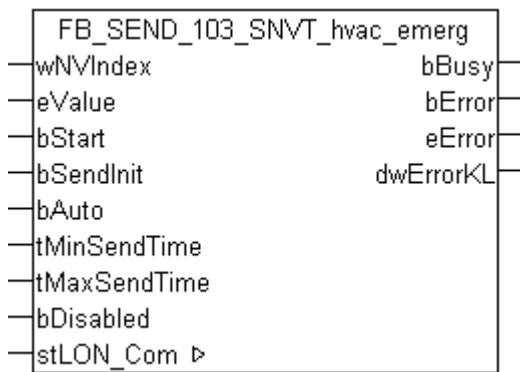
```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.289 FB_SEND_103_SNVT_hvac_emerg



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_hvac_emerg.

SNVT number: 103.

SNVT description: HVAC emergency mode (emergency mode names).

VAR_INPUT

```
wNVIndex      : WORD;
eValue        : E_LON_emerg_t;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

eValue: Enum of the data to be sent (see [E_LON_emerg_t \[▶ 500\]](#)).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

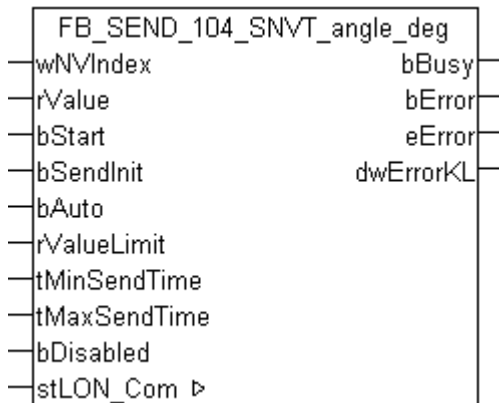
dwErrorKL: Error identifier [▶ 604] of the function block `FB_LON_KL6401()` [▶ 66]. In this case the variable `eError` is set to "eKL6401_Error". Simultaneously `bError` is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects `FB_LON_KL6401()` [▶ 66] with the send/receive function blocks (see `ST_LON_Communication` [▶ 557]).

7.2.290 FB_SEND_104_SNVT_angle_deg



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_angle_deg.

SNVT number: 104.

SNVT description: Angular distance (degree).

VAR_INPUT

```
wNVIndex : WORD;
rValue   : REAL;
bStart   : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto    : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -359.98 / Max: 360.

bStart: A positive edge starts sending (regardless of `bAuto`).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time `tMaxSendTime` runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

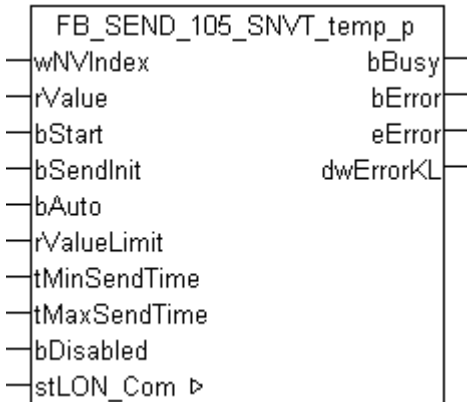
dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.291 FB_SEND_105_SNVT_temp_p



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_temp_p.

SNVT number: 105.

SNVT description: Temperature (degree celsius).

VAR_INPUT

```
wNVIndex   : WORD;
rValue     : REAL;
bStart     : BOOL;
bSendInit  : BOOL := bSendInitDefault;
bAuto      : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled  : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -273.17 / Max: 327.67.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

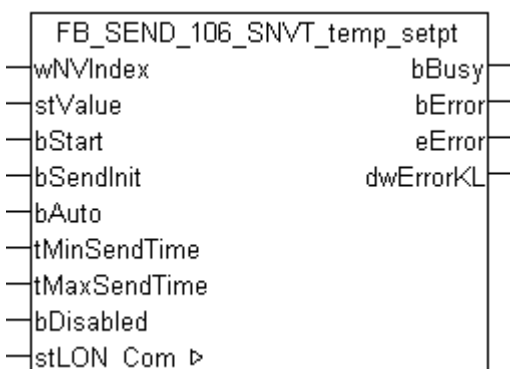
dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.292 FB_SEND_106_SNVT_temp_setpt



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_temp_setpt.

SNVT number: 106.

SNVT description: Temperature (6 temperature values).

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_temp_setpt;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_temp_setpt](#) [► 582]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [► 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier](#) [► 604] of the function block [FB_LON_KL6401\(\)](#) [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [► 66] with the send/receive function blocks (see [ST_LON_Communication](#) [► 557]).

7.2.293 FB_SEND_107_SNVT_time_sec



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_time_sec.

SNVT number: 107.

SNVT description: Elapsed time (seconds).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 6553.5.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending \[▶ 604\]](#). The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

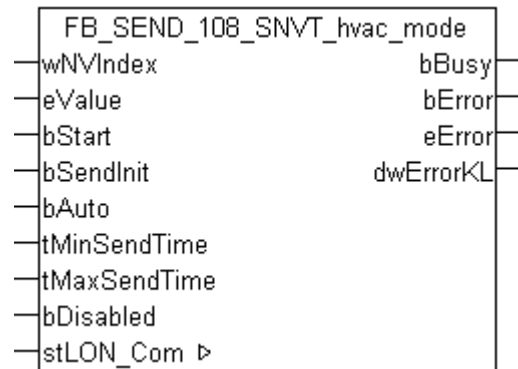
dwErrorKL: [Error identifier \[▶ 604\]](#) of the function block [FB_LON_KL6401\(\) \[▶ 66\]](#). In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.294 FB_SEND_108_SNVT_hvac_mode



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_hvac_mode.

SNVT number: 108.

SNVT description: HVAC mode (HVAC mode names).

VAR_INPUT

```
wNVIndex : WORD;
eValue : E_LON_hvac_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

eValue: Enum of the data to be sent (see [E_LON_hvac_t \[▶ 509\]](#)).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

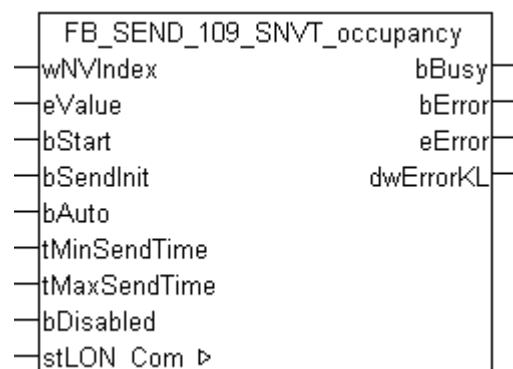
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.295 FB_SEND_109_SNVT_occupancy



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_occupancy.

SNVT number: 109.

SNVT description: Occupancy (occupancy code names).

VAR_INPUT

```
wNVIndex   : WORD;
eValue     : E_LON_occup_t;
bStart     : BOOL;
```

```

bSendInit      : BOOL := bSendInitDefault;
bAuto          : BOOL := bAutoDefault;
tMinSendTime   : TIME := tMinSendTimeDefault;
tMaxSendTime   : TIME := tMaxSendTimeDefault;
bDisabled      : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

eValue: Enum of the data to be sent (see [E_LON_occu_p t](#) [▶ 513]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy          : BOOL;
bError         : BOOL;
eError         : E_LON_ERROR;
dwErrorKL      : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

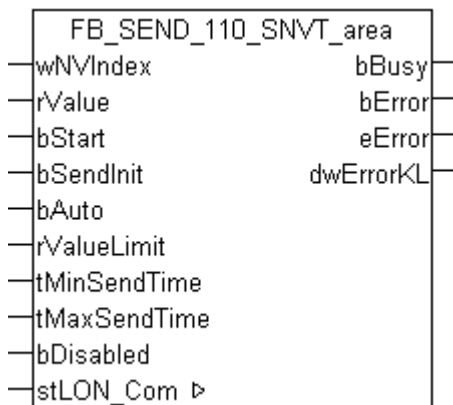
```

stLON_Com      : ST_LON_Communication;

```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.296 FB_SEND_110_SNVT_area



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_area.

SNVT number: 110.

SNVT description: Area (square meter).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 13.107.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

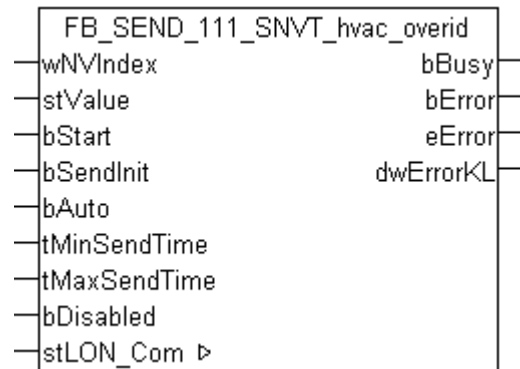
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.297 FB_SEND_111_SNVT_hvac_overid



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_hvac_overid.

SNVT number: 111.

SNVT description: HVAC override (state, pct, flow).

VAR_INPUT

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_hvac_overid;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_hvac_overid](#) [▶ 570]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

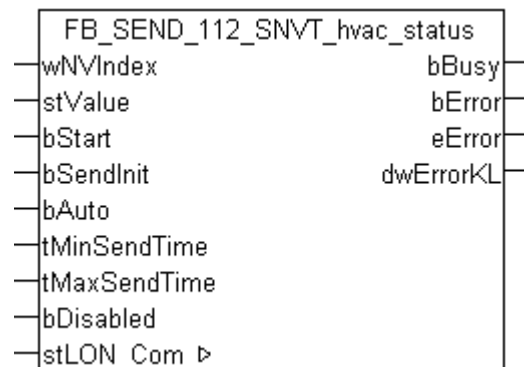
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.298 FB_SEND_112_SNVT_hvac_status



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_hvac_status.

SNVT number: 112.

SNVT description: HVAC status (mode, 5 percents, flag).

VAR_INPUT

```
wNVIndex    : WORD;
stValue     : ST_LON_SNVT_hvac_status;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_hvac_status](#) [▶ 571]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

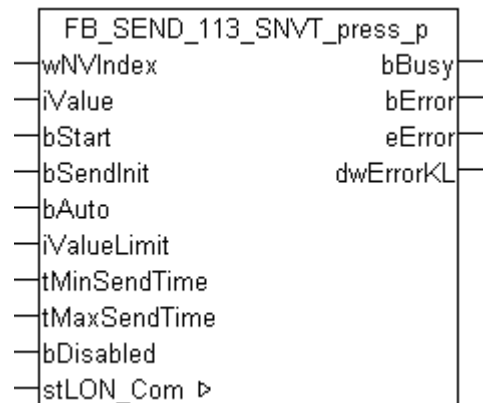
dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.299 FB_SEND_113_SNVT_press_p



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_press_p.

SNVT number: 113.

SNVT description: Pressure (gauge) (pascal).

VAR_INPUT

```

wNVIndex      : WORD;
iValue        : INT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
iValueLimit   : INT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

iValue: Min: -32768 / Max: 32767.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

iValueLimit: Parameter for automatic sending [► 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time, even if the minimum value change (*iValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.300 FB_SEND_114_SNVT_address



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_address.

SNVT number: 114.

SNVT description: Neuron address (16-bit address value).

VAR_INPUT

```
wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

uiValue: Min: 16384 / Max: 64767.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

uiValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*uiValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

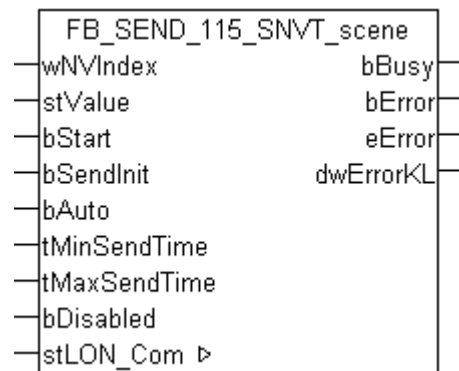
dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.301 FB_SEND_115_SNVT_scene



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_scene.

SNVT number: 115.

SNVT description: Scene control (function, scene number).

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_scene;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_scene](#) [▶ 580]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

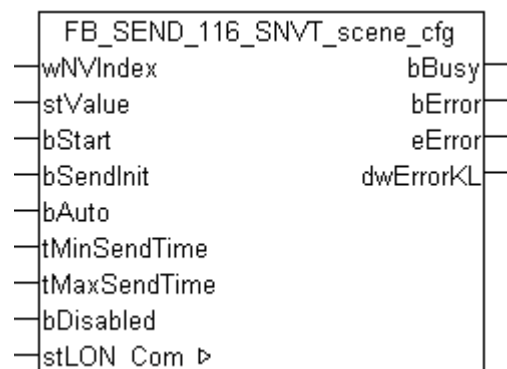
dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.302 FB_SEND_116_SNVT_scene_cfg



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_scene_cfg.

SNVT number: 116.

SNVT description: Scene configuration (function, scene number, setting, rotation, fade, delay, priority).

VAR_INPUT

```
wNVIndex    : WORD;
stValue     : ST_LON_SNVT_scene_cfg;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_scene_cfg](#) [▶ 581]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

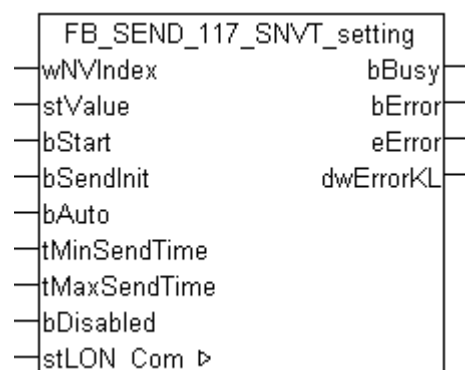
dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.303 FB_SEND_117_SNVT_setting



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_setting.

SNVT number: 117.

SNVT description: Setting control (function, setting, rotation).

VAR_INPUT

```

wNVIndex      : WORD;
stValue       : ST_LON_SNVT_setting;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_setting](#) [▶ 581]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

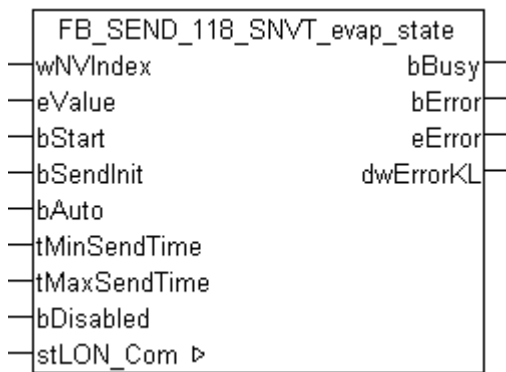
```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.304 FB_SEND_118_SNVT_evap_state



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_evap_state.

SNVT number: 118.

SNVT description: Evaporator state (evaporator state names).

VAR_INPUT

```
wNVIndex      : WORD;
eValue        : E_LON_evap_t;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

eValue: Enum of the data to be sent (see [E_LON_evap_t](#) [▶ 502]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

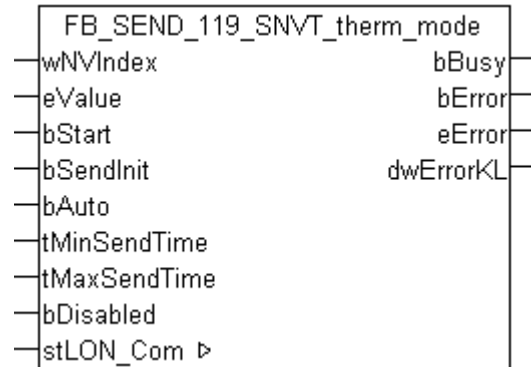
dwErrorKL: Error identifier [▶ 604] of the function block `FB_LON_KL6401()` [▶ 66]. In this case the variable `eError` is set to "eKL6401_Error". Simultaneously `bError` is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects `FB_LON_KL6401()` [▶ 66] with the send/receive function blocks (see `ST_LON_Communication` [▶ 557]).

7.2.305 FB_SEND_119_SNVT_therm_mode



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_therm_mode.

SNVT number: 119.

SNVT description: Thermostat mode (thermostat mode names).

VAR_INPUT

```
wNVIndex : WORD;
eValue : E_LON_therm_mode_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

eValue: Enum of the data to be sent (see `E_LON_therm_mode_t` [▶ 528]).

bStart: A positive edge starts sending (regardless of `bAuto`).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time `tMaxSendTime` runs out (polling).

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier \[▶ 604\]](#) of the function block [FB_LON_KL6401\(\) \[▶ 66\]](#). In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```

stLON_Com  : ST_LON_Communication;

```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.306 FB_SEND_120_SNVT_defr_mode

This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_defr_mode.

SNVT number: 120.

SNVT description: Defrost mode (defrost mode names).

VAR_INPUT

```

wNVIndex    : WORD;
eValue      : E_LON_defrost_mode_t;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

eValue: Enum of the data to be sent (see [E_LON_defrost_mode_t \[▶ 497\]](#)).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

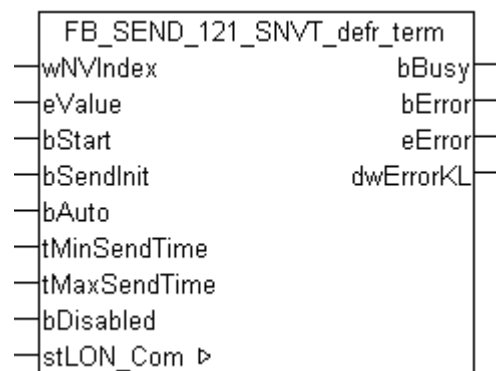
dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.307 FB_SEND_121_SNVT_defr_term



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_defr_term.

SNVT number: 121.

SNVT description: Defrost termination (defrost termination names).

VAR_INPUT

```
wNVIndex   : WORD;
eValue     : E_LON_defrost_term_t;
bStart     : BOOL;
bSendInit  : BOOL := bSendInitDefault;
bAuto     : BOOL := bAutoDefault;
```

```
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled    : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

eValue: Enum of the data to be sent (see [E_LON_defrost term t \[▶ 498\]](#)).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier \[▶ 604\]](#) of the function block [FB_LON_KL6401\(\) \[▶ 66\]](#). In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.308 FB_SEND_122_SNVT_defr_state



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_defr_state.

SNVT number: 122.

SNVT description: Defrost state (defrost state names).

VAR_INPUT

```
wNVIndex      : WORD;
eValue        : E_LON_defrost_state_t;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

eValue: Enum of the data to be sent (see [E_LON_defrost_state_t](#) [▶ 498]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.309 FB_SEND_123_SNVT_time_min



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_time_min.

SNVT number: 123.

SNVT description: Elapsed time (minute).

VAR_INPUT

```
wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

uiValue: Min: 0 / Max: 65535.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

uiValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*uiValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

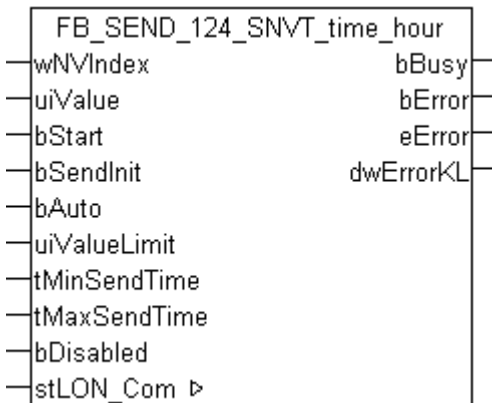
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.310 FB_SEND_124_SNVT_time_hour



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_time_hour.

SNVT number: 124.

SNVT description: Elapsed time (hour).

VAR_INPUT

```
wNVIndex : WORD;
uiValue : UINT;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
uiValueLimit : UINT := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

uiValue: Min: 0 / Max: 65535.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

uiValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*uiValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

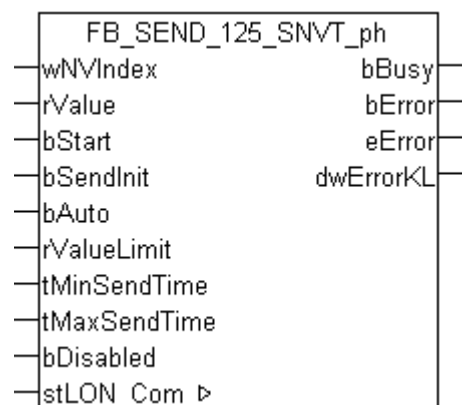
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.311 FB_SEND_125_SNVT_ph



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_ph.

SNVT number: 125.

SNVT description: Acidity (pH). Ratio of concentration of ions.

VAR_INPUT

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -32.768 / Max: 32.767.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [► 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

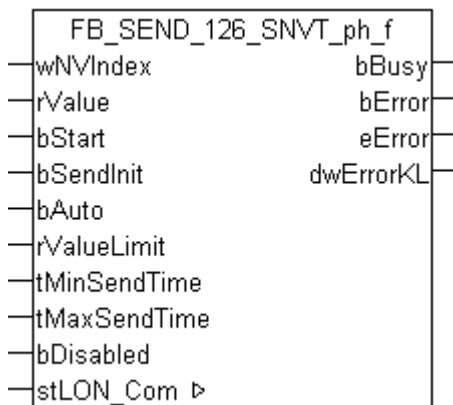
```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.312 FB_SEND_126_SNVT_ph_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_ph_f.

SNVT number: 126.

SNVT description: Acidity (pH). Ratio of concentration of ions.

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -3.40E+38 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending \[▶ 604\]](#). The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

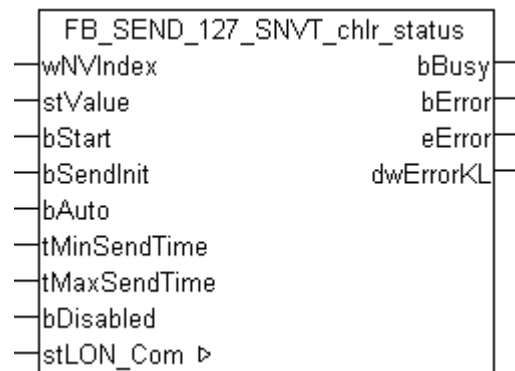
dwErrorKL: Error identifier [[▶ 604](#)] of the function block [FB_LON_KL6401\(\) \[▶ 66\]](#). In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.313 FB_SEND_127_SNVT_chlr_status



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_chlr_status.

SNVT number: 127.

SNVT description: Chiller status (run mode, op mode, state bits).

VAR_INPUT

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_chlr_status;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_chlr_status \[▶ 561\]](#)).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com  : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.314 FB_SEND_128_SNVT_tod_event



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_tod_event.

SNVT number: 128.

SNVT description: Occupancy time (current, next time).

VAR_INPUT

```
wNVIndex   : WORD;
stValue    : ST_LON_SNVT_tod_event;
bStart     : BOOL;
bSendInit  : BOOL := bSendInitDefault;
bAuto      : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled  : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_tod_event](#) [▶ 583]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

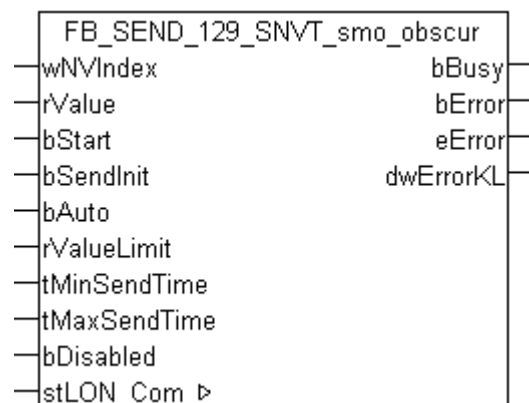
dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.315 FB_SEND_129_SNVT_smo_obscur



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_smo_obscur.

SNVT number: 129.

SNVT description: Smoke obscuration (percent obscuration).

VAR_INPUT

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 5.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [► 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

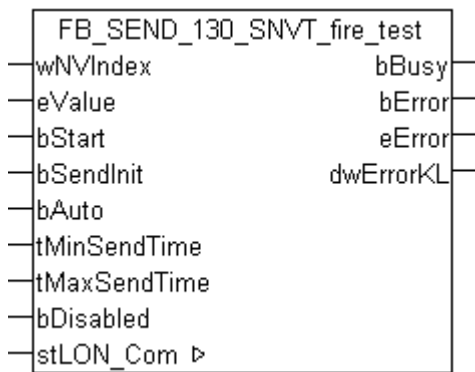
```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.316 FB_SEND_130_SNVT_fire_test



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_fire_test.

SNVT number: 130.

SNVT description: Fire test request (fire test names).

VAR_INPUT

```
wNVIndex      : WORD;
eValue        : E_LON_fire_test_t;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

eValue: Enum of the data to be sent ([E_LON_fire_test_t \[► 505\]](#)).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[► 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending \[► 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[► 604\]](#). The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[► 463\]](#)). Simultaneously *bError* is TRUE.

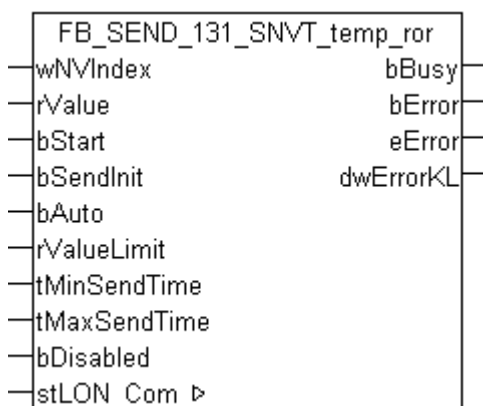
dwErrorKL: Error identifier [▶ 604] of the function block `FB_LON_KL6401()` [▶ 66]. In this case the variable `eError` is set to "eKL6401_Error". Simultaneously `bError` is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects `FB_LON_KL6401()` [▶ 66] with the send/receive function blocks (see `ST_LON_Communication` [▶ 557]).

7.2.317 FB_SEND_131_SNVT_temp_ror



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_temp_ror.

SNVT number: 131.

SNVT description: Temperature rate of change/rise (degree celsius/minute).

VAR_INPUT

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -16384 / Max: 16383.5.

bStart: A positive edge starts sending (regardless of `bAuto`).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time `tMaxSendTime` runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

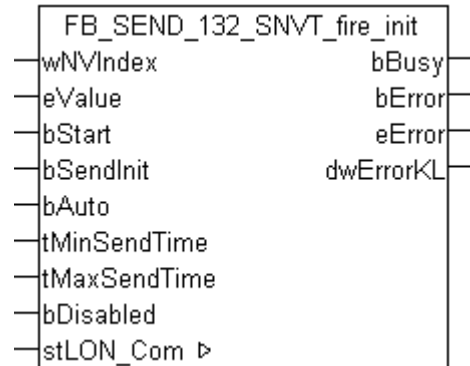
dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.318 FB_SEND_132_SNVT_fire_init



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_fire_init.

SNVT number: 132.

SNVT description: Fire initiator type (fire initiator type names).

VAR_INPUT

```
wNVIndex    : WORD;
eValue      : E_LON_fire_initiator_t;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

eValue: Enum of the data to be sent (see [E_LON_fire_initiator](#) t [▶ 504]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

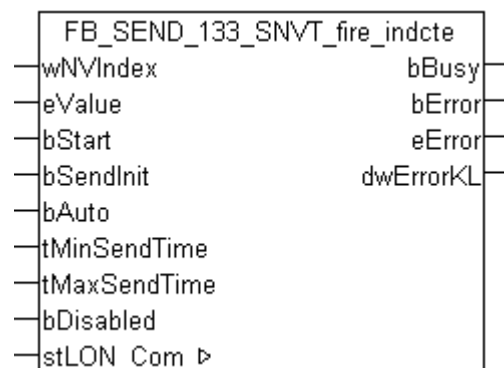
dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.319 FB_SEND_133_SNVF_fire_indcte



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_fire_indcte.

SNVT number: 133.

SNVT description: Fire indicator type (fire indicator type names).

VAR_INPUT

```

wNVIndex      : WORD;
eValue        : E_LON_fire_indicator_t;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

eValue: Enum of the data to be sent (see [E_LON_fire_indicator_t](#) [► 504]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [► 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier](#) [► 604] of the function block [FB_LON_KL6401\(\)](#) [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [► 66] with the send/receive function blocks (see [ST_LON_Communication](#) [► 557]).

7.2.320 FB_SEND_134_SNVT_time_zone



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_time_zone.

SNVT number: 134.

SNVT description: Time zone descriptor (offset, type, startDST, endDST).

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_time_zone;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_time_zone](#) [▶ 583]).

If *stValue.eType_of_description* = eLON_CAL_GREG (0) the variables *stValue.stStart_DST.uiG_day_of_start_DST* and *stValue.stEnd_DST.uiG_day_of_end_DST* are sent.

If *stValue.eType_of_description* = eLON_CAL_JUL (1) the variables *stValue.stStart_DST.uiJ_day_of_start_DST* and *stValue.stEnd_DST.uiJ_day_of_end_DST* are sent.

If *stValue.eType_of_description* = eLON_CAL_MEU(2) the variables *stValue.stStart_DST.stM_start_DST* and *stValue.stEnd_DST.stM_end_DST* are sent.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

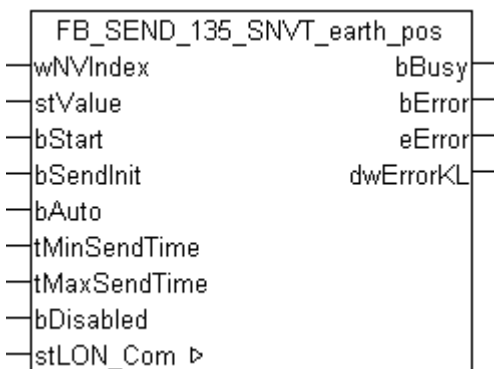
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.321 FB_SEND_135_SNVT_earth_pos



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_earth_pos.

SNVT number: 135.

SNVT description: Earth position (lat & long direction, latitude deg & min, longitude deg & min, height).

VAR_INPUT

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_earth_pos;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_earth_pos](#) [▶ 566]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

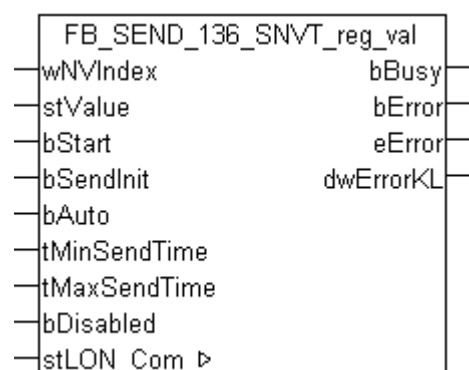
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com  : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.322 FB_SEND_136_SNVT_reg_val



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_reg_val

SNVT number: 136

SNVT description: Register value (raw value, unit code, number of decimals).

VAR_INPUT

```
wNVIndex   : WORD;
stValue    : ST_LON_SNVT_reg_val;
bStart     : BOOL;
bSendInit  : BOOL := bSendInitDefault;
bAuto      : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled  : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_reg_val](#) [▶ 579]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.323 FB_SEND_137_SNVT_reg_val_ts



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_reg_val_ts

SNVT number: 137

SNVT description: Register value (raw value, unit code, number of decimals, status, state, timestamp).

VAR_INPUT

```

wNVIndex      : WORD;
stValue       : ST_LON_SNVT_reg_val_ts;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_reg_val_ts](#) [► 580]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [► 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier](#) [► 604] of the function block [FB_LON_KL6401\(\)](#) [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

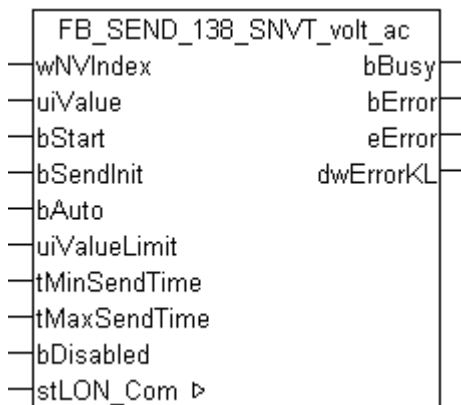
```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [► 66] with the send/receive function blocks (see [ST_LON_Communication](#) [► 557]).

7.2.324 FB_SEND_138_SNVT_volt_ac



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_volt_ac.

SNVT number: 138.

SNVT description: Voltage in alternating current (volt AC).

VAR_INPUT

```
wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

uiValue: Min: 0 / Max: 65535.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

uiValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*uiValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.325 FB_SEND_139_SNVT_amp_ac



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_amp_ac.

SNVT number: 139.

SNVT description: Amperage in alternating current (ampere AC).

VAR_INPUT

```
wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto        : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

uiValue: Min: 0 / Max: 65535.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

uiValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*uiValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

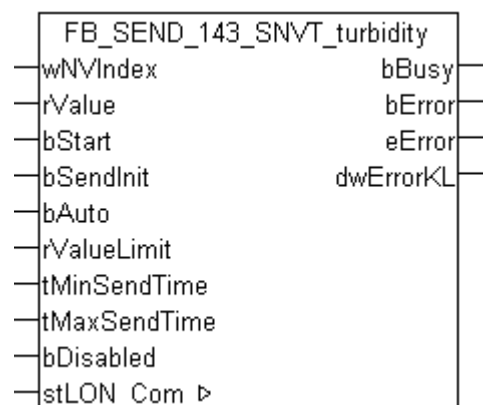
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.326 FB_SEND_143_SNVT_turbidity



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_turbidity.

SNVT number: 143.

SNVT description: Turbidity (nephelometric turbidity unit).

VAR_INPUT

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 65.535.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [► 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

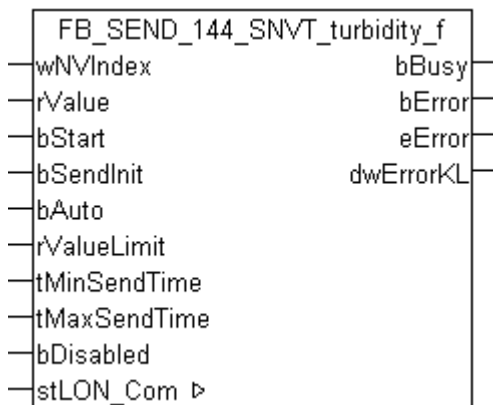
```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.327 FB_SEND_144_SNVT_turbidity_f



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_turbidity_f.

SNVT number: 144.

SNVT description: Turbidity (nephelometric turbidity unit).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 3.40E+38.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.328 FB_SEND_145_SNVT_hvac_type



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_hvac_type.

SNVT number: 145.

SNVT description: HVAC unit type (HVAC unit type names).

VAR_INPUT

```
wNVIndex      : WORD;
eValue        : E_LON_hvac_hvt_t;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

eValue: Enum of the data to be sent (see [E_LON_hvac_hvt_t](#) [▶ 506]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

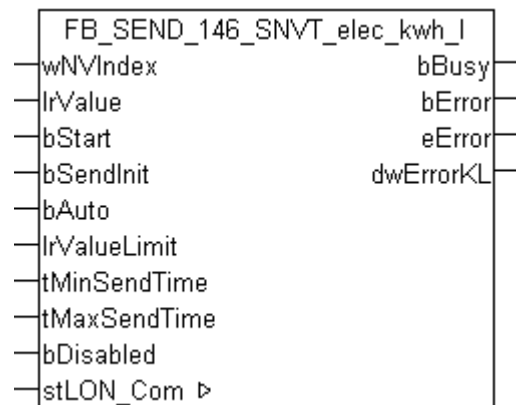
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.329 FB_SEND_146_SNVT_elec_kwh_I



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_elec_kwh_I.

SNVT number: 146.

SNVT description: Electric energy (kilowatt hour).

VAR_INPUT

```
wNVIndex    : WORD;
lrValue     : LREAL;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
lrValueLimit : LREAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

IrValue: Min: -214748364.8 / Max: 214748364.7.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

IrValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*IrValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

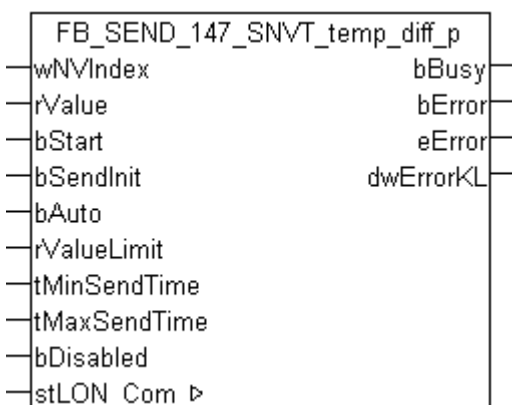
dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.330 FB_SEND_147_SNVT_temp_diff_p



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_temp_diff_p.

SNVT number: 147.

SNVT description: Temp difference (degree celsius).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -327.68 / Max: 327.67.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrororKL   : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

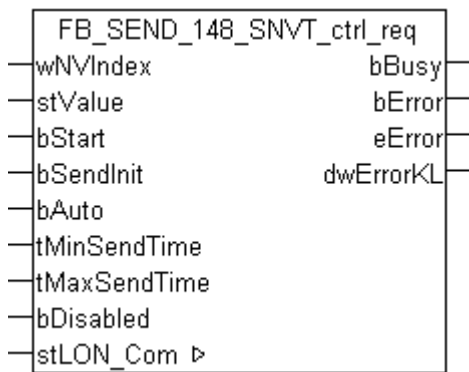
dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.331 FB_SEND_148_SNVT_ctrl_req



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_ctrl_req.

SNVT number: 148.

SNVT description: Control request (receiver ID, sender ID, sender priority).

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_ctrl_req;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_ctrl_req \[▶ 564\]](#)).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

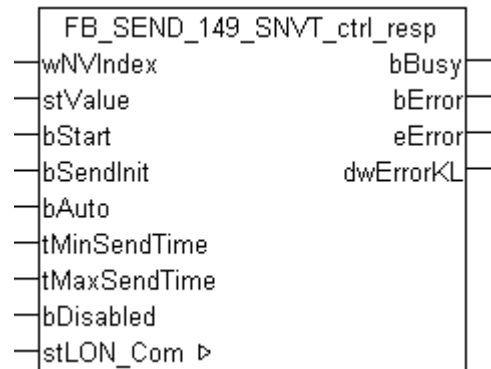
dwErrorKL: Error identifier [▶ 604] of the function block `FB_LON_KL6401()` [▶ 66]. In this case the variable `eError` is set to "eKL6401_Error". Simultaneously `bError` is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects `FB_LON_KL6401()` [▶ 66] with the send/receive function blocks (see `ST_LON_Communication` [▶ 557]).

7.2.332 FB_SEND_149_SNVT_ctrl_resp



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_ctrl_resp.

SNVT number: 149.

SNVT description: Control response (status, sender, controller ID).

VAR_INPUT

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_ctrl_resp;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see `ST_LON_SNVT_ctrl_resp`).

bStart: A positive edge starts sending (regardless of `bAuto`).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time `tMaxSendTime` runs out (polling).

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

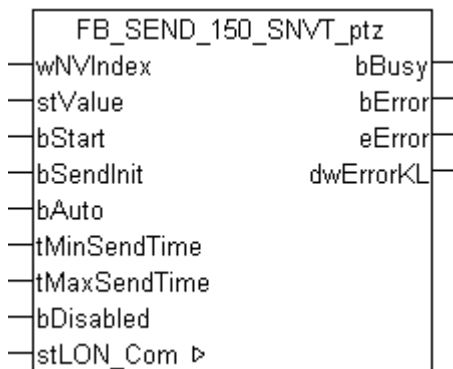
VAR_IN_OUT

```

stLON_Com  : ST_LON_Communication;

```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.333 FB_SEND_150_SNVT_ptz

This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_ptz.

SNVT number: 150.

SNVT description: Camera PTZ (pan, pan speed, tilt, tilt speed, zoom, zoom speed)

VAR_INPUT

```

wNVIndex    : WORD;
stValue     : ST_LON_SNVT_ptz;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_ptz](#) [▶ 576]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.334 FB_SEND_151_SNVT_privacyzone



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_privacyzone.

SNVT number: 151.

SNVT description: Privacy zone (action, zone number, camera ID). Certain areas may be exempted from the camera.

VAR_INPUT

```
wNVIndex   : WORD;
stValue    : ST_LON_SNVT_privacyzone;
bStart     : BOOL;
bSendInit  : BOOL := bSendInitDefault;
```

```

bAuto      : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_privacyzone](#) [▶ 576]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

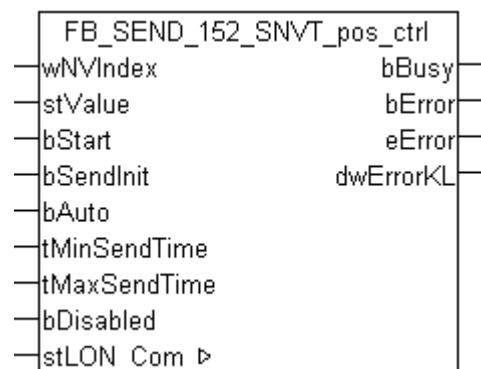
```

stLON_Com : ST_LON_Communication;

```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.335 FB_SEND_152_SNVT_pos_ctrl



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_pos_ctrl.

SNVT number: 152.

SNVT description: Position control (receiver, controller ID, controller priority, function, action, value).

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_pos_ctrl;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_pos_ctrl](#) [► 575]).

If *stValue.eFunction* = eLON_CMF_ABS the structure *stValue.stAbspos* will be sent.

If *stValue.eFunction* <> eLON_CMF_ABS the variable *stValue.byNumber* will be sent.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [► 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [► 463]). Simultaneously *bError* is TRUE.

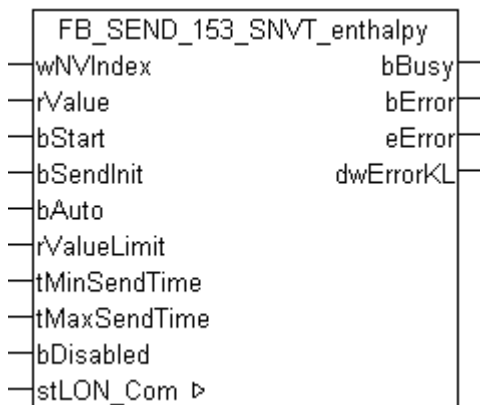
dwErrorKL: [Error identifier](#) [► 604] of the function block [FB_LON_KL6401\(\)](#) [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [► 66] with the send/receive function blocks (see [ST_LON_Communication](#) [► 557]).

7.2.336 FB_SEND_153_SNVT_enthalpy



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_enthalpy.

SNVT number: 153.

SNVT description: Enthalpy (kJ/kg).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -327.68 / Max: 327.67.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```


bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

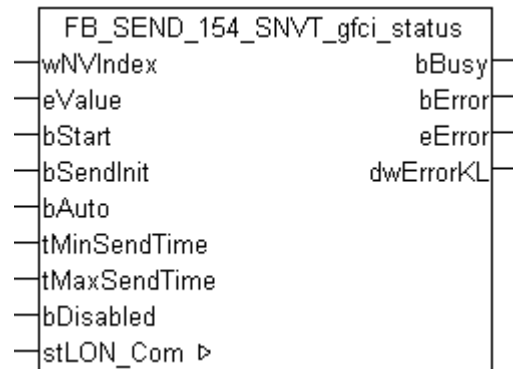
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.337 FB_SEND_154_SNVT_gfci_status



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_gfci_status.

SNVT number: 154.

SNVT description: GFCI status type (GFCI status type names).

VAR_INPUT

```
wNVIndex : WORD;
eValue : E_LON_gfci_status_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

eValue: Enum of the data to be sent (see [E_LON_gfci_status_t](#) [▶ 506]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

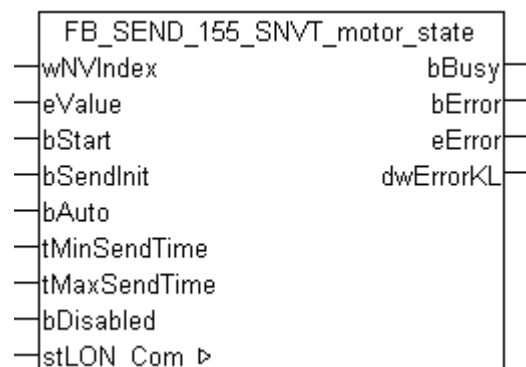
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com  : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.338 FB_SEND_155_SNVT_motor_state



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_motor_state.

SNVT number: 155.

SNVT description: Motor state (motor state names).

VAR_INPUT

```
wNVIndex   : WORD;
eValue     : E_LON_motor_state_t;
bStart     : BOOL;
bSendInit  : BOOL := bSendInitDefault;
bAuto      : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled  : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

eValue: Enum of the data to be sent (see [E_LON_motor_state t \[▶ 511\]](#)).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

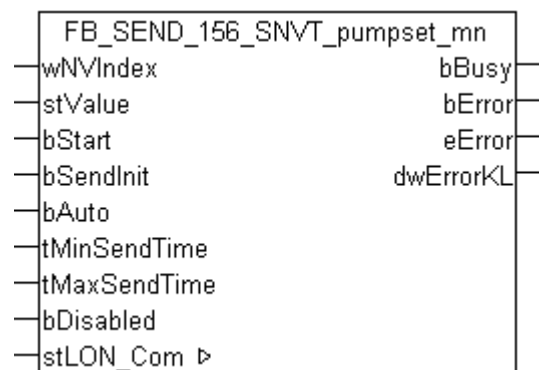
dwErrorKL: [Error identifier \[▶ 604\]](#) of the function block [FB_LON_KL6401\(\) \[▶ 66\]](#). In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.339 FB_SEND_156_SNVT_pumpset_mn



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_pumpset_mn.

SNVT number: 156.

SNVT description: Pumpset (main, booster, priority, ready, emerg, main enabled, booster enabled, maint).

VAR_INPUT

```

wNVIndex      : WORD;
stValue       : ST_LON_SNVT_pumpset_mn;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_pumpset_mn](#) [► 577]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [► 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block [FB_LON_KL6401\(\)](#) [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

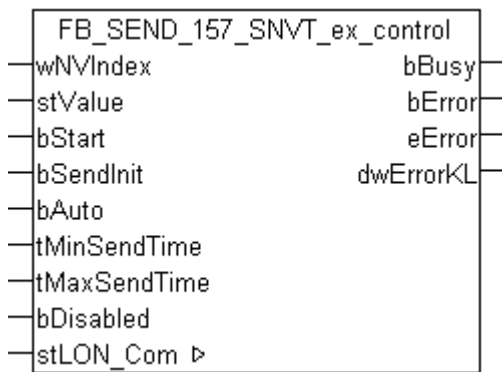
```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [► 66] with the send/receive function blocks (see [ST_LON_Communication](#) [► 557]).

7.2.340 FB_SEND_157_SNVT_ex_control



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_ex_control.

SNVT number: 157.

SNVT description: Exclusive control (status, address). A device has the exclusive control of another device.

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_ex_control;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_ex_control \[▶ 569\]](#)).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

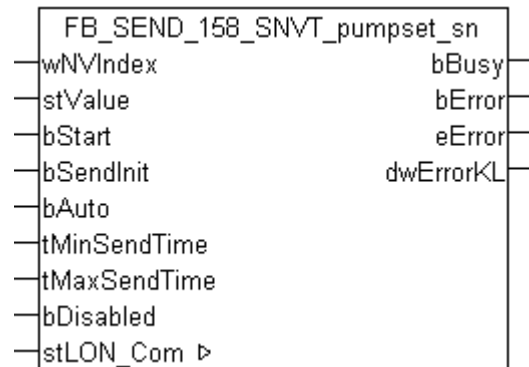
dwErrorKL: Error identifier [▶ 604] of the function block `FB_LON_KL6401()` [▶ 66]. In this case the variable `eError` is set to "eKL6401_Error". Simultaneously `bError` is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects `FB_LON_KL6401()` [▶ 66] with the send/receive function blocks (see `ST_LON_Communication` [▶ 557]).

7.2.341 FB_SEND_158_SNVT_pumpset_sn



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_pumpset_sn.

SNVT number: 158.

SNVT description: Pumpset sensor (dilution, exhaust, pressure, vacuum,...).

VAR_INPUT

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_pumpset_sn;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see `ST_LON_SNVT_pumpset_sn` [▶ 578]).

bStart: A positive edge starts sending (regardless of `bAuto`).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time `tMaxSendTime` runs out (polling).

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

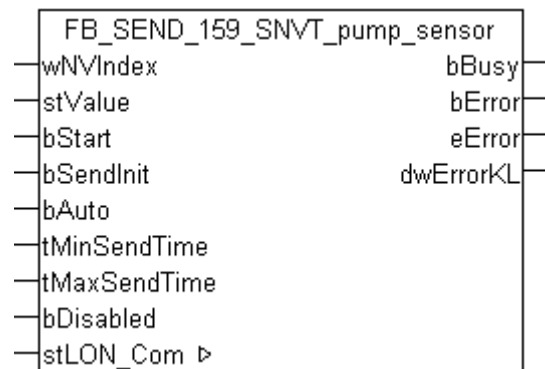
dwErrorKL: [Error identifier \[▶ 604\]](#) of the function block [FB_LON_KL6401\(\) \[▶ 66\]](#). In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.342 FB_SEND_159_SNVT_pump_sensor



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_pump_sensor.

SNVT number: 159.

SNVT description: Pump sensor (speed, temperature, status).

VAR_INPUT

```
wNVIndex    : WORD;
stValue     : ST_LON_SNVT_pump_sensor;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_pump_sensor \[▶ 577\]](#)).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

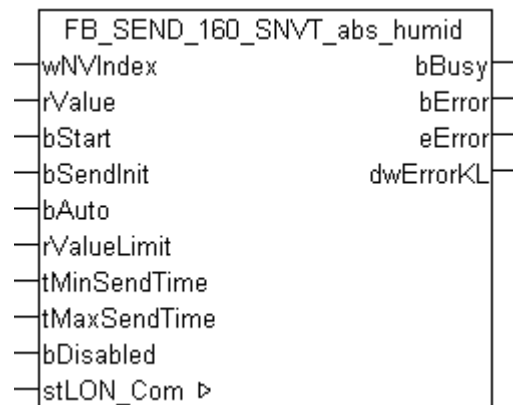
dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com  : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.343 FB_SEND_160_SNVT_abs_humid



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_abs_humid.

SNVT number: 160.

SNVT description: Absolute humidity (gram/kilogram).

VAR_INPUT

```
wNVIndex   : WORD;
rValue     : REAL;
bStart     : BOOL;
bSendInit  : BOOL := bSendInitDefault;
```



```

bAuto      : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled  : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 655.35.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

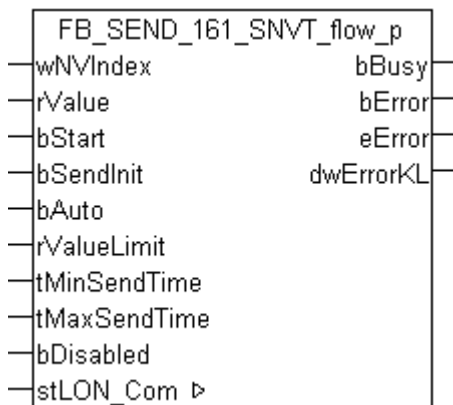
```

stLON_Com : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.344 FB_SEND_161_SNVT_flow_p



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_flow_p.

SNVT number: 161.

SNVT description: Flow volume (cubic meter/hour).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 655.35.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for [automatic sending \[▶ 604\]](#). The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

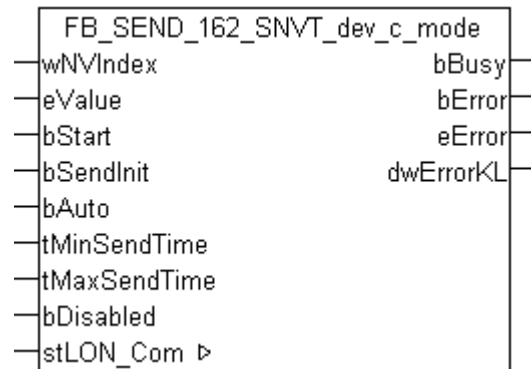
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.345 FB_SEND_162_SNVT_dev_c_mode



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_dev_c_mode.

SNVT number: 162.

SNVT description: Device control mode (device control mode names).

VAR_INPUT

```
wNVIndex : WORD;
eValue : E_LON_device_c_mode_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

eValue: Enum of the data to be sent (see [E_LON_device_c_mode_t](#) [▶ 498]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com  : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.346 FB_SEND_163_SNVT_valve_mode



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_valve_mode.

SNVT number: 163.

SNVT description: Valve mode (valve mode names).

VAR_INPUT

```
wNVIndex   : WORD;
eValue     : E_LON_valve_mode_t;
bStart     : BOOL;
bSendInit  : BOOL := bSendInitDefault;
bAuto      : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled  : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

eValue: Enum of the data to be sent (see [E_LON_valve_mode_t](#) [▶ 529]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

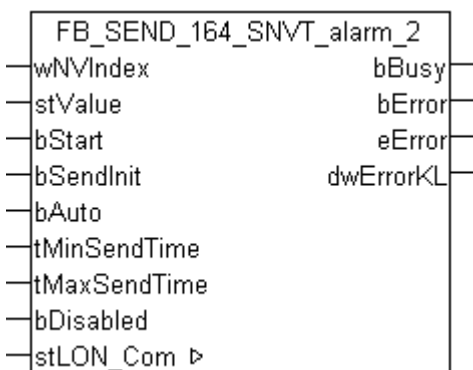
dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.347 FB_SEND_164_SNVT_alarm_2



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_alarm_2.

SNVT number: 164.

SNVT description: Alarm status 2. Used to report alarm status for a functional block or device. Replaces SNVT_alarm.

VAR_INPUT

```

wNVIndex      : WORD;
stValue       : ST_LON_SNVT_alarm_2;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_alarm_2](#) [► 560]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [► 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block [FB_LON_KL6401\(\)](#) [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

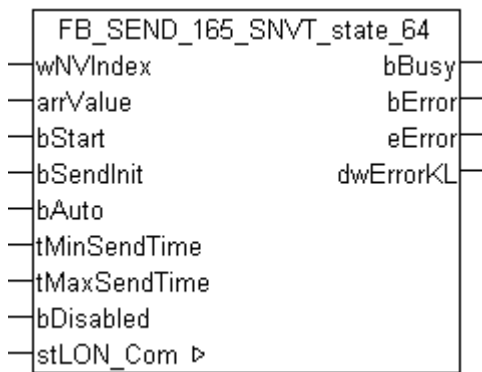
```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [► 66] with the send/receive function blocks (see [ST_LON_Communication](#) [► 557]).

7.2.348 FB_SEND_165_SNVT_state_64



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_state_64.

SNVT number: 165.

SNVT description: State vector (64 individual bit values). Each state is a boolean single-bit value.

VAR_INPUT

```
wNVIndex      : WORD;
arrValue      : ARRAY [0..63] OF BOOL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

arrValue: 0-63 Bit.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

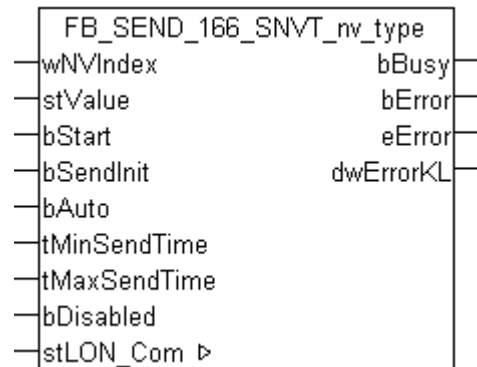
dwErrorKL: Error identifier [▶ 604] of the function block `FB_LON_KL6401()` [▶ 66]. In this case the variable `eError` is set to "eKL6401_Error". Simultaneously `bError` is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects `FB_LON_KL6401()` [▶ 66] with the send/receive function blocks (see `ST_LON_Communication` [▶ 557]).

7.2.349 FB_SEND_166_SNVT_nv_type



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_nv_type.

SNVT number: 166.

SNVT description: Network variable type. Network variable type description for network variables that support changeable types.

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_nv_type;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see `ST_LON_SNVT_nv_type` [▶ 573]).

bStart: A positive edge starts sending (regardless of `bAuto`).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time `tMaxSendTime` runs out (polling).

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

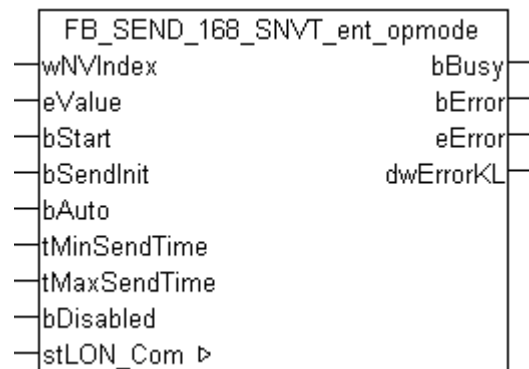
dwErrorKL: [Error identifier \[▶ 604\]](#) of the function block [FB_LON_KL6401\(\) \[▶ 66\]](#). In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.350 FB_SEND_168_SNVT_ent_opmode



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_ent_opmode.

SNVT number: 168.

SNVT description: Entry operation mode (door, lock, sluice, or something which allows/prohibits entry to an area).

VAR_INPUT

```
wNVIndex    : WORD;
eValue      : E_LON_ent_opmode_cmd_t;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

eValue: Enum of the data to be sent (see [E_LON_ent_opmode_cmd_t \[▶ 501\]](#)).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [► 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [► 463]). Simultaneously *bError* is TRUE.

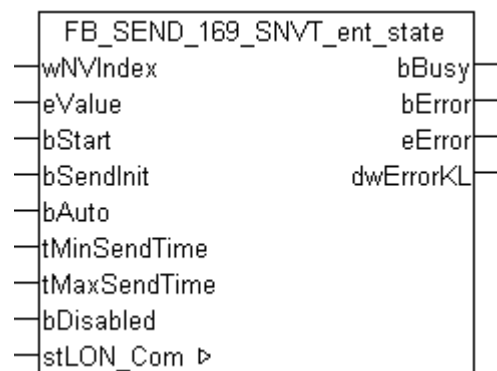
dwErrorKL: Error identifier [► 604] of the function block FB_LON_KL6401() [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com  : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [► 66] with the send/receive function blocks (see ST_LON_Communication [► 557]).

7.2.351 FB_SEND_169_SNVT_ent_state



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_ent_state.

SNVT number: 169.

SNVT description: Entry state (door, lock, sluice, or something that controls entry of an area).

VAR_INPUT

```
wNVIndex   : WORD;
eValue     : E_LON_ent_cmd_t;
bStart     : BOOL;
bSendInit  : BOOL := bSendInitDefault;
bAuto     : BOOL := bAutoDefault;
```

```
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled    : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

eValue: Enum of the data to be sent (see [E_LON_ent_cmd_t](#) [▶ 500]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

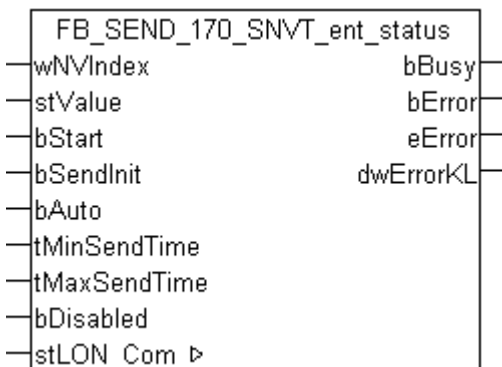
dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.352 FB_SEND_170_SNVT_ent_status



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_ent_status.

SNVT number: 170.

SNVT description: Status of entry objects (door, lock, sluice, or something that allows/prohibits entry into an area).

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_ent_status;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_ent_status \[► 567\]](#)).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[► 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending \[► 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[► 604\]](#). The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[► 463\]](#)). Simultaneously *bError* is TRUE.

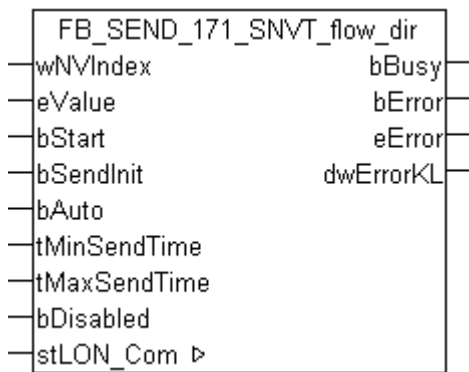
dwErrorKL: [Error identifier \[► 604\]](#) of the function block [FB_LON_KL6401\(\) \[► 66\]](#). In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[► 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[► 557\]](#)).

7.2.353 FB_SEND_171_SNVT_flow_dir



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_flow_dir.

SNVT number: 171.

SNVT description: Flow direction. Direction of allowable flow, or direction of present flow.

VAR_INPUT

```
wNVIndex      : WORD;
eValue        : E_LON_flow_direction_t;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

eValue: Enum of the data to be sent (see [E_LON_flow_direction_t](#) [▶ 506]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block `FB_LON_KL6401()` [► 66]. In this case the variable `eError` is set to "eKL6401_Error". Simultaneously `bError` is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects `FB_LON_KL6401()` [► 66] with the send/receive function blocks (see `ST_LON_Communication` [► 557]).

7.2.354 FB_SEND_172_SNVT_hvac_satsts



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_hvac_satsts.

SNVT number: 172.

SNVT description: HVAC saturation status.

A value of 0 in a field indicates that the resource associated with that field has not saturated or reached an end stop before attaining the required setpoint.

A value of 1 indicates that the resource associated with that field has saturated or reached an end stop without attaining the required setpoint.

VAR_INPUT

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_hvac_satsts;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see `ST_LON_SNVT_hvac_satsts` [► 570]).

bStart: A positive edge starts sending (regardless of `bAuto`).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [► 604] when a value changes or when the time `tMaxSendTime` runs out (polling).

tMinSendTime: Parameter for automatic sending [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.355 FB_SEND_173_SNVT_dev_status



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_dev_status.

SNVT number: 173.

SNVT description: Device status.

VAR_INPUT

```
wNVIndex    : WORD;
stValue     : ST_LON_SNVT_dev_status;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_dev_status](#) [▶ 566]).

If *stValue.eDevice_select* = eLON_DV_PUMP_CTRL (0) the structure *stValue.stDev_type.stPump_ctrl* will be sent.

If *stValue.eDevice_select* = eLON_DV_VALVE_POS (1) the structure *stValue.stDev_type.stValvePos* will be sent.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

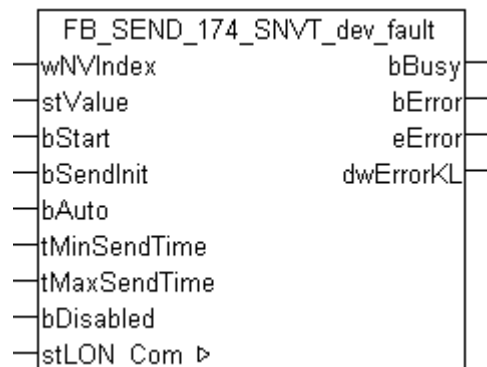
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.356 FB_SEND_174_SNVT_dev_fault



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_dev_fault.

SNVT number: 174.

SNVT description: Device fault states. Fault information for the device.

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_dev_fault;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_dev_fault](#) [▶ 565]).

If *stValue.eDevice_select* = eLON_DV_PUMP_CTRL (0) the structure *stValue.stDev_type.stPump_ctrl* will be sent.

If *stValue.eDevice_select* = eLON_DV_VALVE_POS (1) the structure *stValue.stDev_type.stValvePos* will be sent.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com    : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.357 FB_SEND_175_SNVT_dev_maint



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_dev_maint.

SNVT number: 175.

SNVT description: Device-maintenance states.

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_dev_maint;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_dev_maint \[► 566\]](#)).

Wenn *stValue.eDevice_select* = eLON_DV_PUMP_CTRL (0) wird die Struktur *stValue.stDev_type.stPump_ctrl* gesendet.

Wenn *stValue.eDevice_select* = eLON_DV_VALVE_POS (1) wird die Struktur *stValue.stDev_type.stValvePos* gesendet.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[► 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending \[► 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[► 604\]](#). The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.358 FB_SEND_176_SNVT_date_event



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_date_event.

SNVT number: 176.

SNVT description: Reports the status of a schedule.

VAR_INPUT

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_date_event;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_date_event](#) [▶ 565]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

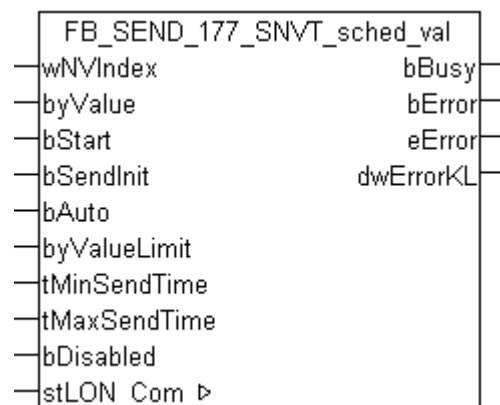
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com  : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.359 FB_SEND_177_SNVT_sched_val



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_sched_val.

SNVT number: 177.

SNVT description: Scheduler value. Index from scheduler that selects entry in SCPTvalueDefinition array, or is a direct value output.

VAR_INPUT

```
wNVIndex   : WORD;
byValue    : BYTE;
bStart     : BOOL;
bSendInit  : BOOL := bSendInitDefault;
bAuto      : BOOL := bAutoDefault;
byValueLimit : BYTE := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled  : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

byValue: Min: 0 / Max: 255.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

byValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*byValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

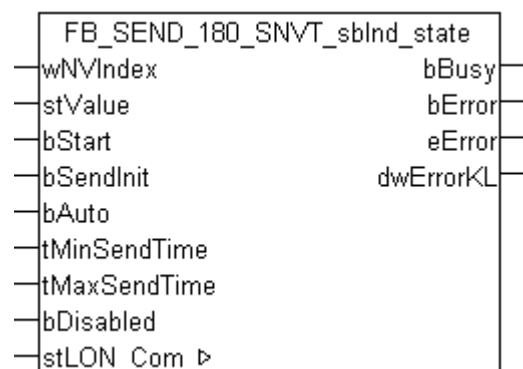
dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.360 FB_SEND_180_SNVT_sbInd_state



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_sblnd_state.

SNVT number: 180.

SNVT description: Sunblind State.

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_sblnd_state;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_sblnd_state](#) [► 580]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [► 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [► 463]). Simultaneously *bError* is TRUE.

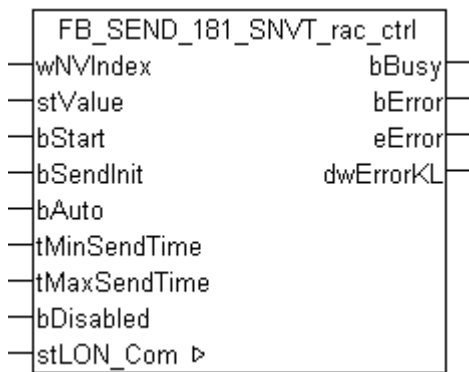
dwErrorKL: [Error identifier](#) [► 604] of the function block [FB_LON_KL6401\(\)](#) [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [► 66] with the send/receive function blocks (see [ST_LON_Communication](#) [► 557]).

7.2.361 FB_SEND_181_SNVT_rac_ctrl



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_rac_ctrl.

SNVT number: 181.

SNVT description: Rail-Audio control. Invokes audio control for a given source.

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_rac_ctrl;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_rac_ctrl](#) [► 579]).

If *stValue.bDest_p2p* = TRUE the structure *stValue.stAddr_dest.stP2p* will be sent.

If *stValue.bDest_p2p* = FALSE the structure *stValue.stAddr_dest.stP2m* will be sent.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [► 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

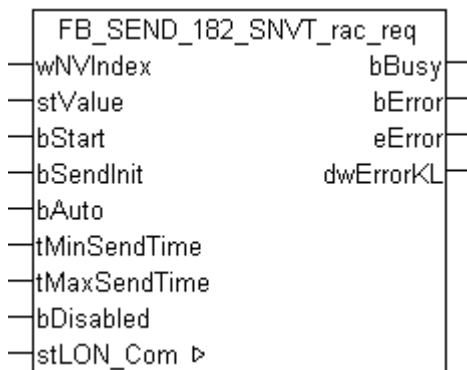
dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.362 FB_SEND_182_SNVT_rac_req



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_rac_req.

SNVT number: 182.

SNVT description: Rail-Audio controller request. Requests audio control for a given source.

VAR_INPUT

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_rac_req;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_rac_req](#) [▶ 579]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

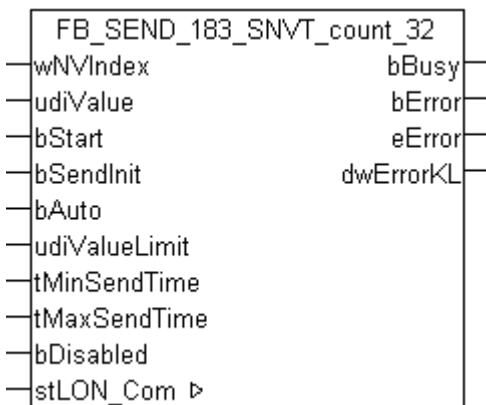
dwErrorKL: [Error identifier \[▶ 604\]](#) of the function block [FB_LON_KL6401\(\) \[▶ 66\]](#). In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.363 FB_SEND_183_SNVT_count_32



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_count_32.

SNVT number: 183.

SNVT description: Absolute count. A 32-bit counter.

VAR_INPUT

```
wNVIndex      : WORD;
udiValue      : UDINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
udiValueLimit : UDINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

udiValue: Min: 0 / Max: 4294967294.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

udiValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*udiValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

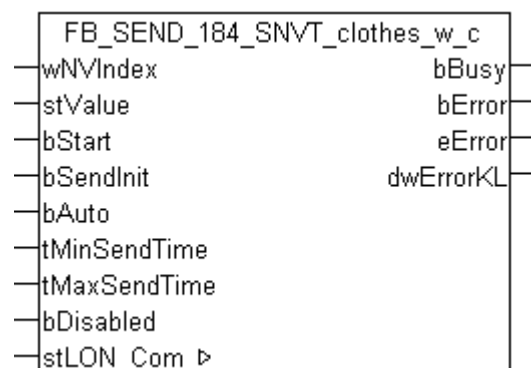
dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.364 FB_SEND_184_SNVT_clothes_w_c



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_clothes_w_c.

SNVT number: 184.

SNVT description: Clothes washer command. Used to program and start a clothes washer.

VAR_INPUT

```

wNVIndex      : WORD;
stValue       : ST_LON_SNVT_clothes_w_c;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_clothes_w_c](#) [► 563]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [► 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [► 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [► 604] of the function block [FB_LON_KL6401\(\)](#) [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

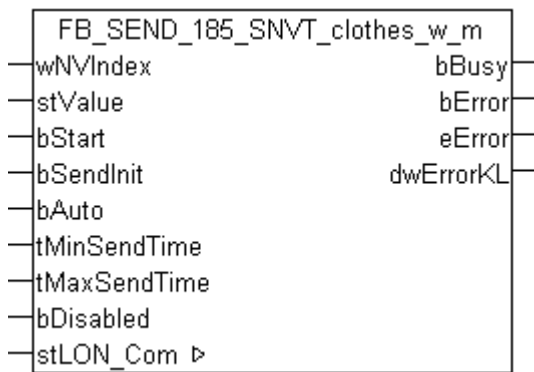
```

stLON_Com     : ST_LON_Communication;

```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [► 66] with the send/receive function blocks (see [ST_LON_Communication](#) [► 557]).

7.2.365 FB_SEND_185_SNVT_clothes_w_m



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_clothes_w_m.

SNVT number: 185.

SNVT description: Clothes Washer-Management Status. Provides status of door/lid and drain.

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_clothes_w_m;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto        : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_clothes_w_m \[▶ 563\]](#)).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

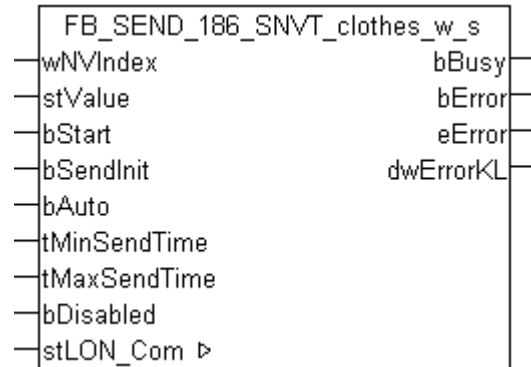
dwErrorKL: Error identifier [▶ 604] of the function block `FB_LON_KL6401()` [▶ 66]. In this case the variable `eError` is set to "eKL6401_Error". Simultaneously `bError` is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects `FB_LON_KL6401()` [▶ 66] with the send/receive function blocks (see `ST_LON_Communication` [▶ 557]).

7.2.366 FB_SEND_186_SNVT_clothes_w_s



This function-block SEND the following LON-output-variables (nvo):

SNVT name: `SNVT_clothes_w_s`.

SNVT number: 186.

SNVT description: Clothes Washer Status. Used to provide present status from a clothes washer, including command and alarm information.

VAR_INPUT

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_clothes_w_s;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see `ST_LON_SNVT_clothes_w_s` [▶ 563]).

bStart: A positive edge starts sending (regardless of `bAuto`).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time `tMaxSendTime` runs out (polling).

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

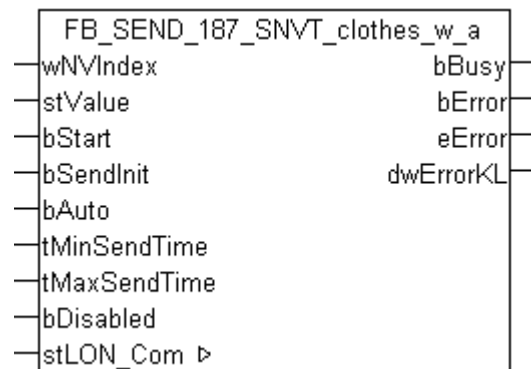
VAR_IN_OUT

```

stLON_Com  : ST_LON_Communication;

```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.367 FB_SEND_187_SNVT_clothes_w_a

This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_clothes_w_a.

SNVT number: 187.

SNVT description: Clothes washer alarm. Used to provide alarm status for a clothes washer.

VAR_INPUT

```

wNVIndex   : WORD;
stValue    : ST_LON_SNVT_clothes_w_a;
bStart     : BOOL;
bSendInit  : BOOL := bSendInitDefault;
bAuto      : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled  : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_clothes_w_a](#) [▶ 561]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

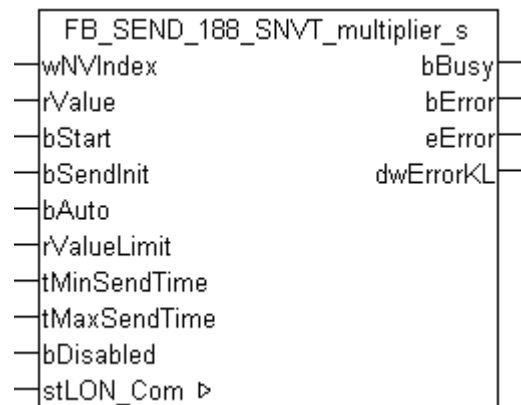
dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.368 FB_SEND_188_SNVT_multiplier_s



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_multiplier_s.

SNVT number: 188.

SNVT description: Multiplier.

VAR_INPUT

```
wNVIndex : WORD;
rValue   : REAL;
bStart   : BOOL;
bSendInit : BOOL := bSendInitDefault;
```

```

bAuto      : BOOL := bAutoDefault;
rValueLimit : REAL := 0.1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled  : BOOL := FALSE;

```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: 0 / Max: 2.54.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```

bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;

```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

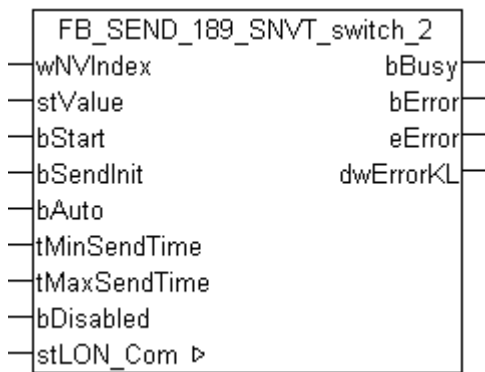
```

stLON_Com : ST_LON_Communication;

```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.369 FB_SEND_189_SNVT_switch_2



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_switch_2.

SNVT number: 189.

SNVT description: Switch with scene and setting control. An enhanced version of SNVT_switch with scene and setting controls similar to SNVT_scene and SNVT_setting.

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_switch_2;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_switch_2](#) [▶ 582]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

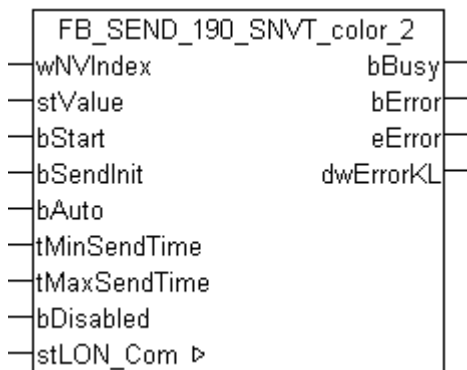
dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.370 FB_SEND_190_SNVT_color_2



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_color_2.

SNVT number: 190.

SNVT description: Color.

VAR_INPUT

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_color_2;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_color_2](#) [▶ 564]).

If *stValue.eEncoding* = eLON_COLOR_CIE31_LUMEN (0) the structure

stValue.stColor_value.stCIE1931_lumen will be sent.

If *stValue.eEncoding* = eLON_COLOR_CIE31_PERCENT (1) the structure

stValue.stColor_value.stCIE1931_percent will be sent.

If *stValue.eEncoding* = eLON_COLOR_RGB (2) the structure *stValue.stColor_value.stRGB* will be sent.

If *stValue.eEncoding* = eLON_COLOR_TEMPERATURE (3) the structure

stValue.stColor_value.uiColor_temperature will be sent.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

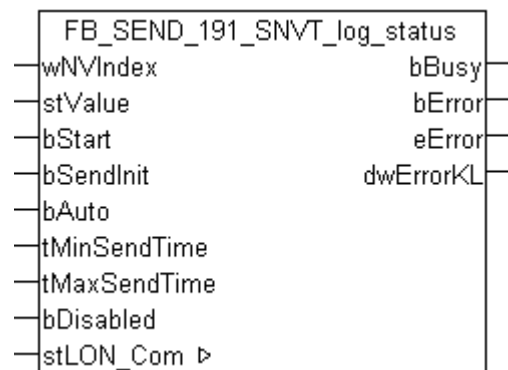
dwErrorKL: [Error identifier \[▶ 604\]](#) of the function block [FB_LON_KL6401\(\) \[▶ 66\]](#). In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\) \[▶ 66\]](#) with the send/receive function blocks (see [ST_LON_Communication \[▶ 557\]](#)).

7.2.371 FB_SEND_191_SNVT_log_status



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_log_status.

SNVT number: 191.

SNVT description: Log status (hundredths of second). Reports the current status of a data log. Updated based on the cpLogNotificationThreshold value. Reports status only. Alarms reported via Node Object nvoAlarm2 output. Required if the Node Object does not include an nvoLogStat output.

VAR_INPUT

```
wNVIndex    : WORD;
stValue     : ST_LON_SNVT_log_status;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
```

```
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled    : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_log_status](#) [▶ 572]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

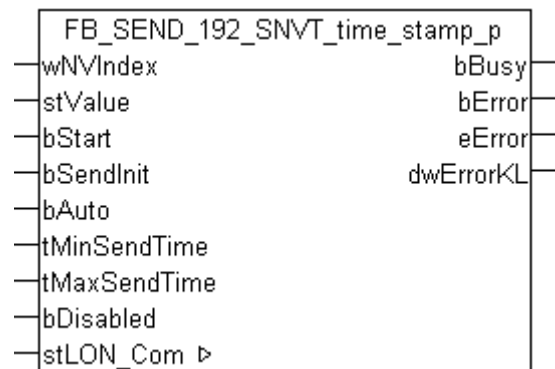
dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.372 FB_SEND_192_SNVT_time_stamp_p



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_time_stamp_p.

SNVT number: 192.

SNVT description: Precision timestamp (seconds). Timestamp with hundredths of a second resolution.

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : Timestruct;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [Timestruct](#)). The structure variable *wDayOfWeek* is not valid here and will not be transferred.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [[▶ 604](#)] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for automatic sending. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [[▶ 463](#)]). Simultaneously *bError* is TRUE.

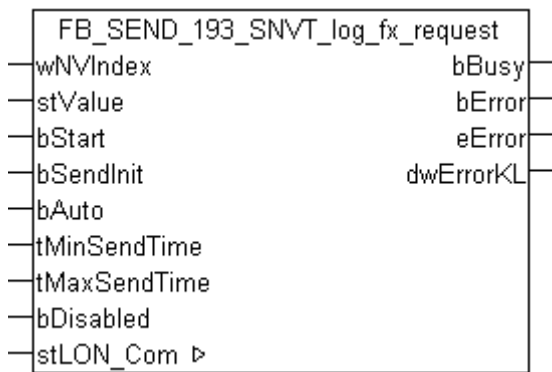
dwErrorKL: Error identifier [[▶ 604](#)] of the function block [FB_LON_KL6401\(\)](#) [[▶ 66](#)]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [[▶ 66](#)] with the send/receive function blocks (see [ST_LON_Communication](#) [[▶ 557](#)]).

7.2.373 FB_SEND_193_SNVT_log_fx_request



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_log_fx_request.

SNVT number: 193.

SNVT description: Log file transfer request. Requests a data log to be transferred via FTP. Must be followed by a standard FTP request to get the data log file. Required on devices implementing the Data Logger functional profile that support data log transfer via FTP.

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_log_fx_request;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_log_fx_request](#) [► 572]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [► 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

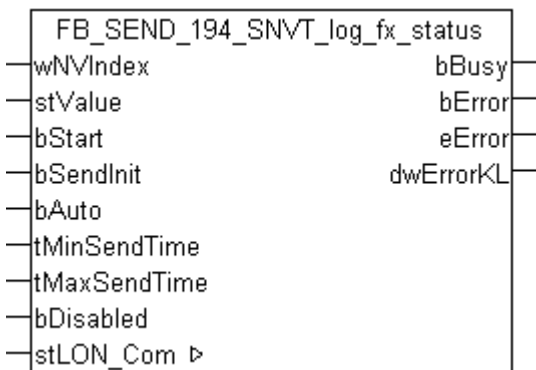
dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.374 FB_SEND_194_SNVT_log_fx_status



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_log_fx_status.

SNVT number: 194.

SNVT description: Log file transfer status. Reports the status of a data log file transfer using FTP. Required on devices implementing the Data Logger functional profile that support data log transfer via FTP.

VAR_INPUT

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_log_fx_status;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_log_fx_status](#) [▶ 572]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

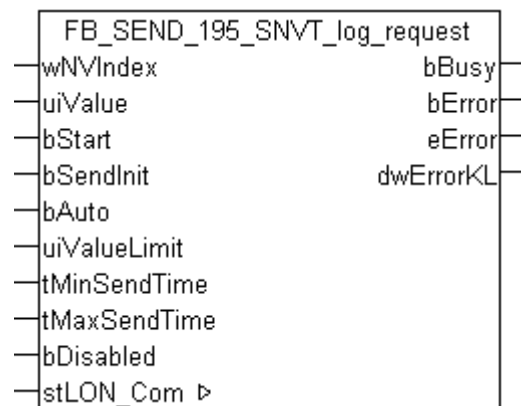
dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.375 FB_SEND_195_SNVT_log_request



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_log_request.

SNVT number: 195.

SNVT description: Log status request. Requests the current status of a data log. Status is reported by a SNVT_log_status output.

VAR_INPUT

```
wNVIndex    : WORD;
uiValue     : UINT;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
uiValueLimit : UINT := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

uiValue: Min: 1 / Max: 65535.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

uiValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*uiValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

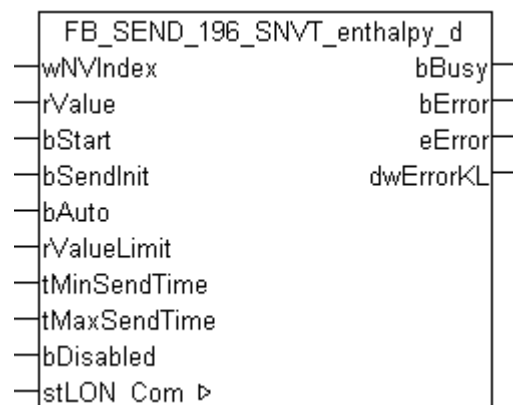
dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.376 FB_SEND_196_SNVT_enthalpy_d



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_enthalpy_d.

SNVT number: 196.

SNVT description: Enthalpy difference (kJ/kg).

VAR_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

rValue: Min: -327.68 / Max: 327.67.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

rValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*rValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see E_LON_ERROR [▶ 463]). Simultaneously *bError* is TRUE.

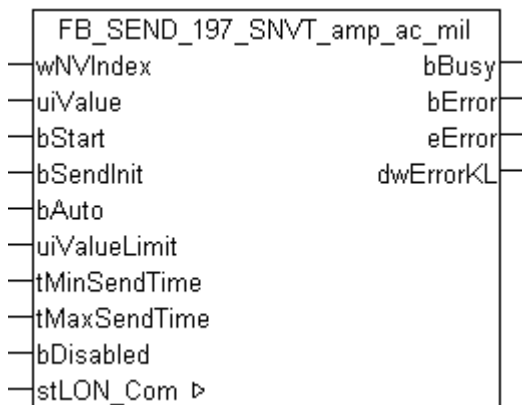
dwErrorKL: Error identifier [▶ 604] of the function block FB_LON_KL6401() [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects FB_LON_KL6401() [▶ 66] with the send/receive function blocks (see ST_LON_Communication [▶ 557]).

7.2.377 FB_SEND_197_SNVT_amp_ac_mil



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_amp_ac_mil.

SNVT number: 197.

SNVT description: Electrical current (milliampere).

VAR_INPUT

```
wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

uiValue: Min: 0 / Max: 65534.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the automatic sending [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

uiValueLimit: Parameter for automatic sending [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for automatic sending [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for automatic sending [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*uiValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

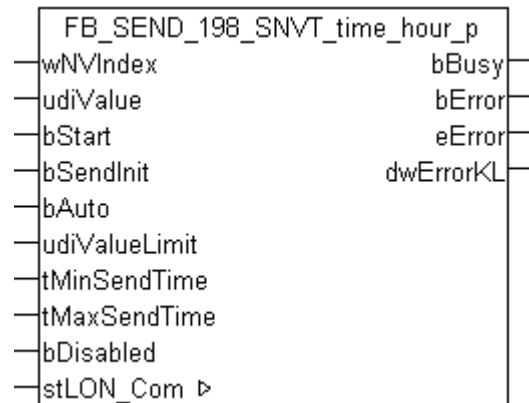
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.378 FB_SEND_198_SNVT_time_hour_p



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_time_hour_p.

SNVT number: 198.

SNVT description: Time hour (hour).

VAR_INPUT

```
wNVIndex      : WORD;
udiValue      : UDINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
udiValueLimit : UDINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

udiValue: Min: 0 / Max: 4294967294.

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

udiValueLimit: Parameter for [automatic sending](#) [▶ 604]. The value is sent only if the change is greater than this parameter since the last transmission. If this value is 0, then is sent after each change in value. This value does not exist in enums and structures. There is sent after each change in value.

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time, even if the minimum value change (*udiValueLimit*) was not reached (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

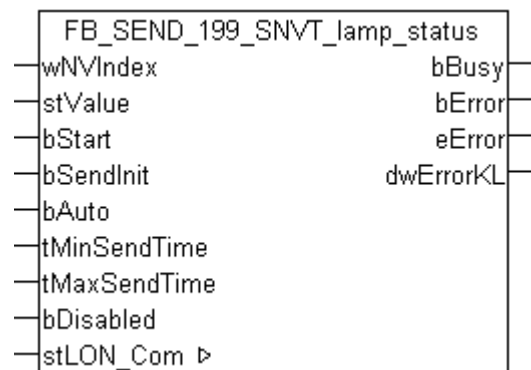
dwErrorKL: Error identifier [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.379 FB_SEND_199_SNVT_lamp_status



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_lamp_status.

SNVT number: 199.

SNVT description: Lamp status.

VAR_INPUT

```
wNVIndex       : WORD;
stValue        : ST_LON_SNVT_lamp_status;
bStart         : BOOL;
```

```
bSendInit      : BOOL := bSendInitDefault;
bAuto          : BOOL := bAutoDefault;
tMinSendTime   : TIME := tMinSendTimeDefault;
tMaxSendTime   : TIME := tMaxSendTimeDefault;
bDisabled      : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_lamp_status](#) [▶ 571]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [▶ 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [▶ 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [▶ 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [▶ 463]). Simultaneously *bError* is TRUE.

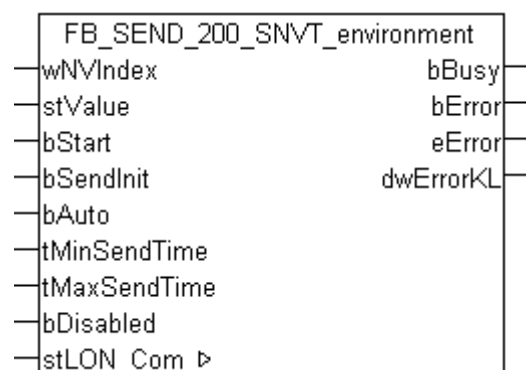
dwErrorKL: [Error identifier](#) [▶ 604] of the function block [FB_LON_KL6401\(\)](#) [▶ 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com      : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [▶ 66] with the send/receive function blocks (see [ST_LON_Communication](#) [▶ 557]).

7.2.380 FB_SEND_200_SNVT_environment



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_environment.

SNVT number: 200.

SNVT description: Environment.

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_environment;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_environment](#) [► 568]).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending](#) [► 604] when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending](#) [► 604]. A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending](#) [► 604]. The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR](#) [► 463]). Simultaneously *bError* is TRUE.

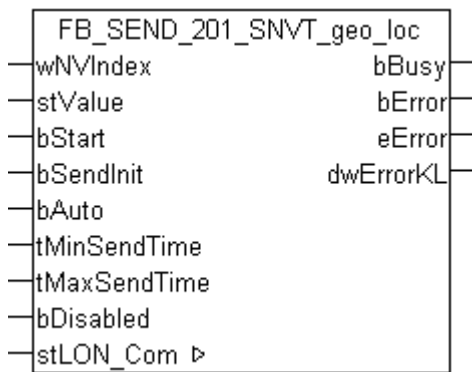
dwErrorKL: [Error identifier](#) [► 604] of the function block [FB_LON_KL6401\(\)](#) [► 66]. In this case the variable *eError* is set to "eKL6401_Error". Simultaneously *bError* is TRUE.

VAR_IN_OUT

```
stLON_Com     : ST_LON_Communication;
```

stLON_Com: This structure connects [FB_LON_KL6401\(\)](#) [► 66] with the send/receive function blocks (see [ST_LON_Communication](#) [► 557]).

7.2.381 FB_SEND_201_SNVT_geo_loc



This function-block SEND the following LON-output-variables (nvo):

SNVT name: SNVT_geo_loc.

SNVT number: 201.

SNVT description: Geographic location.

VAR_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_geo_loc;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

wNVIndex: Unique Index. This is required for the binding of the LON nodes. Per LON terminal a maximum of 62 SNVTs are allowed. Values from 0 to 61 are permissible.

stValue: Structure of the data to be sent (see [ST_LON_SNVT_geo_loc \[▶ 570\]](#)).

bStart: A positive edge starts sending (regardless of *bAuto*).

bSendInit: After restarting the PLC, the values are sent once.

bAuto: Selecting the [automatic sending \[▶ 604\]](#) when a value changes or when the time *tMaxSendTime* runs out (polling).

tMinSendTime: Parameter for [automatic sending \[▶ 604\]](#). A new value is sent no sooner than this time. Thus, a continuous transmission is prevented.

tMaxSendTime: Parameter for [automatic sending \[▶ 604\]](#). The value is sent no later than the end of that time (polling). A value of 0 disables this feature.

bDisabled: TRUE = disables the function block.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

bBusy: If the sending of data is in process, this output will be TRUE.

bError: The output becomes TRUE as soon as an error occurs. The error is described via the variable *eError*.

eError: The output issues an error code when an error occurs (see [E_LON_ERROR \[▶ 463\]](#)). Simultaneously *bError* is TRUE.

dwErrorKL: Error identifier [▶ 604] of the function block `FB_LON_KL6401()` [▶ 66]. In this case the variable `eError` is set to "eKL6401_Error". Simultaneously `bError` is TRUE.

VAR_IN_OUT

```
stLON_Com : ST_LON_Communication;
```

stLON_Com: This structure connects `FB_LON_KL6401()` [▶ 66] with the send/receive function blocks (see `ST_LON_Communication` [▶ 557]).

7.3 Data types

Data types/Enums

Data types	Description
<code>E_LON_alarm_type_t</code> [▶ 488]	Used by: <code>SNVT_alarm</code> / <code>SNVT_alarm_2</code>
<code>E_LON_appl_cwc_t</code> [▶ 490]	Used by: <code>SNVT_clothes_w_c</code> / <code>SNVT_clothes_w_s</code>
<code>E_LON_appl_cwp_t</code> [▶ 490]	Used by: <code>SNVT_clothes_w_c</code>
<code>E_LON_appl_cws_t</code> [▶ 491]	Used by: <code>SNVT_clothes_w_c</code> / <code>SNVT_clothes_w_s</code>
<code>E_LON_appl_rin_t</code> [▶ 491]	Used by: <code>SNVT_clothes_w_c</code>
<code>E_LON_boolean_t</code> [▶ 492]	Used by: <code>SCPTautoAnswer</code> / <code>SCPTcoolingResetEnable</code> / <code>SCPTdefrostHold</code> / <code>SCPTdefrostInternalSchedule</code> / <code>SCPTheatingResetEnable</code> / <code>SCPThighLimit1Enable</code> / <code>SCPThighLimit2Enable</code> / <code>SCPTlowLimit1Enable</code> / <code>SCPTlowLimit2Enable</code> / <code>SCPTscheduleInternal</code> / <code>SNVT_clothes_w_c</code> / <code>SNVT_pump_sensor</code> / <code>SNVT_pumpset_mn</code> / <code>SNVT_pumpset_sn</code>
<code>E_LON_calendar_type_t</code> [▶ 492]	Used by: <code>SNVT_time_zone</code>
<code>E_LON_cam_act_t</code> [▶ 492]	Used by: <code>SNVT_pos_ctrl</code>
<code>E_LON_cam_func_t</code> [▶ 493]	Used by: <code>SNVT_pos_ctrl</code>
<code>E_LON_chiller_t</code> [▶ 493]	Used by: <code>SNVT_chlr_status</code>
<code>E_LON_color_encoding_t</code> [▶ 493]	Used by: <code>SNVT_color_2</code>
<code>E_LON_config_source_t</code> [▶ 494]	Used by: <code>SNVT_config_src</code>
<code>E_LON_control_resp_t</code> [▶ 494]	Used by: <code>SNVT_ctrl_resp</code>
<code>E_LON_currency_t</code> [▶ 494]	Used by: <code>SNVT_currency</code>
<code>E_LON_days_of_week_t</code> [▶ 497]	Used by: <code>SCPTtimePeriod</code> / <code>SNVT_date_day</code> / <code>SNVT_time_zone</code>
<code>E_LON_defrost_mode_t</code> [▶ 497]	Used by: <code>SNVT_defr_mode</code>
<code>E_LON_defrost_state_t</code> [▶ 498]	Used by: <code>SNVT_defr_state</code>
<code>E_LON_defrost_term_t</code> [▶ 498]	Used by: <code>SNVT_defr_term</code>
<code>E_LON_device_c_mode_t</code> [▶ 498]	Used by: <code>SNVT_dev_c_mode</code>
<code>E_LON_device_select_t</code> [▶ 499]	Used by: <code>SNVT_dev_fault</code> / <code>SNVT_dev_maint</code> / <code>SNVT_dev_status</code>
<code>E_LON_discrete_levels_t</code> [▶ 500]	Used by: <code>SNVT_clothes_w_c</code> / <code>SNVT_lev_disc</code>
<code>E_LON_emerg_t</code> [▶ 500]	Used by: <code>SNVT_hvac_emerg</code>
<code>E_LON_ent_cmd_t</code> [▶ 500]	Used by: <code>SNVT_ent_state</code>
<code>E_LON_ent_opmode_cmd_t</code> [▶ 501]	Used by: <code>SNVT_ent_opmode</code> / <code>SNVT_ent_status</code>
<code>E_LON_evap_t</code> [▶ 502]	Used by: <code>SNVT_evap_state</code>
<code>E_LON_ex_control_t</code> [▶ 502]	Used by: <code>SNVT_ex_control</code>

Data types	Description
E_LON_file_request_t [▶ 503]	Used by: SNVT_file_req
E_LON_file_status_t [▶ 503]	Used by: SNVT_file_status
E_LON_fire_indicator_t [▶ 504]	Used by: SNVT_fire_indcte
E_LON_fire_initiator_t [▶ 504]	Used by: SNVT_fire_init
E_LON_fire_test_t [▶ 505]	Used by: SNVT_fire_test
E_LON_flow_direction_t [▶ 506]	Used by: SNVT_flow_dir
E_LON_gfci_status_t [▶ 506]	Used by: SNVT_gfci_status
E_LON_hvac_hvt_t [▶ 506]	Used by: SNVT_hvac_type
E_LON_hvac_overid_t [▶ 507]	Used by: SNVT_hvac_overid
E_LON_hvac_t [▶ 509]	Used by: SNVT_chlr_status / SNVT_hvac_mode / SNVT_hvac_status
E_LON_learn_mode_t [▶ 510]	Used by: SNVT_preset
E_LON_log_status_t [▶ 510]	Used by: SCPTlogRecord / SNVT_log_status
E_LON_motor_state_t [▶ 511]	Used by: SNVT_motor_state / SNVT_pumpset_mn
E_LON_nv_type_category_t [▶ 511]	Used by: SNVT_nv_type
E_LON_object_request_t [▶ 512]	Used by: SNVT_obj_request
E_LON_occup_t [▶ 513]	Used by: SNVT_occupancy / SNVT_tod_event
E_LON_override_t [▶ 513]	Used by: SNVT_override
E_LON_pan_dir_t [▶ 514]	Used by: SNVT_ptz
E_LON_priority_level_t [▶ 514]	Used by: SNVT_alarm / SNVT_alarm_2 / SNVT_pumpset_mn
E_LON_privacyzone_t [▶ 515]	Used by: SNVT_privacyzone
E_LON_rail_audio_sensor_type_t [▶ 515]	Used by: SNVT_rac_ctrl / SNVT_rac_req
E_LON_rail_audio_type_t [▶ 516]	Used by: SNVT_rac_ctrl / SNVT_rac_req
E_LON_reg_val_unit_t [▶ 517]	Used by: SNVT_reg_val / SNVT_reg_val_ts
E_LON_sblnd_cmd_source_t [▶ 519]	Used by: SNVT_sblnd_state
E_LON_sblnd_error_t [▶ 520]	Used by: SNVT_sblnd_state
E_LON_scene_config_t [▶ 521]	Used by: SNVT_scene_cfg
E_LON_scene_t [▶ 522]	Used by: SNVT_scene
E_LON_sec_state_t [▶ 523]	Used by: SNVT_sec_state
E_LON_sec_status_t [▶ 524]	Used by: SNVT_sec_status
E_LON_setting_t [▶ 525]	Used by: SNVT_setting
E_LON_switch_state_t [▶ 525]	Used by: SNVT_switch_2
E_LON_telcom_states_t [▶ 527]	Used by: SNVT_telcom
E_LON_therm_mode_t [▶ 528]	Used by: SNVT_therm_mode
E_LON_tilt_dir_t [▶ 529]	Used by: SNVT_ptz
E_LON_unit_temp_t [▶ 529]	Used by: SNVT_pump_sensor
E_LON_valve_mode_t [▶ 529]	Used by: SNVT_valve_mode
E_LON_zoom_t [▶ 530]	Used by: SNVT_ptz

Data types/Hardware Types

Data types	Description
ST_LON_Parameter_IN_36B [▶ 555]	Process Image of the inputs
ST_LON_Parameter_OUT_36B [▶ 555]	Process Image of the outputs

Data types/LON_TYPES

Data types	Description
E_LON_Parameter_Datatypes [▶ 480]	Enums SNVT types

Data types/Structure/AuxiliaryStructure/SNVT_chlr_status

Data types	Description
ST_LON_chlr_state [▶ 533]	Used by: SNVT_chlr_status

Data types/Structure/AuxiliaryStructure/SNVT_clothes_w_c

Data types	Description
ST_LON_action [▶ 533]	Used by: SNVT_clothes_w_c
ST_LON_dry [▶ 533]	Used by: SNVT_clothes_w_c
ST_LON_duration [▶ 534]	Used by: SNVT_clothes_w_c
ST_LON_function [▶ 534]	Used by: SNVT_clothes_w_c
ST_LON_rinse [▶ 534]	Used by: SNVT_clothes_w_c
ST_LON_spin [▶ 534]	Used by: SNVT_clothes_w_c
ST_LON_wash [▶ 535]	Used by: SNVT_clothes_w_c

Data types/Structure/AuxiliaryStructure/SNVT_clothes_w_s

Data types	Description
ST_LON_alarm [▶ 535]	Used by: SNVT_clothes_w_s

Data types/Structure/AuxiliaryStructure/SNVT_color_2

Data types	Description
ST_LON_CIE1931_lumen [▶ 537]	Used by: SNVT_color_2
ST_LON_CIE1931_percent [▶ 537]	Used by: SNVT_color_2
ST_LON_color_value [▶ 537]	Used by: SNVT_color_2
ST_LON_RGB [▶ 538]	Used by: SNVT_color_2

Data types/Structure/AuxiliaryStructure/SNVT_ctrl_resp

Data types	Description
ST_LON_range [▶ 538]	Used by: SNVT_ctrl_resp
ST_LON_sender [▶ 538]	Used by: SNVT_ctrl_resp

Data types/Structure/AuxiliaryStructure/SNVT_dev_fault

Data types	Description
ST_LON_Dev_type1 [▶ 539]	Used by: SNVT_dev_fault
ST_LON_pump_ctrl1 [▶ 539]	Used by: SNVT_dev_fault
ST_LON_valve_pos1 [▶ 540]	Used by: SNVT_dev_fault

Data types/Structure/AuxiliaryStructure/SNVT_dev_maint

Data types	Description
ST_LON_Dev_type2 [▶ 541]	Used by: SNVT_dev_maint
ST_LON_pump_ctrl2 [▶ 541]	Used by: SNVT_dev_maint

Data types	Description
ST_LON_valve_pos2 [▶ 541]	Used by: SNVT_dev_maint

Data types/Structure/AuxiliaryStructure/SNVT_dev_status

Data types	Description
ST_LON_Dev_type3 [▶ 542]	Used by: SNVT_dev_status
ST_LON_pump_ctrl3 [▶ 542]	Used by: SNVT_dev_status
ST_LON_valve_pos3 [▶ 543]	Used by: SNVT_dev_status

Data types/Structure/AuxiliaryStructure/SNVT_ex_control

Data types	Description
ST_LON_Control_device_addr [▶ 544]	Used by: SNVT_ex_control

Data types/Structure/AuxiliaryStructure/SNVT_file_req

Data types	Description
ST_LON_addrt [▶ 545]	Used by: SNVT_file_req
ST_LON_dest_address [▶ 545]	Used by: SNVT_file_req
ST_LON_gp [▶ 545]	Used by: SNVT_file_req
ST_LON_sn [▶ 545]	Used by: SNVT_file_req

Data types/Structure/AuxiliaryStructure/SNVT_file_status

Data types	Description
ST_LON_address [▶ 546]	Used by: FB_Write_Address_Table / FB_Read_Address_Table
ST_LON_adr [▶ 546]	Used by: SNVT_file_status
ST_LON_descriptor [▶ 547]	Used by: SNVT_file_status

Data types/Structure/AuxiliaryStructure/SNVT_lamp_status

Data types	Description
ST_LON_Alarm_actual [▶ 547]	Used by: SNVT_lamp_status
ST_LON_alarm_previous [▶ 549]	Used by: SNVT_lamp_status

Data types/Structure/AuxiliaryStructure/SNVT_pos_ctrl

Data types	Description
ST_LON_abspos [▶ 550]	Used by: SNVT_pos_ctrl
ST_LON_Value [▶ 551]	Used by: SNVT_pos_ctrl

Data types/Structure/AuxiliaryStructure/SNVT_rac_ctrl

Data types	Description
ST_LON_addr_dest [▶ 551]	Used by: SNVT_rac_req / SNVT_rac_ctrl
ST_LON_addr_init [▶ 551]	Used by: SNVT_rac_req / SNVT_rac_ctrl
ST_LON_addr_talk [▶ 552]	Used by: SNVT_rac_ctrl
ST_LON_p2m [▶ 552]	Used by: SNVT_rac_req / SNVT_rac_ctrl
ST_LON_p2p [▶ 552]	Used by: SNVT_rac_req / SNVT_rac_ctrl

Data types/Structure/AuxiliaryStructure/SNVT_rac_req

Data types	Description
ST_LON_rac_req_addr_dest [▶ 553]	
ST_LON_rac_req_addr_init [▶ 553]	

Data types/Structure/AuxiliaryStructure/SNVT_switch_2

Data types	Description
ST_LON_setting [▶ 553]	Used by: SNVT_switch_2

Data types/Structure/AuxiliaryStructure/SNVT_time_zone

Data types	Description
ST_LON_end_DST [▶ 554]	Used by: SNVT_time_zone
ST_LON_M_end_DST [▶ 554]	Used by: SNVT_time_zone
ST_LON_M_start_DST [▶ 555]	Used by: SNVT_time_zone
ST_LON_start_DST [▶ 555]	Used by: SNVT_time_zone

Data types/Structure

Data types	Description
ST_KL6401 [▶ 558]	Structure for configuration
ST_LON_AddressTable [▶ 558]	Used by: FB_Write_Address_Table / FB_Read_Address_Table
ST_LON_ConfigTable [▶ 559]	Used by: FB_Write_Config_Table / FB_Read_Config_Table
ST_LON_DomainTable [▶ 559]	Used by: FB_Write_Domain_Table / FB_Read_Domain_Table
ST_LON_SNVT_alarm [▶ 560]	Used by: SNVT_alarm
ST_LON_SNVT_alarm_2 [▶ 560]	Used by: SNVT_alarm_2
ST_LON_SNVT_chlr_status [▶ 561]	Used by: SNVT_chlr_status
ST_LON_SNVT_clothes_w_a [▶ 561]	Used by: SNVT_clothes_w_a
ST_LON_SNVT_clothes_w_c [▶ 563]	Used by: SNVT_clothes_w_c
ST_LON_SNVT_clothes_w_m [▶ 563]	Used by: SNVT_clothes_w_m
ST_LON_SNVT_clothes_w_s [▶ 563]	Used by: SNVT_clothes_w_s
ST_LON_SNVT_color [▶ 564]	Used by: SNVT_color
ST_LON_SNVT_color_2 [▶ 564]	Used by: SNVT_color_2
ST_LON_SNVT_ctrl_req [▶ 564]	Used by: SNVT_ctrl_req
ST_LON_SNVT_ctrl_resp [▶ 564]	Used by: SNVT_ctrl_resp
ST_LON_SNVT_currency [▶ 565]	Used by: SNVT_currency
ST_LON_SNVT_date_event [▶ 565]	Used by: SNVT_date_event
ST_LON_SNVT_dev_fault [▶ 565]	Used by: SNVT_dev_fault
ST_LON_SNVT_dev_maint [▶ 566]	Used by: SNVT_dev_maint
ST_LON_SNVT_dev_status [▶ 566]	Used by: SNVT_dev_status
ST_LON_SNVT_earth_pos [▶ 566]	Used by: SNVT_earth_pos
ST_LON_SNVT_elapsed_tm [▶ 566]	Used by: SNVT_elapsed_tm
ST_LON_SNVT_ent_status [▶ 567]	Used by: SNVT_ent_status
ST_LON_SNVT_environment [▶ 568]	Used by: SNVT_environment

Data types	Description
ST_LON_SNVT_ex_control [► 569]	Used by: SNVT_ex_control
ST_LON_SNVT_file_pos [► 569]	Used by: SNVT_file_pos
ST_LON_SNVT_file_req [► 569]	Used by: SNVT_file_req
ST_LON_SNVT_file_status [► 570]	Used by: SNVT_file_status
ST_LON_SNVT_geo_loc [► 570]	Used by: SNVT_geo_loc
ST_LON_SNVT_hvac_overid [► 570]	Used by: SNVT_hvac_overid
ST_LON_SNVT_hvac_satsts [► 570]	Used by: SNVT_hvac_satsts
ST_LON_SNVT_hvac_status [► 571]	Used by: SNVT_hvac_status
ST_LON_SNVT_lamp_status [► 571]	Used by: SNVT_lamp_status
ST_LON_SNVT_log_fx_request [► 572]	Used by: SNVT_log_fx_request
ST_LON_SNVT_log_fx_status [► 572]	Used by: SNVT_log_fx_status
ST_LON_SNVT_log_status [► 572]	Used by: SNVT_log_status
ST_LON_SNVT_muldiv [► 573]	Used by: SNVT_muldiv
ST_LON_SNVT_nv_type [► 573]	Used by: SNVT_nv_type
ST_LON_SNVT_obj_request [► 574]	Used by: SNVT_obj_request
ST_LON_SNVT_obj_status [► 574]	Used by: SNVT_obj_status
ST_LON_SNVT_pos_ctrl [► 575]	Used by: SNVT_pos_ctrl
ST_LON_SNVT_preset [► 576]	Used by: SNVT_preset
ST_LON_SNVT_privacyzone [► 576]	Used by: SNVT_privacyzone
ST_LON_SNVT_ptz [► 576]	Used by: SNVT_ptz
ST_LON_SNVT_pump_sensor [► 577]	Used by: SNVT_pump_sensor
ST_LON_SNVT_pumpset_mn [► 577]	Used by: SNVT_pumpset_mn
ST_LON_SNVT_pumpset_sn [► 578]	Used by: SNVT_pumpset_sn
ST_LON_SNVT_rac_ctrl [► 579]	Used by: SNVT_rac_ctrl
ST_LON_SNVT_rac_req [► 579]	Used by: SNVT_rac_req
ST_LON_SNVT_reg_val [► 579]	Used by: SNVT_rac_val
ST_LON_SNVT_reg_val_ts [► 580]	Used by: SNVT_rac_val_ts
ST_LON_SNVT_sbldnd_state [► 580]	Used by: SNVT_sbldnd_state
ST_LON_SNVT_scene [► 580]	Used by: SNVT_scene
ST_LON_SNVT_scene_cfg [► 581]	Used by: SNVT_scene_cfg
ST_LON_SNVT_setting [► 581]	Used by: SNVT_setting
ST_LON_SNVT_str_int [► 581]	Used by: SNVT_str_int
ST_LON_SNVT_switch [► 582]	Used by: SNVT_switch
ST_LON_SNVT_switch_2 [► 582]	Used by: SNVT_switch_2
ST_LON_SNVT_temp_setpt [► 582]	Used by: SNVT_temp_setpt
ST_LON_SNVT_time_zone [► 583]	Used by: SNVT_time_zone
ST_LON_SNVT_tod_event [► 583]	Used by: SNVT_tod_event
ST_LON_SNVT_trans_table [► 583]	Used by: SNVT_trans_table
ST_LON_SNVT_zerospan [► 584]	Used by: SNVT_zerospan
str_AddressTable [► 584]	adress table
Data types	Description
E_LON_ERROR [► 463]	Error messages
ST_ExplicitMessage [► 556]	Explicit message

Data types	Description
ST_LON_Communication [▶ 557]	Connection between "FB_LON_KL6401" and the read / send function blocks.
ST_LON_ParameterInterface [▶ 557]	LON parameter interface
ST_LON_WriteData [▶ 557]	Structure of transmit buffer
ST_Prm [▶ 558]	Structure for configuration

7.3.1 E_LON_ERROR

Library error messages



The NV index in the PLC is not compared with the NV index (column Id) in the KS2000 during sending. Wrong/invalid values can be sent if the indices do not match.

Sending without binding does no result in an error message.

```

TYPE E_LON_ERROR :
(
  eLON_no_Error                := 0,
  eLON_Value_out_of_range     := 1,
  eLON_Terminal_not_ready     := 2,
  eLON_Wrong_SNV_Typ          := 3,
  eLON_Wrong_wNVIndex         := 4,
  eKL6401_Wrong_Terminal      := 5,
  eKL6401_Error                := 6,
  eKL6401_Terminal_is_not_initialized := 7,

  eLON_L_star_Out_of_range    := 50,
  eLON_A_star_Out_of_range    := 51,
  eLON_B_star_Out_of_range    := 52,

  eLON_eRequest_Out_of_range  := 55,

  eLON_wYear_Out_of_range     := 66,
  eLON_wMonth_Out_of_range    := 67,
  eLON_wDay_Out_of_range      := 68,
  eLON_wHour_Out_of_range     := 69,
  eLON_wMinute_Out_of_range   := 70,
  eLON_wSecond_Out_of_range   := 71,
  eLON_wMillisecond_Out_of_range := 72,

  eLON_rZero_Out_of_range     := 80,
  eLON_rSpan_Out_of_range     := 81,

  eLON_arrValue01_Out_of_range := 85,
  eLON_arrValue02_Out_of_range := 86,
  eLON_arrValue03_Out_of_range := 87,
  eLON_arrValue04_Out_of_range := 88,
  eLON_arrValue05_Out_of_range := 89,
  eLON_arrValue06_Out_of_range := 90,
  eLON_arrValue07_Out_of_range := 91,
  eLON_arrValue08_Out_of_range := 92,
  eLON_arrValue09_Out_of_range := 93,

  eLON_arrValue10_Out_of_range := 100,
  eLON_arrValue11_Out_of_range := 101,
  eLON_arrValue12_Out_of_range := 102,
  eLON_arrValue13_Out_of_range := 103,
  eLON_arrValue14_Out_of_range := 104,
  eLON_arrValue15_Out_of_range := 105,
  eLON_arrValue16_Out_of_range := 106,
  eLON_arrValue17_Out_of_range := 107,
  eLON_arrValue18_Out_of_range := 108,
  eLON_arrValue19_Out_of_range := 109,

  eLON_arrValue20_Out_of_range := 115,
  eLON_arrValue21_Out_of_range := 116,
  eLON_arrValue22_Out_of_range := 117,
  eLON_arrValue23_Out_of_range := 118,
  eLON_arrValue24_Out_of_range := 119,
  eLON_arrValue25_Out_of_range := 120,
)

```

```

eLON_arrValue26_Out_of_range := 121,
eLON_arrValue27_Out_of_range := 122,
eLON_arrValue28_Out_of_range := 123,
eLON_arrValue29_Out_of_range := 124,

eLON_arrValue30_Out_of_range := 130,
eLON_arrValue31_Out_of_range := 131,
eLON_arrValue32_Out_of_range := 132,
eLON_arrValue33_Out_of_range := 133,
eLON_arrValue34_Out_of_range := 134,
eLON_arrValue35_Out_of_range := 135,
eLON_arrValue36_Out_of_range := 136,
eLON_arrValue37_Out_of_range := 137,
eLON_arrValue38_Out_of_range := 138,
eLON_arrValue39_Out_of_range := 139,
eLON_arrValue40_Out_of_range := 140,

eLON_087uiDay_Out_of_range := 145,
eLON_087uiHour_Out_of_range := 146,
eLON_087uiMinute_Out_of_range := 147,
eLON_087uiSecond_Out_of_range := 148,
eLON_087uiMillisecond_Out_of_range := 149,

eLON_ePriority_level_Out_of_range := 155,
eLON_eAlarm_type_Out_of_range := 156,

eLON_Currency_Out_of_range := 160,

eLON_diRw_ptr_Out_of_range := 165,

eLON_Object_request_Out_of_range := 170,

eLON_094eLearn_Out_of_range := 175,
eLON_094uiHour_Out_of_range := 176,
eLON_094uiMinute_Out_of_range := 177,
eLON_094uiSecond_Out_of_range := 178,
eLON_094uiMillisecond_Out_of_range := 179,

eLON_095rValue_Out_of_range := 185,
eLON_095siState_Out_of_range := 186,

eLON_byInterp_pts_0_to_1_Out_of_range := 190,
eLON_byInterp_pts_1_to_2_Out_of_range := 191,
eLON_byInterp_pts_2_to_3_Out_of_range := 192,
eLON_byInterp_pts_3_to_4_Out_of_range := 193,
eLON_byInterp_pts_4_to_5_Out_of_range := 194,
eLON_byInterp_pts_5_to_6_Out_of_range := 195,
eLON_byInterp_pts_6_to_0_Out_of_range := 196,

eLON_rOccupied_cool_Out_of_range := 200,
eLON_rStandby_cool_Out_of_range := 201,
eLON_rUnoccupied_cool_Out_of_range := 202,
eLON_rOccupied_heat_Out_of_range := 203,
eLON_rStandby_heat_Out_of_range := 204,
eLON_rUnoccupied_heat_Out_of_range := 205,

eLON_111rPercent_Out_of_range := 210,
eLON_111eState_Out_of_range := 211,

eLON_eMode_Out_of_range := 215,
eLON_rHeat_output_primary_Out_of_range := 216,
eLON_rHeat_output_secondary_Out_of_range := 217,
eLON_rCool_output_Out_of_range := 218,
eLON_rEcon_output_Out_of_range := 219,
eLON_rFan_output_Out_of_range := 220,

eLON_115eFunction_Out_of_range := 225,

eLON_eFunction_Out_of_range := 226,
eLON_rSetting_Out_of_range := 227,
eLON_rRotation_Out_of_range := 228,
eLON_rFade_time_Out_of_range := 229,
eLON_rDelay_time_Out_of_range := 230,

eLON_eChlr_run_mode_Out_of_range := 235,
eLON_echlr_op_mode_Out_of_range := 236,

eLON_eNext_state_Out_of_range := 240,
eLON_eCurrent_state_Out_of_range := 241,

```



```

eLON_diSecond_time_offset_Out_of_range := 245,
eLON_eType_of_description_Out_of_range := 246,
eLON_byHour_of_start_DST_Out_of_range := 250,
eLON_byMinute_of_start_DST_Out_of_range := 251,
eLON_bySecond_of_start_DST_Out_of_range := 252,
eLON_byHour_of_end_DST_Out_of_range := 260,
eLON_byMinute_of_end_DST_Out_of_range := 261,
eLON_bySecond_of_end_DST_Out_of_range := 262,
eLON_stStart_DST_uiG_day_of_start_DST_Out_of_range := 263,
eLON_stStart_DST_uiJ_day_of_start_DST_Out_of_range := 264,
eLON_stStart_DST_stM_start_DST_byMonth_of_start_DST_Out_of_range := 265,
eLON_stStart_DST_stM_start_DST_byWeek_of_start_DST_Out_of_range := 266,
eLON_stStart_DST_stM_start_DST_eDateday_of_start_DST_Out_of_range := 267,
eLON_stEnd_DST_uiG_day_of_end_DST_Out_of_range := 268,
eLON_stEnd_DST_uiJ_day_of_end_DST_Out_of_range := 269,
eLON_stEnd_DST_stM_end_DST_byMonth_of_end_DST_Out_of_range := 270,
eLON_stEnd_DST_stM_end_DST_byWeek_of_end_DST_Out_of_range := 271,
eLON_stEnd_DST_stM_end_DST_eDateday_of_end_DST_Out_of_range := 272,

eLON_byLatitude_deg_Out_of_range := 280,
eLON_rLatitude_min_Out_of_range := 281,
eLON_byLongitude_deg_Out_of_range := 282,
eLON_rLongitude_min_Out_of_range := 283,

eLON_byNr_decimals_Out_of_range := 290,
eLON_eUnit_Out_of_range := 291,

eLON_137eUnit_Out_of_range := 295,
eLON_137byNr_decimals_Out_of_range := 296,
eLON_137byStatus_Out_of_range := 297,
eLON_137uiYear_Out_of_range := 298,
eLON_137uiMonth_Out_of_range := 299,
eLON_137uiDay_Out_of_range := 300,
eLON_137uiHour_Out_of_range := 301,
eLON_137uiMinute_Out_of_range := 302,
eLON_137uiSecond_Out_of_range := 303,

eLON_bySender_prio_Out_of_range := 310,

eLON_eStatus_Out_of_range := 315,
eLON_stSender_uiID_Out_of_range := 316,
eLON_stSender_stRange_uiLower_Out_of_range := 317,
eLON_stSender_stRange_uiUpper_Out_of_range := 318,
eLON_uiController_id_Out_of_range := 319,

eLON_ePan_dir_Out_of_range := 325,
eLON_rPan_speed_Out_of_range := 326,
eLON_eTilt_dir_Out_of_range := 327,
eLON_rTilt_speed_Out_of_range := 328,
eLON_eZoom_Out_of_range := 329,
eLON_rZoom_speed_Out_of_range := 330,

eLON_eAction_Out_of_range := 335,

eLON_byController_prio_Out_of_range := 340,
eLON_152eFunction_Out_of_range := 341,
eLON_152eAction_Out_of_range := 342,
eLON_stValue_stAbspos_rZoom_Out_of_range := 343,
eLON_stValue_stAbspos_rTilt_Out_of_range := 344,
eLON_stValue_stAbspos_rPan_Out_of_range := 345,

eLON_eMain_pump_Out_of_range := 350,
eLON_eBooster_pump_Out_of_range := 351,
eLON_ePriority_level_Out_of_range := 352,
eLON_eProcess_ready_Out_of_range := 353,
eLON_eEmergency_stop_activated_Out_of_range := 354,
eLON_eMain_pump_drive_enabled_Out_of_range := 355,
eLON_eBooster_pump_drive_enabled_Out_of_range := 356,
eLON_eMaintenance_required_Out_of_range := 357,

eLON_eControl_status_Out_of_range := 365,
eLON_stControl_device_addr_byDomain_length_Out_of_range := 366,
eLON_stControl_device_addr_bySubnet_Out_of_range := 367,
eLON_stControl_device_addr_byNode_Out_of_range := 368,

eLON_rExhaust_temperature_Out_of_range := 375,
eLON_rExhaust_pressure_Out_of_range := 376,
eLON_rShaft_seal_purge_pressure_Out_of_range := 377,
eLON_rSupply_voltage_Out_of_range := 378,
eLON_eCoolant_flow_low_Out_of_range := 379,

```

```

eLON_eDilution_active_Out_of_range           := 380,
eLON_eBallast_dilution_active_Out_of_range   := 381,
eLON_eInlet_purge_dilution_active_Out_of_range := 382,
eLON_eExhaust_dilution_active_Out_of_range   := 383,
eLON_eDilution_flow_Out_of_range            := 384,
eLON_ePower_supply_on_Out_of_range           := 385,

eLON_rRotational_speed_Out_of_range           := 390,
eLON_rBody_temperature_Out_of_range           := 391,
eLON_rMotor_external_temperature_Out_of_range := 392,
eLON_rMotor_internal_temperature_Out_of_range := 393,
eLON_eMotor_overloaded_Out_of_range           := 394,
eLON_eOil_level_low_Out_of_range              := 395,
eLON_ePhase_imbalance_detected_Out_of_range  := 396,
eLON_rCurrent_usage_Out_of_range             := 397,
eLON_rPower_usage_Out_of_range                := 398,
eLON_eTemperature_control_Out_of_range        := 399,
eLON_eElectromagnetic_brake_active_Out_of_range := 400,
eLON_eFriction_brake_active_Out_of_range     := 401,
eLON_eGas_brake_active_Out_of_range          := 402,

eLON_164iMilliseconds_Out_of_range           := 410,
eLON_164ePriority_level_Out_of_range          := 411,
eLON_164eAlarm_type_Out_of_range             := 412,

eLON_byType_scope_Out_of_range               := 420,
eLON_uiType_index_Out_of_range               := 421,
eLON_eType_category_Out_of_range             := 422,
eLON_byType_length_Out_of_range              := 423,

eLON_eCmd_fb_Out_of_range                     := 430,

eLON_byManufacturer_Out_of_range             := 435,

eLON_eDevice_select_Out_of_range             := 440,

eLON_stPos_eFunction_Out_of_range            := 445,
eLON_stPos_rSetting_Out_of_range             := 446,
eLON_stPos_rRotation_Out_of_range           := 447,
eLON_eCmd_source_Out_of_range                := 448,
eLON_eError_code_Out_of_range                := 449,

eLON_181stAddr_talk_eAudio_sensor_type_Out_of_range := 455,
eLON_181stAddr_talk_byCar_id_Out_of_range   := 456,
eLON_181stAddr_talk_byLocation_Out_of_range := 457,
eLON_181stAddr_talk_byUnit_id_Out_of_range  := 458,
eLON_181stAddr_init_eAudio_sensor_type_Out_of_range := 459,
eLON_181stAddr_init_byCar_id_Out_of_range   := 450,
eLON_181stAddr_init_byLocation_Out_of_range := 461,
eLON_181stAddr_init_byUnit_id_Out_of_range  := 462,
eLON_181eAudio_type_Out_of_range            := 463,
eLON_181byAudio_line_Out_of_range           := 464,
eLON_181stAddr_dest_stP2p_eAudio_sensor_type_Out_of_range := 465,
eLON_181stAddr_dest_stP2p_byCar_id_Out_of_range := 466,
eLON_181stAddr_dest_stP2p_byLocation_Out_of_range := 467,
eLON_181stAddr_dest_stP2p_byUnit_id_Out_of_range := 468,

eLON_stAddr_dest_stP2p_eAudio_sensor_type_Out_of_range := 475,
eLON_stAddr_dest_stP2p_byCar_id_Out_of_range := 476,
eLON_stAddr_dest_stP2p_byLocation_Out_of_range := 477,
eLON_stAddr_dest_stP2p_byUnit_id_Out_of_range := 478,
eLON_stAddr_init_eAudio_sensor_type_Out_of_range := 479,
eLON_stAddr_init_byCar_id_Out_of_range := 480,
eLON_stAddr_init_byLocation_Out_of_range := 481,
eLON_stAddr_init_byUnit_id_Out_of_range := 482,
eLON_eAudio_type_Out_of_range := 483,

eLON_eCycle_Out_of_range := 490,
eLON_eSubcycle_Out_of_range := 491,
eLON_stFunction_eProgram_Out_of_range := 492,
eLON_stFunction_stWash_eLoad_level_Out_of_range := 493,
eLON_stFunction_stWash_ePrewash_Out_of_range := 494,
eLON_stFunction_stRinse_eOption_Out_of_range := 495,
eLON_stFunction_stRinse_byRepeat_Out_of_range := 496,
eLON_stFunction_stSpin_eHold_Out_of_range := 497,
eLON_stFunction_stDry_byTemp_Out_of_range := 498,
eLON_stFunction_stDry_stDuration_eDryness_Out_of_range := 499,

eLON_186eCycle_Out_of_range := 505,

```

```

eLON_186eSubcycle_Out_of_range           := 506,
eLON_stWasher_command_data_eCycle_Out_of_range := 507,
eLON_stWasher_command_data_eSubcycle_Out_of_range := 508,
eLON_stWasher_command_data_stFunction_eProgram_Out_of_range := 509,
eLON_stWasher_command_data_stFunction_stWash_eLoad_level_Out_of_range := 510,
eLON_stWasher_command_data_stFunction_stWash_ePrewash_Out_of_range := 511,
eLON_stWasher_command_data_stFunction_stRinse_eOption_Out_of_range := 512,
eLON_stWasher_command_data_stFunction_stRinse_byRepeat_Out_of_range := 513,
eLON_stWasher_command_data_stFunction_stSpin_eHold_Out_of_range := 514,
eLON_stWasher_command_data_stFunction_stDry_byTemp_Out_of_range := 515,
eLON_stWasher_command_data_stFunction_stDry_stDuration_eDryness_Out_of_range := 516,

eLON_eState_Out_of_range                 := 518,
eLON_stSetting_rValue_Out_of_range       := 519,
eLON_stSetting_rChange_Out_of_range      := 520,
eLON_stSetting_rMultiplier_Out_of_range := 521,
eLON_stSetting_iAngle_Out_of_range       := 522,
eLON_stSetting_byGroup_number_Out_of_range := 523,
eLON_stSetting_siFan_level_Out_of_range  := 524,

eLON_stColor_value_stCIE1931_lumen_rX_Out_of_range := 525,
eLON_stColor_value_stCIE1931_lumen_rY_Out_of_range := 526,
eLON_stColor_value_stCIE1931_lumen_udiAbsolute_Y_Out_of_range := 527,
eLON_stColor_value_stCIE1931_percent_rX_Out_of_range := 528,
eLON_stColor_value_stCIE1931_percent_rY_Out_of_range := 529,
eLON_stColor_value_stCIE1931_percent_rPercent_Y_Out_of_range := 530,
eLON_stColor_value_uiColor_temperature_Out_of_range := 531,

eLON_191eStatus_Out_of_range             := 535,
eLON_uiLog_number_Out_of_range            := 536,
eLON_rLevel_Out_of_range                  := 537,
eLON_stCurrent_notify_time_rHundredths_Out_of_range := 538,
eLON_stPrevious_notify_time_rHundredths_Out_of_range := 539,

eLON_rHundredths_Out_of_range             := 545,

eLON_stStart_time_rHundredths_Out_of_range := 550,
eLON_stEnd_time_rHundredths_Out_of_range := 551,

eLON_rComplete_Out_of_range              := 565,

eLON_stTime_actual_rHundredths_Out_of_range := 570,
eLON_stTime_previous_rHundredths_Out_of_range := 571,

eLON_lrEnergy_Out_of_range                := 585,
eLON_rPowerFactor_Out_of_range            := 586,
eLON_rPower_Out_of_range                  := 587,
eLON_rBallastTemp_Out_of_range            := 588,

eLON_lrLongitude_Out_of_range             := 595,
eLON_lrLatitude_Out_of_range              := 596
)
END_TYPE

```

eLON_no_Error: there is no error pending.

eLON_Value_out_of_range: the input variable "Value" is outside the permitted range. The value was not sent. "Value" can have different formats with corresponding prefix (e.g. LREAL = lValue).

eLON_Terminal_not_ready: the function block "FB_LON_KL6401" passes through an initialization step sequence (query terminal type, query firmware etc.) when the PLC is started. This message is issued as along as the initialization is in progress. If an error is pending after a PLC reset, the controller should be de-energized once.

eLON_Wrong_SNVT_Typ: the received SNVT type does not match the SNVT type of the addressed NV index (input variable "wld")

eLON_Wrong_wNVIndex: wrong NV index.

eKL6401_Wrong_Terminal: no KL6401 was detected.

eKL6401_Error: the function block "FB_LON_KL6401" has an error. The error code is shown at output "dwErrorKL".

eKL6401_Terminal_is_not_initialized: the terminal is not initialized. This message usually means that there is no connection to the terminal. Terminal linked to the variables in the System Manager? Terminal plugged in incorrectly? Clean all, rebuild all and read again in the System Manager?

eLON_L_star_Out_of_range: SNVT 70 / The input variable "stValue.L_star" is outside the permitted range. The value was not sent.

eLON_A_star_Out_of_range: SNVT 70 / The input variable "stValue.A_star" is outside the permitted range. The value was not sent.

eLON_B_star_Out_of_range: SNVT 70 / The input variable "stValue.B_star" is outside the permitted range. The value was not sent.

eLON_eRequest_Out_of_range: SNVT 73 / The input variable "stValue.eRequest" is outside the permitted range. The value was not sent.

eLON_wYear_Out_of_range: SNVT 084 / 088 / The input variable "stValue.wYear" is outside the permitted range. The value was not sent.

eLON_wMonth_Out_of_range: SNVT 084 / 088 / The input variable "stValue.wMonth" is outside the permitted range. The value was not sent.

eLON_wDay_Out_of_range: SNVT 084 / 088 / The input variable "stValue.wDay" is outside the permitted range. The value was not sent.

eLON_wHour_Out_of_range: SNVT 084 / 088 / The input variable "stValue.wHour" is outside the permitted range. The value was not sent.

eLON_wMinute_Out_of_range: SNVT 084 / 088 / The input variable "stValue.wMinute" is outside the permitted range. The value was not sent.

eLON_wSecond_Out_of_range: SNVT 084 / 088 / The input variable "stValue.wSecond" is outside the permitted range. The value was not sent.

eLON_wMillisecond_Out_of_range: SNVT 088 / The input variable "stValue.wMillisecond" is outside the permitted range. The value was not sent.

eLON_rZero_Out_of_range: SNVT 085 / The input variable "stValue.rZero" is outside the permitted range. The value was not sent.

eLON_rSpan_Out_of_range: SNVT 085 / The input variable "stValue.rSpan" is outside the permitted range. The value was not sent.

eLON_arrValue01_Out_of_range: SNVT 086 / The input variable "arrValue[1]" is outside the permitted range. The value was not sent.

eLON_arrValue02_Out_of_range: SNVT 086 / The input variable "arrValue[2]" is outside the permitted range. The value was not sent.

eLON_arrValue03_Out_of_range: SNVT 086 / The input variable "arrValue[3]" is outside the permitted range. The value was not sent.

eLON_arrValue04_Out_of_range: SNVT 086 / The input variable "arrValue[4]" is outside the permitted range. The value was not sent.

eLON_arrValue05_Out_of_range: SNVT 086 / The input variable "arrValue[5]" is outside the permitted range. The value was not sent.

eLON_arrValue06_Out_of_range: SNVT 086 / The input variable "arrValue[6]" is outside the permitted range. The value was not sent.

eLON_arrValue07_Out_of_range: SNVT 086 / The input variable "arrValue[7]" is outside the permitted range. The value was not sent.

eLON_arrValue08_Out_of_range: SNVT 086 / The input variable "arrValue[8]" is outside the permitted range. The value was not sent.

eLON_arrValue09_Out_of_range: SNVT 086 / The input variable "arrValue[9]" is outside the permitted range. The value was not sent.

eLON_arrValue10_Out_of_range: SNVT 086 / The input variable "arrValue[10]" is outside the permitted range. The value was not sent.

eLON_arrValue11_Out_of_range: SNVT 086 / The input variable "arrValue[11]" is outside the permitted range. The value was not sent.

eLON_arrValue12_Out_of_range: SNVT 086 / The input variable "arrValue[12]" is outside the permitted range. The value was not sent.

eLON_arrValue13_Out_of_range: SNVT 086 / The input variable "arrValue[13]" is outside the permitted range. The value was not sent.

eLON_arrValue14_Out_of_range: SNVT 086 / The input variable "arrValue[14]" is outside the permitted range. The value was not sent.

eLON_arrValue15_Out_of_range: SNVT 086 / The input variable "arrValue[15]" is outside the permitted range. The value was not sent.

eLON_arrValue16_Out_of_range: SNVT 086 / The input variable "arrValue[16]" is outside the permitted range. The value was not sent.

eLON_arrValue17_Out_of_range: SNVT 086 / The input variable "arrValue[17]" is outside the permitted range. The value was not sent.

eLON_arrValue18_Out_of_range: SNVT 086 / The input variable "arrValue[18]" is outside the permitted range. The value was not sent.

eLON_arrValue19_Out_of_range: SNVT 086 / The input variable "arrValue[19]" is outside the permitted range. The value was not sent.

eLON_arrValue20_Out_of_range: SNVT 086 / The input variable "arrValue[20]" is outside the permitted range. The value was not sent.

eLON_arrValue21_Out_of_range: SNVT 086 / The input variable "arrValue[21]" is outside the permitted range. The value was not sent.

eLON_arrValue22_Out_of_range: SNVT 086 / The input variable "arrValue[22]" is outside the permitted range. The value was not sent.

eLON_arrValue23_Out_of_range: SNVT 086 / The input variable "arrValue[23]" is outside the permitted range. The value was not sent.

eLON_arrValue24_Out_of_range: SNVT 086 / The input variable "arrValue[24]" is outside the permitted range. The value was not sent.

eLON_arrValue25_Out_of_range: SNVT 086 / The input variable "arrValue[25]" is outside the permitted range. The value was not sent.

eLON_arrValue26_Out_of_range: SNVT 086 / The input variable "arrValue[26]" is outside the permitted range. The value was not sent.

eLON_arrValue27_Out_of_range: SNVT 086 / The input variable "arrValue[27]" is outside the permitted range. The value was not sent.

eLON_arrValue28_Out_of_range: SNVT 086 / The input variable "arrValue[28]" is outside the permitted range. The value was not sent.

eLON_arrValue29_Out_of_range: SNVT 086 / The input variable "arrValue[29]" is outside the permitted range. The value was not sent.

eLON_arrValue30_Out_of_range: SNVT 086 / The input variable "arrValue[30]" is outside the permitted range. The value was not sent.

eLON_arrValue31_Out_of_range: SNVT 086 / The input variable "arrValue[31]" is outside the permitted range. The value was not sent.

eLON_arrValue32_Out_of_range: SNVT 086 / The input variable "arrValue[32]" is outside the permitted range. The value was not sent.

eLON_arrValue33_Out_of_range: SNVT 086 / The input variable "arrValue[33]" is outside the permitted range. The value was not sent.

eLON_arrValue34_Out_of_range: SNVT 086 / The input variable "arrValue[34]" is outside the permitted range. The value was not sent.

eLON_arrValue35_Out_of_range: SNVT 086 / The input variable "arrValue[35]" is outside the permitted range. The value was not sent.

eLON_arrValue36_Out_of_range: SNVT 086 / The input variable "arrValue[36]" is outside the permitted range. The value was not sent.

eLON_arrValue37_Out_of_range: SNVT 086 / The input variable "arrValue[37]" is outside the permitted range. The value was not sent.

eLON_arrValue38_Out_of_range: SNVT 086 / The input variable "arrValue[38]" is outside the permitted range. The value was not sent.

eLON_arrValue39_Out_of_range: SNVT 086 / The input variable "arrValue[39]" is outside the permitted range. The value was not sent.

eLON_arrValue40_Out_of_range: SNVT 086 / The input variable "arrValue[40]" is outside the permitted range. The value was not sent.

eLON_087uiDay_Out_of_range: SNVT 087 / The input variable "stValue.uiDay" is outside the permitted range. The value was not sent.

eLON_087uiHour_Out_of_range: SNVT 087 / The input variable "stValue.uiHour" is outside the permitted range. The value was not sent.

eLON_087uiMinute_Out_of_range: SNVT 087 / The input variable "stValue.uiMinute" is outside the permitted range. The value was not sent.

eLON_087uiSecond_Out_of_range: SNVT 087 / The input variable "stValue.uiSecond" is outside the permitted range. The value was not sent.

eLON_087uiMillisecond_Out_of_range: SNVT 087 / The input variable "stValue.uiMillisecond" is outside the permitted range. The value was not sent.

eLON_ePriority_level_Out_of_range: SNVT 088 / The input variable "stValue.ePriority_level" is outside the permitted range. The value was not sent.

eLON_eAlarm_type_Out_of_range: SNVT 088 / The input variable "stValue.eAlarm_type" is outside the permitted range. The value was not sent.

eLON_Currency_Out_of_range: SNVT 089 / The input variable "stValue.Currency" is outside the permitted range. The value was not sent.

eLON_diRw_ptr_Out_of_range: SNVT 090 / The input variable "stValue.diRw_ptr" is outside the permitted range. The value was not sent.

eLON_Object_request_Out_of_range: SNVT 092 / The input variable "stValue.Object_request" is outside the permitted range. The value was not sent.

eLON_094eLearn_Out_of_range: SNVT 094 / The input variable "stValue.eLearn" is outside the permitted range. The value was not sent.

eLON_094uiHour_Out_of_range: SNVT 094 / The input variable "stValue.uiHour" is outside the permitted range. The value was not sent.

eLON_094uiMinute_Out_of_range: SNVT 094 / The input variable "stValue.uiMinute" is outside the permitted range. The value was not sent.

eLON_094uiSecond_Out_of_range: SNVT 094 / The input variable "stValue.uiSecond" is outside the permitted range. The value was not sent.

eLON_094uiMillisecond_Out_of_range: SNVT 094 / The input variable "stValue.uiMillisecond" is outside the permitted range. The value was not sent.

eLON_095rValue_Out_of_range: SNVT 095 / The input variable "stValue.rValue" is outside the permitted range. The value was not sent.

eLON_095siState_Out_of_range: SNVT 095 / The input variable "stValue.siState" is outside the permitted range. The value was not sent.

eLON_byInterp_pts_0_to_1_Out_of_range: SNVT 096 / The input variable "stValue.byInterp_pts_0_to_1" is outside the permitted range. The value was not sent.

eLON_byInterp_pts_1_to_2_Out_of_range: SNVT 096 / The input variable "stValue.byInterp_pts_1_to_2" is outside the permitted range. The value was not sent.

eLON_byInterp_pts_2_to_3_Out_of_range: SNVT 096 / The input variable "stValue.byInterp_pts_2_to_3" is outside the permitted range. The value was not sent.

eLON_byInterp_pts_3_to_4_Out_of_range: SNVT 096 / The input variable "stValue.byInterp_pts_3_to_4" is outside the permitted range. The value was not sent.

eLON_byInterp_pts_4_to_5_Out_of_range: SNVT 096 / The input variable "stValue.byInterp_pts_4_to_5" is outside the permitted range. The value was not sent.

eLON_byInterp_pts_5_to_6_Out_of_range: SNVT 096 / The input variable "stValue.byInterp_pts_5_to_6" is outside the permitted range. The value was not sent.

eLON_byInterp_pts_6_to_0_Out_of_range: SNVT 096 / The input variable "stValue.byInterp_pts_6_to_0" is outside the permitted range. The value was not sent.

eLON_rOccupied_cool_Out_of_range: SNVT 106 / The input variable "stValue.rOccupied_cool" is outside the permitted range. The value was not sent.

eLON_rStandby_cool_Out_of_range: SNVT 106 / The input variable "stValue.rStandby_cool" is outside the permitted range. The value was not sent.

eLON_rUnoccupied_cool_Out_of_range: SNVT 106 / The input variable "stValue.rUnoccupied_cool" is outside the permitted range. The value was not sent.

eLON_rOccupied_heat_Out_of_range: SNVT 106 / The input variable "stValue.rOccupied_heat" is outside the permitted range. The value was not sent.

eLON_rStandby_heat_Out_of_range: SNVT 106 / The input variable "stValue.rStandby_heat" is outside the permitted range. The value was not sent.

eLON_rUnoccupied_heat_Out_of_range: SNVT 106 / The input variable "stValue.rUnoccupied_heat" is outside the permitted range. The value was not sent.

eLON_111rPercent_Out_of_range: SNVT 111 / The input variable "stValue.rPercent" is outside the permitted range. The value was not sent.

eLON_111eState_Out_of_range: SNVT 111 / The input variable "stValue.eState" is outside the permitted range. The value was not sent.

eLON_eMode_Out_of_range: SNVT 112 / The input variable "stValue.eMode" is outside the permitted range. The value was not sent.

eLON_rHeat_output_primary_Out_of_range: SNVT 112 / The input variable "stValue.rHeat_output_primary" is outside the permitted range. The value was not sent.

eLON_rHeat_output_secondary_Out_of_range: SNVT 112 / The input variable "stValue.rHeat_output_secondary" is outside the permitted range. The value was not sent.

eLON_rCool_output_Out_of_range: SNVT 112 / The input variable "stValue.rCool_output" is outside the permitted range. The value was not sent.

eLON_rEcon_output_Out_of_range: SNVT 112 / The input variable "stValue.rEcon_output" is outside the permitted range. The value was not sent.

eLON_rFan_output_Out_of_range: SNVT 112 / The input variable "stValue.rFan_output" is outside the permitted range. The value was not sent.

eLON_115eFunction_Out_of_range: SNVT 115 / The input variable "stValue.eFunction" is outside the permitted range. The value was not sent.

eLON_eFunction_Out_of_range: SNVT 116 / 117 / The input variable "stValue.eFunction" is outside the permitted range. The value was not sent.

eLON_rSetting_Out_of_range: SNVT 116 / 117 / The input variable "stValue.rSetting" is outside the permitted range. The value was not sent.

eLON_rRotation_Out_of_range: SNVT 116 / 117 / The input variable "stValue.rRotation" is outside the permitted range. The value was not sent.

eLON_rFade_time_Out_of_range: SNVT 116 / The input variable "stValue.rFade_time" is outside the permitted range. The value was not sent.

eLON_rDelay_time_Out_of_range: SNVT 116 / The input variable "stValue.rDelay_time" is outside the permitted range. The value was not sent.

eLON_eChlr_run_mode_Out_of_range: SNVT 127 / The input variable "stValue.eChlr_run_mode" is outside the permitted range. The value was not sent.

eLON_echlr_op_mode_Out_of_range: SNVT 127 / The input variable "stValue.echlr_op_mode" is outside the permitted range. The value was not sent.

eLON_eNext_state_Out_of_range: SNVT 128 / The input variable "stValue.eNext_state" is outside the permitted range. The value was not sent.

eLON_eCurrent_state_Out_of_range: SNVT 128 / The input variable "stValue.eCurrent_state" is outside the permitted range. The value was not sent.

eLON_diSecond_time_offset_Out_of_range: SNVT 134 / The input variable "stValue.diSecond_time_offset" is outside the permitted range. The value was not sent.

eLON_eType_of_description_Out_of_range: SNVT 134 / The input variable "stValue.eType_of_description" is outside the permitted range. The value was not sent.

eLON_byHour_of_start_DST_Out_of_range: SNVT 134 / The input variable "stValue.byHour_of_start_DST" is outside the permitted range. The value was not sent.

eLON_byMinute_of_start_DST_Out_of_range: SNVT 134 / The input variable "stValue.byMinute_of_start_DST" is outside the permitted range. The value was not sent.

eLON_bySecond_of_start_DST_Out_of_range: SNVT 134 / The input variable "stValue.bySecond_of_start_DST" is outside the permitted range. The value was not sent.

eLON_byHour_of_end_DST_Out_of_range: SNVT 134 / The input variable "stValue.byHour_of_end_DST" is outside the permitted range. The value was not sent.

eLON_byMinute_of_end_DST_Out_of_range: SNVT 134 / The input variable "stValue.byMinute_of_end_DST" is outside the permitted range. The value was not sent.

eLON_bySecond_of_end_DST_Out_of_range: SNVT 134 / The input variable "stValue.bySecond_of_end_DST" is outside the permitted range. The value was not sent.

eLON_stStart_DST_uiG_day_of_start_DST_Out_of_range: SNVT 134 / The input variable "stValue.stStart_DST.uiG_day_of_start_DST" is outside the permitted range. The value was not sent.

eLON_stStart_DST_uiJ_day_of_start_DST_Out_of_range: SNVT 134 / The input variable "stValue.stStart_DST.uiJ_day_of_start_DST" is outside the permitted range. The value was not sent.

eLON_stStart_DST_stM_start_DST_byMonth_of_start_DST_Out_of_range: SNVT 134 / The input variable "stValue.stStart_DST.stM_start_DST.byMonth_of_start_DST" is outside the permitted range. The value was not sent.

eLON_stStart_DST_stM_start_DST_byWeek_of_start_DST_Out_of_range: SNVT 134 / The input variable "stValue.stStart_DST.stM_start_DST.byWeek_of_start_DST" is outside the permitted range. The value was not sent.

eLON_stStart_DST_stM_start_DST_eDateday_of_start_DST_Out_of_range: SNVT 134 / The input variable "stValue.stStart_DST.stM_start_DST.eDateday_of_start_DST" is outside the permitted range. The value was not sent.

eLON_stEnd_DST_uiG_day_of_end_DST_Out_of_range: SNVT 134 / The input variable "stValue.stEnd_DST.uiG_day_of_end_DST" is outside the permitted range. The value was not sent.

eLON_stEnd_DST_uiJ_day_of_end_DST_Out_of_range: SNVT 134 / The input variable "stValue.stEnd_DST.uiJ_day_of_end_DST" is outside the permitted range. The value was not sent.

eLON_stEnd_DST_stM_end_DST_byMonth_of_end_DST_Out_of_range: SNVT 134 / The input variable "stValue.stEnd_DST.stM_end_DST.byMonth_of_end_DST" is outside the permitted range. The value was not sent.

eLON_stEnd_DST_stM_end_DST_byWeek_of_end_DST_Out_of_range: SNVT 134 / The input variable "stValue.stEnd_DST.stM_end_DST.byWeek_of_end_DST" is outside the permitted range. The value was not sent.

eLON_stEnd_DST_stM_end_DST_eDateday_of_end_DST_Out_of_range: SNVT 134 / The input variable "stValue.stEnd_DST.stM_end_DST.eDateday_of_end_DST" is outside the permitted range. The value was not sent.

eLON_byLatitude_deg_Out_of_range: SNVT 135 / The input variable "stValue.byLatitude" is outside the permitted range. The value was not sent.

eLON_rLatitude_min_Out_of_range: SNVT 135 / The input variable "stValue.rLatitude_min" is outside the permitted range. The value was not sent.

eLON_bylongitude_deg_Out_of_range: SNVT 135 / The input variable "stValue.bylongitude_deg" is outside the permitted range. The value was not sent.

eLON_rLongitude_min_Out_of_range: SNVT 135 / The input variable "stValue.rLongitude_min" is outside the permitted range. The value was not sent.

eLON_byNr_decimals_Out_of_range: SNVT 136 / The input variable "stValue.byNr_decimals" is outside the permitted range. The value was not sent.

eLON_eUnit_Out_of_range: SNVT 136 / The input variable "stValue.eUnit" is outside the permitted range. The value was not sent.

eLON_137eUnit_Out_of_range: SNVT 137 / The input variable "stValue.eUnit" is outside the permitted range. The value was not sent.

eLON_137byNr_decimals_Out_of_range: SNVT 137 / The input variable "stValue.byNr_decimals" is outside the permitted range. The value was not sent.

eLON_137byStatus_Out_of_range: SNVT 137 / The input variable "stValue.byStatus" is outside the permitted range. The value was not sent.

eLON_137uiYear_Out_of_range: SNVT 137 / The input variable "stValue.uiYear" is outside the permitted range. The value was not sent.

eLON_137uiMonth_Out_of_range: SNVT 137 / The input variable "stValue.uiMonth" is outside the permitted range. The value was not sent.

eLON_137uiDay_Out_of_range: SNVT 137 / The input variable "stValue.uiDay" is outside the permitted range. The value was not sent.

eLON_137uiHour_Out_of_range: SNVT 137 / The input variable "stValue.uiHour" is outside the permitted range. The value was not sent.

eLON_137uiMinute_Out_of_range: SNVT 137 / The input variable "stValue.uiMinute" is outside the permitted range. The value was not sent.

eLON_137uiSecond_Out_of_range: SNVT 137 / The input variable "stValue.uiSecond" is outside the permitted range. The value was not sent.

eLON_bySender_prio_Out_of_range: SNVT 148 / The input variable "stValue.bySender_prio" is outside the permitted range. The value was not sent.

eLON_eStatus_Out_of_range: SNVT 149 / The input variable "stValue.eStatus" is outside the permitted range. The value was not sent.

eLON_stSender_uiID_Out_of_range: SNVT 149 / The input variable "stValue.stSender.uiID" is outside the permitted range. The value was not sent.

eLON_stSender_stRange_uiLower_Out_of_range: SNVT 149 / The input variable "stValue.stSender.stRange.uiLower" is outside the permitted range. The value was not sent.

eLON_stSender_stRange_uiUpper_Out_of_range: SNVT 149 / The input variable "stValue.stSender.stRange.uiUpper" is outside the permitted range. The value was not sent.

eLON_uiController_id_Out_of_range: SNVT 149 / The input variable "stValue.uiController" is outside the permitted range. The value was not sent.

eLON_ePan_dir_Out_of_range: SNVT 150 / The input variable "stValue." is outside the permitted range. The value was not sent.

eLON_rPan_speed_Out_of_range: SNVT 150 / The input variable "stValue." is outside the permitted range. The value was not sent.

eLON_eTilt_dir_Out_of_range: SNVT 150 / The input variable "stValue." is outside the permitted range. The value was not sent.

eLON_rTilt_speed_Out_of_range: SNVT 150 / The input variable "stValue." is outside the permitted range. The value was not sent.

eLON_eZoom_Out_of_range: SNVT 150 / The input variable "stValue." is outside the permitted range. The value was not sent.

eLON_rZoom_speed_Out_of_range: SNVT 150 / The input variable "stValue." is outside the permitted range. The value was not sent.

eLON_eAction_Out_of_range: SNVT 151 / The input variable "stValue.eAction" is outside the permitted range. The value was not sent.

eLON_byController_prio_Out_of_range: SNVT 152 / The input variable "stValue.byController" is outside the permitted range. The value was not sent.

eLON_152eFunction_Out_of_range: SNVT 152 / The input variable "stValue.eFunction" is outside the permitted range. The value was not sent.

eLON_152eAction_Out_of_range: SNVT 152 / The input variable "stValue.eAction" is outside the permitted range. The value was not sent.

eLON_stValue_stAbspos_rZoom_Out_of_range: SNVT 152 / The input variable "stValue.stValue.stAbspos.rZoom" is outside the permitted range. The value was not sent.

eLON_stValue_stAbspos_rTilt_Out_of_range: SNVT 152 / The input variable "stValue.stValue.stAbspos.rTilt" is outside the permitted range. The value was not sent.

eLON_stValue_stAbspos_rPan_Out_of_range: SNVT 152 / The input variable "stValue.stValue.stAbspos.rPan" is outside the permitted range. The value was not sent.

eLON_eMain_pump_Out_of_range: SNVT 156 / The input variable "stValue.eMain_pump" is outside the permitted range. The value was not sent.

eLON_eBooster_pump_Out_of_range: SNVT 156 / The input variable "stValue.eBooster_pump" is outside the permitted range. The value was not sent.

eLON_ePriority_level_Out_of_range: SNVT 156 / The input variable "stValue.ePriority_level" is outside the permitted range. The value was not sent.

eLON_eProcess_ready_Out_of_range: SNVT 156 / The input variable "stValue.eProcess_ready" is outside the permitted range. The value was not sent.

eLON_eEmergency_stop_activated_Out_of_range: SNVT 156 / The input variable "stValue.eEmergency_stop_activated" is outside the permitted range. The value was not sent.

eLON_eMain_pump_drive_enabled_Out_of_range: SNVT 156 / The input variable "stValue.eMain_pump_drive_enabled" is outside the permitted range. The value was not sent.

eLON_eBooster_pump_drive_enabled_Out_of_range: SNVT 156 / The input variable "stValue.eBooster_pump_drive_enabled" is outside the permitted range. The value was not sent.

eLON_eMaintenance_required_Out_of_range: SNVT 156 / The input variable "stValue.eMaintenance_required" is outside the permitted range. The value was not sent.

eLON_eControl_status_Out_of_range: SNVT 157 / The input variable "stValue.eControl_status" is outside the permitted range. The value was not sent.

eLON_stControl_device_addr_byDomain_length_Out_of_range: SNVT 157 / The input variable "stValue.stControl_device_addr.byDomain_length" is outside the permitted range. The value was not sent.

eLON_stControl_device_addr_bySubnet_Out_of_range: SNVT 157 / The input variable "stValue.stControl_device_addr.bySubnet" is outside the permitted range. The value was not sent.

eLON_stControl_device_addr_byNode_Out_of_range: SNVT 157 / The input variable "stValue.stControl_device_addr.byNode" is outside the permitted range. The value was not sent.

eLON_rExhaust_temperature_Out_of_range: SNVT 158 / The input variable "stValue.rExhaust_temperature" is outside the permitted range. The value was not sent.

eLON_rExhaust_pressure_Out_of_range: SNVT 158 / The input variable "stValue.rExhaust_pressure" is outside the permitted range. The value was not sent.

eLON_rShaft_seal_purge_pressure_Out_of_range: SNVT 158 / The input variable "stValue.rShaft_seal_purge_pressure" is outside the permitted range. The value was not sent.

eLON_rSupply_voltage_Out_of_range: SNVT 158 / The input variable "stValue.rSupply_voltage" is outside the permitted range. The value was not sent.

eLON_eCoolant_flow_low_Out_of_range: SNVT 158 / The input variable "stValue.eCoolant_flow_low" is outside the permitted range. The value was not sent.

eLON_eDilution_active_Out_of_range: SNVT 158 / The input variable "stValue.eDilution_active" is outside the permitted range. The value was not sent.

eLON_eBallast_dilution_active_Out_of_range: SNVT 158 / The input variable "stValue.eBallast_dilution_active" is outside the permitted range. The value was not sent.

eLON_eInlet_purge_dilution_active_Out_of_range: SNVT 158 / The input variable "stValue.eInlet_purge_dilution_active" is outside the permitted range. The value was not sent.

eLON_eExhaust_dilution_active_Out_of_range: SNVT 158 / The input variable "stValue.eExhaust_dilution_active" is outside the permitted range. The value was not sent.

eLON_eDilution_flow_Out_of_range: SNVT 158 / The input variable "stValue.eDilution_flow" is outside the permitted range. The value was not sent.

eLON_ePower_supply_on_Out_of_range: SNVT 158 / The input variable "stValue.ePower_supply_on" is outside the permitted range. The value was not sent.

eLON_rRotational_speed_Out_of_range: SNVT 159 / The input variable "stValue.rRotational_speed" is outside the permitted range. The value was not sent.

eLON_rBody_temperature_Out_of_range: SNVT 159 / The input variable "stValue.rBody" is outside the permitted range. The value was not sent.

eLON_rMotor_external_temperature_Out_of_range: SNVT 159 / The input variable "stValue.rMotor_external_temperature" is outside the permitted range. The value was not sent.

eLON_rMotor_internal_temperature_Out_of_range: SNVT 159 / The input variable "stValue.eMotor_overloaded" is outside the permitted range. The value was not sent.

eLON_eMotor_overloaded_Out_of_range: SNVT 159 / The input variable "stValue." is outside the permitted range. The value was not sent.

eLON_eOil_level_low_Out_of_range: SNVT 159 / The input variable "stValue.ePhase_imbalance_detected" is outside the permitted range. The value was not sent.

eLON_ePhase_imbalance_detected_Out_of_range: SNVT 159 / The input variable "stValue." is outside the permitted range. The value was not sent.

eLON_rCurrent_usage_Out_of_range: SNVT 159 / The input variable "stValue.rCurrent_usage" is outside the permitted range. The value was not sent.

eLON_rPower_usage_Out_of_range: SNVT 159 / The input variable "stValue.Power_usage" is outside the permitted range. The value was not sent.

eLON_eTemperature_control_Out_of_range: SNVT 159 / The input variable "stValue.eElectromagnetic_brake_active" is outside the permitted range. The value was not sent.

eLON_eElectromagnetic_brake_active_Out_of_range: SNVT 159 / The input variable "stValue." is outside the permitted range. The value was not sent.

eLON_eFriction_brake_active_Out_of_range: SNVT 159 / The input variable "stValue.eFriction_brake_active" is outside the permitted range. The value was not sent.

eLON_eGas_brake_active_Out_of_range: SNVT 159 / The input variable "stValue.eGas_brake_active" is outside the permitted range. The value was not sent.

eLON_164iMilliseconds_Out_of_range: SNVT 164 / The input variable "stValue.iMilliseconds" is outside the permitted range. The value was not sent.

eLON_164ePriority_level_Out_of_range: SNVT 164 / The input variable "stValue.ePriority_level" is outside the permitted range. The value was not sent.

eLON_164eAlarm_type_Out_of_range: SNVT 164 / The input variable "stValue.eAlarm" is outside the permitted range. The value was not sent.

eLON_byType_scope_Out_of_range: SNVT 166 / The input variable "stValue.byType_scope" is outside the permitted range. The value was not sent.

eLON_uiType_index_Out_of_range: SNVT 166 / The input variable "stValue.uiType_index" is outside the permitted range. The value was not sent.

eLON_eType_category_Out_of_range: SNVT 166 / The input variable "stValue.eType_category" is outside the permitted range. The value was not sent.

eLON_byType_length_Out_of_range: SNVT 166 / The input variable "stValue.byType" is outside the permitted range. The value was not sent.

eLON_eCmd_fb_Out_of_range: SNVT 170 / The input variable "stValue.eCmd_fb" is outside the permitted range. The value was not sent.

eLON_byManufacturer_Out_of_range: SNVT 172 / The input variable "stValue.byManufacturer" is outside the permitted range. The value was not sent.

eLON_eDevice_select_Out_of_range: SNVT 175 / The input variable "stValue.eDevice_select" is outside the permitted range. The value was not sent.

eLON_stPos_eFunction_Out_of_range: SNVT 180 / The input variable "stValue.stPos_eFunction" is outside the permitted range. The value was not sent.

eLON_stPos_rSetting_Out_of_range: SNVT 180 / The input variable "stValue.stPos.rSetting" is outside the permitted range. The value was not sent.

eLON_stPos_rRotation_Out_of_range: SNVT 180 / The input variable "stValue.stPos.rRotation" is outside the permitted range. The value was not sent.

eLON_eCmd_source_Out_of_range: SNVT 180 / The input variable "stValue.eCmd_source" is outside the permitted range. The value was not sent.

eLON_eError_code_Out_of_range: SNVT 180 / The input variable "stValue._eError_code" is outside the permitted range. The value was not sent.

eLON_181stAddr_talk_eAudio_sensor_type_Out_of_range: SNVT 181 / The input variable "stValue.stAddr_talk.eAudio_sensor" is outside the permitted range. The value was not sent.

eLON_181stAddr_talk_byCar_id_Out_of_range: SNVT 181 / The input variable "stValue.stAddr_talk.byCar_id" is outside the permitted range. The value was not sent.

eLON_181stAddr_talk_byLocation_Out_of_range: SNVT 181 / The input variable "stValue.stAddr_talk.byLocation" is outside the permitted range. The value was not sent.

eLON_181stAddr_talk_byUnit_id_Out_of_range: SNVT 181 / The input variable "stValue.stAddr_talk.byUnit" is outside the permitted range. The value was not sent.

eLON_181stAddr_init_eAudio_sensor_type_Out_of_range: SNVT 181 / The input variable "stValue.stAddr_init.eAudio_sensor_type" is outside the permitted range. The value was not sent.

eLON_181stAddr_init_byCar_id_Out_of_range: SNVT 181 / The input variable "stValue.stAddr_init.byCar" is outside the permitted range. The value was not sent.

eLON_181stAddr_init_byLocation_Out_of_range: SNVT 181 / The input variable "stValue.stAddr_init.byLocation" is outside the permitted range. The value was not sent.

eLON_181stAddr_init_byUnit_id_Out_of_range: SNVT 181 / The input variable "stValue.stAddr_init.byUnit" is outside the permitted range. The value was not sent.

eLON_181eAudio_type_Out_of_range: SNVT 181 / The input variable "stValue.eAudio_type" is outside the permitted range. The value was not sent.

eLON_181byAudio_line_Out_of_range: SNVT 181 / The input variable "stValue.byAudio_line" is outside the permitted range. The value was not sent.

eLON_181stAddr_dest_stP2p_eAudio_sensor_type_Out_of_range: SNVT 181 / The input variable "stValue.stAddr_dest.stP2p.eAudio_sensor_type" is outside the permitted range. The value was not sent.

eLON_181stAddr_dest_stP2p_byCar_id_Out_of_range: SNVT 181 / The input variable "stValue.stAddr_dest.stP2p.byLocation" is outside the permitted range. The value was not sent.

eLON_181stAddr_dest_stP2p_byLocation_Out_of_range: SNVT 181 / The input variable "stValue." is outside the permitted range. The value was not sent.

eLON_181stAddr_dest_stP2p_byUnit_id_Out_of_range: SNVT 181 / The input variable "stValue.stAddr_dest.stP2p.byUnit_id" is outside the permitted range. The value was not sent.

eLON_stAddr_dest_stP2p_eAudio_sensor_type_Out_of_range: SNVT 182 / The input variable "stValue.stAddr_dest.stP2p.eAudio_sensor_type" is outside the permitted range. The value was not sent.

eLON_stAddr_dest_stP2p_byCar_id_Out_of_range: SNVT 182 / The input variable "stValue.stAddr_dest.stP2p.byCar" is outside the permitted range. The value was not sent.

eLON_stAddr_dest_stP2p_byLocation_Out_of_range: SNVT 182 / The input variable "stValue.stAddr_dest.stP2p.byLocation" is outside the permitted range. The value was not sent.

eLON_stAddr_dest_stP2p_byUnit_id_Out_of_range: SNVT 182 / The input variable "stValue.stAddr_init.eAudio_sensor_type" is outside the permitted range. The value was not sent.

eLON_stAddr_init_eAudio_sensor_type_Out_of_range: SNVT 182 / The input variable "stValue.stAddr_init.byCar_id" is outside the permitted range. The value was not sent.

eLON_stAddr_init_byCar_id_Out_of_range: SNVT 182 / The input variable "stValue." is outside the permitted range. The value was not sent.

eLON_stAddr_init_byLocation_Out_of_range: SNVT 182 / The input variable "stValue.stAddr_init.byLocation" is outside the permitted range. The value was not sent.

eLON_stAddr_init_byUnit_id_Out_of_range: SNVT 182 / The input variable "stValue.stAddr_init.byUnit_id" is outside the permitted range. The value was not sent.

eLON_eAudio_type_Out_of_range: SNVT 182 / The input variable "stValue.eAudio_type" is outside the permitted range. The value was not sent.

eLON_eCycle_Out_of_range: SNVT 184 / The input variable "stValue.eCycle" is outside the permitted range. The value was not sent.

eLON_eSubcycle_Out_of_range: SNVT 184 / The input variable "stValue.eSubcycle" is outside the permitted range. The value was not sent.

eLON_stFunction_eProgram_Out_of_range: SNVT 184 / The input variable "stValue.stFunction.eProgram" is outside the permitted range. The value was not sent.

eLON_stFunction_stWash_eLoad_level_Out_of_range: SNVT 184 / The input variable "stValue.stFunction.stWash_eLoad_level" is outside the permitted range. The value was not sent.

eLON_stFunction_stWash_ePrewash_Out_of_range: SNVT 184 / The input variable "stValue.stFunction.stWash.ePrewash" is outside the permitted range. The value was not sent.

eLON_stFunction_stRinse_eOption_Out_of_range: SNVT 184 / The input variable "stValue.stFunction.stRinse.eOption" is outside the permitted range. The value was not sent.

eLON_stFunction_stRinse_byRepeat_Out_of_range: SNVT 184 / The input variable "stValue.stFunction.stRinse.byRepeat" is outside the permitted range. The value was not sent.

eLON_stFunction_stSpin_eHold_Out_of_range: SNVT 184 / The input variable "stValue.stFunction.stSpin.eHold" is outside the permitted range. The value was not sent.

eLON_stFunction_stDry_byTemp_Out_of_range: SNVT 184 / The input variable "stValue.stFunction.stDry.byTemp" is outside the permitted range. The value was not sent.

eLON_stFunction_stDry_stDuration_eDryness_Out_of_range: SNVT 184 / The input variable "stValue.stFunction.stDry.stDuration.eDryness" is outside the permitted range. The value was not sent.

eLON_186eCycle_Out_of_range: SNVT 186 / The input variable "stValue.eCycle" is outside the permitted range. The value was not sent.

eLON_186eSubcycle_Out_of_range: SNVT 186 / The input variable "stValue.eSubcycle" is outside the permitted range. The value was not sent.

eLON_stWasher_command_data_eCycle_Out_of_range: SNVT 186 / The input variable "stValue.stWasher_command_data.eCycle" is outside the permitted range. The value was not sent.

eLON_stWasher_command_data_eSubcycle_Out_of_range: SNVT 186 / The input variable "stValue.stWasher_command_data.eSubcycle" is outside the permitted range. The value was not sent.

eLON_stWasher_command_data_stFunction_eProgram_Out_of_range: SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.eProgram" is outside the permitted range. The value was not sent.

eLON_stWasher_command_data_stFunction_stWash_eLoad_level_Out_of_range: SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.stWash.eLoad" is outside the permitted range. The value was not sent.

eLON_stWasher_command_data_stFunction_stWash_ePrewash_Out_of_range: SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.stWash.ePrewash" is outside the permitted range. The value was not sent.

eLON_stWasher_command_data_stFunction_stRinse_eOption_Out_of_range: SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.stRinse.eOption" is outside the permitted range. The value was not sent.

eLON_stWasher_command_data_stFunction_stRinse_byRepeat_Out_of_range: SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.stRinse.byRepeat" is outside the permitted range. The value was not sent.

eLON_stWasher_command_data_stFunction_stSpin_eHold_Out_of_range: SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.stSpin.eHold" is outside the permitted range. The value was not sent.

eLON_stWasher_command_data_stFunction_stDry_byTemp_Out_of_range: SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.stDry.byTemp" is outside the permitted range. The value was not sent.

eLON_stWasher_command_data_stFunction_stDry_eDryness_Out_of_range: SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.stDry.stDuration.eDryness" is outside the permitted range. The value was not sent.

eLON_eState_Out_of_range: SNVT 189 / The input variable "stValue.eState" is outside the permitted range. The value was not sent.

eLON_stSetting_rValue_Out_of_range: SNVT 189 / The input variable "stValue.stSettings.rValue" is outside the permitted range. The value was not sent.

eLON_stSetting_rChange_Out_of_range: SNVT 189 / The input variable "stValue.stSettings.rChange" is outside the permitted range. The value was not sent.

eLON_stSetting_rMultiplier_Out_of_range: SNVT 189 / The input variable "stValue.stSettings.rMultiplier" is outside the permitted range. The value was not sent.

eLON_stSetting_iAngle_Out_of_range: SNVT 189 / The input variable "stValue.stSettings.iAngle" is outside the permitted range. The value was not sent.

eLON_stSetting_byGroup_number_Out_of_range: SNVT 189 / The input variable "stValue.stSettings.byGroup_number" is outside the permitted range. The value was not sent.

eLON_stSetting_siFan_level_Out_of_range: SNVT 189 / The input variable "stValue.stSettings.siFan_level" is outside the permitted range. The value was not sent.

eLON_stColor_value_stCIE1931_lumen_rX_Out_of_range: SNVT 190 / The input variable "stValue.stColor_value.stCIE1931_lumen.rX" is outside the permitted range. The value was not sent.

eLON_stColor_value_stCIE1931_lumen_rY_Out_of_range: SNVT 190 / The input variable "stValue.stColor_value.stCIE1931_lumen.rY" is outside the permitted range. The value was not sent.

eLON_stColor_value_stCIE1931_lumen_udiAbsolute_Y_Out_of_range: SNVT 190 / The input variable "stValue.stColor_value.stCIE1931_lumen.udiAbsolute_Y" is outside the permitted range. The value was not sent.

eLON_stColor_value_stCIE1931_percent_rX_Out_of_range: SNVT 190 / The input variable "stValue.stColor_value.stCIE1931_percent.rX" is outside the permitted range. The value was not sent.

eLON_stColor_value_stCIE1931_percent_rY_Out_of_range: SNVT 190 / The input variable "stValue.stColor_value.stCIE1931_percent.rY" is outside the permitted range. The value was not sent.

eLON_stColor_value_stCIE1931_percent_rPercent_Y_Out_of_range: SNVT 190 / The input variable "stValue.stColor_value.stCIE1931_percent.rPercent_Y" is outside the permitted range. The value was not sent.

eLON_stColor_value_uiColor_temperature_Out_of_range: SNVT 190 / The input variable "stValue.stColor_value.uiColor_temperature" is outside the permitted range. The value was not sent.

eLON_191eStatus_Out_of_range: SNVT 191 / The input variable "stValue.Status" is outside the permitted range. The value was not sent.

eLON_uiLog_number_Out_of_range: SNVT 191 / The input variable "stValue.uiLog_number" is outside the permitted range. The value was not sent.

eLON_rLevel_Out_of_range: SNVT 191 / The input variable "stValue.rLevel" is outside the permitted range. The value was not sent.

eLON_stCurrent_notify_time_rHundredths_Out_of_range: SNVT 191 / The input variable "stValue.stCurrent_notify_time.rHundredths" is outside the permitted range. The value was not sent.

eLON_stPrevious_notify_time_rHundredths_Out_of_range: SNVT 191 / The input variable "stValue.stPrevious_notify_time.rHundredths" is outside the permitted range. The value was not sent.

eLON_rHundredths_Out_of_range: SNVT 192 / The input variable "stValue.rHundredths" is outside the permitted range. The value was not sent.

eLON_stStart_time_rHundredths_Out_of_range: SNVT 193 / The input variable "stValue.stStart_time.rHundredths" is outside the permitted range. The value was not sent.

eLON_stEnd_time_rHundredths_Out_of_range: SNVT 193 / The input variable "stValue.stEnd_time.rHundredths" is outside the permitted range. The value was not sent.

eLON_rComplete_Out_of_range: SNVT 194 / The input variable "stValue.rComplete" is outside the permitted range. The value was not sent.

eLON_stTime_actual_rHundredths_Out_of_range: SNVT 199 / The input variable "stValue.stTime_actual.rHundredths" is outside the permitted range. The value was not sent.

eLON_stTime_previous_rHundredths_Out_of_range: SNVT 199 / The input variable "stValue.stTime_previous.rHundredths" is outside the permitted range. The value was not sent.

eLON_lrEnergy_Out_of_range: SNVT 200 / The input variable "stValue.lrEnergy" is outside the permitted range. The value was not sent.

eLON_rPowerFactor_Out_of_range: SNVT 200 / The input variable "stValue.rPowerFactor" is outside the permitted range. The value was not sent.

eLON_rPower_Out_of_range: SNVT 200 / The input variable "stValue.rPower" is outside the permitted range. The value was not sent.

eLON_rBallastTemp_Out_of_range: SNVT 200 / The input variable "stValue.rBallastTemp" is outside the permitted range. The value was not sent.

eLON_lrLongitude_Out_of_range: SNVT 201 / The input variable "stValue.lrLongitude" is outside the permitted range. The value was not sent.

eLON_lrLatitude_Out_of_range: SNVT 201 / The input variable "stValue.lrLatitude" is outside the permitted range. The value was not sent.

7.3.2 E_LON_Parameter_Datatypes

Enums SNVT types

```

TYPE E_LON_Parameter_Datatypes :
(
  eEmpty           := 0,
  eSNVT_amp        := 1,
  eSNVT_amp_mil    := 2,
  eSNVT_angle      := 3,
  eSNVT_angle_vel  := 4,
  eSNVT_btu_kilo   := 5,
  eSNVT_btu_mega   := 6,
  eSNVT_char_ascii := 7,
  eSNVT_count      := 8,
  eSNVT_count_inc  := 9,
  eSNVT_date_cal   := 10,
  eSNVT_date_day   := 11,
  eSNVT_date_time  := 12,
  eSNVT_elec_kwh   := 13,
  eSNVT_elec_whr   := 14,
  eSNVT_flow       := 15,
  eSNVT_flow_mil  := 16,
  eSNVT_length     := 17,
  eSNVT_length_kilo := 18,
  eSNVT_length_micr := 19,
  eSNVT_length_mil := 20,
  eSNVT_lev_cont   := 21,
  eSNVT_lev_disc   := 22,
  eSNVT_mass       := 23,
  eSNVT_mass_kilo  := 24,
  eSNVT_mass_mega  := 25,
  eSNVT_mass_mil   := 26,
  eSNVT_power      := 27,
  eSNVT_power_kilo := 28,
  eSNVT_ppm        := 29,

```



```

eSNVT_press      := 30,
eSNVT_res        := 31,
eSNVT_res_kilo   := 32,
eSNVT_sound_db  := 33,
eSNVT_speed      := 34,
eSNVT_speed_mil := 35,
eSNVT_str_asc    := 36,
eSNVT_str_int    := 37,
eSNVT_telcom     := 38,
eSNVT_temp       := 39,
eSNVT_time_passed := 40,
eSNVT_vol        := 41,
eSNVT_vol_kilo   := 42,
eSNVT_vol_mil    := 43,
eSNVT_volt       := 44,
eSNVT_volt_dbmv  := 45,
eSNVT_volt_kilo  := 46,
eSNVT_volt_mil   := 47,
eSNVT_amp_f      := 48,
eSNVT_angle_f    := 49,
eSNVT_angle_vel_f := 50,
eSNVT_count_f    := 51,
eSNVT_count_inc_f := 52,
eSNVT_flow_f     := 53,
eSNVT_length_f   := 54,
eSNVT_lev_cont_f := 55,
eSNVT_mass_f     := 56,
eSNVT_power_f    := 57,
eSNVT_ppm_f      := 58,
eSNVT_press_f    := 59,
eSNVT_res_f      := 60,
eSNVT_sound_db_f := 61,
eSNVT_speed_f    := 62,
eSNVT_temp_f     := 63,
eSNVT_time_f     := 64,
eSNVT_vol_f      := 65,
eSNVT_volt_f     := 66,
eSNVT_btu_f      := 67,
eSNVT_elec_whr_f := 68,
eSNVT_config_src := 69,
eSNVT_color      := 70,
eSNVT_grammage   := 71,
eSNVT_grammage_f := 72,
eSNVT_file_req   := 73,
eSNVT_file_status := 74,
eSNVT_freq_f     := 75,
eSNVT_freq_hz    := 76,
eSNVT_freq_kilohz := 77,
eSNVT_freq_milhz := 78,
eSNVT_lux        := 79,
eSNVT_ISO_7811   := 80,
eSNVT_lev_percent := 81,
eSNVT_multiplier := 82,
eSNVT_state      := 83,
eSNVT_time_stamp := 84,
eSNVT_zerospans := 85,
eSNVT_magcard    := 86,
eSNVT_elapsed_tm := 87,
eSNVT_alarm      := 88,
eSNVT_currency   := 89,
eSNVT_file_pos   := 90,
eSNVT_muldiv     := 91,
eSNVT_obj_request := 92,
eSNVT_obj_status := 93,
eSNVT_preset     := 94,
eSNVT_switch     := 95,
eSNVT_trans_table := 96,
eSNVT_override   := 97,
eSNVT_pwr_fact   := 98,
eSNVT_pwr_fact_f := 99,
eSNVT_density    := 100,
eSNVT_density_f  := 101,
eSNVT_rpm        := 102,
eSNVT_hvac_emerg := 103,
eSNVT_angle_deg  := 104,
eSNVT_temp_p     := 105,
eSNVT_temp_setpt := 106,
eSNVT_time_sec   := 107,
eSNVT_hvac_mode  := 108,
eSNVT_occupancy  := 109,

```

```

eSNVT_area           := 110,
eSNVT_hvac_overid   := 111,
eSNVT_hvac_status   := 112,
eSNVT_press_p       := 113,
eSNVT_address       := 114,
eSNVT_scene         := 115,
eSNVT_scene_cfg     := 116,
eSNVT_setting       := 117,
eSNVT_evap_state    := 118,
eSNVT_therm_mode    := 119,
eSNVT_defr_mode     := 120,
eSNVT_defr_term     := 121,
eSNVT_defr_state    := 122,
eSNVT_time_min      := 123,
eSNVT_time_hour     := 124,
eSNVT_ph            := 125,
eSNVT_ph_f          := 126,
eSNVT_chlr_status   := 127,
eSNVT_tod_event     := 128,
eSNVT_smo_obscur    := 129,
eSNVT_fire_test     := 130,
eSNVT_temp_ror      := 131,
eSNVT_fire_init     := 132,
eSNVT_fire_indcte   := 133,
eSNVT_time_zone     := 134,
eSNVT_earth_pos     := 135,
eSNVT_reg_val       := 136,
eSNVT_reg_val_ts    := 137,
eSNVT_volt_ac       := 138,
eSNVT_amp_ac        := 139,

eSNVT_turbidity     := 143,
eSNVT_turbidity_f   := 144,
eSNVT_hvac_type     := 145,
eSNVT_elec_kwh_l    := 146,
eSNVT_temp_diff_p   := 147,
eSNVT_ctrl_req      := 148,
eSNVT_ctrl_resp     := 149,
eSNVT_ptz           := 150,
eSNVT_privacyzone   := 151,
eSNVT_pos_ctrl      := 152,
eSNVT_enthalpy     := 153,
eSNVT_gfci_status   := 154,
eSNVT_motor_state   := 155,
eSNVT_pumpset_mn    := 156,
eSNVT_ex_control    := 157,
eSNVT_pumpset_sn    := 158,
eSNVT_pump_sensor   := 159,
eSNVT_abs_humid     := 160,
eSNVT_flow_p        := 161,
eSNVT_dev_c_mode    := 162,
eSNVT_valve_mode    := 163,
eSNVT_alarm_2       := 164,
eSNVT_state_64      := 165,
eSNVT_nv_type       := 166,

eSNVT_ent_opmode    := 168,
eSNVT_ent_state     := 169,
eSNVT_ent_status    := 170,
eSNVT_flow_dir      := 171,
eSNVT_hvac_satsts   := 172,
eSNVT_dev_status    := 173,
eSNVT_dev_fault     := 174,
eSNVT_dev_maint     := 175,
eSNVT_date_event    := 176,
eSNVT_sched_val     := 177,
eSNVT_sec_state     := 178,
eSNVT_sec_status    := 179,
eSNVT_sblnd_state   := 180,
eSNVT_rac_ctrl      := 181,
eSNVT_rac_req       := 182,
eSNVT_count_32      := 183,
eSNVT_clothes_w_c   := 184,
eSNVT_clothes_w_m   := 185,
eSNVT_clothes_w_s   := 186,
eSNVT_clothes_w_a   := 187,
eSNVT_multiplier_s  := 188,
eSNVT_switch_2      := 189,
eSNVT_color_2       := 190,
eSNVT_log_status    := 191,

```

```
eSNVT_time_stamp_p      := 192,  
eSNVT_log_fx_request    := 193,  
eSNVT_log_fx_status     := 194,  
eSNVT_log_request       := 195,  
eSNVT_enthalpy_d        := 196,  
eSNVT_amp_ac_mil        := 197,  
eSNVT_time_hour_p       := 198,  
eSNVT_lamp_status       := 199,  
eSNVT_environment       := 200,  
eSNVT_geo_loc           := 201  
)  
END_TYPE
```

eEmpty:

eSNVT_amp: SNVT_amp

eSNVT_amp_mil: SNVT_amp_mil

eSNVT_angle: SNVT_angle

eSNVT_angle_vel: SNVT_angle_vel

eSNVT_btu_kilo: SNVT_btu_kilo

eSNVT_btu_mega: SNVT_btu_mega

eSNVT_char_ascii: SNVT_char_ascii

eSNVT_count: SNVT_count

eSNVT_count_inc: SNVT_count_inc

eSNVT_date_cal: SNVT_date_cal

eSNVT_date_day: SNVT_date_day

eSNVT_date_time: SNVT_date_time

eSNVT_elec_kwh: SNVT_elec_kwh

eSNVT_elec_whr: SNVT_elec_whr

eSNVT_flow: SNVT_flow

eSNVT_flow_mil: SNVT_flow_mil

eSNVT_length: SNVT_length

eSNVT_length_kilo: SNVT_length_kilo

eSNVT_length_micr: SNVT_length_micr

eSNVT_length_mil: SNVT_length_mil

eSNVT_lev_cont: SNVT_lev_cont

eSNVT_lev_disc: SNVT_lev_disc

eSNVT_mass: SNVT_mass

eSNVT_mass_kilo: SNVT_mass_kilo

eSNVT_mass_mega: SNVT_mass_mega

eSNVT_mass_mil: SNVT_mass_mil

eSNVT_power: SNVT_power

eSNVT_power_kilo: SNVT_power_kilo

eSNVT_ppm: SNVT_ppm

eSNVT_press: SNVT_press
eSNVT_res: SNVT_res
eSNVT_res_kilo: SNVT_res_kilo
eSNVT_sound_db: SNVT_sound_db
eSNVT_speed: SNVT_speed
eSNVT_speed_mil: SNVT_speed_mil
eSNVT_str_asc: SNVT_str_asc
eSNVT_str_int: SNVT_str_int
eSNVT_telcom: SNVT_telcom
eSNVT_temp: SNVT_temp
eSNVT_time_passed: SNVT_time_passed
eSNVT_vol: SNVT_vol
eSNVT_vol_kilo: SNVT_vol_kilo
eSNVT_vol_mil: SNVT_vol_mil
eSNVT_volt: SNVT_volt
eSNVT_volt_dbmv: SNVT_volt_dbmv
eSNVT_volt_kilo: SNVT_volt_kilo
eSNVT_volt_mil: SNVT_volt_mil
eSNVT_amp_f: SNVT_amp_f
eSNVT_angle_f: SNVT_angle_f
eSNVT_angle_vel_f: SNVT_angle_vel_f
eSNVT_count_f: SNVT_count_f
eSNVT_count_inc_f: SNVT_count_inc_f
eSNVT_flow_f: SNVT_flow_f
eSNVT_length_f: SNVT_length_f
eSNVT_lev_cont_f: SNVT_lev_cont_f
eSNVT_mass_f: SNVT_mass_f
eSNVT_power_f: SNVT_power_f
eSNVT_ppm_f: SNVT_ppm_f
eSNVT_press_f: SNVT_press_f
eSNVT_res_f: SNVT_res_f
eSNVT_sound_db_f: SNVT_sound_db_f
eSNVT_speed_f: SNVT_speed_f
eSNVT_temp_f: SNVT_temp_f
eSNVT_time_f: SNVT_time_f
eSNVT_vol_f: SNVT_vol_f

eSNVT_volt_f: SNVT_volt_f
eSNVT_btu_f: SNVT_btu_f
eSNVT_elec_whr_f: SNVT_elec_whr_f
eSNVT_config_src: SNVT_config_src
eSNVT_color: SNVT_color
eSNVT_grammage: SNVT_grammage
eSNVT_grammage_f: SNVT_grammage_f
eSNVT_file_req: SNVT_file_req
eSNVT_file_status: SNVT_file_status
eSNVT_freq_f: SNVT_freq_f
eSNVT_freq_hz: SNVT_freq_hz
eSNVT_freq_kilohz: SNVT_freq_kilohz
eSNVT_freq_milhz: SNVT_freq_milhz
eSNVT_lux: SNVT_lux
eSNVT_ISO_7811: SNVT_ISO_7811
eSNVT_lev_percent: SNVT_lev_percent
eSNVT_multiplier: SNVT_multiplier
eSNVT_state: SNVT_state
eSNVT_time_stamp: SNVT_time_stamp
eSNVT_zerospan: SNVT_zerospan
eSNVT_magcard: SNVT_magcard
eSNVT_elapsed_tm: SNVT_elapsed_tm
eSNVT_alarm: SNVT_alarm
eSNVT_currency: SNVT_currency
eSNVT_file_pos: SNVT_file_pos
eSNVT_muldiv: SNVT_muldiv
eSNVT_obj_request: SNVT_obj_request
eSNVT_obj_status: SNVT_obj_status
eSNVT_preset: SNVT_preset
eSNVT_switch: SNVT_switch
eSNVT_trans_table: SNVT_trans_table
eSNVT_override: SNVT_override
eSNVT_pwr_fact: SNVT_pwr_fact
eSNVT_pwr_fact_f: SNVT_pwr_fact_f
eSNVT_density: SNVT_density
eSNVT_density_f: SNVT_density_f

eSNVT_rpm: SNVT_rpm
eSNVT_hvac_emerg: SNVT_hvac_emerg
eSNVT_angle_deg: SNVT_angle_deg
eSNVT_temp_p: SNVT_temp_p
eSNVT_temp_setpt: SNVT_temp_setpt
eSNVT_time_sec: SNVT_time_sec
eSNVT_hvac_mode: SNVT_hvac_mode
eSNVT_occupancy: SNVT_occupancy
eSNVT_area: SNVT_area
eSNVT_hvac_overid: SNVT_hvac_overid
eSNVT_hvac_status: SNVT_hvac_status
eSNVT_press_p: SNVT_press_p
eSNVT_address: SNVT_address
eSNVT_scene: SNVT_scene
eSNVT_scene_cfg: SNVT_scene_cfg
eSNVT_setting: SNVT_setting
eSNVT_evap_state: SNVT_evap_state
eSNVT_therm_mode: SNVT_therm_mode
eSNVT_defr_mode: SNVT_defr_mode
eSNVT_defr_term: SNVT_defr_term
eSNVT_defr_state: SNVT_defr_state
eSNVT_time_min: SNVT_time_min
eSNVT_time_hour: SNVT_time_hour
eSNVT_ph: SNVT_ph
eSNVT_ph_f: SNVT_ph_f
eSNVT_chlr_status: SNVT_chlr_status
eSNVT_tod_event: SNVT_tod_event
eSNVT_smo_obscur: SNVT_smo_obscur
eSNVT_fire_test: SNVT_fire_test
eSNVT_temp_ror: SNVT_temp_ror
eSNVT_fire_init: SNVT_fire_init
eSNVT_fire_indcte: SNVT_fire_indcte
eSNVT_time_zone: SNVT_time_zone
eSNVT_earth_pos: SNVT_earth_pos
eSNVT_reg_val: SNVT_reg_val
eSNVT_reg_val_ts: SNVT_reg_val_ts

eSNVT_volt_ac: SNVT_volt_ac
eSNVT_amp_ac: SNVT_amp_ac
eSNVT_turbidity: SNVT_turbidity
eSNVT_turbidity_f: SNVT_turbidity_f
eSNVT_hvac_type: SNVT_hvac_type
eSNVT_elec_kwh_l: SNVT_elec_kwh_l
eSNVT_temp_diff_p: SNVT_temp_diff_p
eSNVT_ctrl_req: SNVT_ctrl_req
eSNVT_ctrl_resp: SNVT_ctrl_resp
eSNVT_ptz: SNVT_ptz
eSNVT_privacyzone: SNVT_privacyzone
eSNVT_pos_ctrl: SNVT_pos_ctrl
eSNVT_enthalpy: SNVT_enthalpy
eSNVT_gfci_status: SNVT_gfci_status
eSNVT_motor_state: SNVT_motor_state
eSNVT_pumpset_mn: SNVT_pumpset_mn
eSNVT_ex_control: SNVT_ex_control
eSNVT_pumpset_sn: SNVT_pumpset_sn
eSNVT_pump_sensor: SNVT_pump_sensor
eSNVT_abs_humid: SNVT_abs_humid
eSNVT_flow_p: SNVT_flow_p
eSNVT_dev_c_mode: SNVT_dev_c_mode
eSNVT_valve_mode: SNVT_valve_mode
eSNVT_alarm_2: SNVT_alarm_2
eSNVT_state_64: SNVT_state_64
eSNVT_nv_type: SNVT_nv_type
eSNVT_ent_opmode: SNVT_ent_opmode
eSNVT_ent_state: SNVT_ent_state
eSNVT_ent_status: SNVT_ent_status
eSNVT_flow_dir: SNVT_flow_dir
eSNVT_hvac_satsts: SNVT_hvac_satsts
eSNVT_dev_status: SNVT_dev_status
eSNVT_dev_fault: SNVT_dev_fault
eSNVT_dev_maint: SNVT_dev_maint
eSNVT_date_event: SNVT_date_event
eSNVT_sched_val: SNVT_sched_val

eSNVT_sec_state: SNVT_sec_state
 eSNVT_sec_status: SNVT_sec_status
 eSNVT_sblnd_state: SNVT_sblnd_state
 eSNVT_rac_ctrl: SNVT_rac_ctrl
 eSNVT_rac_req: SNVT_rac_req
 eSNVT_count_32: SNVT_count_32
 eSNVT_clothes_w_c: SNVT_clothes_w_c
 eSNVT_clothes_w_m: SNVT_clothes_w_m
 eSNVT_clothes_w_s: SNVT_clothes_w_s
 eSNVT_clothes_w_a: SNVT_clothes_w_a
 eSNVT_multiplier_s: SNVT_multiplier_s
 eSNVT_switch_2: SNVT_switch_2
 eSNVT_color_2: SNVT_color_2
 eSNVT_log_status: SNVT_log_status
 eSNVT_time_stamp_p: SNVT_time_stamp_p
 eSNVT_log_fx_request: SNVT_log_fx_request
 eSNVT_log_fx_status: SNVT_log_fx_status
 eSNVT_log_request: SNVT_log_request
 eSNVT_enthalpy_d: SNVT_enthalpy_d
 eSNVT_amp_ac_mil: SNVT_amp_ac_mil
 eSNVT_time_hour_p: SNVT_time_hour_p
 eSNVT_lamp_status: SNVT_lamp_status
 eSNVT_environment: SNVT_environment
 eSNVT_geo_loc: SNVT_geo_loc

7.3.3 E_LON_alarm_type_t

Used by: SNVT_alarm / SNVT_alarm_2

```

TYPE E_LON_alarm_type_t :
(
  eLON_AL_HEADER           := -13,
  eLON_AL_FOOTER           := -12,
  eLON_AL_DEBUG            := -11,
  eLON_AL_INFO             := -10,
  eLON_AL_SYSTEM_INFO     := -6,
  eLON_AL_VALUE_INVALID   := -5,
  eLON_AL_CONSTANT        := -4,
  eLON_AL_OFFLINE         := -3,
  eLON_AL_UNKNOWN         := -2,
  eLON_AL_NUL              := -1,
  eLON_AL_NO_CONDITION    := 0,
  eLON_AL_ALM_CONDITION   := 1,
  eLON_AL_TOT_SVC_ALM_1   := 2,
  eLON_AL_TOT_SVC_ALM_2   := 3,
  eLON_AL_TOT_SVC_ALM_3   := 4,
  eLON_AL_LOW_LMT_CLR_1   := 5,
  eLON_AL_LOW_LMT_CLR_2   := 6,
  eLON_AL_HIGH_LMT_CLR_1  := 7,
  eLON_AL_HIGH_LMT_CLR_2  := 8,

```



```

eLON_AL_LOW_LMT_ALM_1      := 9,
eLON_AL_LOW_LMT_ALM_2      := 10,
eLON_AL_HIGH_LMT_ALM_1     := 11,
eLON_AL_HIGH_LMT_ALM_2     := 12,
eLON_AL_FIR_ALM            := 13,
eLON_AL_FIR_PRE_ALM        := 14,
eLON_AL_FIR_TRBL           := 15,
eLON_AL_FIR_SUPV           := 16,
eLON_AL_FIR_TEST_ALM       := 17,
eLON_AL_FIR_TEST_PRE_ALM   := 18,
eLON_AL_FIR_ENVCOMP_MAX    := 19,
eLON_AL_FIR_MONITOR_COND   := 20,
eLON_AL_FIR_MAINT_ALERT    := 21,
eLON_AL_FATAL_ERROR        := 30,
eLON_AL_ERROR              := 31,
eLON_AL_WARNING            := 32
)
END_TYPE

```

eLON_AL_HEADER: Update sequence header

eLON_AL_FOOTER: Update sequence footer

eLON_AL_DEBUG: Debug information (not an alarm)

eLON_AL_INFO: Information update (not an alarm)

eLON_AL_SYSTEM_INFO: System information (not an alarm)

eLON_AL_VALUE_INVALID: The value is invalid

eLON_AL_CONSTANT: The value is a constant value (not an alarm)

eLON_AL_OFFLINE: The device is offline

eLON_AL_UNKNOWN: Alarm condition unknown (may be due to a communication failure or hardware failure)

eLON_AL_NUL: Invalid alarm type value (alarm condition not specified)

eLON_AL_NO_CONDITION: No alarm condition present

eLON_AL_ALM_CONDITION: Unspecified alarm condition present

eLON_AL_TOT_SVC_ALM_1: Total/service interval alarm 1 (component requires service or maintenance)

eLON_AL_TOT_SVC_ALM_2: Total/service interval alarm 2

eLON_AL_TOT_SVC_ALM_3: Total/service interval alarm 3

eLON_AL_LOW_LMT_CLR_1: Alarm low limit alarm clear 1

eLON_AL_LOW_LMT_CLR_2: Alarm low limit alarm clear 2

eLON_AL_HIGH_LMT_CLR_1: Alarm high limit alarm clear 1

eLON_AL_HIGH_LMT_CLR_2: Alarm high limit alarm clear 2

eLON_AL_LOW_LMT_ALM_1: Alarm low limit alarm 1

eLON_AL_LOW_LMT_ALM_2: Alarm low limit alarm 2

eLON_AL_HIGH_LMT_ALM_1: Alarm high limit alarm 1

eLON_AL_HIGH_LMT_ALM_2: Alarm high limit alarm 2

eLON_AL_FIR_ALM: Fire alarm condition

eLON_AL_FIR_PRE_ALM: Fire pre-alarm condition

eLON_AL_FIR_TRBL: Fire-related trouble (fault) condition

eLON_AL_FIR_SUPV: Fire-related supervisory condition (e.g., sprinkler pressure)

eLON_AL_FIR_TEST_ALM: Fire-related test-mode alarm condition

eLON_AL_FIR_TEST_PRE_ALM: Fire-related test-mode pre-alarm condition

eLON_AL_FIR_ENVCOMP_MAX: Fire-related maximum environmental compensation level reached

eLON_AL_FIR_MONITOR_COND: Fire-related abnormal input condition

eLON_AL_FIR_MAINT_ALERT: Fire-related maintenance alert

eLON_AL_FATAL_ERROR: Fatal application error

eLON_AL_ERROR: Other error condition

eLON_AL_WARNING: Other warning condition

7.3.4 E_LON_appl_cwc_t

Used by: SNVT_clothes_w_c / SNVT_clothes_w_s

```
TYPE E_LON_appl_cwc_t :
(
  eLON_CWC_NUL      := -1,
  eLON_CWC_WASH     := 0,
  eLON_CWC_RINSE    := 1,
  eLON_CWC_SPIN     := 2,
  eLON_CWC_DRY      := 3
)
END_TYPE
```

eLON_CWC_NUL: Invalid Value

eLON_CWC_WASH: Wash

eLON_CWC_RINSE: Rinse

eLON_CWC_SPIN: Spin

eLON_CWC_DRY: Dry

7.3.5 E_LON_appl_cwp_t

Used by: SNVT_clothes_w_c

```
TYPE E_LON_appl_cwp_t :
(
  eLON_CWP_NUL      := -1,
  eLON_CWP_GENERAL  := 0,
  eLON_CWP_BOIL     := 1,
  eLON_CWP_FAST_WASH := 2,
  eLON_CWP_LINGERIE := 3,
  eLON_CWP_WOOL     := 4,
  eLON_CWP_TOWEL    := 5,
  eLON_CWP_BED_LINENS := 6,
  eLON_CWP_CURTAIN  := 7,
  eLON_CWP_RINSE_SPIN_ONLY := 8,
  eLON_CWP_DELICATE_RINSE := 9,
  eLON_CWP_SPIN_ONLY := 10,
  eLON_CWP_DRY_ONLY := 11
)
END_TYPE
```

eLON_CWP_NUL: Invalid Value

eLON_CWP_GENERAL: Normal Wash

eLON_CWP_BOIL: Boil

eLON_CWP_FAST_WASH: Fast Wash

eLON_CWP_LINGERIE: Lingerie

eLON_CWP_WOOL: Wool

eLON_CWP_TOWEL: Towel
 eLON_CWP_BED_LINENS: Bed Linens
 eLON_CWP_CURTAIN: Curtain
 eLON_CWP_RINSE_SPIN_ONLY: Rinse and Spin Only
 eLON_CWP_DELICATE_RINSE: Delicate Rinse
 eLON_CWP_SPIN_ONLY: Spin Only
 eLON_CWP_DRY_ONLY: Dry Only

7.3.6 E_LON_appl_cws_t

Used by: SNVT_clothes_w_c / SNVT_clothes_w_s

```

TYPE E_LON_appl_cws_t :
(
  eLON_CWS_NUL           := -1,
  eLON_CWS_LOAD_SENSING := 0,
  eLON_CWS_WETTING      := 1,
  eLON_CWS_DETERGENT    := 2,
  eLON_CWS_WASHING      := 3,
  eLON_CWS_WATERING     := 4,
  eLON_CWS_RINSING      := 5,
  eLON_CWS_ARRANGING    := 6,
  eLON_CWS_DRAIN        := 7,
  eLON_CWS_SPINNING     := 8,
  eLON_CWS_FINAL_SPINNING := 9,
  eLON_CWS_FLUFFING     := 10,
  eLON_CWS_DRYING       := 11,
  eLON_CWS_COOLING      := 12
)
END_TYPE

```

eLON_CWS_NUL: Invalid Value
 eLON_CWS_LOAD_SENSING: Sensing Load
 eLON_CWS_WETTING: Wetting
 eLON_CWS_DETERGENT: Detergent
 eLON_CWS_WASHING: Washing
 eLON_CWS_WATERING: Watering
 eLON_CWS_RINSING: Rinsing
 eLON_CWS_ARRANGING: Arranging
 eLON_CWS_DRAIN: Drain
 eLON_CWS_SPINNING: Spinning
 eLON_CWS_FINAL_SPINNING: In Final Spin
 eLON_CWS_FLUFFING: Fluffing
 eLON_CWS_DRYING: Drying
 eLON_CWS_COOLING: Cooling

7.3.7 E_LON_appl_rin_t

Used by: SNVT_clothes_w_c

```

TYPE E_LON_appl_rin_t :
(
  eLON_RIN_NUL           := -1,

```

```
eLON_RIN_PRE_WASH      := 0,
eLON_RIN_WATER_PLUS   := 1,
eLON_RIN_DETERGENT_PLUS := 2,
eLON_RIN_RINSE_HOLD   := 3
)
END_TYPE
```

eLON_RIN_NUL: Invalid Value

eLON_RIN_PRE_WASH: Pre-wash

eLON_RIN_WATER_PLUS: Water Plus

eLON_RIN_DETERGENT_PLUS: Detergent Plus

eLON_RIN_RINSE_HOLD: Rinse Hold

7.3.8 E_LON_boolean_t

Used by: SCPTautoAnswer / SCPTcoolingResetEnable / SCPTdefrostHold / SCPTdefrostInternalSchedule / SCPTheatingResetEnable / SCPThighLimit1Enable / SCPThighLimit2Enable / SCPTlowLimit1Enable / SCPTlowLimit2Enable / SCPTscheduleInternal / SNVT_clothes_w_c / SNVT_pump_sensor / SNVT_pumpset_mn / SNVT_pumpset_sn

```
TYPE E_LON_boolean_t :
(
  eLON_BOOL_NUL      := -1,
  eLON_BOOL_FALSE   := 0,
  eLON_BOOL_TRUE     := 1
)
END_TYPE
```

eLON_BOOL_NUL: Invalid Value

eLON_BOOL_FALSE: False

eLON_BOOL_TRUE: True

7.3.9 E_LON_calendar_type_t

Used by: SNVT_time_zone

```
TYPE E_LON_calendar_type_t :
(
  eLON_CAL_NUL      := -1,
  eLON_CAL_GREG     := 0,
  eLON_CAL_JUL      := 1,
  eLON_CAL_MEU      := 2
)
END_TYPE
```

eLON_CAL_NUL: Invalid Value

eLON_CAL_GREG: Gregorian calendar

eLON_CAL_JUL: Julian calendar

eLON_CAL_MEU: Calendar Method European/US "MEU"

7.3.10 E_LON_cam_act_t

Used by: SNVT_pos_ctrl

```
TYPE E_LON_cam_act_t :
(
  eLON_CMA_NUL      := -1,
  eLON_CMA_SAVE     := 0,
  eLON_CMA_CALL     := 1,
  eLON_CMA_READ     := 2
)
END_TYPE
```

eLON_CMA_NUL: Invalid action call response

eLON_CMA_SAVE: Save the values defined by the function

eLON_CMA_CALL: Preposition tour tables

eLON_CMA_READ: Absolute positions

7.3.11 E_LON_cam_func_t

Used by: SNVT_pos_ctrl

```
TYPE E_LON_cam_func_t :
(
  eLON_CMF_NUL      := -1,
  eLON_CMF_REL      := 0,
  eLON_CMF_TOUR     := 1,
  eLON_CMF_ABS      := 2
)
END_TYPE
```

eLON_CMF_NUL: Invalid function call response

eLON_CMF_REL: Relative positions, prepositions

eLON_CMF_TOUR: Preposition tour tables

eLON_CMF_ABS: Absolute positions

7.3.12 E_LON_chiller_t

Used by: SNVT_chlr_status

```
TYPE E_LON_chiller_t :
(
  eLON_CHLR_NUL     := -1,
  eLON_CHLR_OFF     := 0,
  eLON_CHLR_START   := 1,
  eLON_CHLR_RUN     := 2,
  eLON_CHLR_PRESHUTDN := 3,
  eLON_CHLR_SERVICE := 4
)
END_TYPE
```

eLON_CHLR_NUL: Invalid Value

eLON_CHLR_OFF: Chiller off

eLON_CHLR_START: Chiller in start mode

eLON_CHLR_RUN: Chiller in run mode

eLON_CHLR_PRESHUTDN: Chiller in pre shutdown mode

eLON_CHLR_SERVICE: Chiller in service mode

7.3.13 E_LON_color_encoding_t

Used by: SNVT_color_2

```
TYPE E_LON_color_encoding_t :
(
  eLON_COLOR_NUL           := -1,
  eLON_COLOR_CIE31_LUMEN   := 0,
  eLON_COLOR_CIE31_PERCENT := 1,
  eLON_COLOR_RGB           := 2,
  eLON_COLOR_TEMPERATURE   := 3
)
END_TYPE
```

eLON_COLOR_NUL: Invalid value

eLON_COLOR_CIE31_LUMEN: CIE 1931 color space; Y output in lumen

eLON_COLOR_CIE31_PERCENT: CIE 1931 color space; Y output in percent of maximum lumen output of the lamp

eLON_COLOR_RGB: No color space, RGB color value

eLON_COLOR_TEMPERATURE: Color temperature

7.3.14 E_LON_config_source_t

Used by: SNVT_config_src

```
TYPE E_LON_config_source_t :
(
  eLON_CFG_NUL      := -1,
  eLON_CFG_LOCAL    := 0,
  eLON_CFG_EXTERNAL := 1
)
END_TYPE
```

eLON_CFG_NUL: Invalid Value

eLON_CFG_LOCAL: Device will use self-installation functions to set its own network image

eLON_CFG_EXTERNAL: Device's network image will be set by an outside source

7.3.15 E_LON_control_resp_t

Used by: SNVT_ctrl_resp

```
TYPE E_LON_control_resp_t :
(
  eLON_CTRLR_NUL      := -1,
  eLON_CTRLR_NO       := 0,
  eLON_CTRLR_PEND     := 1,
  eLON_CTRLR_REL      := 2,
  eLON_CTRLR_QUERY    := 3,
  eLON_CTRLR_RES      := 4,
  eLON_CTRLR_ERR      := 5
)
END_TYPE
```

eLON_CTRLR_NUL: Invalid value

eLON_CTRLR_NO: Number of current controller

eLON_CTRLR_PEND: Request pending due to control query to current operator

eLON_CTRLR_REL: Current control released

eLON_CTRLR_QUERY: Query to current controller

eLON_CTRLR_RES: Controllable device has been reset

eLON_CTRLR_ERR: Error in control

7.3.16 E_LON_currency_t

Used by: SNVT_currency

```
TYPE E_LON_currency_t :
(
  eLON_CU_NUL          := -1,
  eLON_CU_ARGENTINA_PESO := 0,
  eLON_CU_AUSTRALIA_DOLLAR := 1,
  eLON_CU_AUSTRIA_SCHILLING := 2,
  eLON_CU_BAHRAIN_DINAR := 3,
  eLON_CU_BELGIUM_FRANC := 4,
  eLON_CU_BRAZIL_CRUZEIRO_REAL := 5,
  eLON_CU_BRITAIN_POUND := 6,
  eLON_CU_CANADA_DOLLAR := 7,

```

```

eLON_CU_CZECH_KORUNA      := 8,
eLON_CU_CHILE_PESO       := 9,
eLON_CU_CHINA_RENMINBI   := 10,
eLON_CU_COLOMBIA_PESO    := 11,
eLON_CU_DENMARK_KRONE    := 12,
eLON_CU_ECUADOR_SUCRE   := 13,
eLON_CU_EUROPEAN_CURRENCY_UNIT := 14,
eLON_CU_FINLAND_MARKKA   := 15,
eLON_CU_FRANCE_FRANC     := 16,
eLON_CU_GERMANY_MARK     := 17,
eLON_CU_GREECE_DRACHMA  := 18,
eLON_CU_HONG_KONG_DOLLAR := 19,
eLON_CU_HUNGARY_FORINT   := 20,
eLON_CU_INDIA_RUPEE     := 21,
eLON_CU_INDONESIA_RUPIAH  := 22,
eLON_CU_IRELAND_PUNT     := 23,
eLON_CU_ISRAEL_SHEKEL   := 24,
eLON_CU_ITALY_LIRA       := 25,
eLON_CU_JAPAN_YEN       := 26,
eLON_CU_JORDAN_DINAR     := 27,
eLON_CU_KUWAIT_DINAR    := 28,
eLON_CU_LEBANON_POUND   := 29,
eLON_CU_MALAYSIA_RINGGIT := 30,
eLON_CU_MALTA_LIRA       := 31,
eLON_CU_MEXICO_PESO      := 32,
eLON_CU_NETHERLANDS_GUILDER := 33,
eLON_CU_NEW_ZEALAND_DOLLAR := 34,
eLON_CU_NORWAY_KRONE     := 35,
eLON_CU_PAKISTAN_RUPEE  := 36,
eLON_CU_PERU_NEW_SOL     := 37,
eLON_CU_PHILIPPINES_PESO := 38,
eLON_CU_POLAND_ZLOTY    := 39,
eLON_CU_PORTUGAL_ESCUDO  := 40,
eLON_CU_SAUDI_ARABIA_RIYAL := 41,
eLON_CU_SINGAPORE_DOLLAR := 42,
eLON_CU_SLOVAK_KORUNA   := 43,
eLON_CU_SOUTH_AFRICA_RAND := 44,
eLON_CU_SOUTH_KOREA_WON := 45,
eLON_CU_SPAIN_PESETA    := 46,
eLON_CU_SPECIAL_DRAWING_RIGHTS := 47,
eLON_CU_SWEDEN_KRONA    := 48,
eLON_CU_SWITZERLAND_FRANC := 49,
eLON_CU_TAIWAN_DOLLAR   := 50,
eLON_CU_THAILAND_BAHT   := 51,
eLON_CU_TURKEY_LIRA     := 52,
eLON_CU_UNITED_ARAB_DIRHAM := 53,
eLON_CU_UNITED_STATES_DOLLAR := 54,
eLON_CU_URUGUAY_NEW_PESO := 55,
eLON_CU_VENEZUELA_BOLIVAR := 56
)
END_TYPE

```

eLON_CU_NUL: Invalid Value

eLON_CU_ARGENTINA_PESO: Argentine Peso

eLON_CU_AUSTRALIA_DOLLAR: Australian Dollar

eLON_CU_AUSTRIA_SCHILLING: Austrian Schilling

eLON_CU_BAHRAIN_DINAR: Bahraini Dinar

eLON_CU_BELGIUM_FRANC: Belgian Franc

eLON_CU_BRAZIL_CRUZEIRO_REAL: Brazilian Cruzeiro Real

eLON_CU_BRITAIN_POUND: British Pound

eLON_CU_CANADA_DOLLAR: Canadian Dollar

eLON_CU_CZECH_KORUNA: Czechoslovakian Koruna

eLON_CU_CHILE_PESO: Chilean Peso

eLON_CU_CHINA_RENMINBI: Chinese Renminbi Yuan

eLON_CU_COLOMBIA_PESO: Colombian Peso

eLON_CU_DENMARK_KRONE: Danish Krone
eLON_CU_ECUADOR_SUCRE: Ecuadorian Sucre
eLON_CU_EUROPEAN_CURRENCY_UNIT: European Euro
eLON_CU_FINLAND_MARKKA: Finnish Markka
eLON_CU_FRANCE_FRANC: French Franc
eLON_CU_GERMANY_MARK: German Mark
eLON_CU_GREECE_DRACHMA: Greek Drachma
eLON_CU_HONG_KONG_DOLLAR: Hong Kong Dollar
eLON_CU_HUNGARY_FORINT: Hungarian Forint
eLON_CU_INDIA_RUPEE: Indian Rupee
eLON_CU_INDONESIA_RUPIAH: Indonesian Rupiah
eLON_CU_IRELAND_PUNT: Irish Punt
eLON_CU_ISRAEL_SHEKEL: Israeli Shekel
eLON_CU_ITALY_LIRA: Italian Lira
eLON_CU_JAPAN_YEN: Japanese Yen
eLON_CU_JORDAN_DINAR: Jordanian Dinar
eLON_CU_KUWAIT_DINAR: Kuwaiti Dinar
eLON_CU_LEBANON_POUND: Lebanese Pound
eLON_CU_MALAYSIA_RINGGIT: Malaysian Ringgit
eLON_CU_MALTA_LIRA: Maltese Lira
eLON_CU_MEXICO_PESO: Mexican New Peso
eLON_CU_NETHERLANDS_GUILDER: Netherlands Guilder
eLON_CU_NEW_ZEALAND_DOLLAR: New Zealand Dollar
eLON_CU_NORWAY_KRONE: Norwegian Krone
eLON_CU_PAKISTAN_RUPEE: Pakistani Rupee
eLON_CU_PERU_NEW_SOL: Peruvian New Sol
eLON_CU_PHILIPPINES_PESO: Philippine Peso
eLON_CU_POLAND_ZLOTY: Polish Zloty
eLON_CU_PORTUGAL_ESCUDO: Portuguese Escudo
eLON_CU_SAUDI_ARABIA_RIYAL: Saudi Arabian Riyal
eLON_CU_SINGAPORE_DOLLAR: Singaporean Dollar
eLON_CU_SLOVAK_KORUNA: Slavic Koruna
eLON_CU_SOUTH_AFRICA_RAND: South African Rand
eLON_CU_SOUTH_KOREA_WON: South Korean Won
eLON_CU_SPAIN_PESETA: Spanish Peseta
eLON_CU_SPECIAL_DRAWING_RIGHTS: international governmental exchange

eLON_CU_SWEDEN_KRONA: Swedish Krona

eLON_CU_SWITZERLAND_FRANC: Swiss Franc

eLON_CU_TAIWAN_DOLLAR: Taiwanese Dollar

eLON_CU_THAILAND_BAHT: Thai Baht

eLON_CU_TURKEY_LIRA: Turkish Lira

eLON_CU_UNITED_ARAB_DIRHAM: United Arab Emirates Dirham

eLON_CU_UNITED_STATES_DOLLAR: United States Dollar

eLON_CU_URUGUAY_NEW_PESO: Uruguayan New Peso

eLON_CU_VENEZUELA_BOLIVAR: Venezuelan Bolivar

7.3.17 E_LON_days_of_week_t

Used by: SCPTtimePeriod / SNVT_date_day / SNVT_time_zone

```
TYPE E_LON_days_of_week_t :
(
  eLON_DAY_NUL   := -1,
  eLON_DAY_SUN   := 0,
  eLON_DAY_MON   := 1,
  eLON_DAY_TUE   := 2,
  eLON_DAY_WED   := 3,
  eLON_DAY_THU   := 4,
  eLON_DAY_FRI   := 5,
  eLON_DAY_SAT   := 6
)
END_TYPE
```

eLON_DAY_NUL: Invalid Value

eLON_DAY_SUN: Sunday

eLON_DAY_MON: Monday

eLON_DAY_TUE: Tuesday

eLON_DAY_WED: Wednesday

eLON_DAY_THU: Thursday

eLON_DAY_FRI: Friday

eLON_DAY_SAT: Saturday

7.3.18 E_LON_defrost_mode_t

Used by: SNVT_defr_mode

```
TYPE E_LON_defrost_mode_t :
(
  eLON_DFM_NUL           := -1,
  eLON_DFM_MODE_AMBIENT := 0,
  eLON_DFM_MODE_FORCED  := 1,
  eLON_DFM_MODE_SYNC    := 2
)
END_TYPE
```

eLON_DFM_NUL: Invalid Value

eLON_DFM_MODE_AMBIENT: No forced heating required

eLON_DFM_MODE_FORCED: Start-up after defrost ignored

eLON_DFM_MODE_SYNC: Synchronized

7.3.19 E_LON_defrost_state_t

Used by: SNVT_defr_state

```

TYPE E_LON_defrost_state_t :
(
  eLON_DFS_NUL      := -1,
  eLON_DFS_STANDBY  := 0,
  eLON_DFS_PUMPDOWN := 1,
  eLON_DFS_DEFROST  := 2,
  eLON_DFS_DRAINDOWN := 3,
  eLON_DFS_INJECT_DLY := 4
)
END_TYPE

```

eLON_DFS_NUL: Invalid Value

eLON_DFS_STANDBY: Defrost in standby

eLON_DFS_PUMPDOWN: Defrost in pump-down mode

eLON_DFS_DEFROST: In defrost mode

eLON_DFS_DRAINDOWN: Defrost in drain-down

eLON_DFS_INJECT_DLY: Defrost in injection delay

7.3.20 E_LON_defrost_term_t

Used by: SNVT_defr_term

```

TYPE E_LON_defrost_term_t :
(
  eLON_DFT_NUL      := -1,
  eLON_DFT_TERM_TEMP := 0,
  eLON_DFT_TERM_TIME := 1,
  eLON_DFT_TERM_FIRST := 2,
  eLON_DFT_TERM_LAST := 3,
  eLON_DFT_TERM_SENSOR := 4,
  eLON_DFT_TERM_DISCHARGE := 5,
  eLON_DFT_TERM_RETURN := 6,
  eLON_DFT_TERM_SW_OPEN := 7,
  eLON_DFT_TERM_SW_CLOSE := 8,
  eLON_DFT_TERM_MANUF := 100
)
END_TYPE

```

eLON_DFT_NUL: Invalid Value

eLON_DFT_TERM_TEMP: Terminate on temperature

eLON_DFT_TERM_TIME: Terminate on time

eLON_DFT_TERM_FIRST: Terminate on first occurring

eLON_DFT_TERM_LAST: Terminate on last occurring

eLON_DFT_TERM_SENSOR: Terminate on sensor

eLON_DFT_TERM_DISCHARGE: Terminate on discharge

eLON_DFT_TERM_RETURN: Terminate on return

eLON_DFT_TERM_SW_OPEN: Terminate on "Switch Open"

eLON_DFT_TERM_SW_CLOSE: Terminate on "Switch Closed"

eLON_DFT_TERM_MANUF: Manufacturer-Defined termination state

7.3.21 E_LON_device_c_mode_t

Used by: SNVT_dev_c_mode

```

TYPE E_LON_device_c_mode_t :
(
  eLON_DCM_NUL           := -1,
  eLON_DCM_SPEED_CONST  := 0,
  eLON_DCM_PRESS_CONST  := 1,
  eLON_DCM_PRESS_COMP   := 2,
  eLON_DCM_FLOW_CONST   := 3,
  eLON_DCM_FLOW_COMP    := 4,
  eLON_DCM_TEMP_CONST   := 5,
  eLON_DCM_TEMP_COMP    := 6,
  eLON_DCM_PRESS_AUTO   := 7,
  eLON_DCM_QUICK_OPEN   := 20,
  eLON_DCM_LINEAR       := 21,
  eLON_DCM_EQUAL_PERCENT := 22,
  eLON_DCM_QUADRATIC    := 23,
  eLON_DCM_FREE_DEFINED := 24,
  eLON_DCM_2WAY_VALVE   := 27,
  eLON_DCM_MIXING_VALVE := 28,
  eLON_DCM_DIVERTING_VALVE := 29,
  eLON_DCM_INVFNC_QCK_OPN := 30,
  eLON_DCM_INVFNC_EQL_PERC := 31,
  eLON_DCM_INVFNC_QUAD  := 32
)
END_TYPE

```

eLON_DCM_NUL: Invalid Value

eLON_DCM_SPEED_CONST:

eLON_DCM_PRESS_CONST:

eLON_DCM_PRESS_COMP:

eLON_DCM_FLOW_CONST:

eLON_DCM_FLOW_COMP:

eLON_DCM_TEMP_CONST:

eLON_DCM_TEMP_COMP:

eLON_DCM_PRESS_AUTO:

eLON_DCM_QUICK_OPEN: Valve works with Quick-Open flow characteristic

eLON_DCM_LINEAR: Valve works with Linear flow characteristic

eLON_DCM_EQUAL_PERCENT: Valve works with Equal Percent flow characteristic

eLON_DCM_QUADRATIC: Valve works with Quadratic flow characteristic

eLON_DCM_FREE_DEFINED: Valve works with free defined flow characteristic

eLON_DCM_2WAY_VALVE:

eLON_DCM_MIXING_VALVE:

eLON_DCM_DIVERTING_VALVE:

eLON_DCM_INVFNC_QCK_OPN:

eLON_DCM_INVFNC_EQL_PERC:

eLON_DCM_INVFNC_QUAD:

7.3.22 E_LON_device_select_t

Used by: SNVT_dev_fault / SNVT_dev_maint / SNVT_dev_status

```

TYPE E_LON_device_select_t :
(
  eLON_DV_NUL           := -1,
  eLON_DV_PUMP_CTRL    := 0,

```

```
eLON_DV_VALVE_POS := 1
)
END_TYPE
```

eLON_DV_NUL: Invalid value

eLON_DV_PUMP_CTRL: Use union for SFPTpumpController values

eLON_DV_VALVE_POS: Use union for SFPTvalvePositioner values

7.3.23 E_LON_discrete_levels_t

Used by: SNVT_clothes_w_c / SNVT_lev_disc

```
TYPE E_LON_discrete_levels_t :
(
  eLON_ST_NUL    := -1,
  eLON_ST_OFF    := 0,
  eLON_ST_LOW    := 1,
  eLON_ST_MED    := 2,
  eLON_ST_HIGH   := 3,
  eLON_ST_ON     := 4
)
END_TYPE
```

eLON_ST_NUL:

eLON_ST_OFF:

eLON_ST_LOW:

eLON_ST_MED:

eLON_ST_HIGH:

eLON_ST_ON:

7.3.24 E_LON_emerg_t

Used by: SNVT_hvac_emerg

```
TYPE E_LON_emerg_t :
(
  eLON_EMERG_NUL      := -1,
  eLON_EMERG_NORMAL   := 0,
  eLON_EMERG_PRESSURIZE := 1,
  eLON_EMERG_DEPRESSURIZE := 2,
  eLON_EMERG_PURGE    := 3,
  eLON_EMERG_SHUTDOWN := 4,
  eLON_EMERG_FIRE     := 5
)
END_TYPE
```

eLON_EMERG_NUL: Invalid Value

eLON_EMERG_NORMAL: No emergency mode

eLON_EMERG_PRESSURIZE: Emergency pressurize mode

eLON_EMERG_DEPRESSURIZE: Emergency depressurize mode

eLON_EMERG_PURGE: Emergency purge mode

eLON_EMERG_SHUTDOWN: Emergency shutdown mode

eLON_EMERG_FIRE: Emergency fire mode

7.3.25 E_LON_ent_cmd_t

Used by: SNVT_ent_state

```

TYPE E_LON_ent_cmd_t :
(
  eLON_ES_NUL           := -1,
  eLON_ES_UNDEFINED    := 0,
  eLON_ES_OPEN_PULS    := 1,
  eLON_ES_OPEN         := 2,
  eLON_ES_CLOSE        := 3,
  eLON_ES_STOP         := 4,
  eLON_ES_STOP_RESUME  := 5,
  eLON_ES_ENTRY_REQ    := 6,
  eLON_ES_EXIT_REQ     := 7,
  eLON_ES_KEY_REQ      := 8,
  eLON_ES_SAFETY_EXT_REQ := 9,
  eLON_ES_EMERGENCY_REQ := 10,
  eLON_ES_UPDATE_STATE := 11,
  eLON_ES_SAF_EXT_RESUME := 12,
  eLON_ES_EMERG_RESUME := 13
)
END_TYPE

```

eLON_ES_NUL: Invalid Value

eLON_ES_UNDEFINED: State is not yet defined

eLON_ES_OPEN_PULS: Open the device and close it when back in normal position

eLON_ES_OPEN: Open the device if not locked

eLON_ES_CLOSE: Close the device

eLON_ES_STOP: Stop the device

eLON_ES_STOP_RESUME: Continue after stop command

eLON_ES_ENTRY_REQ: Entry request, access in to the area

eLON_ES_EXIT_REQ: Exit request, access out from the area

eLON_ES_KEY_REQ: Exit request, access out from the area

eLON_ES_SAFETY_EXT_REQ: Safety request, the device will go to a pre defined safety position/mode

eLON_ES_EMERGENCY_REQ: Emergency request, the device will go to an pre defined emergency position/mode

eLON_ES_UPDATE_STATE: Update the current state and mode

eLON_ES_SAF_EXT_RESUME: Resume after Safety function

eLON_ES_EMERG_RESUME: Resume after Emergency function

7.3.26 E_LON_ent_opmode_cmd_t

Used by: SNVT_ent_opmode / SNVT_ent_status

```

TYPE E_LON_ent_opmode_cmd_t :
(
  eLON_EM_NUL           := -1,
  eLON_EM_UNDEFINED    := 0,
  eLON_EM_AUTO         := 1,
  eLON_EM_AUTO_RED     := 2,
  eLON_EM_CLOSE_LOCK   := 3,
  eLON_EM_CLOSE_UNLOCK := 4,
  eLON_EM_EXIT_ONLY    := 5,
  eLON_EM_OPEN         := 6,
  eLON_EM_OPEN_ONCE    := 7,
  eLON_EM_MANUAL       := 8,
  eLON_EM_FIRE         := 9,
  eLON_EM_EVAC         := 10,
  eLON_EM_WEATHER      := 11,
  eLON_EM_DAY_LOCKING  := 12,
  eLON_EM_NIGHT_LOCKING := 13,
  eLON_EM_BLOCKED     := 14,
  eLON_EM_SERVICE      := 15,
)

```

```
eLON_EM_ENTRY_ONLY      := 16
)
END_TYPE
```

eLON_EM_NUL: Invalid Value

eLON_EM_UNDEFINED: Operation mode is not defined

eLON_EM_AUTO: Operation mode is AUTOMATIC

eLON_EM_AUTO_RED: Operation mode is AUTOMATIC with reduced width

eLON_EM_CLOSE_LOCK: Operation mode is CLOSE AND LOCK

eLON_EM_CLOSE_UNLOCK: Operation mode is CLOSE AND UNLOCK

eLON_EM_EXIT_ONLY: Operation mode is EXIT ONLY

eLON_EM_OPEN: Operation mode is OPEN

eLON_EM_OPEN_ONCE: Operation mode is OPEN AND CLOSE ONCE

eLON_EM_MANUAL: Operation mode is MANUAL

eLON_EM_FIRE: Operation mode is FIRE

eLON_EM_EVAC: Operation mode is EVACUATION

eLON_EM_WEATHER: Operation mode is WEATHER MODE

eLON_EM_DAY_LOCKING: Operation mode is DAY_LOCKING, locking with reduced level of security

eLON_EM_NIGHT_LOCKING: Operation mode is NIGHT_LOCKING, locking with maximum level of security

eLON_EM_BLOCKED: Operation mode is BLOCKED, no operations is allowed

eLON_EM_SERVICE: Operation mode is SERVICE

eLON_EM_ENTRY_ONLY: Operation mode is ENTRY_ONLY

7.3.27 E_LON_evap_t

Used by: SNVT_evap_state

```
TYPE E_LON_evap_t :
(
  eLON_EVAP_NUL           := -1,
  eLON_EVAP_NO_COOLING   := 0,
  eLON_EVAP_COOLING      := 1,
  eLON_EVAP_EMERG_COOLING := 2
)
END_TYPE
```

eLON_EVAP_NUL: Invalid Value

eLON_EVAP_NO_COOLING: Object not performing cooling (off cycle or disabled)

eLON_EVAP_COOLING: Object currently cooling

eLON_EVAP_EMERG_COOLING: Object performing emergency cooling

7.3.28 E_LON_ex_control_t

Used by: SNVT_ex_control

```
TYPE E_LON_ex_control_t :
(
  eLON_EX_CONTROL_NUL      := -1,
  eLON_EX_CONTROL_NONE     := 0,
  eLON_EX_CONTROL_OTHER    := 1,

```

```
eLON_EX_CONTROL_THIS_ADDR := 2
)
END_TYPE
```

eLON_EX_CONTROL_NUL: The control status of the item is unknown

eLON_EX_CONTROL_NONE: Nothing has control of the item.

eLON_EX_CONTROL_OTHER: Some unidentified entity has control of the item.

eLON_EX_CONTROL_THIS_ADDR: A device has control of the item. The network address of this device is specified in the control_device_addr

7.3.29 E_LON_file_request_t

Used by: SNVT_file_req

```
TYPE E_LON_file_request_t :
(
  eLON_FR_NUL           := -1,
  eLON_FR_OPEN_TO_SEND := 0,
  eLON_FR_OPEN_TO_RECEIVE := 1,
  eLON_FR_CLOSE_FILE   := 2,
  eLON_FR_CLOSE_DELETE_FILE := 3,
  eLON_FR_DIRECTORY_LOOKUP := 4,
  eLON_FR_OPEN_TO_SEND_RA := 5,
  eLON_FR_OPEN_TO_RECEIVE_RA := 6
)
END_TYPE
```

eLON_FR_NUL: Invalid Value

eLON_FR_OPEN_TO_SEND: Sequential access read

eLON_FR_OPEN_TO_RECEIVE: Sequential access write

eLON_FR_CLOSE_FILE: Close and save file

eLON_FR_CLOSE_DELETE_FILE: Close and delete file

eLON_FR_DIRECTORY_LOOKUP: Retrieve directory entry

eLON_FR_OPEN_TO_SEND_RA: Random access read

eLON_FR_OPEN_TO_RECEIVE_RA: Random access write

7.3.30 E_LON_file_status_t

Used by: SNVT_file_status

```
TYPE E_LON_file_status_t :
(
  eLON_FS_NUL           := -1,
  eLON_FS_XFER_OK      := 0,
  eLON_FS_LOOKUP_OK    := 1,
  eLON_FS_OPEN_FAIL    := 2,
  eLON_FS_LOOKUP_ERR   := 3,
  eLON_FS_XFER_UNDERWAY := 4,
  eLON_FS_IO_ERR       := 5,
  eLON_FS_TIMEOUT_ERR  := 6,
  eLON_FS_WINDOW_ERR   := 7,
  eLON_FS_AUTH_ERR     := 8,
  eLON_FS_ACCESS_UNAVAIL := 9,
  eLON_FS_SEEK_INVALID := 10,
  eLON_FS_SEEK_WAIT    := 11
)
END_TYPE
```

eLON_FS_NUL: Invalid Value

eLON_FS_XFER_OK: File transfer successful

eLON_FS_LOOKUP_OK: Directory lookup successful

eLON_FS_OPEN_FAIL: Error on opening file
 eLON_FS_LOOKUP_ERR: Error on directory lookup
 eLON_FS_XFER_UNDERWAY: File transfer in progress
 eLON_FS_IO_ERR: Error on reading/writing file
 eLON_FS_TIMEOUT_ERR: File transfer timed out
 eLON_FS_WINDOW_ERR: Window sequence error
 eLON_FS_AUTH_ERR: Authentication failure
 eLON_FS_ACCESS_UNAVAIL: Access mode not supported
 eLON_FS_SEEK_INVALID: Random access beyond EOF
 eLON_FS_SEEK_WAIT:

7.3.31 E_LON_fire_indicator_t

Used by: SNVT_fire_indcte

```

TYPE E_LON_fire_indicator_t :
(
  eLON_FN_NUL      := -1,
  eLON_FN_UNDEFINED := 0,
  eLON_FN_STROBE_U := 1,
  eLON_FN_STROBE_S := 2,
  eLON_FN_HORN     := 3,
  eLON_FN_CHIME    := 4,
  eLON_FN_BELL     := 5,
  eLON_FN_SOUNDER  := 6,
  eLON_FN_SPEAKER  := 7,
  eLON_FN_UNIVERSAL := 8
)
END_TYPE

```

eLON_FN_NUL: Invalid Value
 eLON_FN_UNDEFINED: Undefined indicator
 eLON_FN_STROBE_U: The indicator is un-synchronized
 eLON_FN_STROBE_S: The indicator is synchronized
 eLON_FN_HORN: The indicator is a DC input, pre coded Horn
 eLON_FN_CHIME: The indicator is a DC input, pre coded Chime
 eLON_FN_BELL: The indicator is a DC input
 eLON_FN_SOUNDER: The indicator is powered from the device
 eLON_FN_SPEAKER: The indicator is an AC input for the speaker
 eLON_FN_UNIVERSAL: General purpose indicator

7.3.32 E_LON_fire_initiator_t

Used by: SNVT_fire_init

```

TYPE E_LON_fire_initiator_t :
(
  eLON_FI_NUL          := -1,
  eLON_FI_UNDEFINED    := 0,
  eLON_FI_THERMAL_FIXED := 1,
  eLON_FI_SMOKE_ION    := 2,
  eLON_FI_MULTI_ION_THERMAL := 3,
  eLON_FI_SMOKE_PHOTO  := 4,
  eLON_FI_MULTI_PHOTO_THERMAL := 5,
)

```



```

eLON_FI_MULTI_PHOTO_ION      := 6,
eLON_FI_MULTI_PHOTO_ION_THERMAL := 7,
eLON_FI_THERMAL_ROR          := 8,
eLON_FI_MULTI_THERMAL_ROR    := 9,
eLON_FI_MANUAL_PULL         := 10,
eLON_FI_WATER_FLOW          := 11,
eLON_FI_WATER_FLOW_TAMPER    := 12,
eLON_FI_STATUS_ONLY         := 13,
eLON_FI_MANUAL_CALL         := 14,
eLON_FI_FIREMAN_CALL        := 15,
eLON_FI_UNIVERSAL           := 16
)
END_TYPE

```

eLON_FI_NUL: Invalid Value

eLON_FI_UNDEFINED: Initiator is undefined

eLON_FI_THERMAL_FIXED: Initiator is thermal fixed (heat)

eLON_FI_SMOKE_ION: Initiator is smoke and ion

eLON_FI_MULTI_ION_THERMAL: Initiator is multi-ion and thermal

eLON_FI_SMOKE_PHOTO: Initiator is smoke and photo

eLON_FI_MULTI_PHOTO_THERMAL: Initiator is multi-photo and thermal

eLON_FI_MULTI_PHOTO_ION: Initiator is multi-photo and ion

eLON_FI_MULTI_PHOTO_ION_THERMAL: Initiator is multi-photo, ion and thermal

eLON_FI_THERMAL_ROR: Initiator is thermal fixed and Rate of Rise

eLON_FI_MULTI_THERMAL_ROR: Initiator is multi-thermal and Rate of Rise

eLON_FI_MANUAL_PULL: Initiator is manual pull

eLON_FI_WATER_FLOW: Initiator is water flow

eLON_FI_WATER_FLOW_TAMPER: Initiator is water flow and tamper

eLON_FI_STATUS_ONLY: Initiator is status only

eLON_FI_MANUAL_CALL: Initiator is a manual call point

eLON_FI_FIREMAN_CALL: Initiator is a fireman call point

eLON_FI_UNIVERSAL: General purpose initiator definition

7.3.33 E_LON_fire_test_t

Used by: SNVT_fire_test

```

TYPE E_LON_fire_test_t :
(
  eLON_FT_NUL      := -1,
  eLON_FT_NORMAL   := 0,
  eLON_FT_RESET    := 1,
  eLON_FT_TEST     := 2,
  eLON_FT_NOTEST   := 3
)
END_TYPE

```

eLON_FT_NUL: Invalid Value

eLON_FT_NORMAL: Return object to normal status

eLON_FT_RESET: Perform a RESET function (for smoke detectors)

eLON_FT_TEST: Go into TEST mode

eLON_FT_NOTEST: Exit TEST mode

7.3.34 E_LON_flow_direction_t

Used by: SNVT_flow_dir

```
TYPE E_LON_flow_direction_t :
(
  eLON_FD_NUL      := -1,
  eLON_FD_NONE     := 0,
  eLON_FD_OUT      := 1,
  eLON_FD_IN       := 2,
  eLON_FD_ANY      := 3
)
END_TYPE
```

eLON_FD_NUL: Invalid Value

eLON_FD_NONE: No flow/movement allowed

eLON_FD_OUT: Exit/out/away direction only

eLON_FD_IN: Entry/in/toward direction only

eLON_FD_ANY: No restriction on flow/movement

7.3.35 E_LON_gfci_status_t

Used by: SNVT_gfci_status

```
TYPE E_LON_gfci_status_t :
(
  eLON_GFCI_NUL      := -1,
  eLON_GFCI_UNKNOWN  := 0,
  eLON_GFCI_NORMAL   := 1,
  eLON_GFCI_TRIPPED  := 2,
  eLON_GFCI_TEST_FAILED := 3,
  eLON_GFCI_TEST_PASSED := 4,
  eLON_GFCI_TEST_NOW   := 5
)
END_TYPE
```

eLON_GFCI_NUL: Invalid Value

eLON_GFCI_UNKNOWN: Unknown response

eLON_GFCI_NORMAL: Normal GFCI operating condition

eLON_GFCI_TRIPPED: A ground-fault has caused the GFCI to interrupt the circuit

eLON_GFCI_TEST_FAILED: The GFCI failed testing

eLON_GFCI_TEST_PASSED: The GFCI passed testing

eLON_GFCI_TEST_NOW: The GFCI needs to be tested

7.3.36 E_LON_hvac_hvt_t

Used by: SNVT_hvac_type

```
TYPE E_LON_hvac_hvt_t :
(
  eLON_HVT_NUL      := -1,
  eLON_HVT_GENERIC  := 0,
  eLON_HVT_FAN_COIL := 1,
  eLON_HVT_VAV      := 2,
  eLON_HVT_HEAT_PUMP := 3,
  eLON_HVT_ROOFTOP  := 4,
  eLON_HVT_UNIT_VENT := 5,
  eLON_HVT_CHILL_CEIL := 6,
  eLON_HVT_RADIATOR := 7,
  eLON_HVT_AHU      := 8,
  eLON_HVT_SELF_CONT := 9
)
END_TYPE
```

eLON_HVT_NUL: Invalid Value
 eLON_HVT_GENERIC: Generic
 eLON_HVT_FAN_COIL: Fan Coil
 eLON_HVT_VAV: Variable Air Volume Terminal
 eLON_HVT_HEAT_PUMP: Heat Pump
 eLON_HVT_ROOFTOP: Rooftop Unit
 eLON_HVT_UNIT_VENT: Unit Ventilator
 eLON_HVT_CHILL_CEIL: Chilled Ceiling
 eLON_HVT_RADIATOR: Radiator
 eLON_HVT_AHU: Air Handling Unit
 eLON_HVT_SELF_CONT: Self-Contained Unit

7.3.37 E_LON_hvac_overid_t

Used by: SNVT_hvac_overid

```

TYPE E_LON_hvac_overid_t :
(
  eLON_HVO_NUL           := -1,
  eLON_HVO_OFF           := 0,
  eLON_HVO_POSITION     := 1,
  eLON_HVO_FLOW_VALUE   := 2,
  eLON_HVO_FLOW_PERCENT := 3,
  eLON_HVO_OPEN         := 4,
  eLON_HVO_CLOSE       := 5,
  eLON_HVO_MINIMUM     := 6,
  eLON_HVO_MAXIMUM     := 7,
  eLON_HVO_UNUSED8     := 8,
  eLON_HVO_UNUSED9     := 9,
  eLON_HVO_UNUSED10    := 10,
  eLON_HVO_UNUSED11    := 11,
  eLON_HVO_UNUSED12    := 12,
  eLON_HVO_UNUSED13    := 13,
  eLON_HVO_UNUSED14    := 14,
  eLON_HVO_UNUSED15    := 15,
  eLON_HVO_UNUSED16    := 16,
  eLON_HVO_POSITION_1  := 17,
  eLON_HVO_FLOW_VALUE_1 := 18,
  eLON_HVO_FLOW_PERCENT_1 := 19,
  eLON_HVO_OPEN_1     := 20,
  eLON_HVO_CLOSE_1    := 21,
  eLON_HVO_MINIMUM_1  := 22,
  eLON_HVO_MAXIMUM_1  := 23,
  eLON_HVO_UNUSED24   := 24,
  eLON_HVO_UNUSED25   := 25,
  eLON_HVO_UNUSED26   := 26,
  eLON_HVO_UNUSED27   := 27,
  eLON_HVO_UNUSED28   := 28,
  eLON_HVO_UNUSED29   := 29,
  eLON_HVO_UNUSED30   := 30,
  eLON_HVO_UNUSED31   := 31,
  eLON_HVO_UNUSED32   := 32,
  eLON_HVO_POSITION_2 := 33,
  eLON_HVO_FLOW_VALUE_2 := 34,
  eLON_HVO_FLOW_PERCENT_2 := 35,
  eLON_HVO_OPEN_2     := 36,
  eLON_HVO_CLOSE_2    := 37,
  eLON_HVO_MINIMUM_2  := 38,
  eLON_HVO_MAXIMUM_2  := 39,
  eLON_HVO_UNUSED40   := 40,
  eLON_HVO_UNUSED41   := 41,
  eLON_HVO_UNUSED42   := 42,
  eLON_HVO_UNUSED43   := 43,
  eLON_HVO_UNUSED44   := 44,
  eLON_HVO_UNUSED45   := 45,
  eLON_HVO_UNUSED46   := 46,

```

```
eLON_HVO_UNUSED47 := 47,  
eLON_HVO_UNUSED48 := 48  
)  
END_TYPE
```

eLON_HVO_NUL: Invalid Value

eLON_HVO_OFF: Not overridden

eLON_HVO_POSITION:

eLON_HVO_FLOW_VALUE: Override flow in liters/sec - use flow field

eLON_HVO_FLOW_PERCENT: Override flow percentage - use percent field

eLON_HVO_OPEN: Override to position = 100%

eLON_HVO_CLOSE: Override to position = 0%

eLON_HVO_MINIMUM: Override to configured minimum

eLON_HVO_MAXIMUM: Override to configured maximum

eLON_HVO_UNUSED8:

eLON_HVO_UNUSED9:

eLON_HVO_UNUSED10:

eLON_HVO_UNUSED11:

eLON_HVO_UNUSED12:

eLON_HVO_UNUSED13:

eLON_HVO_UNUSED14:

eLON_HVO_UNUSED15:

eLON_HVO_UNUSED16:

eLON_HVO_POSITION_1:

eLON_HVO_FLOW_VALUE_1: Override flow in liters/sec - use flow field

eLON_HVO_FLOW_PERCENT_1: Override flow percentage - use percent field

eLON_HVO_OPEN_1: Override to position = 100%

eLON_HVO_CLOSE_1: Override to position = 0%

eLON_HVO_MINIMUM_1: Override to configured minimum

eLON_HVO_MAXIMUM_1: Override to configured maximum

eLON_HVO_UNUSED24:

eLON_HVO_UNUSED25:

eLON_HVO_UNUSED26:

eLON_HVO_UNUSED27:

eLON_HVO_UNUSED28:

eLON_HVO_UNUSED29:

eLON_HVO_UNUSED30:

eLON_HVO_UNUSED31:

eLON_HVO_UNUSED32:

eLON_HVO_POSITION_2:
 eLON_HVO_FLOW_VALUE_2: Override flow in liters/sec - use flow field
 eLON_HVO_FLOW_PERCENT_2: Override flow percentage - use percent field
 eLON_HVO_OPEN_2: Override to position = 100%
 eLON_HVO_CLOSE_2: Override to position = 0%
 eLON_HVO_MINIMUM_2: Override to configured minimum
 eLON_HVO_MAXIMUM_2: Override to configured maximum
 eLON_HVO_UNUSED40:
 eLON_HVO_UNUSED41:
 eLON_HVO_UNUSED42:
 eLON_HVO_UNUSED43:
 eLON_HVO_UNUSED44:
 eLON_HVO_UNUSED45:
 eLON_HVO_UNUSED46:
 eLON_HVO_UNUSED47:
 eLON_HVO_UNUSED48:

7.3.38 E_LON_hvac_t

Used by: SNVT_chlr_status / SNVT_hvac_mode / SNVT_hvac_status

```

TYPE E_LON_hvac_t :
(
  eLON_HVAC_NUL      := -1,
  eLON_HVAC_AUTO     := 0,
  eLON_HVAC_HEAT     := 1,
  eLON_HVAC_MRNG_WRMUP := 2,
  eLON_HVAC_COOL     := 3,
  eLON_HVAC_NIGHT_PURGE := 4,
  eLON_HVAC_PRE_COOL := 5,
  eLON_HVAC_OFF      := 6,
  eLON_HVAC_TEST     := 7,
  eLON_HVAC_EMERG_HEAT := 8,
  eLON_HVAC_FAN_ONLY := 9,
  eLON_HVAC_FREE_COOL := 10,
  eLON_HVAC_ICE      := 11,
  eLON_HVAC_MAX_HEAT := 12,
  eLON_HVAC_ECONOMY := 13,
  eLON_HVAC_DEHUMID := 14,
  eLON_HVAC_CALIBRATE := 15,
  eLON_HVAC_EMERG_COOL := 16,
  eLON_HVAC_EMERG_STEAM := 17,
  eLON_HVAC_MAX_COOL := 18,
  eLON_HVAC_HVC_LOAD := 19,
  eLON_HVAC_NO_LOAD := 20
)
END_TYPE

```

eLON_HVAC_NUL: Invalid value
 eLON_HVAC_AUTO: Controller automatically changes between application modes
 eLON_HVAC_HEAT: Heating only
 eLON_HVAC_MRNG_WRMUP: Application-specific morning warm-up
 eLON_HVAC_COOL: Cooling only
 eLON_HVAC_NIGHT_PURGE: Application-specific night purge

eLON_HVAC_PRE_COOL: Application-specific pre-cool
 eLON_HVAC_OFF: Controller not controlling outputs
 eLON_HVAC_TEST: Equipment being tested
 eLON_HVAC_EMERG_HEAT: Emergency heat mode (heat pump)
 eLON_HVAC_FAN_ONLY: Air not conditioned, fan turned on
 eLON_HVAC_FREE_COOL: Cooling with compressor not running
 eLON_HVAC_ICE: Ice-making mode
 eLON_HVAC_MAX_HEAT: Maximum heating mode
 eLON_HVAC_ECONOMY: Economic Heat/Cool mode
 eLON_HVAC_DEHUMID: Dehumidification mode
 eLON_HVAC_CALIBRATE: Calibration mode
 eLON_HVAC_EMERG_COOL: Emergency cool mode
 eLON_HVAC_EMERG_STEAM: Emergency steam mode
 eLON_HVAC_MAX_COOL:
 eLON_HVAC_HVC_LOAD:
 eLON_HVAC_NO_LOAD:

7.3.39 E_LON_learn_mode_t

Used by: SNVT_preset

```
TYPE E_LON_learn_mode_t :
(
  eLON_LN_NUL           := -1,
  eLON_LN_RECALL       := 0,
  eLON_LN_LEARN_CURRENT := 1,
  eLON_LN_LEARN_VALUE  := 2,
  eLON_LN_REPORT_VALUE := 3
)
END_TYPE
```

eLON_LN_NUL: Invalid Value

eLON_LN_RECALL: Recall

eLON_LN_LEARN_CURRENT: Learn present value

eLON_LN_LEARN_VALUE: Learn given value

eLON_LN_REPORT_VALUE: Report the value

7.3.40 E_LON_log_status_t

Used by: SCPTlogRecord / SNVT_log_status

```
TYPE E_LON_log_status_t :
(
  eLON_LS_NUL           := -1,
  eLON_LS_ENABLED      := 0,
  eLON_LS_DISABLED     := 1,
  eLON_LS_FULL         := 2,
  eLON_LS_OVERFLOW_ERR := 3,
  eLON_LS_INVALID_LOG_ERR := 4,
  eLON_LS_APP_ERR      := 5
)
END_TYPE
```

eLON_LS_NUL: Invalid value
 eLON_LS_ENABLED: Log enabled
 eLON_LS_DISABLED: Log disabled
 eLON_LS_FULL: Log enabled and full
 eLON_LS_OVERFLOW_ERR: Log enabled, overflow occurred
 eLON_LS_INVALID_LOG_ERR: Invalid log selected
 eLON_LS_APP_ERR: Other application error

7.3.41 E_LON_motor_state_t

Used by: SNVT_motor_state / SNVT_pumpset_mn

```
TYPE E_LON_motor_state_t :
(
  eLON_MOTOR_NUL           := -1,
  eLON_MOTOR_STOPPED      := 0,
  eLON_MOTOR_STARTING     := 1,
  eLON_MOTOR_ACCELERATING := 2,
  eLON_MOTOR_AT_STANDBY   := 3,
  eLON_MOTOR_AT_NORMAL    := 4,
  eLON_MOTOR_AT_REFERENCE := 5,
  eLON_MOTOR_DECELERATING := 6,
  eLON_MOTOR_STOPPING     := 7
)
END_TYPE
```

eLON_MOTOR_NUL: The state of the motor is unknown (invalid value)
 eLON_MOTOR_STOPPED: The motor is not running
 eLON_MOTOR_STARTING: The motor is performing its start-up sequence
 eLON_MOTOR_ACCELERATING: The motor is running. Speed is increasing.
 eLON_MOTOR_AT_STANDBY: The motor is running in its standby mode
 eLON_MOTOR_AT_NORMAL: The motor is running in its normal operational mode
 eLON_MOTOR_AT_REFERENCE: The motor is running at its reference speed.
 eLON_MOTOR_DECELERATING: The motor is running. Speed is decreasing.
 eLON_MOTOR_STOPPING: The motor is running, beginning its shutdown sequence.

7.3.42 E_LON_nv_type_category_t

Used by: SNVT_nv_type

```
TYPE E_LON_nv_type_category_t :
(
  eLON_NVT_CAT_NUL           := -1,
  eLON_NVT_CAT_INITIAL      := 0,
  eLON_NVT_CAT_SIGNED_CHAR  := 1,
  eLON_NVT_CAT_UNSIGNED_CHAR := 2,
  eLON_NVT_CAT_SIGNED_SHORT := 3,
  eLON_NVT_CAT_UNSIGNED_SHORT := 4,
  eLON_NVT_CAT_SIGNED_LONG  := 5,
  eLON_NVT_CAT_UNSIGNED_LONG := 6,
  eLON_NVT_CAT_ENUM         := 7,
  eLON_NVT_CAT_ARRAY        := 8,
  eLON_NVT_CAT_STRUCT       := 9,
  eLON_NVT_CAT_UNION        := 10,
  eLON_NVT_CAT_BITFIELD     := 11,
  eLON_NVT_CAT_FLOAT        := 12,
  eLON_NVT_CAT_SIGNED_QUAD  := 13,
)
```

```
eLON_NVT_CAT_REFERENCE      := 14
)
END_TYPE
```

eLON_NVT_CAT_NUL: Invalid Value

eLON_NVT_CAT_INITIAL:

eLON_NVT_CAT_SIGNED_CHAR: 8-bit signed character

eLON_NVT_CAT_UNSIGNED_CHAR: 8-bit unsigned character

eLON_NVT_CAT_SIGNED_SHORT: 8-bit signed integer

eLON_NVT_CAT_UNSIGNED_SHORT: 8-bit unsigned integer

eLON_NVT_CAT_SIGNED_LONG: 16-bit signed integer

eLON_NVT_CAT_UNSIGNED_LONG: 16-bit unsigned integer

eLON_NVT_CAT_ENUM: 8-bit enumeration

eLON_NVT_CAT_ARRAY: Array

eLON_NVT_CAT_STRUCT: Structure

eLON_NVT_CAT_UNION: Union

eLON_NVT_CAT_BITFIELD: Bitfield

eLON_NVT_CAT_FLOAT: 32-bit IEC 60559 (IEEE 754) floating-point value

eLON_NVT_CAT_SIGNED_QUAD: 32-bit signed integer

eLON_NVT_CAT_REFERENCE: Reference type

7.3.43 E_LON_object_request_t

Used by: SNVT_obj_request

```
TYPE E_LON_object_request_t :
(
  eLON_RQ_NUL           := -1,
  eLON_RQ_NORMAL       := 0,
  eLON_RQ_DISABLED     := 1,
  eLON_RQ_UPDATE_STATUS := 2,
  eLON_RQ_SELF_TEST    := 3,
  eLON_RQ_UPDATE_ALARM := 4,
  eLON_RQ_REPORT_MASK  := 5,
  eLON_RQ_OVERRIDE     := 6,
  eLON_RQ_ENABLE       := 7,
  eLON_RQ_RMV_OVERRIDE := 8,
  eLON_RQ_CLEAR_STATUS := 9,
  eLON_RQ_CLEAR_ALARM  := 10,
  eLON_RQ_ALARM_NOTIFY_ENABLED := 11,
  eLON_RQ_ALARM_NOTIFY_DISABLED := 12,
  eLON_RQ_MANUAL_CTRL  := 13,
  eLON_RQ_REMOTE_CTRL  := 14,
  eLON_RQ_PROGRAM      := 15,
  eLON_RQ_CLEAR_RESET  := 16,
  eLON_RQ_RESET        := 17,
  eLON_RQ_CLEAR_LOG    := 18
)
END_TYPE
```

eLON_RQ_NUL: Invalid Value

eLON_RQ_NORMAL: Enable object and remove override

eLON_RQ_DISABLED: Disable object

eLON_RQ_UPDATE_STATUS: Report object status

eLON_RQ_SELF_TEST: Perform object self-test

eLON_RQ_UPDATE_ALARM: Update alarm status
 eLON_RQ_REPORT_MASK: Report status bit mask
 eLON_RQ_OVERRIDE: Override object
 eLON_RQ_ENABLE: Enable object
 eLON_RQ_RMV_OVERRIDE: Remove object override
 eLON_RQ_CLEAR_STATUS: Clear object status
 eLON_RQ_CLEAR_ALARM: Clear object alarm
 eLON_RQ_ALARM_NOTIFY_ENABLED: Enable alarm notification
 eLON_RQ_ALARM_NOTIFY_DISABLED: Disable alarm notification
 eLON_RQ_MANUAL_CTRL: Enable object for manual control
 eLON_RQ_REMOTE_CTRL: Enable object for remote control
 eLON_RQ_PROGRAM: Enable programming of special configuration properties
 eLON_RQ_CLEAR_RESET: Clear reset-complete flag (reset_complete)
 eLON_RQ_RESET: Execute reset-sequence of object
 eLON_RQ_CLEAR_LOG: Clear data log

7.3.44 E_LON_occup_t

Used by: SNVT_occupancy / SNVT_tod_event

```
TYPE E_LON_occup_t :
(
  eLON_OC_NUL      := -1,
  eLON_OC_OCCUPIED := 0,
  eLON_OC_UNOCCUPIED := 1,
  eLON_OC_BYPASS  := 2,
  eLON_OC_STANDBY := 3
)
END_TYPE
```

eLON_OC_NUL: Invalid Value
 eLON_OC_OCCUPIED: Area is occupied
 eLON_OC_UNOCCUPIED: Area is unoccupied
 eLON_OC_BYPASS: Area is temporarily occupied for the bypass period
 eLON_OC_STANDBY: Area is temporarily unoccupied

7.3.45 E_LON_override_t

Used by: SNVT_override

```
TYPE E_LON_override_t :
(
  eLON_OV_NUL      := -1,
  eLON_OV_RETAIN   := 0,
  eLON_OV_SPECIFIED := 1,
  eLON_OV_DEFAULT  := 2
)
END_TYPE
```

eLON_OV_NUL: Invalid Value
 eLON_OV_RETAIN: Retain current level
 eLON_OV_SPECIFIED: Go to specified level

eLON_OV_DEFAULT: Go to default level

7.3.46 E_LON_pan_dir_t

Used by: SNVT_ptz

```
TYPE E_LON_pan_dir_t :
(
  eLON_PAN_NUL      := -1,
  eLON_PAN_STOP    := 0,
  eLON_PAN_RIGHT   := 1,
  eLON_PAN_LEFT    := 2
)
END_TYPE
```

eLON_PAN_NUL: Invalid Value

eLON_PAN_STOP: Stop panning

eLON_PAN_RIGHT: Pan to the right

eLON_PAN_LEFT: Pan to the left

7.3.47 E_LON_priority_level_t

Used by: SNVT_alarm / SNVT_alarm_2 / SNVT_pumpset_mn

```
TYPE E_LON_priority_level_t :
(
  eLON_PR_NUL      := -1,
  eLON_PR_LEVEL_0 := 0,
  eLON_PR_LEVEL_1 := 1,
  eLON_PR_LEVEL_2 := 2,
  eLON_PR_LEVEL_3 := 3,
  eLON_PR_1        := 4,
  eLON_PR_2        := 5,
  eLON_PR_3        := 6,
  eLON_PR_4        := 7,
  eLON_PR_6        := 8,
  eLON_PR_8        := 9,
  eLON_PR_10       := 10,
  eLON_PR_16       := 11
)
END_TYPE
```

eLON_PR_NUL: Invalid Value

eLON_PR_LEVEL_0: Lowest alarm priority level

eLON_PR_LEVEL_1:

eLON_PR_LEVEL_2:

eLON_PR_LEVEL_3: Highest alarm priority level

eLON_PR_1: Life Safety Fire Alarms (BACnet Priority 2)

eLON_PR_2: Property Safety Fire Alarms (BACnet Priority 3)

eLON_PR_3: Fire Supervisory Alarm (BACnet Priority 4)

eLON_PR_4: Fire Trouble/Fault (Display) (BACnet Priority 5)

eLON_PR_6: Fire Pre-Alarm, HVAC Critical Equipment Alarm (BACnet Priority 6)

eLON_PR_8: HVAC Alarms (BACnet Priority 8)

eLON_PR_10: HVAC Critical Equipment RTN, Fire RTN (Display) (BACnet Priority 10)

eLON_PR_16: HVAC RTN (lowest priority) (BACnet Priority 16)

7.3.48 E_LON_privacyzone_t

Used by: SNVT_privacyzone

```

TYPE E_LON_privacyzone_t :
(
  eLON_PZ_NUL      := -1,
  eLON_PZ_DISABLE := 0,
  eLON_PZ_ENABLE  := 1,
  eLON_PZ_UPPER_LEFT := 2,
  eLON_PZ_LOWER_RIGHT := 3,
  eLON_PZ_ENTER   := 4,
  eLON_PZ_EXIT    := 5
)
END_TYPE

```

eLON_PZ_NUL: Invalid value

eLON_PZ_DISABLE: Disable privacy zone warning

eLON_PZ_ENABLE: Enable privacy zone warning

eLON_PZ_UPPER_LEFT: Set upper left corner

eLON_PZ_LOWER_RIGHT: Set lower right corner

eLON_PZ_ENTER: Privacy zone enter warning

eLON_PZ_EXIT: Privacy zone exit message

7.3.49 E_LON_rail_audio_sensor_type_t

Used by: SNVT_rac_ctrl / SNVT_rac_req

```

TYPE E_LON_rail_audio_sensor_type_t :
(
  eLON_RAST_NUL      := -1,
  eLON_RAST_CU_TYPE_1 := 0,
  eLON_RAST_CU_TYPE_2 := 1,
  eLON_RAST_CU_TYPE_3 := 2,
  eLON_RAST_CU_TYPE_4 := 3,
  eLON_RAST_LS_LINE_1 := 4,
  eLON_RAST_LS_LINE_2 := 5,
  eLON_RAST_LS_LINE_3 := 6,
  eLON_RAST_LS_LINE_4 := 7,
  eLON_RAST_LS_LINE_5 := 8,
  eLON_RAST_LS_LINE_6 := 9,
  eLON_RAST_LS_LINE_7 := 10,
  eLON_RAST_LS_LINE_8 := 11,
  eLON_RAST_PAÜ      := 12,
  eLON_RAST_CFA_TYPE_1 := 13,
  eLON_RAST_CFA_TYPE_2 := 14,
  eLON_RAST_CFA_TYPE_3 := 15,
  eLON_RAST_CFA_TYPE_4 := 16,
  eLON_RAST_DVA      := 17,
  eLON_RAST_ET_TYPE_1 := 18,
  eLON_RAST_ET_TYPE_2 := 19,
  eLON_RAST_USERDEF_TYPE_1 := 20,
  eLON_RAST_USERDEF_TYPE_2 := 21,
  eLON_RAST_USERDEF_TYPE_3 := 22,
  eLON_RAST_USERDEF_TYPE_4 := 23
)
END_TYPE

```

eLON_RAST_NUL: Invalid Value

eLON_RAST_CU_TYPE_1: CU Type 1

eLON_RAST_CU_TYPE_2: CU Type 2

eLON_RAST_CU_TYPE_3:

eLON_RAST_CU_TYPE_4: CU Type 4

eLON_RAST_LS_LINE_1: LS Line 1

eLON_RAST_LS_LINE_2: LS Line 2
 eLON_RAST_LS_LINE_3: LS Line 3
 eLON_RAST_LS_LINE_4: LS Line 4
 eLON_RAST_LS_LINE_5: LS Line 5
 eLON_RAST_LS_LINE_6: LS Line 6
 eLON_RAST_LS_LINE_7: LS Line 7
 eLON_RAST_LS_LINE_8: LS Line 8
 eLON_RAST_PAU: Public-Address Unit
 eLON_RAST_CFA_TYPE_1: CFA Type 1
 eLON_RAST_CFA_TYPE_2: CFA Type 2
 eLON_RAST_CFA_TYPE_3: CFA Type 3
 eLON_RAST_CFA_TYPE_4: CFA Type 4
 eLON_RAST_DVA: DVA
 eLON_RAST_ET_TYPE_1: ET Type 1
 eLON_RAST_ET_TYPE_2: ET Type 2
 eLON_RAST_USERDEF_TYPE_1: User-defined Type 1
 eLON_RAST_USERDEF_TYPE_2: User-defined Type 2
 eLON_RAST_USERDEF_TYPE_3: User-defined Type 3
 eLON_RAST_USERDEF_TYPE_4: User-defined Type 4

7.3.50 E_LON_rail_audio_type_t

Used by: SNVT_rac_ctrl / SNVT_rac_req

```

TYPE E_LON_rail_audio_type_t :
(
  eLON_RAT_NUL          := -1,
  eLON_RAT_IC_REQ      := 0,
  eLON_RAT_IC_JOIN     := 1,
  eLON_RAT_IC_QUIT    := 2,
  eLON_RAT_IC_END     := 3,
  eLON_RAT_HW_RADIO_REQ := 4,
  eLON_RAT_HW_RADIO_END := 5,
  eLON_RAT_HW_PA_REQ  := 6,
  eLON_RAT_HW_PA_END  := 7,
  eLON_RAT_SW_PA_REQ  := 8,
  eLON_RAT_SW_PA_END  := 9,
  eLON_RAT_SW_PA_OR_REQ := 10,
  eLON_RAT_SW_PA_OR_END := 11,
  eLON_RAT_PAU_REQ    := 12,
  eLON_RAT_PAU_ACCEPT := 13,
  eLON_RAT_PAU_CALL   := 14,
  eLON_RAT_PAU_END    := 15,
  eLON_RAT_ENTERT_REQ := 16,
  eLON_RAT_ENTERT_END := 17
)
END_TYPE
  
```

eLON_RAT_NUL:

eLON_RAT_IC_REQ:

eLON_RAT_IC_JOIN:

eLON_RAT_IC_QUIT:

eLON_RAT_IC_END:
 eLON_RAT_HW_RADIO_REQ:
 eLON_RAT_HW_RADIO_END:
 eLON_RAT_HW_PA_REQ:
 eLON_RAT_HW_PA_END:
 eLON_RAT_SW_PA_REQ:
 eLON_RAT_SW_PA_END:
 eLON_RAT_SW_PA_OR_REQ:
 eLON_RAT_SW_PA_OR_END:
 eLON_RAT_PAU_REQ:
 eLON_RAT_PAU_ACCEPT:
 eLON_RAT_PAU_CALL:
 eLON_RAT_PAU_END:
 eLON_RAT_ENTERT_REQ:
 eLON_RAT_ENTERT_END:

7.3.51 E_LON_reg_val_unit_t

Used by: SNVT_reg_val / SNVT_reg_val_ts

```

TYPE E_LON_reg_val_unit_t :
(
  eLON_RVU_NUL      := -1,
  eLON_RVU_NONE    := 0,
  eLON_RVU_W       := 1,
  eLON_RVU_KW      := 2,
  eLON_RVU_MW      := 3,
  eLON_RVU_GW      := 4,
  eLON_RVU_VAR     := 5,
  eLON_RVU_KVAR    := 6,
  eLON_RVU_MVAR    := 7,
  eLON_RVU_GVAR    := 8,
  eLON_RVU_WH      := 9,
  eLON_RVU_KWH     := 10,
  eLON_RVU_MWH     := 11,
  eLON_RVU_GWH     := 12,
  eLON_RVU_VARH    := 13,
  eLON_RVU_KVARH   := 14,
  eLON_RVU_MVARH   := 15,
  eLON_RVU_GVARH   := 16,
  eLON_RVU_V       := 17,
  eLON_RVU_A       := 18,
  eLON_RVU_COSF    := 19,
  eLON_RVU_M3      := 20,
  eLON_RVU_L       := 21,
  eLON_RVU_ML      := 22,
  eLON_RVU_USGAL   := 23,
  eLON_RVU_GJ      := 24,
  eLON_RVU_MJ      := 25,
  eLON_RVU_MCAL    := 26,
  eLON_RVU_KCAL    := 27,
  eLON_RVU_MBTU    := 28,
  eLON_RVU_KBTU    := 29,
  eLON_RVU_MJH     := 30,
  eLON_RVU_MLS     := 31,
  eLON_RVU_LS      := 32,
  eLON_RVU_M3S     := 33,
  eLON_RVU_C       := 34,
  eLON_RVU_LH      := 35,
  eLON_RVU_VA      := 36,
  eLON_RVU_KVA     := 37,

```

```

eLON_RVU_MVA      := 38,
eLON_RVU_GVA      := 39,
eLON_RVU_VAH      := 40,
eLON_RVU_KVAH     := 41,
eLON_RVU_MVAH     := 42,
eLON_RVU_GVAH     := 43
)
END_TYPE

```

eLON_RVU_NUL: invalid unit of measure (INVALID)

eLON_RVU_NONE: no units specified ()

eLON_RVU_W: Watts (W)

eLON_RVU_KW: kiloWatts (kW)

eLON_RVU_MW: megaWatts (MW)

eLON_RVU_GW: gigaWatts (GW)

eLON_RVU_VAR: Volt-Amperes reactive (var)

eLON_RVU_KVAR: kilo-Volt-Amperes reactive (kvar)

eLON_RVU_MVAR: mega-Volt-Amperes reactive (Mvar)

eLON_RVU_GVAR: giga-Volt-Amperes reactive (Gvar)

eLON_RVU_WH: Watt-hour (Wh)

eLON_RVU_KWH: kiloWatt-hour (kWh)

eLON_RVU_MWH: megaWatt-hour (MWh)

eLON_RVU_GWH: gigaWatt-hour (GWh)

eLON_RVU_VARH: Volt-Amperes reactive -hour (varh)

eLON_RVU_KVARH: kilo-Volt-Amperes reactive -hour (kvarh)

eLON_RVU_MVARH: mega-Volt-Amperes reactive -hour (Mvarh)

eLON_RVU_GVARH: giga-Volt-Amperes reactive -hour (Gvarh)

eLON_RVU_V: Volts (V)

eLON_RVU_A: Amps (A)

eLON_RVU_COSF: (cosf)

eLON_RVU_M3: cubic meters (m³)(cu.m)

eLON_RVU_L: liters (l)

eLON_RVU_ML: milliliters (ml)

eLON_RVU_USGAL: U.S. Gallons (USG)

eLON_RVU_GJ: giga-Joules (GJ)

eLON_RVU_MJ: mega-Joules (MJ)

eLON_RVU_MCAL: megacalories (Mcal)

eLON_RVU_KCAL: kilocalories (kcal) / Calories (Cal)

eLON_RVU_MBTU: mega-British thermal units (mBtu)

eLON_RVU_KBTU: kilo-British thermal units (kBtu)

eLON_RVU_MJH: mega-Joules per hour (MJ/h)

eLON_RVU_MLS: milliliters per second (ml/s)
 eLON_RVU_LS: liters per second (l/s)
 eLON_RVU_M3S: cubic-meters per second (m³/s) (cu.m/s)
 eLON_RVU_C: (C)
 eLON_RVU_LH: liters per hour (l/h)
 eLON_RVU_VA: Volt-Amperes (VA)
 eLON_RVU_KVA: kiloVolt-Amperes (kVA)
 eLON_RVU_MVA: megaVolt-Amperes (MVA)
 eLON_RVU_GVA: gigaVolt-Amperes (GVA)
 eLON_RVU_VAH: Volt-Ampere hours (VAh)
 eLON_RVU_KVAH: kiloVolt-Ampere hours (kVAh)
 eLON_RVU_MVAH: megaVolt-Ampere hours (MVAh)
 eLON_RVU_GVAH: giga-Volt-Ampere hours (GVAh)

7.3.52 E_LON_sblnd_cmd_source_t

Used by: SNVT_sblnd_state

```

TYPE E_LON_sblnd_cmd_source_t :
(
  eLON_SBCS_NUL           := -1,
  eLON_SBCS_LOCAL        := 0,
  eLON_SBCS_GROUP        := 1,
  eLON_SBCS_WIND_SPEED   := 2,
  eLON_SBCS_SUN_LUX      := 3,
  eLON_SBCS_RAIN         := 4,
  eLON_SBCS_FROST        := 5,
  eLON_SBCS_DAWN         := 6,
  eLON_SBCS_DUSK        := 7,
  eLON_SBCS_OUTSIDE_TEMP := 8,
  eLON_SBCS_INDOOR_TEMP  := 9,
  eLON_SBCS_OUTDOOR_RH   := 10,
  eLON_SBCS_INDOOR_RH    := 11,
  eLON_SBCS_ILLUM_LEVEL  := 12,
  eLON_SBCS_SCENE        := 13,
  eLON_SBCS_GLOBAL       := 14,
  eLON_SBCS_WINDOW_CONTACT := 15,
  eLON_SBCS_AUTOMODE_CHANGED := 16,
  eLON_SBCS_OVERRIDE     := 17,
  eLON_SBCS_EMERGENCY    := 18,
  eLON_SBCS_MAINTENANCE  := 19,
  eLON_SBCS_INTRUSION    := 20,
  eLON_SBCS_TERMINAL_LOAD := 21,
  eLON_SBCS_ALARM        := 22,
  eLON_SBCS_OCC_SENSOR   := 23,
  eLON_SBCS_OCC_MAN_CMD  := 24,
  eLON_SBCS_GLARE        := 25,
  eLON_SBCS_ALARM_2      := 26,
  eLON_SBCS_NOTIFY       := 27,
  eLON_SBCS_ELEVATION    := 28,
  eLON_SBCS_AZIMUTH      := 29,
  eLON_SBCS_SET_OVERRIDE := 30,
  eLON_SBCS_SET_MAINTENANCE := 31,
  eLON_SBCS_TIMER        := 32,
  eLON_SBCS_UNKNOWN      := 127
)
END_TYPE

```

eLON_SBCS_NUL: Invalid value

eLON_SBCS_LOCAL: Local

eLON_SBCS_GROUP: Group

eLON_SBCS_WIND_SPEED: Wind speed
eLON_SBCS_SUN_LUX: Sun lux level
eLON_SBCS_RAIN: Rain
eLON_SBCS_FROST: Frost
eLON_SBCS_DAWN: Dawn
eLON_SBCS_DUSK: Dusk
eLON_SBCS_OUTSIDE_TEMP: Outside temperature
eLON_SBCS_INDOOR_TEMP: Indoor temperature
eLON_SBCS_OUTDOOR_RH: Outdoor relative humidity
eLON_SBCS_INDOOR_RH: Indoor relative humidity
eLON_SBCS_ILLUM_LEVEL: Illumination level
eLON_SBCS_SCENE: Scene
eLON_SBCS_GLOBAL: Global
eLON_SBCS_WINDOW_CONTACT: Window contact
eLON_SBCS_AUTOMODE_CHANGED: Auto-mode changed
eLON_SBCS_OVERRIDE: Override
eLON_SBCS_EMERGENCY: Emergency
eLON_SBCS_MAINTENANCE: Maintenance
eLON_SBCS_INTRUSION: Intrusion
eLON_SBCS_TERMINAL_LOAD: Terminal load
eLON_SBCS_ALARM: Alarm
eLON_SBCS_OCC_SENSOR: Occupancy sensor
eLON_SBCS_OCC_MAN_CMD: Occupancy manual command
eLON_SBCS_GLARE: Glare
eLON_SBCS_ALARM_2: Alarm 2
eLON_SBCS_NOTIFY: Notify
eLON_SBCS_ELEVATION: Elevation
eLON_SBCS_AZIMUTH: Azimuth
eLON_SBCS_SET_OVERRIDE: Set override
eLON_SBCS_SET_MAINTENANCE: Set maintenance
eLON_SBCS_TIMER: Timer
eLON_SBCS_UNKNOWN: Unknown command source

7.3.53 E_LON_sblnd_error_t

Used by: SNVT_sblnd_state

```
TYPE E_LON_sblnd_error_t :  
(  
    eLON_SBE_NUL           := -1,  
    eLON_SBE_NO_ERROR     := 0,
```



```

eLON_SBE_IN_PROGRESS      := 1,
eLON_SBE_LIMITS          := 2,
eLON_SBE_OBSTACLE_UP     := 3,
eLON_SBE_OBSTACLE_DOWN  := 4,
eLON_SBE_OVERHEAT       := 5,
eLON_SBE_POWER           := 6,
eLON_SBE_SENSOR          := 7,
eLON_SBE_MOTOR_CIRCUIT  := 8,
eLON_SBE_FUSE            := 9,
eLON_SBE_REFERENCE_LOST := 10,
eLON_SBE_HOST_COMM       := 11,
eLON_SBE_VOLTAGE_1       := 12,
eLON_SBE_VOLTAGE_2       := 13,
eLON_SBE_CONTROLLER      := 14
)
END_TYPE

```

eLON_SBE_NUL: Invalid Value

eLON_SBE_NO_ERROR: No error

eLON_SBE_IN_PROGRESS: In progress

eLON_SBE_LIMITS: Limits

eLON_SBE_OBSTACLE_UP: Obstacle up

eLON_SBE_OBSTACLE_DOWN: Obstacle down

eLON_SBE_OVERHEAT: Overheat

eLON_SBE_POWER: Power

eLON_SBE_SENSOR: Sensor

eLON_SBE_MOTOR_CIRCUIT: Motor circuit

eLON_SBE_FUSE: Fuse

eLON_SBE_REFERENCE_LOST: Reference lost

eLON_SBE_HOST_COMM: Host communication

eLON_SBE_VOLTAGE_1: Voltage 1

eLON_SBE_VOLTAGE_2: Voltage 2

eLON_SBE_CONTROLLER: Controller

7.3.54 E_LON_scene_config_t

Used by: SNVT_scene_cfg

```

TYPE E_LON_scene_config_t :
(
  eLON_SCF_NUL      := -1,
  eLON_SCF_SAVE     := 0,
  eLON_SCF_CLEAR    := 1,
  eLON_SCF_REPORT   := 2,
  eLON_SCF_SIZE     := 3,
  eLON_SCF_FREE     := 4
)
END_TYPE

```

eLON_SCF_NUL: Invalid Value

eLON_SCF_SAVE: Overwrite this scene with new data

eLON_SCF_CLEAR: Delete this scene from the list

eLON_SCF_REPORT: Display this scene's data

eLON_SCF_SIZE: Report the number of programmed scenes

eLON_SCF_FREE: Report the number of free scene storage spaces

7.3.55 E_LON_scene_t

Used by: SNVT_scene

```

TYPE E_LON_scene_t :
(
  eLON_SC_NUL           := -1,
  eLON_SC_RECALL       := 0,
  eLON_SC_LEARN        := 1,
  eLON_SC_DISPLAY      := 2,
  eLON_SC_GROUP_OFF    := 3,
  eLON_SC_GROUP_ON     := 4,
  eLON_SC_STATUS_OFF   := 5,
  eLON_SC_STATUS_ON    := 6,
  eLON_SC_STATUS_MIXED := 7,
  eLON_SC_GROUP_STATUS := 8,
  eLON_SC_FLICK        := 9,
  eLON_SC_TIMEOUT      := 10,
  eLON_SC_TIMEOUT_FLICK := 11,
  eLON_SC_DELAYOFF     := 12,
  eLON_SC_DELAYOFF_FLICK := 13,
  eLON_SC_DELAYON      := 14,
  eLON_SC_ENABLE_GROUP := 15,
  eLON_SC_DISABLE_GROUP := 16,
  eLON_SC_CLEANON      := 17,
  eLON_SC_CLEANOFF     := 18,
  eLON_SC_WINK         := 19,
  eLON_SC_RESET        := 20,
  eLON_SC_MODE1        := 21,
  eLON_SC_MODE2        := 22,
  eLON_SC_MODE3        := 23
)
END_TYPE

```

eLON_SC_NUL: Invalid value

eLON_SC_RECALL: Recall a specified scene.

eLON_SC_LEARN: Store the current setting in the specified scene.

eLON_SC_DISPLAY: Display the current scene.

eLON_SC_GROUP_OFF: Report current group is off.

eLON_SC_GROUP_ON: Report current group is on.

eLON_SC_STATUS_OFF: Report current status is off.

eLON_SC_STATUS_ON: Report current status is on.

eLON_SC_STATUS_MIXED: Report current status is mixed.

eLON_SC_GROUP_STATUS: Get group status.

eLON_SC_FLICK: Toggle state off and then on.

eLON_SC_TIMEOUT: Report a timeout occurred.

eLON_SC_TIMEOUT_FLICK: Report a timeout occurred for a flick warning.

eLON_SC_DELAYOFF: Set the state to off after a delay.

eLON_SC_DELAYOFF_FLICK: Flick and then set the state to off after a delay.

eLON_SC_DELAYON: Set the state to on after a delay.

eLON_SC_ENABLE_GROUP: Enable the current group.

eLON_SC_DISABLE_GROUP: Disable the current group.

eLON_SC_CLEANON: Recall the cleaning scene.

eLON_SC_CLEANOFF: Restore the previous scene.

eLON_SC_WINK: Toggle to the opposite state and then restore the state.

eLON_SC_RESET: Restore the factory default scene table.

eLON_SC_MODE1: Manufacturer-specific mode 1.

eLON_SC_MODE2: Manufacturer-specific mode 2.

eLON_SC_MODE3: Manufacturer-specific mode 3.

7.3.56 E_LON_sec_state_t

Used by: SNVT_sec_state

```

TYPE E_LON_sec_state_t :
(
  eLON_SSE_NUL           := -1,
  eLON_SSE_OFF           := 0,
  eLON_SSE_ON            := 1,
  eLON_SSE_INHIBIT_RESET := 2,
  eLON_SSE_INHIBIT       := 3,
  eLON_SSE_WALK_TEST_OFF := 4,
  eLON_SSE_WALK_TEST_ON  := 5,
  eLON_SSE_TEST_MODE_OFF := 6,
  eLON_SSE_TEST_MODE_ON  := 7,
  eLON_SSE_POLL_STATUS   := 8,
  eLON_SSE_POLL_STATE    := 9,
  eLON_SSE_CONFIRM_ALARM_RESET := 10,
  eLON_SSE_CONFIRM_ALARM   := 11,
  eLON_SSE_CONFIRM_TAMPER_RESET := 12,
  eLON_SSE_CONFIRM_TAMPER  := 13,
  eLON_SSE_CONFIRM_MAINTENANCE := 14,
  eLON_SSE_CONFIRM_TROUBLE := 15,
  eLON_SSE_CONFIRM_FAULT   := 16,
  eLON_SSE_CONFIRM_RECOVERED_SENSOR := 17,
  eLON_SSE_LOST_SENSOR     := 18,
  eLON_SSE_CONFIRM_UNSUPPORTED := 19
)
END_TYPE

```

eLON_SSE_NUL:

eLON_SSE_OFF:

eLON_SSE_ON:

eLON_SSE_INHIBIT_RESET:

eLON_SSE_INHIBIT:

eLON_SSE_WALK_TEST_OFF:

eLON_SSE_WALK_TEST_ON:

eLON_SSE_TEST_MODE_OFF:

eLON_SSE_TEST_MODE_ON:

eLON_SSE_POLL_STATUS:

eLON_SSE_POLL_STATE:

eLON_SSE_CONFIRM_ALARM_RESET:

eLON_SSE_CONFIRM_ALARM:

eLON_SSE_CONFIRM_TAMPER_RESET:

eLON_SSE_CONFIRM_TAMPER:

eLON_SSE_CONFIRM_MAINTENANCE:

eLON_SSE_CONFIRM_TROUBLE:

eLON_SSE_CONFIRM_FAULT:

eLON_SSE_CONFIRM_RECOVERED_SENSOR:

eLON_SSE_LOST_SENSOR:

eLON_SSE_CONFIRM_UNSUPPORTED:

7.3.57 E_LON_sec_status_t

Used by: SNVT_sec_status

```

TYPE E_LON_sec_status_t :
(
  eLON_SSS_NUL                := -1,
  eLON_SSS_POWER_UP          := 0,
  eLON_SSS_ALARM_RESET       := 1,
  eLON_SSS_ALARM             := 2,
  eLON_SSS_TAMPER_RESET      := 3,
  eLON_SSS_TAMPER            := 4,
  eLON_SSS_MAINTENANCE       := 5,
  eLON_SSS_TROUBLE           := 6,
  eLON_SSS_FAULT             := 7,
  eLON_SSS_RECOVERED_SENSOR  := 8,
  eLON_SSS_LOST_SENSOR       := 9,
  eLON_SSS_POLL_ACTIVE       := 10,
  eLON_SSS_POLL_INACTIVE     := 11,
  eLON_SSS_POLL_TAMPER      := 12,
  eLON_SSS_POLL_ON           := 13,
  eLON_SSS_POLL_OFF          := 14,
  eLON_SSS_POLL_INHIBIT     := 15,
  eLON_SSS_POLL_TEST         := 16,
  eLON_SSS_CONFIRM_OFF      := 17,
  eLON_SSS_CONFIRM_ON       := 18,
  eLON_SSS_CONFIRM_INHIBIT_RESET := 19,
  eLON_SSS_CONFIRM_INHIBIT := 20,
  eLON_SSS_CONFIRM_WALK_TEST_OFF := 21,
  eLON_SSS_CONFIRM_WALK_TEST_ON := 22,
  eLON_SSS_CONFIRM_TEST_MODE_OFF := 23,
  eLON_SSS_CONFIRM_TEST_MODE_ON := 24,
  eLON_SSS_CONFIRM_UNSUPPORTED := 25
)
END_TYPE

```

eLON_SSS_NUL:

eLON_SSS_POWER_UP:

eLON_SSS_ALARM_RESET:

eLON_SSS_ALARM:

eLON_SSS_TAMPER_RESET:

eLON_SSS_TAMPER:

eLON_SSS_MAINTENANCE:

eLON_SSS_TROUBLE:

eLON_SSS_FAULT:

eLON_SSS_RECOVERED_SENSOR:

eLON_SSS_LOST_SENSOR:

eLON_SSS_POLL_ACTIVE:

eLON_SSS_POLL_INACTIVE:

eLON_SSS_POLL_TAMPER:

eLON_SSS_POLL_ON:
 eLON_SSS_POLL_OFF:
 eLON_SSS_POLL_INHIBIT:
 eLON_SSS_POLL_TEST:
 eLON_SSS_CONFIRM_OFF:
 eLON_SSS_CONFIRM_ON:
 eLON_SSS_CONFIRM_INHIBIT_RESET:
 eLON_SSS_CONFIRM_INHIBIT:
 eLON_SSS_CONFIRM_WALK_TEST_OFF:
 eLON_SSS_CONFIRM_WALK_TEST_ON:
 eLON_SSS_CONFIRM_TEST_MODE_OFF:
 eLON_SSS_CONFIRM_TEST_MODE_ON:
 eLON_SSS_CONFIRM_UNSUPPORTED:

7.3.58 E_LON_setting_t

Used by: SNVT_setting

```

TYPE E_LON_setting_t :
(
  eLON_SET_NUL      := -1,
  eLON_SET_OFF     := 0,
  eLON_SET_ON      := 1,
  eLON_SET_DOWN    := 2,
  eLON_SET_UP      := 3,
  eLON_SET_STOP    := 4,
  eLON_SET_STATE   := 5
)
END_TYPE

```

eLON_SET_NUL: Invalid value

eLON_SET_OFF: Change state to off

eLON_SET_ON: Change state to on, restoring the last on setting

eLON_SET_DOWN: Decrease the setting by the offset supplied in the setting field

eLON_SET_UP: Increase the setting by the offset supplied in the setting field

eLON_SET_STOP: Stop any motion, for example for blinds

eLON_SET_STATE: Change the setting to the value specified

7.3.59 E_LON_switch_state_t

Used by: SNVT_switch_2

```

TYPE E_LON_switch_state_t :
(
  eLON_SW_NUL          := -1,
  eLON_SW_SET_OFF     := 0,
  eLON_SW_SET_ON      := 1,
  eLON_SW_REPORT_OFF  := 2,
  eLON_SW_REPORT_ON   := 3,
  eLON_SW_TOGGLE_STATE := 4,
  eLON_SW_SET_LEVEL   := 5,
  eLON_SW_INCREASE_LEVEL := 6,
  eLON_SW_DECREASE_LEVEL := 7,
  eLON_SW_RECALL_SCENE := 8,
)

```

```

eLON_SW_STORE_SCENE      := 9,
eLON_SW_LEARN_SCENE     := 10,
eLON_SW_SET_OCCUPIED    := 11,
eLON_SW_SET_UNOCCUPIED := 12,
eLON_SW_SET_MULTIPLIER  := 13,
eLON_SW_ENABLE_GROUP    := 14,
eLON_SW_DISABLE_GROUP   := 15,
eLON_SW_WINK            := 16,
eLON_SW_RESET           := 17,
eLON_SW_RESET_ENERGY_USAGE := 18,
eLON_SW_RESET_RUNTIME   := 19,
eLON_SW_INCREASE_HUE    := 20,
eLON_SW_DECREASE_HUE   := 21,
eLON_SW_SET_BUTTON     := 22,
eLON_SW_SET_FAN_UP     := 32,
eLON_SW_SET_FAN_DOWN   := 33,
eLON_SW_TOGGLE_FAN_DIRECTION := 34,
eLON_SW_INCREASE_FAN_LEVEL := 35,
eLON_SW_DECREASE_FAN_LEVEL := 36,
eLON_SW_SET_FAN_ON     := 37,
eLON_SW_SET_FAN_OFF    := 38,
eLON_SW_TOGGLE_FAN_STATE := 39,
eLON_SW_MOVE_OPEN     := 48,
eLON_SW_MOVE_CLOSED   := 49,
eLON_SW_SET_ANGLE     := 50,
eLON_SW_ROTATE_OPEN   := 51,
eLON_SW_ROTATE_CLOSED := 52,
eLON_SW_STOP          := 53,
eLON_SW_SET_STANDBY   := 54,
eLON_SW_TOGGLE_STANDBY := 55,
eLON_SW_SET_POSITION  := 56,
eLON_SW_REPORT_POSITION := 57,
eLON_SW_REPORT_FAN_LEVEL := 58
)
END_TYPE

```

eLON_SW_NUL: Invalid value

eLON_SW_SET_OFF: Set the state to off; ignored for blinds, drapes, shades, and fans

eLON_SW_SET_ON: Set the state to on; ignored for blinds, drapes, shades, and fans

eLON_SW_REPORT_OFF: Report that the state is off; output only; ignored for input

eLON_SW_REPORT_ON: Report that the state is on; output only; ignored for input

eLON_SW_TOGGLE_STATE: Toggle on-off state; same action as SW_SET_OFF if the on/off state was on, and SW_SET_ON if the on/off state was off; ignored for blinds, drapes, shades, and fans

eLON_SW_SET_LEVEL: Set the level to the specified value; ignored for blinds, drapes, shades, and fans

eLON_SW_INCREASE_LEVEL: Increase the level by the specified value; ignored for blinds, drapes, shades, and fans

eLON_SW_DECREASE_LEVEL: Decrease the level by the specified amount; ignored for blinds, drapes, shades, and fans

eLON_SW_RECALL_SCENE: Recall the state and level from the specified scene

eLON_SW_STORE_SCENE: Store setting for the specified scene

eLON_SW_LEARN_SCENE: Learn setting for the specified scene

eLON_SW_SET_OCCUPIED: Set the occupancy state

eLON_SW_SET_UNOCCUPIED: Clear the occupancy state

eLON_SW_SET_MULTIPLIER: Set a multiplier for the level for 60 minutes; ignored for blinds, drapes, shades, and fans

eLON_SW_ENABLE_GROUP: Enable a group; all groups are enabled by default

eLON_SW_DISABLE_GROUP: Disable a group

eLON_SW_WINK: Blink state (toggle on-off state; pause; toggle on-off state again)

eLON_SW_RESET: Reset scene definitions, multiplier, occupancy state, group enable flags, and settings to factory defaults

eLON_SW_RESET_ENERGY_USAGE: Reset energy usage value to zero

eLON_SW_RESET_RUNTIME: Reset runtime value to zero

eLON_SW_INCREASE_HUE: Increase color hue

eLON_SW_DECREASE_HUE: Decrease color hue

eLON_SW_SET_BUTTON: Trigger the actions for pressing and releasing the button specified in the value field

eLON_SW_SET_FAN_UP: Set ceiling fan direction to up, with specified level

eLON_SW_SET_FAN_DOWN: Set ceiling fan direction to down, with specified level

eLON_SW_TOGGLE_FAN_DIRECTION: Toggle fan up-down direction

eLON_SW_INCREASE_FAN_LEVEL: Increase fan speed by the setting

eLON_SW_DECREASE_FAN_LEVEL: Decrease fan speed by the setting

eLON_SW_SET_FAN_ON: Set the fan state to on

eLON_SW_SET_FAN_OFF: Set the fan state to off

eLON_SW_TOGGLE_FAN_STATE: Toggle the fan on-off state

eLON_SW_MOVE_OPEN: Move blinds, drapes, or shades open by the setting

eLON_SW_MOVE_CLOSED: Move blinds, drapes, or shades closed by the setting

eLON_SW_SET_ANGLE: Set the rotation angle of blinds to the setting

eLON_SW_ROTATE_OPEN: Rotate blinds open by the setting

eLON_SW_ROTATE_CLOSED: Rotate blinds closed by the setting

eLON_SW_STOP: Stop any motion of blinds, drapes, or shades

eLON_SW_SET_STANDBY: Set Standby mode

eLON_SW_TOGGLE_STANDBY: Toggle the standby state

eLON_SW_SET_POSITION: Set blinds, drapes, or shades to the specified position; 100% is fully open, 0% is fully closed

eLON_SW_REPORT_POSITION: Report the position of blinds, drapes, or shades output only; ignored for input

eLON_SW_REPORT_FAN_LEVEL: Report the fan speed in percent of full level output only; ignored for input

7.3.60 E_LON_telcom_states_t

Used by: SNVT_telcom

```

TYPE E_LON_telcom_states_t :
(
  eLON_TEL_NUL      := -1,
  eLON_TEL_NOTINUSE := 0,
  eLON_TEL_OFFHOOK  := 1,
  eLON_TEL_DIALING  := 2,
  eLON_TEL_DIALCOMP := 3,
  eLON_TEL_RINGBACK := 4,
  eLON_TEL_INCOMING := 5,
  eLON_TEL_RINGING  := 6,
  eLON_TEL_ANSWERED := 7,
  eLON_TEL_CONNECTED := 8,
  eLON_TEL_TALKING  := 9,

```

```

eLON_TEL_HANGINGUP := 10,
eLON_TEL_HUNGUPX   := 11,
eLON_TEL_HOLD      := 12,
eLON_TEL_UNHOLD    := 13,
eLON_TEL_RELEASE   := 14,
eLON_TEL_FULLDUP   := 15,
eLON_TEL_BLOCKED   := 16,
eLON_TEL_CWAIT     := 17,
eLON_TEL_DESTBUSY  := 18,
eLON_TEL_NETBUSY   := 19,
eLON_TEL_ERROR     := 20
)
END_TYPE

```

eLON_TEL_NUL: Invalid Value

eLON_TEL_NOTINUSE: "Null State (U0)" not in use

eLON_TEL_OFFHOOK: "Call Initiated (U1)"

eLON_TEL_DIALING: "Overlap Sending (U2)"

eLON_TEL_DIALCOMP: "Outgoing Call Proceeding (U3)"

eLON_TEL_RINGBACK: "Call Delivered (U4)" hearing ringback

eLON_TEL_INCOMING: "Call Present (U6)" incoming call has not yet started ringing (only on ISDN line)

eLON_TEL_RINGING: "Call Received (U7)" incoming call when the user has indicated alerting but has not yet answered

eLON_TEL_ANSWERED: "Connect Request (U8)" user has answered the call and is waiting to be awarded the call

eLON_TEL_CONNECTED:

eLON_TEL_TALKING: "Active (U10)" two parties are exchanging data

eLON_TEL_HANGINGUP: "Disconnect Request (U11)" user has hung up

eLON_TEL_HUNGUPX: "Disconnect Indication (U12)" the other side hung up

eLON_TEL_HOLD: "Suspend Request (U15)" user has requested the network suspend the call

eLON_TEL_UNHOLD: "Resume Request (U17)" resume a held call (usually go back to TEL_TALKING)

eLON_TEL_RELEASE: "Release Request (U19)" user has requested the network to release

eLON_TEL_FULLDUP: "Overlap Receiving (U25)" user has acknowledged the call and is prepared to receive additional

eLON_TEL_BLOCKED: connection with blocking, (call-waiting disabled)

eLON_TEL_CWAIT: call-waiting coming in

eLON_TEL_DESTBUSY: destination busy

eLON_TEL_NETBUSY: problem, network

eLON_TEL_ERROR: problem, non-network

7.3.61 E_LON_therm_mode_t

Used by: SNVT_therm_mode

```

TYPE E_LON_therm_mode_t :
(
eLON_THERM_NUL      := -1,
eLON_THERM_NO_CONTROL := 0,
eLON_THERM_IN_OUT   := 1,
eLON_THERM_MODULATING := 2
)
END_TYPE

```


eLON_THERM_NUL: Invalid value.

eLON_THERM_NO_CONTROL: Thermostat disabled.

eLON_THERM_IN_OUT: Cut in/out control.

eLON_THERM_MODULATING: Modulating control.

7.3.62 E_LON_tilt_dir_t

Used by: SNVT_ptz

```
TYPE E_LON_tilt_dir_t :
(
  eLON_TILT_NUL      := -1,
  eLON_TILT_STOP    := 0,
  eLON_TILT_UP      := 1,
  eLON_TILT_DOWN    := 2
)
END_TYPE
```

eLON_TILT_NUL: Invalid value

eLON_TILT_STOP: Stop tilting.

eLON_TILT_UP: Tilt up.

eLON_TILT_DOWN: Tilt down.

7.3.63 E_LON_unit_temp_t

Used by: SNVT_pump_sensor

```
TYPE E_LON_unit_temp_t :
(
  eLON_TEMP_NUL          := -1,
  eLON_TEMP_INACTIVE    := 0,
  eLON_TEMP_AT_DESIRED  := 1,
  eLON_TEMP_TOO_HOT     := 2,
  eLON_TEMP_TOO_COLD   := 3
)
END_TYPE
```

eLON_TEMP_NUL: The status of the apparatus or unit is unknown, or not applicable (Invalid Value).

eLON_TEMP_INACTIVE: The temperature-sensing apparatus is present, but not currently operating.

eLON_TEMP_AT_DESIRED: The unit temperature is within the desired range.

eLON_TEMP_TOO_HOT: The unit temperature is above the upper limit of the desired range.

eLON_TEMP_TOO_COLD: The unit temperature is below the lower limit of the desired range.

7.3.64 E_LON_valve_mode_t

Used by: SNVT_valve_mode

```
TYPE E_LON_valve_mode_t :
(
  eLON_VALVE_NUL          := -1,
  eLON_VALVE_NORMAL      := 0,
  eLON_VALVE_COOLING     := 1,
  eLON_VALVE_HEATING     := 2,
  eLON_VALVE_EMERGENCY   := 3,
  eLON_VALVE_STROKE_ADP  := 4,
  eLON_VALVE_STROKE_SYN  := 5,
  eLON_VALVE_ERROR       := 6,
  eLON_VALVE_OVERRIDDEN  := 7
)
END_TYPE
```

eLON_VALVE_NUL: Invalid value.

eLON_VALVE_NORMAL: Valve works as normal valve.

eLON_VALVE_COOLING: Valve works as cooling valve only.

eLON_VALVE_HEATING: Valve works as heating valve only.

eLON_VALVE_EMERGENCY: Valve works in emergency operation.

eLON_VALVE_STROKE_ADP: Valve adapt its stroke and its end positions.

eLON_VALVE_STROKE_SYN: Valve re-synchronizes its position.

eLON_VALVE_ERROR: Valve is in error mode.

eLON_VALVE_OVERRIDDEN: Value is overridden.

7.3.65 E_LON_zoom_t

Used by: SNVT_ptz

```

TYPE E_LON_zoom_t :
(
  eLON_ZOOM_NUL   := -1,
  eLON_ZOOM_STOP  := 0,
  eLON_ZOOM_TELE  := 1,
  eLON_ZOOM_WIDE  := 2
)
END_TYPE

```

eLON_ZOOM_NUL: Invalid Value.

eLON_ZOOM_STOP: Stop zooming.

eLON_ZOOM_TELE: Telephoto zoom / zoom in.

eLON_ZOOM_WIDE: Wide zoom / zoom out.

7.3.66 AuxiliaryStructure

SNVT_chlr_status

Data types	Description
ST_LON_chlr_state [▶ 533]	Used by: SNVT_chlr_status

SNVT_clothes_w_c

Data types	Description
ST_LON_action [▶ 533]	Used by: SNVT_clothes_w_c
ST_LON_dry [▶ 533]	Used by: SNVT_clothes_w_c
ST_LON_duration [▶ 534]	Used by: SNVT_clothes_w_c
ST_LON_function [▶ 534]	Used by: SNVT_clothes_w_c
ST_LON_rinse [▶ 534]	Used by: SNVT_clothes_w_c
ST_LON_spin [▶ 534]	Used by: SNVT_clothes_w_c
ST_LON_wash [▶ 535]	Used by: SNVT_clothes_w_c

SNVT_clothes_w_s

Data types	Description
ST_LON_alarm [▶ 535]	Used by: SNVT_clothes_w_s

SNVT_color_2

Data types	Description
ST_LON_CIE1931_lumen [► 537]	Used by: SNVT_color_2
ST_LON_CIE1931_percent [► 537]	Used by: SNVT_color_2
ST_LON_color_value [► 537]	Used by: SNVT_color_2
ST_LON_RGB [► 538]	Used by: SNVT_color_2

SNVT_ctrl_resp

Data types	Description
ST_LON_range [► 538]	Used by: SNVT_ctrl_resp
ST_LON_sender [► 538]	Used by: SNVT_ctrl_resp

SNVT_dev_fault

Data types	Description
ST_LON_Dev_type1 [► 539]	Used by: SNVT_dev_fault
ST_LON_pump_ctrl1 [► 539]	Used by: SNVT_dev_fault
ST_LON_valve_pos1 [► 540]	Used by: SNVT_dev_fault

SNVT_dev_maint

Data types	Description
ST_LON_Dev_type2 [► 541]	Used by: SNVT_dev_maint
ST_LON_pump_ctrl2 [► 541]	Used by: SNVT_dev_maint
ST_LON_valve_pos2 [► 541]	Used by: SNVT_dev_maint

SNVT_dev_status

Data types	Description
ST_LON_Dev_type3 [► 542]	Used by: SNVT_dev_status
ST_LON_pump_ctrl3 [► 542]	Used by: SNVT_dev_status
ST_LON_valve_pos3 [► 543]	Used by: SNVT_dev_status

SNVT_ex_control

Data types	Description
ST_LON_Control_device_addr [► 544]	Used by: SNVT_ex_control

SNVT_file_req

Data types	Description
ST_LON_addr [► 545]	Used by: SNVT_file_req
ST_LON_dest_address [► 545]	Used by: SNVT_file_req
ST_LON_gp [► 545]	Used by: SNVT_file_req
ST_LON_sn [► 545]	Used by: SNVT_file_req

SNVT_file_status

Data types	Description
ST_LON_address [► 546]	Used by: FB_Write_Address_Table / FB_Read_Address_Table
ST_LON_adr [► 546]	Used by: SNVT_file_status
ST_LON_descriptor [► 547]	Used by: SNVT_file_status

SNVT_lamp_status

Data types	Description
ST_LON_Alarm_actual [► 547]	Used by: SNVT_lamp_status
ST_LON_alarm_previous [► 549]	Used by: SNVT_lamp_status

SNVT_pos_ctrl

Data types	Description
ST_LON_abspos [► 550]	Used by: SNVT_pos_ctrl
ST_LON_Value [► 551]	Used by: SNVT_pos_ctrl

SNVT_rac_ctrl

Data types	Description
ST_LON_addr_dest [► 551]	Used by: SNVT_rac_req / SNVT_rac_ctrl
ST_LON_addr_init [► 551]	Used by: SNVT_rac_req / SNVT_rac_ctrl
ST_LON_addr_talk [► 552]	Used by: SNVT_rac_ctrl
ST_LON_p2m [► 552]	Used by: SNVT_rac_req / SNVT_rac_ctrl
ST_LON_p2p [► 552]	Used by: SNVT_rac_req / SNVT_rac_ctrl

SNVT_rac_req

Data types	Description
ST_LON_rac_req_addr_dest [► 553]	
ST_LON_rac_req_addr_init [► 553]	

SNVT_switch_2

Data types	Description
ST_LON_setting [► 553]	Used by: SNVT_switch_2

SNVT_time_zone

Data types	Description
ST_LON_end_DST [► 554]	Used by: SNVT_time_zone
ST_LON_M_end_DST [► 554]	Used by: SNVT_time_zone
ST_LON_M_start_DST [► 555]	Used by: SNVT_time_zone
ST_LON_start_DST [► 555]	Used by: SNVT_time_zone

7.3.66.1 SNVT_chlr_status

Data types	Description
ST_LON_chlr_state [► 533]	Used by: SNVT_chlr_status

7.3.66.1.1 ST_LON_chlr_state

Used by: SNVT_chlr_status

```

TYPE ST_LON_chlr_state :
STRUCT
  bIn_alarm      : BOOL;
  bRun_enabled   : BOOL;
  bLocal         : BOOL;
  bLimited       : BOOL;
  bChw_flow     : BOOL;
  bCondw_flow    : BOOL;
END_STRUCT
END_TYPE
    
```

bIn_alarm: Alarm flag (boolean).

bRun_enabled: Run-enabled flag (boolean).

bLocal: Locally-controlled flag (boolean).

bLimited: Limited-condition flag (boolean). Conditions may exist that prevent reaching the setpoint

bChw_flow: Chiller-water-flow flag (boolean).

bCondw_flow: Condenser-water-flow flag (boolean).

7.3.66.2 SNVT_clothes_w_c

Data types	Description
ST_LON_action [► 533]	Used by: SNVT_clothes_w_c
ST_LON_dry [► 533]	Used by: SNVT_clothes_w_c
ST_LON_duration [► 534]	Used by: SNVT_clothes_w_c
ST_LON_function [► 534]	Used by: SNVT_clothes_w_c
ST_LON_rinse [► 534]	Used by: SNVT_clothes_w_c
ST_LON_spin [► 534]	Used by: SNVT_clothes_w_c
ST_LON_wash [► 535]	Used by: SNVT_clothes_w_c

7.3.66.2.1 ST_LON_action

Used by: SNVT_clothes_w_c

```

TYPE ST_LON_action :
STRUCT
  bPower_on      : BOOL;
  bRun_mode      : BOOL;
  byRsrvd2_7     : BYTE;
END_STRUCT
END_TYPE
    
```

bPower_on:

bRun_mode:

byRsrvd2_7:

7.3.66.2.2 ST_LON_dry

Used by: SNVT_clothes_w_c

```

TYPE ST_LON_dry :
STRUCT
  byTemp         : BYTE;
  stDuration     : ST_LON_Duration;
END_STRUCT
END_TYPE
    
```

byTemp: Min: 0 / Max: 1.

stDuration: (see [ST_LON_Duration](#) [[▶ 534](#)])

7.3.66.2.3 ST_LON_duration

Used by: SNVT_clothes_w_c

```
TYPE ST_LON_duration :
STRUCT
  byTime      : BYTE;
  eDryness    : E_LON_discrete_levels_t;
END_STRUCT
END_TYPE
```

byTime: Min: 0 / Max: 255

eDryness: see [E_LON_discrete_levels_t](#) [[▶ 500](#)]

7.3.66.2.4 ST_LON_function

Used by: SNVT_clothes_w_c

```
TYPE ST_LON_function :
STRUCT
  eProgram    : E_LON_appl_cwp_t;
  stWash      : ST_LON_wash;
  stRinse     : ST_LON_rinse;
  stSpin      : ST_LON_spin;
  stDry       : ST_LON_dry;
END_STRUCT
END_TYPE
```

eProgram: (see [E_LON_appl_cwp_t](#) [[▶ 490](#)])

stWash: (see [ST_LON_wash](#) [[▶ 535](#)])

stRinse: (see [ST_LON_rinse](#) [[▶ 534](#)])

stSpin: (see [ST_LON_spin](#) [[▶ 534](#)])

stDry: (see [ST_LON_dry](#) [[▶ 533](#)])

7.3.66.2.5 ST_LON_rinse

Used by: SNVT_clothes_w_c

```
TYPE ST_LON_rinse :
STRUCT
  byTemp      : BYTE;
  byRepeat    : BYTE;
  eOption     : E_LON_appl_rin_t;
END_STRUCT
END_TYPE
```

byTemp: Min: 0 / Max: 255

byRepeat: Min: 0 / Max: 9

eOption: see [E_LON_appl_rin_t](#) [[▶ 491](#)]

7.3.66.2.6 ST_LON_spin

Used by: SNVT_clothes_w_c

```
TYPE ST_LON_spin :
STRUCT
  uiSpeed     : UINT;
  byTime      : BYTE;
END_STRUCT
```

```
eHold      : E_LON_boolean_t;
END_STRUCT
END_TYPE
```

uiSpeed: Min: 0 / Max: 65535

byTime: Min: 0 / Max: 255

eHold: see [E_LON_boolean_t \[▶ 492\]](#)

7.3.66.2.7 ST_LON_wash

Used by: SNVT_clothes_w_c

```
TYPE ST_LON_wash :
STRUCT
  eLoad_level  : E_LON_discrete_levels_t;
  byTemp       : BYTE;
  byTime       : BYTE;
  ePrewash     : E_LON_boolean_t;
END_STRUCT
END_TYPE
```

eLoad_level: see [E_LON_discrete_levels_t \[▶ 500\]](#)

byTemp: Min: 0 / Max: 255

byTime: Min: 0 / Max: 255

ePrewash: see [E_LON_boolean_t \[▶ 492\]](#)

7.3.66.3 SNVT_clothes_w_s

Data types	Description
ST_LON_alarm [▶ 535]	Used by: SNVT_clothes_w_s

7.3.66.3.1 ST_LON_alarm

Used by: SNVT_clothes_w_s

```
TYPE ST_LON_alarm :
STRUCT
  bAlarm_reset      : BOOL;
  bWar_water_supply : BOOL;
  bWar_drain_slow   : BOOL;
  bWar_door_open    : BOOL;
  bWar_load_unbalanced : BOOL;
  bWar_filter_cleaning : BOOL;
  bWar_hoses_reversed : BOOL;
  bWar_voltage_low  : BOOL;
  bWar_power_failure : BOOL;
  bWar_drain_open   : BOOL;
  bWar_execute_fail : BOOL;
  bWar_door_locked  : BOOL;
  bWar_service      : BOOL;
  bWar_rsrvd5       : BOOL;
  bWar_rsrvd6       : BOOL;
  bWar_rsrvd7       : BOOL;
  bErr_motor_stall  : BOOL;
  bErr_water_temp   : BOOL;
  bErr_pressure     : BOOL;
  bErr_overflow     : BOOL;
  bErr_water_heat   : BOOL;
  bErr_water_leak   : BOOL;
  bErr_motor_speed  : BOOL;
  bErr_wash_thermistor : BOOL;
  bErr_dry_thermistor : BOOL;
  bErr_dry_overheat : BOOL;
  bErr_dry_heating  : BOOL;
  bErr_dry_fan      : BOOL;
  bErr_rsrvd4       : BOOL;
  bErr_rsrvd5       : BOOL;
```

```
bErr_rsrvd6      : BOOL;  
bErr_rsrvd7      : BOOL;  
byErr_rsrvd0_7   : BYTE;  
byErr_manuf_code : BYTE;  
END_STRUCT  
END_TYPE
```

bAlarm_reset:

bWar_water_supply:

bWar_drain_slow:

bWar_door_open:

bWar_load_unbalanced:

bWar_filter_cleaning:

bWar_hoses_reversed:

bWar_voltage_low:

bWar_power_failure:

bWar_drain_open:

bWar_execute_fail:

bWar_door_locked:

bWar_service:

bWar_rsrvd5:

bWar_rsrvd6:

bWar_rsrvd7:

bErr_motor_stall:

bErr_water_temp:

bErr_pressure:

bErr_overflow:

bErr_water_heat:

bErr_water_leak:

bErr_motor_speed:

bErr_wash_thermistor:

bErr_dry_thermistor:

bErr_dry_overheat:

bErr_dry_heating:

bErr_dry_fan:

bErr_rsrvd4:

bErr_rsrvd5:

bErr_rsrvd6:

bErr_rsrvd7:

byErr_rsrvd0_7:

byErr_manuf_code: Min: 0 / Max: 255

7.3.66.4 SNVT_color_2

Data types	Description
ST_LON_CIE1931_lumen [▶ 537]	Used by: SNVT_color_2
ST_LON_CIE1931_percent [▶ 537]	Used by: SNVT_color_2
ST_LON_color_value [▶ 537]	Used by: SNVT_color_2
ST_LON_RGB [▶ 538]	Used by: SNVT_color_2

7.3.66.4.1 ST_LON_CIE1931_lumen

Used by: SNVT_color_2

```
TYPE ST_LON_CIE1931_lumen :
STRUCT
  rX          : REAL;
  rY          : REAL;
  udiAbsolute_Y : UDINT;
END_STRUCT
END_TYPE
```

rX: Min: 0.0 / Max: 0.740 / Invalid: 1.275 / CIE 1931 x value (CIE 1931 color space coordinate). CIE 1931 x-axis color value

rY: Min: 0.0 / Max: 0.840 / Invalid: 1.275 / CIE 1931 y value (CIE 1931 color space coordinate). CIE 1931 y-axis color value

udiAbsolute_Y: Min: 0 / Max: 6553400 / Invalid: 6553500 / Absolute luminance (lumen). Absolute luminance

7.3.66.4.2 ST_LON_CIE1931_percent

Used by: SNVT_color_2

```
TYPE ST_LON_CIE1931_percent :
STRUCT
  rX          : REAL;
  rY          : REAL;
  rPercent_Y  : REAL;
END_STRUCT
END_TYPE
```

rX: Min: 0.0 / Max: 0.740 / Invalid: 1.275 / CIE 1931 x value (CIE 1931 color space coordinate). CIE 1931 x-axis color value

rY: Min: 0.0 / Max: 0.840 / Invalid: 1.275 / CIE 1931 y value (CIE 1931 color space coordinate). CIE 1931 y-axis color value

rPercent_Y: Min: 0.0 / Max: 100.0 / Invalid: 655.35 / Luminance (% of full level). Y output in percent of maximum lumen output of the lamp

7.3.66.4.3 ST_LON_color_value

Used by: SNVT_color_2

```
TYPE ST_LON_color_value :
STRUCT
  stCIE1931_lumen      : ST_LON_CIE1931_lumen;
  stCIE1931_percent    : ST_LON_CIE1931_percent;
  stRGB                : ST_LON_RGB;
  uiColor_temperature  : UINT;
END_STRUCT
END_TYPE
```

stCIE1931_lumen: CIE 1931 color space with lumen. CIE 1931 color space with Y output in lumen (see [ST_LON_CIE1931_lumen](#) [[▶ 537](#)]).

stCIE1931_percent: CIE 1931 color space with percent. CIE 1931 color space with Y output in percent of maximum lumen output of the lamp (see [ST_LON_CIE1931_percent](#) [[▶ 537](#)]).

stRGB: RGB color value (see [ST_LON_RGB](#) [[▶ 538](#)]).

uiColor_temperature: Min: 2800 / Max: 7500 / Invalid: 12750 / Color temperature (degrees Kelvin).

7.3.66.4 ST_LON_RGB

Used by: SNVT_color_2

```
TYPE ST_LON_RGB :
STRUCT
  byRed      : BYTE;
  byGreen    : BYTE;
  byBlue     : BYTE;
END_STRUCT
END_TYPE
```

byRed: Min: 0 / Max: 250 / Red component. Red component for RGB color

byGreen: Min: 0 / Max: 250 / Green component. Green component for RGB color

byBlue: Min: 0 / Max: 250 / Blue component. Blue component for RGB color

7.3.66.5 SNVT_ctrl_resp

Data types	Description
ST_LON_range [▶ 538]	Used by: SNVT_ctrl_resp
ST_LON_sender [▶ 538]	Used by: SNVT_ctrl_resp

7.3.66.5.1 ST_LON_range

Used by: SNVT_ctrl_resp

```
TYPE ST_LON_range :
STRUCT
  uiLower    : UINT;
  uiUpper    : UINT;
END_STRUCT
END_TYPE
```

uiLower: Min: 1 / Max: 65535 / Invalid: 65535 / Sender range lower ID (ID number).

uiUpper: Min: 1 / Max: 65535 / Invalid: 65535 / Sender range upper ID (ID number).

7.3.66.5.2 ST_LON_sender

Used by: SNVT_ctrl_resp

```
TYPE ST_LON_sender :
STRUCT
  uiID       : UINT;
  stRange    : ST_LON_range;
END_STRUCT
END_TYPE
```

uiID: Min: 1 / Max: 65535 / Invalid: 65535 / Sender ID (ID number).

stRange: Sender ID range (lower, upper) (see [ST_LON_range](#) [[▶ 538](#)]).

7.3.66.6 SNVT_dev_fault

Data types	Description
ST_LON_Dev_type1 [▶ 539]	Used by: SNVT_dev_fault

Data types	Description
ST_LON_pump_ctrl1 [▶ 539]	Used by: SNVT_dev_fault
ST_LON_valve_pos1 [▶ 540]	Used by: SNVT_dev_fault

7.3.66.6.1 ST_LON_Dev_type1

Used by: SNVT_dev_fault

```

TYPE ST_LON_Dev_type1 :
STRUCT
  stPump_ctrl   : ST_LON_pump_ctrl1;
  stValvePos    : ST_LON_valve_pos1;
END_STRUCT
END_TYPE
    
```

stPump_ctrl: Pump controller device fault information (see [ST_LON_pump_ctrl1 \[▶ 539\]](#)).

stValvePos: Valve positioner device fault information (see [ST_LON_valve_pos1 \[▶ 540\]](#)).

7.3.66.6.2 ST_LON_pump_ctrl1

Used by: SNVT_dev_fault

```

TYPE ST_LON_pump_ctrl1 :
STRUCT
  bSf_voltage_low      : BOOL;
  bSf_voltage_high     : BOOL;
  bSf_phase             : BOOL;
  bSf_no_fluid         : BOOL;
  bSf_press_low        : BOOL;
  bSf_press_high       : BOOL;
  bSf_general_fault    : BOOL;
  bSf_reserved1_7      : BOOL;
  bDf_motor_temp       : BOOL;
  bDf_motor_failure    : BOOL;
  bDf_pump_blocked     : BOOL;
  bDf_elect_temp       : BOOL;
  bDf_elect_failure_nf : BOOL;
  bDf_elect_failure    : BOOL;
  bDf_sensor_failure   : BOOL;
  bDf_general_fault    : BOOL;
  byReserved3_0_7     : BYTE;
END_STRUCT
END_TYPE
    
```

bSf_voltage_low: Supply fault - low voltage (boolean). Supply voltage is too low.

bSf_voltage_high: Supply fault - high voltage (boolean). Supply voltage is too high.

bSf_phase: Supply fault - power phase (boolean). Supply power is missing phase.

bSf_no_fluid: Supply fault - no fluid (boolean). There is no fluid in the pump.

bSf_press_low: Supply fault - low pressure (boolean). System pressure is too low

bSf_press_high: Supply fault - high pressure (boolean). System pressure is too high

bSf_general_fault: General supply fault.

bSf_reserved1_7:

bDf_motor_temp: Device fault - motor temperature (boolean). Motor temperature is too high

bDf_motor_failure: Device fault - motor fatal failure (boolean). Motor has encountered a fatal failure

bDf_pump_blocked: Device fault - pump blocked (boolean). Pump is presently blocked

bDf_elect_temp: Device fault - electronics temperature (boolean). Temperature of the electronic circuitry is too high

bDf_elect_failure_nf: Device fault - electronics failure (boolean). Electronic circuitry has encountered a non-fatal failure

bDf_elect_failure: Device fault - electronics fatal failure (boolean). Electronic circuitry has encountered a fatal failure

bDf_sensor_failure: Device fault - sensor failure (boolean). Sensor has failed on the device

bDf_general_fault: General device fault.

byReserved3_0_7:

7.3.66.6.3 ST_LON_valve_pos1

Used by: SNVT_dev_fault

```

TYPE ST_LON_valve_pos1 :
STRUCT
  bDf_valve_blocked           : BOOL;
  bDf_blocked_direction_open  : BOOL;
  bDf_blocked_direction_close : BOOL;
  bDf_position_error         : BOOL;
  bDf_stroke_Out_of_range    : BOOL;
  bDf_initialization         : BOOL;
  bDf_vibration_cavitation    : BOOL;
  bDf_ed_too_high            : BOOL;
  byReserved1_0_2           : BYTE;
  bEe_oscillating            : BOOL;
  bEe_valve_too_large        : BOOL;
  bEe_valve_too_small        : BOOL;
  byReserved2_6_7           : BYTE;
  bReserved3_0_7            : BOOL;
  bSf_voltage_Out_of_range   : BOOL;
  bSf_electronic_high_temp   : BOOL;
  bSf_frictional_resistance   : BOOL;
  byReserved4_4_6           : BYTE;
  bGeneral_fault             : BOOL;
END_STRUCT
END_TYPE

```

bDf_valve_blocked: Device fault - valve blocked. The valve is presently blocked.

bDf_blocked_direction_open: Device fault - blocked direction open. The device is blocked while attempting to open.

bDf_blocked_direction_close: Device fault - blocked direction close. The device is blocked while attempting to close

bDf_position_error: Device fault - position error. The valve position is not correct

bDf_stroke_Out_of_range: Device fault - stroke out of range. The valve stroke is out of operating range

bDf_initialization: Device fault - initialization error. There was an error during initialization of the device

bDf_vibration_cavitation: Device fault - vibration / cavitation. There are excessive vibrations or cavitations detected

bDf_ed_too_high: Device fault - ED too high. The ED is too high

byReserved1_0_2: This field is reserved.. This field is reserved.

bEe_oscillating: Engineering error - oscillating. There is an oscillating error

bEe_valve_too_large: Engineering error - valve too big. The valve size is too large

bEe_valve_too_small: Engineering error - valve too small. The valve size is too small

byReserved2_6_7: This field is reserved.. This field is reserved.

bReserved3_0_7: This field is reserved.. This field is reserved.

bSf_voltage_Out_of_range: Supply fault - voltage out of range. The voltage is out of the specified acceptable range

bSf_electronic_high_temp: Supply fault - electronics temperature. The temperature of the electronics is too high

bSf_frictional_resistance: Supply fault - frictional resistance. Resistance due to friction is detected

byReserved4_4_6: This field is reserved.. This field is reserved.

bGeneral_fault: General Fault. A General Fault has occurred. Please consult the documentation or contact the valve-controller manufacturer.

7.3.66.7 SNVT_dev_maint

Data types	Description
ST_LON_Dev_type2 [▶ 541]	Used by: SNVT_dev_maint
ST_LON_pump_ctrl2 [▶ 541]	Used by: SNVT_dev_maint
ST_LON_valve_pos2 [▶ 541]	Used by: SNVT_dev_maint

7.3.66.7.1 ST_LON_Dev_type2

Used by: SNVT_dev_maint

```

TYPE ST_LON_Dev_type2 :
STRUCT
  stPump_ctrl1   : ST_LON_pump_ctrl2;
  stValvePos     : ST_LON_valve_pos2;
END_STRUCT
END_TYPE
    
```

stPump_ctrl1: Pump controller device maintenance state (see [ST_LON_pump_ctrl2 \[\[▶ 541\]\(#\)\]](#)).

stValvePos: Valve positioner device maintenance information (see [ST_LON_valve_pos2 \[\[▶ 541\]\(#\)\]](#)).

7.3.66.7.2 ST_LON_pump_ctrl2

Used by: SNVT_dev_maint

```

TYPE ST_LON_pump_ctrl2 :
STRUCT
  bService_required      : BOOL;
  bBearings_change       : BOOL;
  bBearings_lubricate     : BOOL;
  bShaftseal_change      : BOOL;
  byReserved1_4_7        : BYTE;
  byReserved2_0_7        : BYTE;
  byReserved3_0_7        : BYTE;
END_STRUCT
END_TYPE
    
```

bService_required: Service required (boolean) . Service/maintenance is required

bBearings_change: Change bearings (boolean) . Bearings need to be replaced

bBearings_lubricate: Lubricate bearings (boolean) . Bearings need to be greased

bShaftseal_change: Change shaft seal (boolean) . Seal on the shaft needs to be replaced

byReserved1_4_7: Reserve

byReserved2_0_7: Reserve

byReserved3_0_7: Reserve

7.3.66.7.3 ST_LON_valve_pos2

Used by: SNVT_dev_maint

```

TYPE ST_LON_valve_pos2 :
STRUCT
  bMotor_maint      : BOOL;
  bPacking_change   : BOOL;
  bElectronics_check : BOOL;
  bPositioning_check : BOOL;
  bLubrication_check : BOOL;
  bReturn_check     : BOOL;
  battery_check     : BOOL;
  bReserved1_7      : BOOL;
  byReserved2_0_7   : BYTE;
  byReserved3_0_6   : BYTE;
  bGeneral_maint    : BOOL;
END_STRUCT
END_TYPE

```

bMotor_maint: Motor Maintenance. The motor requires servicing

bPacking_change: Packing Change. The packing needs to be controlled or changed

bElectronics_check: Check Electronics. The electronics need to be checked (temperature too high)

bPositioning_check: Check Position. The positioning needs to be checked (mechanical or electronic)

bLubrication_check: Check Lubrication. The lubrication need to be checked

bReturn_check: Check Spring-Return Function. The spring-return function needs to be checked

battery_check: Check battery. The battery needs to be checked

bReserved1_7: This field is reserved.. This field is reserved.

byReserved2_0_7: This field is reserved.. This field is reserved.

byReserved3_0_6: This field is reserved.. This field is reserved.

bGeneral_maint: General Maintenance. General Maintenance needs to be performed. Please consult the documentation or your Maintenance Department.

7.3.66.8 SNVT_dev_status

Data types	Description
ST_LON_Dev_type3 [▶ 542]	Used by: SNVT_dev_status
ST_LON_pump_ctrl3 [▶ 542]	Used by: SNVT_dev_status
ST_LON_valve_pos3 [▶ 543]	Used by: SNVT_dev_status

7.3.66.8.1 ST_LON_Dev_type3

Used by: SNVT_dev_status

```

TYPE ST_LON_Dev_type3 :
STRUCT
  stPump_ctrl  : ST_LON_pump_ctrl3;
  stValvePos   : ST_LON_valve_pos3;
END_STRUCT
END_TYPE

```

stPump_ctrl: Pump controller device status (see [ST_LON_pump_ctrl3](#) [[▶ 542](#)]).

stValvePos: Valve positioner device status (see [ST_LON_valve_pos3](#) [[▶ 543](#)]).

7.3.66.8.2 ST_LON_pump_ctrl3

Used by: SNVT_dev_status

```

TYPE ST_LON_pump_ctrl3 :
STRUCT
  bDevice_fault   : BOOL;
  bSupply_fault   : BOOL;

```

```

bReserved1_2      : BOOL;
bSpeed_low       : BOOL;
bSpeed_high      : BOOL;
bReserved1_5     : BOOL;
bSetpt_Out_of_range : BOOL;
bReserved1_7     : BOOL;
bLocal_control   : BOOL;
bReserved2_1     : BOOL;
bRunning         : BOOL;
bReserved2_3     : BOOL;
bRemote_press    : BOOL;
bRemote_flow     : BOOL;
bRemote_temp     : BOOL;
bReserved2_7     : BOOL;
byReserved3_0_7  : BYTE;
END_STRUCT
END_TYPE

```

bDevice_fault: Pump controller fault (boolean). See SNVT_pump_fault network variable declaration on device

bSupply_fault: Supply fault (boolean). No electrical power, no fluid in pump, etc. See SNVT_pump_fault network variable declaration on device.

bReserved1_2:

bSpeed_low: Low-speed limit of pump (boolean). Pump is running at the lowest possible speed, therefore the requested performance is not possible.

bSpeed_high: High-speed limit of pump (boolean). Pump is running at the highest possible speed, therefore the requested performance is not possible.

bReserved1_5:

bSetpt_Out_of_range: Setpoint out of range (boolean). Chosen override setpoint value is lower than the manufacturer-defined low-setpoint limit or higher than the manufacturer-defined high-setpoint limit.

bReserved1_7:

bLocal_control: Locally controlled pump (boolean). Pump is locally operated (hardware override)

bReserved2_1:

bRunning: Running pump (boolean). Pump is presently running

bReserved2_3:

bRemote_press: Remote pressure sensor (boolean). Pump controller is using a remote pressure sensor

bRemote_flow: Remote flow sensor (boolean). Pump controller is using a remote flow sensor

bRemote_temp: Remote temperature sensor (boolean). Pump controller is using a remote temperature sensor

bReserved2_7:

byReserved3_0_7:

7.3.66.8.3 ST_LON_valve_pos3

Used by: SNVT_dev_status

```

TYPE ST_LON_valve_pos3 :
STRUCT
  bRunning          : BOOL;
  bAdapting         : BOOL;
  bInitializing     : BOOL;
  bLocal_control    : BOOL;
  bSetpt_Out_of_range : BOOL;
  bRemote_ctrl_signal : BOOL;
  byReserved1_6_7   : BYTE;
  bHw_emergency     : BOOL;
  bSw_emergency     : BOOL;

```

```

    byReserved2_2_7      : BYTE;
    byReserved3_0_7      : BYTE;
END_STRUCT
END_TYPE

```

bRunning: Valve Running. Valve is presently being positioned.

bAdapting: Adapting. Valve is presently adapting.

bInitializing: Initializing. Valve is presently initializing.

bLocal_control: Local Control. The valve operation is being locally controlled.

bSetpt_Out_of_range: Setpoint out of range. Chosen override setpoint value is lower than the manufacturer-defined low-setpoint limit or higher than the manufacturer-defined high-setpoint limit.

bRemote_ctrl_signal: Remote Control Signal. The remote-control signal is active.

byReserved1_6_7: This field is reserved.. This field is reserved.

bHw_emergency: Hardware Emergency. The hardware-emergency state is active

bSw_emergency: Software Emergency. The software-emergency state is active

byReserved2_2_7: This field is reserved.. This field is reserved.

byReserved3_0_7: This field is reserved.. This field is reserved.

7.3.66.9 SNVT_ex_control

Data types	Description
ST_LON_Control_device_addr [► 544]	Used by: SNVT_ex_control

7.3.66.9.1 ST_LON_Control_device_addr

Used by: SNVT_ex_control

```

TYPE ST_LON_Control_device_addr :
STRUCT
    arrDomain_id      : ARRAY [0..5] OF BYTE;
    byDomain_length    : BYTE;
    bySubnet           : BYTE;
    byNode             : BYTE;
END_STRUCT
END_TYPE

```

arrDomain_id: Domain ID (array of 6 bytes). ANSI/CEA-709.1 domain ID

byDomain_length: Domain length (ANSI/CEA-709.1 domain length). Valid domain lengths are 0, 1, 3, and 6.

bySubnet: Min: 1 / Max: 255 / Subnet (subnet number). There can be 255 subnets (1-255) in a domain.

byNode: Min: 1 / Max: 127 / Node (node number). There can be 127 nodes (1-127) in a subnet.

7.3.66.10 SNVT_file_req

Data types	Description
ST_LON_addrt [► 545]	Used by: SNVT_file_req
ST_LON_dest_address [► 545]	Used by: SNVT_file_req
ST_LON_gp [► 545]	Used by: SNVT_file_req
ST_LON_sn [► 545]	Used by: SNVT_file_req

7.3.66.10.1 ST_LON_addrt

Used by: SNVT_file_req

```
TYPE ST_LON_addrt :
STRUCT
  byType      : BYTE;
  uiIndex     : UINT;
END_STRUCT
END_TYPE
```

byType: Min: 0 / Max: 33 / Address type (8-bit unsigned value). The address-table address type is 33 (0x21).

uiIndex: Min: 0 / Max: 65535 / Address table index (16-bit unsigned value).

7.3.66.10.2 ST_LON_dest_address

Used by: SNVT_file_req

```
TYPE ST_LON_dest_address :
STRUCT
  stAddr      : ST_LON_addrt;
  stSn        : ST_LON_sn;
  stGp        : ST_LON_gp;
END_STRUCT
END_TYPE
```

stAddr: Address table entry. ANSI/CEA-709.1 address in device's internal address table entry (see [ST_LON_addrt](#) [► 545]).

stSn: Subnet-node address (LonWorks subnet-node address) (see [ST_LON_sn](#) [► 545]).

stGp: Group address (LonWorks group address) (see [ST_LON_gp](#) [► 545]).

7.3.66.10.3 ST_LON_gp

Used by: SNVT_file_req

```
TYPE ST_LON_gp :
STRUCT
  byType      : BYTE;
  bySize      : BYTE;
  bDomain     : BOOL;
  byUnused    : BYTE;
  byRetry     : BYTE;
  byTx_timer  : BYTE;
  byGroup     : BYTE;
END_STRUCT
END_TYPE
```

byType: Min: 0 / Max: 1 / Address type (boolean). The group address type is 1.

bySize: Min: 0 / Max: 65 / Size (LonWorks group size). An acknowledged group can have from 0-64 addressees, plus the sender.

bDomain: Min: 0 / Max: 1 / Domain (LonWorks domain index).

byUnused: Unused field. This field is reserved.

byRetry: Min: 0 / Max: 15 / Retry count (number of retries).

byTx_timer: Min: 0 / Max: 15 / Transaction timer (timer code value).

byGroup: Min: 0 / Max: 255 / Group. There can be 256 groups (0-255) in a domain.

7.3.66.10.4 ST_LON_sn

Used by: SNVT_file_req

```

TYPE ST_LON_sn :
STRUCT
  byType      : BYTE;
  bDomain     : BOOL;
  byNode      : BYTE;
  byRetry     : BYTE;
  byTx_timer  : BYTE;
  bySubnet    : BYTE;
END_STRUCT
END_TYPE

```

byType: Min: 1 / Max: 1 / Address type (8-bit unsigned value). The subnet-node address type is 1.

bDomain: Domain (LonWorks domain index).

byNode: Min: 0 / Max: 127 / Node (node number). There can be 127 nodes (1-127) in a subnet.

byRetry: Min: 0 / Max: 15 / Retry count (number of retries).

byTx_timer: Min: 0 / Max: 15 / Transaction timer (timer code value).

bySubnet: Min: 0 / Max: 255 / Subnet (subnet number). There can be 255 subnets (1-255) in a domain.

7.3.66.11 SNVT_file_status

Data types	Description
<u>ST_LON_address</u> [► 546]	Used by: FB_Write_Address_Table / FB_Read_Address_Table
<u>ST_LON_adr</u> [► 546]	Used by: SNVT_file_status
<u>ST_LON_descriptor</u> [► 547]	Used by: SNVT_file_status

7.3.66.11.1 ST_LON_address

Used by: FB_Write_Address_Table / FB_Read_Address_Table

```

TYPE ST_LON_address :
STRUCT
  arrDomain_id  : ARRAY [0..5] OF BYTE;
  byDomain_length : BYTE;
  bySubnet      : BYTE;
  byNode        : BYTE;
END_STRUCT
END_TYPE

```

arrDomain_id: Domain ID (array of 6 bytes). ANSI/CEA-709.1 domain ID

byDomain_length: Min: 0 / Max: 6 / Domain length (ANSI/CEA-709.1 domain length). Valid domain lengths are 0, 1, 3, and 6.

bySubnet: Min: 0 / Max: 255 / Subnet (subnet number). There can be 255 subnets (1-255) in a domain.

byNode: Min: 0 / Max: 127 / Node (node number). There can be 127 nodes (1-127) in a subnet.

7.3.66.11.2 ST_LON_adr

Used by: SNVT_file_status

```

TYPE ST_LON_adr :
STRUCT
  stDescriptor : ST_LON_descriptor;
  stAddress    : ST_LON_address;
END_STRUCT
END_TYPE

```

stDescriptor: Descriptor (see ST_LON_descriptor [► 547]).

stAddress: Address (see ST_LON_address [► 546]).

7.3.66.11.3 ST_LON_descriptor

Used by: SNVT_file_status

```

TYPE ST_LON_descriptor :
STRUCT
  sFile_info   : STRING(16);
  udiSize      : UDINT;
  uiType       : UINT;
END_STRUCT
END_TYPE
    
```

sFile_info: File info (array of 16 characters)

udiSize: Min: 0 / Max: 2147483647 / Size (bytes)

uiType: Min: 0 / Max: 65535 / Type

7.3.66.12 SNVT_lamp_status

Data types	Description
ST_LON_Alarm_actual [▶ 547]	Used by: SNVT_lamp_status
ST_LON_alarm_previous [▶ 549]	Used by: SNVT_lamp_status

7.3.66.12.1 ST_LON_Alarm_actual

Used by: SNVT_lamp_status

```

TYPE ST_LON_Alarm_actual :
STRUCT
  bLamp_current_high   : BOOL;
  bLamp_current_low    : BOOL;
  bMain_current_high   : BOOL;
  bMain_current_low    : BOOL;
  bLamp_voltage_high   : BOOL;
  bLamp_voltage_low    : BOOL;
  bMain_voltage_high   : BOOL;
  bMain_voltage_low    : BOOL;
  bPowerfactor_low     : BOOL;
  bOLC_temp_high       : BOOL;
  bPower_high          : BOOL;
  bPower_low           : BOOL;
  bRelay_failure       : BOOL;
  bCap_failure         : BOOL;
  bLamp_failure        : BOOL;
  bBallast_failure     : BOOL;
  bInter_com_failure   : BOOL;
  bExter_com_failure   : BOOL;
  bMain_volt_below_spec : BOOL;
  bLamp_restart_count  : BOOL;
  bFading_ready        : BOOL;
  bBallast_temp_high   : BOOL;
  bDigi_in_A           : BOOL;
  bDigi_in_B           : BOOL;
  bBit_25_res          : BOOL;
  bBit_26_res          : BOOL;
  bBit_27_res          : BOOL;
  bBit_28_res          : BOOL;
  bBit_29_res          : BOOL;
  bBit_30_res          : BOOL;
  bBit_31_res          : BOOL;
  bBit_32_res          : BOOL;
  bBit_33_res          : BOOL;
  bBit_34_res          : BOOL;
  bBit_35_res          : BOOL;
  bBit_36_res          : BOOL;
  bBit_37_res          : BOOL;
  bBit_38_res          : BOOL;
  bBit_39_res          : BOOL;
  bBit_40_res          : BOOL;
END_STRUCT
END_TYPE
    
```

bLamp_current_high: Lamp current too high.

bLamp_current_low: Lamp current too low.
bMain_current_high: Main current too high.
bMain_current_low: Main current too low.
bLamp_voltage_high: Lamp voltage too high.
bLamp_voltage_low: Lamp voltage too low.
bMain_voltage_high: Main voltage too high.
bMain_voltage_low: Main voltage too low.
bPowerfactor_low: Powerfactor too low.
bOLC_temp_high: Temperature too high.
bPower_high: Power too high.
bPower_low: Power too low.
bRelay_failure: Relay failure.
bCap_failure: Capacitor failure.
bLamp_failure: Lamp failure.
bBallast_failure: Ballast failure.
bInter_com_failure: Internal communication failure.
bExter_com_failure: External communication failure.
bMain_volt_below_spec: Main voltage below performance specification.
bLamp_restart_count: Lamp restart retry counter / cycling failure.
bFading_ready: Fading ready.
bBallast_temp_high: Ballast temperature too high.
bDigi_in_A: digital input A active.
bDigi_in_B: digital input B active.
bBit_25_res:
bBit_26_res:
bBit_27_res:
bBit_28_res:
bBit_29_res:
bBit_30_res:
bBit_31_res:
bBit_32_res:
bBit_33_res:
bBit_34_res:
bBit_35_res:
bBit_36_res:
bBit_37_res:

bBit_38_res:

bBit_39_res:

bBit_40_res:

7.3.66.12.2 ST_LON_alarm_previous

Used by: SNVT_lamp_status

```

TYPE ST_LON_alarm_previous :
STRUCT
  bLamp_current_high      : BOOL;
  bLamp_current_low       : BOOL;
  bMain_current_high      : BOOL;
  bMain_current_low       : BOOL;
  bLamp_voltage_high      : BOOL;
  bLamp_voltage_low       : BOOL;
  bMain_voltage_high      : BOOL;
  bMain_voltage_low       : BOOL;
  bPowerfactor_low        : BOOL;
  bOLC_temp_high          : BOOL;
  bPower_high             : BOOL;
  bPower_low              : BOOL;
  bRelay_failure          : BOOL;
  bCap_failure            : BOOL;
  bLamp_failure           : BOOL;
  bBallast_failure        : BOOL;
  bInter_com_failure      : BOOL;
  bExter_com_failure      : BOOL;
  bMain_volt_below_spec   : BOOL;
  bLamp_restart_count     : BOOL;
  bFading_ready           : BOOL;
  bBallast_temp_high      : BOOL;
  bDigi_in_A              : BOOL;
  bDigi_in_B              : BOOL;
  bBit_25_res             : BOOL;
  bBit_26_res             : BOOL;
  bBit_27_res             : BOOL;
  bBit_28_res             : BOOL;
  bBit_29_res             : BOOL;
  bBit_30_res             : BOOL;
  bBit_31_res             : BOOL;
  bBit_32_res             : BOOL;
  bBit_33_res             : BOOL;
  bBit_34_res             : BOOL;
  bBit_35_res             : BOOL;
  bBit_36_res             : BOOL;
  bBit_37_res             : BOOL;
  bBit_38_res             : BOOL;
  bBit_39_res             : BOOL;
  bBit_40_res             : BOOL;
END_STRUCT
END_TYPE

```

bLamp_current_high: Lamp current too high.

bLamp_current_low: Lamp current too low.

bMain_current_high: Main current too high.

bMain_current_low: Main current too low.

bLamp_voltage_high: Lamp voltage too high.

bLamp_voltage_low: Lamp voltage too low.

bMain_voltage_high: Main voltage too high.

bMain_voltage_low: Main voltage too low.

bPowerfactor_low: Powerfactor too low.

bOLC_temp_high: Temperature too high.

bPower_high: Power too high.

bPower_low: Power too low.
bRelay_failure: Relay failure.
bCap_failure: Capacitor failure.
bLamp_failure: Lamp failure.
bBallast_failure: Ballast failure.
bInter_com_failure: Internal communication failure.
bExter_com_failure: External communication failure.
bMain_volt_below_spec: Main voltage below performance specification.
bLamp_restart_count: Lamp restart retry counter / cycling failure.
bFading_ready: Fading ready.
bBallast_temp_high: Ballast temperature too high.
bDigi_in_A: digital input A active.
bDigi_in_B: digital input B active.
bBit_25_res:
bBit_26_res:
bBit_27_res:
bBit_28_res:
bBit_29_res:
bBit_30_res:
bBit_31_res:
bBit_32_res:
bBit_33_res:
bBit_34_res:
bBit_35_res:
bBit_36_res:
bBit_37_res:
bBit_38_res:
bBit_39_res:
bBit_40_res:

7.3.66.13 SNVT_pos_ctrl

Data types	Description
ST_LON_abspos [► 550]	Used by: SNVT_pos_ctrl
ST_LON_Value [► 551]	Used by: SNVT_pos_ctrl

7.3.66.13.1 ST_LON_abspos

Used by: SNVT_pos_ctrl

```

TYPE ST_LON_abspos :
STRUCT
  rPan    : REAL;
  rTilt   : REAL;
  rZoom   : REAL;
END_STRUCT
END_TYPE
    
```

rPan: Min: -359.98 / Max: 360 / Pan position.

rTilt: Min: -359.98 / Max: 360 / Tilt position.

rZoom: Min: -163.84 / Max: 163.835 / Zoom position.

7.3.66.13.2 ST_LON_Value

Used by: SNVT_pos_ctrl

```

TYPE ST_LON_Value :
STRUCT
  byNumber : BYTE;
  stAbspos : ST_LON_abspos;
END_STRUCT
END_TYPE
    
```

byNumber: Min: 0 / Max: 255 / Action number (action number).

stAbspos: Function absolute values (pan, tilt, zoom) (see [ST_LON_abspos \[► 550\]](#)).

7.3.66.14 SNVT_rac_ctrl

Data types	Description
ST_LON_addr_dest [► 551]	Used by: SNVT_rac_req / SNVT_rac_ctrl
ST_LON_addr_init [► 551]	Used by: SNVT_rac_req / SNVT_rac_ctrl
ST_LON_addr_talk [► 552]	Used by: SNVT_rac_ctrl
ST_LON_p2m [► 552]	Used by: SNVT_rac_req / SNVT_rac_ctrl
ST_LON_p2p [► 552]	Used by: SNVT_rac_req / SNVT_rac_ctrl

7.3.66.14.1 ST_LON_addr_dest

Used by: SNVT_rac_req / SNVT_rac_ctrl

```

TYPE ST_LON_addr_dest :
STRUCT
  stP2p : ST_LON_p2p;
  stP2m : ST_LON_p2m;
END_STRUCT
END_TYPE
    
```

stP2p: (see [ST_LON_p2p \[► 552\]](#))

stP2m: (see [ST_LON_p2m \[► 552\]](#))

7.3.66.14.2 ST_LON_addr_init

Used by: SNVT_rac_req / SNVT_rac_ctrl

```

TYPE ST_LON_addr_init :
STRUCT
  byUnit_id      : BYTE;
  byLocation     : BYTE;
  byCar_id       : BYTE;
  byReserved     : BYTE;
  eAudio_sensor_type : E_LON_rail_audio_sensor_type_t;
END_STRUCT
END_TYPE
    
```

byUnit_id: Min: 0 / Max: 15

byLocation: Min: 0 / Max: 15

byCar_id: Min: 0 / Max: 31

byReserved:

eAudio_sensor_type: see [E_LON_rail_audio_sensor_type_t \[► 515\]](#)

7.3.66.14.3 ST_LON_addr_talk

Used by: SNVT_rac_ctrl

```
TYPE ST_LON_addr_talk :
STRUCT
  byUnit_id      : BYTE;
  byLocation     : BYTE;
  byCar_id      : BYTE;
  byReserved     : BYTE;
  eAudio_sensor_type : E_LON_rail_audio_sensor_type_t;
END_STRUCT
END_TYPE
```

byUnit_id: Min: 0 / Max: 15

byLocation: Min: 0 / Max: 15

byCar_id: Min: 0 / Max: 31

byReserved:

eAudio_sensor_type: see [E_LON_rail_audio_sensor_type_t \[► 515\]](#)

7.3.66.14.4 ST_LON_p2m

Used by: SNVT_rac_req / SNVT_rac_ctrl

```
TYPE ST_LON_p2m :
STRUCT
  byMask_unit    : BYTE;
  arrMask_car    : ARRAY [0..3] OF BYTE;
  arrMask_location : ARRAY [0..1] OF BYTE;
  arrMask_audio  : ARRAY [0..2] OF BYTE;
END_STRUCT
END_TYPE
```

byMask_unit: Min: 0 / Max: 255

arrMask_car: unsigned char [4] ??

arrMask_location: unsigned char [2] ??

arrMask_audio: unsigned char [3] ??

7.3.66.14.5 ST_LON_p2p

Used by: SNVT_rac_req / SNVT_rac_ctrl

```
TYPE ST_LON_p2p :
STRUCT
  byUnit_id      : BYTE;
  byLocation     : BYTE;
  byCar_id      : BYTE;
  byReserved     : BYTE;
  eAudio_sensor_type : E_LON_rail_audio_sensor_type_t;
END_STRUCT
END_TYPE
```

byUnit_id: Min: 0 / Max: 15

byLocation: Min: 0 / Max: 15

byCar_id: Min: 0 / Max: 31

byReserved:

eAudio_sensor_type: see [E_LON_rail_audio_sensor_type_t \[▶ 515\]](#)

7.3.66.15 SNVT_rac_req

Data types	Description
ST_LON_rac_req_addr_dest [▶ 553]	
ST_LON_rac_req_addr_init [▶ 553]	

7.3.66.15.1 ST_LON_rac_req_addr_dest

```

TYPE ST_LON_rac_req_addr_dest :
STRUCT
  stP2p : ST_LON_p2p;
  stP2m : ST_LON_p2m;
END_STRUCT
END_TYPE
    
```

stP2p: (see [ST_LON_p2p \[▶ 552\]](#))

stP2m: (see [ST_LON_p2m \[▶ 552\]](#))

7.3.66.15.2 ST_LON_rac_req_addr_init

```

TYPE ST_LON_rac_req_addr_init :
STRUCT
  byUnit_id : BYTE;
  byLocation : BYTE;
  byCar_id : BYTE;
  byReserved : BYTE;
  eAudio_sensor_type : E_LON_rail_audio_sensor_type_t;
END_STRUCT
END_TYPE
    
```

byUnit_id:

byLocation:

byCar_id:

byReserved:

eAudio_sensor_type: see [E_LON_rail_audio_sensor_type_t \[▶ 515\]](#)

7.3.66.16 SNVT_switch_2

Data types	Description
ST_LON_setting [▶ 553]	Used by: SNVT_switch_2

7.3.66.16.1 ST_LON_setting

Used by: SNVT_switch_2

```

TYPE ST_LON_setting :
STRUCT
  rValue : REAL;
  rChange : REAL;
  byDelay : BYTE;
  byGroup_number : BYTE;
  rMultiplier : REAL;
  iAngle : INT;
  siFan_level : SINT;
    
```

```

byButton_number : BYTE;
END_STRUCT
END_TYPE

```

rValue: Min: 0 / Max: 100 / Value. Percent of full level when state is on. Reports last level for outputs when state is off.

rChange: Min: 0 / Max: 100 / Percent change. Percent change to level.

byDelay: Min: 0 / Max: 255 / On or off delay (seconds) (seconds). Time delay before changing state to on or off.

byGroup_number: Min: 0 / Max: 65 / Group number. Group number that is enabled or disabled by the SW_ENABLE_GROUP and SW_DISABLE_GROUP states in the state field; if 0, all groups are enabled or disabled.

rMultiplier: Min: 0 / Max: 2.54 / Factor (percent). Multiplier for the level

iAngle: Min:-180 / Max:180 / Rotation angle (degrees). Rotation angle for devices that support a rotation setting such as blinds.

siFan_level: Min: -100 / Max: 100 / Fan level. Percent of full level fan speed when state is on. Reports last fan speed for outputs when state is off. Positive values represent the down direction, and negative values represent the up direction.

byButton_number: Min: 0 / Max: 255 / Button number. The button number to activate when the state field is set to SW_SET_BUTTON, no invalid value

7.3.66.17 SNVT_time_zone

Data types	Description
ST_LON_end_DST [► 554]	Used by: SNVT_time_zone
ST_LON_M_end_DST [► 554]	Used by: SNVT_time_zone
ST_LON_M_start_DST [► 555]	Used by: SNVT_time_zone
ST_LON_start_DST [► 555]	Used by: SNVT_time_zone

7.3.66.17.1 ST_LON_end_DST

Used by: SNVT_time_zone

```

TYPE ST_LON_end_DST :
STRUCT
  uiG_day_of_end_DST : UINT;
  uiJ_day_of_end_DST : UINT;
  stM_end_DST       : ST_LON_M_end_DST;
END_STRUCT
END_TYPE

```

uiG_day_of_end_DST: Min: 0 / Max: 365 / Gregorian calendar day of end DST (days).

uiJ_day_of_end_DST: Min: 1 / Max: 365 / Julian calendar day of end DST (days).

stM_end_DST: Meu calendar day of end DST (month, week, dateday) (see [ST_LON_M_end_DST \[► 554\]](#)).

7.3.66.17.2 ST_LON_M_end_DST

Used by: SNVT_time_zone

```

TYPE ST_LON_M_end_DST :
STRUCT
  byMonth_of_end_DST : BYTE;
  byWeek_of_end_DST  : BYTE;
  eDateday_of_end_DST : E_LON_days_of_week_t;
END_STRUCT
END_TYPE

```

byMonth_of_end_DST: Min: 1 / Max: 12 / Month of end DST (months).

byWeek_of_end_DST: Min: 1 / Max: 5 / Week of end DST (weeks).

eDateday_of_end_DST: Day of week (day names, see [E_LON_days_of_week_t \[► 497\]](#)).

7.3.66.17.3 ST_LON_M_start_DST

Used by: SNVT_time_zone

```
TYPE ST_LON_M_start_DST :
STRUCT
  byMonth_of_start_DST   : BYTE;
  byWeek_of_start_DST    : BYTE;
  eDateday_of_start_DST  : E_LON_days_of_week_t;
END_STRUCT
END_TYPE
```

byMonth_of_start_DST: Min: 1 / Max: 12 / Month of start DST (months).

byWeek_of_start_DST: Min: 1 / Max: 5 / Week of start DST (weeks).

eDateday_of_start_DST: Day of week (day names, see [E_LON_days_of_week_t \[► 497\]](#)).

7.3.66.17.4 ST_LON_start_DST

Used by: SNVT_time_zone

```
TYPE ST_LON_start_DST :
STRUCT
  uiG_day_of_start_DST   : UINT;
  uiJ_day_of_start_DST   : UINT;
  stM_start_DST         : ST_LON_M_start_DST;
END_STRUCT
END_TYPE
```

uiG_day_of_start_DST: Min: 0 / Max: 365 / Gregorian calendar day of start DST (days).

uiJ_day_of_start_DST: Min: 1 / Max: 365 / Julian calendar day of start DST (days).

stM_start_DST: Meu calendar day of start DST (month, week, dateday) (see [ST_LON_M_start_DST \[► 555\]](#)).

7.3.67 ST_LON_Parameter_IN_36B

Process Image of the inputs.

This variable must be linked to the System Manager with the KL6401.

```
TYPE ST_LON_Parameter_IN_36B :
STRUCT
  wParameterStatus      : WORD;
  wDummy                : WORD;
  stParameterReadValue  : ST_LON_ParameterInterface;
  byParameterType       : BYTE;
  byLONStatus           : BYTE;
END_STRUCT
END_TYPE
```

wParameterStatus: Status word.

wDummy: Dummy for CX9000.

stParameterReadValue: Data structure (see [ST_LON_ParameterInterface \[► 557\]](#)).

byParameterType: Parameter type.

byLONStatus: LON state.

7.3.68 ST_LON_Parameter_OUT_36B

Process Image of the outputs.

This variable must be linked to the System Manager with the KL6401.

```

TYPE ST_LON_Parameter_OUT_36B :
STRUCT
  wParameterControl : WORD;
  wDummy            : WORD;
  stParametervalue  : ST_LON_ParameterInterface;
  byCMD             : BYTE;
  byIdx            : BYTE;
END_STRUCT
END_TYPE

```

wParameterControl: Control word.

wDummy: Dummy for CX9000.

stParametervalue: Data structure (see [ST_LON_ParameterInterface](#) [▶ 557]).

byCMD: LON command.

byIdx: LON index.

7.3.69 ST_ExplicitMessage

Explicit message

```

TYPE ST_ExplicitMessage :
STRUCT
  byEcpM_1 : BYTE;
  byEcpM_2 : BYTE;
  byLen     : BYTE;
  byAddressingTyp : BYTE;
  byRetry   : BYTE;
  byRptTimer : BYTE;
  byTxTimer : BYTE;
  byRcvTimer : BYTE;
  bySubNet  : BYTE;
  byDestSubNet : BYTE;
  byNode    : BYTE;
  arrNeuronId : ARRAY[0..7] OF BYTE;
  byTyp      : BYTE;
  arrData    : ARRAY [0..47] OF BYTE;
END_STRUCT
END_TYPE

```

byEcpM_1: Ecp M1

byEcpM_2: Ecp M2

byLen: len

byAddressingTyp: adress type

byRetry: retry

byRptTimer: Rpt timer

byTxTimer: Tx timer

byRcvTimer: Rcv timer

bySubNet: Sub net

byDestSubNet: destination Sub net

byNode: node

arrNeuronId: Neuron address

byTyp: type

arrData: 48 data bytes

7.3.70 ST_LON_Communication

Connection between [FB_LON_KL6401 \[► 66\]](#) and the send/receive blocks.

```

TYPE ST_LON_Communication :
STRUCT
  arrWriteLONdata      : ARRAY[0..iLONBufferSize] OF ST_LON_WriteData;
  bWrite               : BOOL;
  bWriteBusy          : BOOL;
  bReadBusy           : BOOL;
  arrParameterReadValue : ARRAY[1..32] OF BYTE;
  wNV_Index           : WORD;
  eParameterDataType  : E_LON_Parameter_Datatypes;
  bWriteLONdataToTable : BOOL;
  bTerminalOk         : BOOL;
  byActBuffer         : BYTE;
  rActBuffer           : REAL;
  rMaxBuffer           : REAL;
END_STRUCT
END_TYPE

```

arrWriteLONdata: Transmit buffer.

bWrite: Data are sent.

bWriteBusy: Write-OnChange active.

bReadBusy: Read-OnChange active.

arrParameterReadValue: 32 data bytes.

wNV_Index: NV index.

eParameterDataType: Data type of the LON variable (SNVT), see [E_LON_Parameter_Datatypes \[► 480\]](#).

bWriteLONdataToTable: Writing the data into the table is active.

bTerminalOk: Initialization of the KL6401 was completed successfully.

byActBuffer: Number of jobs in the transmit buffer.

rActBuffer: Current utilization of the transmit buffer in percent.

rMaxBuffer: Maximum utilization of the transmit buffer in percent. The value can be reset with the input variable *bResetMaxBuffer*.

7.3.71 ST_LON_ParameterInterface

LON parameter interface.

```

TYPE ST_LON_ParameterInterface :
STRUCT
  arrParameterInterface : ARRAY[1..8] OF DWORD;
END_STRUCT
END_TYPE

```

arrParameterInterface: 8 word data.

7.3.72 ST_LON_WriteData

Structure of transmit buffer

```

TYPE ST_LON_WriteData :
STRUCT
  wNVIndex           : WORD;
  udiSrcAddrWriteValue : UDINT;
  uiLenWriteValue    : UINT;
  udiAdrBusy         : UDINT;
  udiAdrErrorKL      : UDINT;
END_STRUCT
END_TYPE

```

wNVIndex: NV-Index

udiSrcAddrWriteValue: Address of the value to be sent

uiLenWriteValue: Length of the value to be sent

udiAdrBusy: Address of the output "bBusy"

udiAdrErrorKL: Address of the output "dwErrorKL"

7.3.73 ST_Prm

Structure for configuration

```

TYPE ST_Prm :
STRUCT
  byCMD          : BYTE;
  byIDX          : BYTE;
  wControl       : WORD;
  wStatus        : WORD;
  wParameterControl : WORD;
  wError         : WORD;
  arrParameterInterface : ARRAY[1..8] OF DWORD;
END_STRUCT
END_TYPE

```

byCMD: LON command

byIDX: LON index

wControl: control word

wStatus: control state

wParameterControl: control word

wError: Error information

arrParameterInterface: 8 data word

7.3.74 ST_KL6401

Structure for configuration.

```

TYPE ST_KL6401 :
STRUCT
  wStatus        : WORD;
  wControl       : WORD;
  dwPointer_IN   : DWORD;
  dwPointer_OUT  : DWORD;
  arrParameterInterface : ARRAY[1..8] OF DWORD;
END_STRUCT
END_TYPE

```

wStatus: State word.

wControl: Control word.

dwPointer_IN: Pointer in.

dwPointer_OUT: Pointer out.

arrParameterInterface: 8 word data.

7.3.75 ST_LON_AddressTable

Used by: FB_Write_Address_Table / FB_Read_Address_Table

```

TYPE ST_LON_AddressTable :
STRUCT
  bType         : BOOL;

```

```

usiNode      : USINT;
bDomain      : BOOL;
usiMember    : USINT;
usiRPT_Timer : USINT;
usiRetry     : USINT;
usiRCV_Timer : USINT;
usiTx_Timer  : USINT;
byGroup      : BYTE;
END_STRUCT
END_TYPE

```

bType: type

usiNode: node

bDomain: domain

usiMember: member

usiRPT_Timer: rpt timer

usiRetry: retry

usiRCV_Timer: Rcv timer

usiTx_Timer: Tx timer

byGroup: group

7.3.76 ST_LON_ConfigTable

Used by: FB_Write_Config_Table / FB_Read_Config_Table

```

TYPE ST_LON_ConfigTable :
STRUCT
  bPriority      : BOOL;
  bDirection    : BOOL;
  wSelector     : WORD;
  bTurnaround   : BOOL;
  usiService    : USINT;
  bAuth         : BOOL;
  usiAddrIndex  : USINT;
END_STRUCT
END_TYPE

```

bPriority: priority

bDirection: direction

wSelector: selector

bTurnaround: turn around

usiService: service

bAuth: auth

usiAddrIndex: address index

7.3.77 ST_LON_DomainTable

Used by: FB_Write_Domain_Table / FB_Read_Domain_Table

```

TYPE ST_LON_DomainTable :
STRUCT
  arrDomainID   : ARRAY[0..5] OF BYTE;
  bySubNet      : BYTE;
  bCloneDomainBit : BOOL;
  byNode        : BYTE;
  byLen         : BYTE;
  arrKey        : ARRAY[0..5] OF BYTE := 16#FF;
END_STRUCT
END_TYPE

```

arrDomainID: Domain Id

bySubNet: Sub Net

bCloneDomainBit: clone domain bit

byNode: node

byLen: Len

arrKey: key

7.3.78 ST_LON_SNVT_alarm

Used by: SNVT_alarm

```

TYPE ST_LON_SNVT_alarm :
STRUCT
  arrLocation      : ARRAY[0..5] OF BYTE;
  uiObject_Id     : UINT;
  eAlarm_type     : E_LON_alarm_type_t;
  ePriority_level  : E_LON_priority_level_t;
  uiIndex_To_SNVT : WORD;
  arrValue        : ARRAY [0..3] OF BYTE;
  uiYear          : UINT;
  uiMonth         : UINT;
  uiDay           : UINT;
  uiHour          : UINT;
  uiMinute        : UINT;
  uiSecond        : UINT;
  uiMilliseconds  : UINT;
  arrAlarm_limit  : ARRAY [0..3] OF BYTE;
END_STRUCT
END_TYPE

```

arrLocation: Location (array of 6 bytes). Location code for the node

uiObject_Id: Object ID (object index). ID of object within node

eAlarm_type: Alarm type (alarm type names, see [E_LON_alarm_type_t \[▶ 488\]](#)).

ePriority_level: Priority level (priority level names, see [E_LON_priority_level_t \[▶ 514\]](#)).

uiIndex_To_SNVT: Index of NV (index of NV causing alarm).

arrValue: Value (array of 4 bytes). The type of this field is dependent on the NV causing the alarm condition.

uiYear: Year (years). Zero (0) means year not specified.

uiMonth: Month (months). Zero (0) means month not specified.

uiDay: Day (days). Zero (0) means day not specified.

uiHour: Hour (hours). This field uses a 24-hour value.

uiMinute: Minute (minutes).

uiSecond: Second (seconds).

uiMilliseconds: Millisecond (milliseconds).

arrAlarm_limit: Alarm limit (array of 4 bytes). The type of this field is dependent on the NV causing the alarm condition.

7.3.79 ST_LON_SNVT_alarm_2

Used by: SNVT_alarm_2

```

TYPE ST_LON_SNVT_alarm_2 :
STRUCT
  eAlarm_type     : E_LON_alarm_type_t;
  ePriority_level  : E_LON_priority_level_t;

```



```

udiAlarm_time      : UDINT;
iMilliseconds      : INT;
bySequence_number  : BYTE;
sDescription        : STRING(22);
END_STRUCT
END_TYPE

```

eAlarm_type: Alarm type (alarm type names, see [E_LON_alarm_type_t](#) [► 488]). Alarm condition reported by this update

ePriority_level: Priority level (priority level names, see [E_LON_priority_level_t](#) [► 514]). Priority level of the alarm reported by this update

udiAlarm_time: Alarm time (seconds). Alarm time in seconds since 2000-01-01T00:00:00Z (the 0 hour of 1 January 2000, Coordinated Universal Time)

iMilliseconds: Milliseconds (milliseconds). Alarm time in milliseconds since the second specified by the alarm_time field

bySequence_number: Sequence number(count). Sequence number for this update. Incremented by one for each update from an alarm source. Wraps to zero after reaching 255. An alarm receiver can use the sequence number to detect missed alarm messages.

sDescription: Description (array of 22 characters). Alarm description with NUL terminator. The terminator is not required if the description requires 22 characters. May include a reference to a language string, delimited by a 0x80 value.

7.3.80 ST_LON_SNVt_chlr_status

Used by: SNVT_chlr_status

```

TYPE ST_LON_SNVt_chlr_status :
STRUCT
  eChlr_run_mode      : E_LON_chiller_t;
  echlr_op_mode       : E_LON_hvac_t;
  stChlr_state        : ST_LON_chlr_state;
END_STRUCT
END_TYPE

```

eChlr_run_mode: Chiller run mode (chiller run mode names) (see [E_LON_chiller_t](#) [► 493]).

echlr_op_mode: Chiller operating mode (HVAC mode names) (see [E_LON_hvac_t](#) [► 509]).

stChlr_state: Chiller state flags (alarm, enabled, local, limited, chiller water flow, condenser water flow) (see [ST_LON_chlr_state](#) [► 533]).

7.3.81 ST_LON_SNVt_clothes_w_a

Used by: SNVT_clothes_w_a

```

TYPE ST_LON_SNVt_clothes_w_a :
STRUCT
  bAlarm_reset        : BOOL;
  bWar_water_supply    : BOOL;
  bWar_drain_slow      : BOOL;
  bWar_door_open       : BOOL;
  bWar_load_unbalanced : BOOL;
  bWar_filter_cleaning : BOOL;
  bWar_hoses_reversed  : BOOL;
  bWar_voltage_low     : BOOL;
  bWar_power_failure   : BOOL;
  bWar_drain_open      : BOOL;
  bWar_execute_fail    : BOOL;
  bWar_door_locked     : BOOL;
  bWar_service         : BOOL;
  bRsrvd5              : BOOL;
  bRsrvd6              : BOOL;
  bRsrvd7              : BOOL;
  bErr_motor_stall     : BOOL;
  bErr_water_temp      : BOOL;
  bErr_pressure        : BOOL;

```

```
bErr_overflow      : BOOL;
bErr_water_heat   : BOOL;
bErr_water_leak   : BOOL;
bErr_motor_speed  : BOOL;
bErr_wash_thermistor : BOOL;
bErr_dry_thermistor : BOOL;
bErr_dry_overheat : BOOL;
bErr_dry_heating  : BOOL;
bErr_dry_fan      : BOOL;
bErr_rsrvd4       : BOOL;
bErr_rsrvd5       : BOOL;
bErr_rsrvd6       : BOOL;
bErr_rsrvd7       : BOOL;
byErr_rsrvd0_7    : BYTE;
byManuf_code      : BYTE;
END_STRUCT
END_TYPE
```

bAlarm_reset:

bWar_water_supply:

bWar_drain_slow:

bWar_door_open:

bWar_load_unbalanced:

bWar_filter_cleaning:

bWar_hoses_reversed:

bWar_voltage_low:

bWar_power_failure:

bWar_drain_open:

bWar_execute_fail:

bWar_door_locked:

bWar_service:

bRsrvd5:

bRsrvd6:

bRsrvd7:

bErr_motor_stall:

bErr_water_temp:

bErr_pressure:

bErr_overflow:

bErr_water_heat:

bErr_water_leak:

bErr_motor_speed:

bErr_wash_thermistor:

bErr_dry_thermistor:

bErr_dry_overheat:

bErr_dry_heating:

bErr_dry_fan:

bErr_rsrvd4:**bErr_rsrvd5:****bErr_rsrvd6:****bErr_rsrvd7:****byErr_rsrvd0_7:****byManuf_code:** Min: 0 / Max: 255

7.3.82 ST_LON_SNVT_clothes_w_c

Used by: SNVT_clothes_w_c

```

TYPE ST_LON_SNVT_clothes_w_c :
STRUCT
  eCycle      : E_LON_appl_cwc_t;
  eSubcycle   : E_LON_appl_cws_t;
  byRervd     : BYTE;
  stAction    : ST_LON_action;
  stFunction  : ST_LON_function;
  uiTime_remaining : UINT;
END_STRUCT
END_TYPE

```

eCycle: (see [E_LON_appl_cwc_t](#) [▶ 490])**eSubcycle:** (see [E_LON_appl_cws_t](#) [▶ 491])**byRervd:****stAction:** (see [ST_LON_action](#) [▶ 533])**stFunction:** (see [ST_LON_function](#) [▶ 534])**uiTime_remaining:** Min: 0 / Max: 65535.

7.3.83 ST_LON_SNVT_clothes_w_m

Used by: SNVT_clothes_w_m

```

TYPE ST_LON_SNVT_clothes_w_m :
STRUCT
  bDoor_ajar : BOOL;
  bDrain_on  : BOOL;
  byReserved : BYTE;
END_STRUCT
END_TYPE

```

bDoor_ajar: Door/Lid Ajar. The door/lid of the washer is not fully closed.**bDrain_on:** Drain On. The drain is on**byReserved:** Reserve

7.3.84 ST_LON_SNVT_clothes_w_s

Used by: SNVT_clothes_w_s

```

TYPE ST_LON_SNVT_clothes_w_s :
STRUCT
  eCycle      : E_LON_appl_cwc_t;
  eSubcycle   : E_LON_appl_cws_t;
  stWasher_command_data : ST_LON_SNVT_clothes_w_c;
  uiTime_remaining : UINT;
  stAlarm      : ST_LON_alarm;
END_STRUCT
END_TYPE

```

eCycle: (see [E_LON_appl_cwc_t](#) [▶ 490])

eSubcycle: (see [E_LON_appl_cws_t](#) [▶ 491])

stWasher_command_data: (see [ST_LON_SNVT_clothes_w_c](#) [▶ 563])

uiTime_remaining: Min: 0 / Max: 65535.

stAlarm: (see [ST_LON_alarm](#) [▶ 535])

7.3.85 ST_LON_SNVT_color

Used by: SNVT_color

```
TYPE ST_LON_SNVT_color :
STRUCT
  rL_star : REAL;
  rA_star : REAL;
  rB_star : REAL;
END_STRUCT
END_TYPE
```

rL_star: Min: 0 / Max: 100.0 / L*

rA_star: Min: -200.0 / Max: 200.0 / a*

rB_star: Min: -200.0 / Max: 200.0 / b*

7.3.86 ST_LON_SNVT_color_2

Used by: SNVT_color_2

```
TYPE ST_LON_SNVT_color_2 :
STRUCT
  eEncoding : E_LON_color_encoding_t;
  stColor_value : ST_LON_color_value;
END_STRUCT
END_TYPE
```

eEncoding: Color encoding. Color encoding specified by the color_value union; additional encodings may be added (see [E_LON_color_encoding_t](#) [▶ 493]).

stColor_value: Color value. Color value encoded as specified by the encoding field (see [ST_LON_color_value](#) [▶ 537]).

7.3.87 ST_LON_SNVT_ctrl_req

Used by: SNVT_ctrl_req

```
TYPE ST_LON_SNVT_ctrl_req :
STRUCT
  uiReceiver_id : UINT;
  uiSender_id : UINT;
  bySender_prio : BYTE;
END_STRUCT
END_TYPE
```

uiReceiver_id: Min: 1 / Max: 65535 / Invalid / 0 / Receiver ID (ID number)

uiSender_id: Min: 1 / Max: 65535 / Invalid / 65535 / Sender ID (ID number)

bySender_prio: Min: 0 / Max: 200 / Sender priority (priority value)

7.3.88 ST_LON_SNVT_ctrl_resp

Used by: SNVT_ctrl_resp

```

TYPE ST_LON_SNVT_ctrl_resp :
STRUCT
  eStatus      : E_LON_control_resp_t;
  stSender     : ST_LON_sender;
  uiController_id : UINT;
END_STRUCT
END_TYPE

```

eStatus: Control response type (control response type names) (see [E_LON_control_resp_t](#) [▶ 494]).

stSender: Sender ID (see [ST_LON_sender](#) [▶ 538]).

uiController_id: Min: 1 / Max: 65535 / Invalid: 65535 / Controller ID (ID number).

7.3.89 ST_LON_SNVT_currency

Used by: SNVT_currency

```

TYPE ST_LON_SNVT_currency :
STRUCT
  eCurrency    : E_LON_currency_t;
  siPower_of_10 : SINT;
  diValue      : DINT;
END_STRUCT
END_TYPE

```

eCurrency: Currency (currency names, see [E_LON_currency_t](#) [▶ 494])

siPower_of_10: Min: -128 / Max: 127 / Magnitude (power of 10)

diValue: Min: -2147483648 / Max: 2147483647 / Value (currency value). Credit is positive, debit is negative.

7.3.90 ST_LON_SNVT_date_event

Used by: SNVT_date_event

```

TYPE ST_LON_SNVT_date_event :
STRUCT
  iDays_to_active      : INT;
  iDays_to_inactive   : INT;
  sName                : STRING(22);
END_STRUCT
END_TYPE

```

iDays_to_active: Min: -32768 / Max: 32767 / Invalid: 32767 / Days to active (days). Number of days until this schedule will be active. Positive if a schedule is inactive; zero or negative if a schedule is active.

iDays_to_inactive: Min: -32768 / Max: 32767 / Invalid: -32768 / Days to inactive (days). Number of days until this schedule will be inactive. Positive if a schedule is active; zero or negative if a schedule is inactive.

sName: 22 characters / Schedule name (array of 22 characters). Nul-terminated schedule name. The nul terminator is not required if the name is 22 characters.

7.3.91 ST_LON_SNVT_dev_fault

Used by: SNVT_dev_fault

```

TYPE ST_LON_SNVT_dev_fault :
STRUCT
  eDevice_select : E_LON_device_select_t;
  stDev_type     : ST_LON_Dev_type1;
END_STRUCT
END_TYPE

```

eDevice_select: Device selection (device selection names). Determines the interpretation of the network-variable content (see [E_LON_device_select_t](#) [▶ 499]).

stDev_type: Union of device fault structures for various devices (see [ST_LON_Dev_type1](#) [▶ 539]).

7.3.92 ST_LON_SNVT_dev_maint

Used by: SNVT_dev_maint

```
TYPE ST_LON_SNVT_dev_maint :
STRUCT
  eDevice_select : E_LON_device_select_t;
  stDev_type     : ST_LON_Dev_type2;
END_STRUCT
END_TYPE
```

eDevice_select: Device selection (device selection names). Determines the interpretation of the network-variable content (see [E_LON_device_select_t](#) [▶ 499]).

stDev_type: Union of device maintenance state structures for various devices (see [ST_LON_Dev_type2](#) [▶ 541]).

7.3.93 ST_LON_SNVT_dev_status

Used by: SNVT_dev_status

```
TYPE ST_LON_SNVT_dev_status :
STRUCT
  eDevice_select : E_LON_device_select_t;
  stDev_type     : ST_LON_Dev_type3;
END_STRUCT
END_TYPE
```

eDevice_select: Device selection (device selection names). Determines the interpretation of the network-variable content (see [E_LON_device_select_t](#) [▶ 499]).

stDev_type: Union of device status for various devices (see [ST_LON_Dev_type3](#) [▶ 542]).

7.3.94 ST_LON_SNVT_earth_pos

Used by: SNVT_earth_pos

```
TYPE ST_LON_SNVT_earth_pos :
STRUCT
  bLatitude_direction : BOOL;
  bLongitude_direction : BOOL;
  byLatitude_deg      : BYTE;
  rLatitude_min       : REAL;
  byLongitude_deg     : BYTE;
  rLongitude_min      : REAL;
  rHeight_above_sea  : REAL;
END_STRUCT
END_TYPE
```

bLatitude_direction: FALSE = South latitude, TRUE = North latitude

bLongitude_direction: FALSE = East longitude, TRUE = West longitude

byLatitude_deg: Min: 0 / Max: 90 / Invalid: 255 / Latitude degrees (degrees).

rLatitude_min: Min: 0.0 / Max: 59.999 / Invalid: 65.535/ Latitude minutes (minutes).

byLongitude_deg: Min: 0 / Max: 180 / Invalid: 255 / Longitude degrees (degrees).

rLongitude_min: Min: 0.0 / Max: 59.999 / Invalid: 65.535/ Longitude minutes (minutes).

rHeight_above_sea: Min: -3.40E+38 / Max: 3.40E+38 / Height above sea level (meters).

7.3.95 ST_LON_SNVT_elapsed_tm

Used by: SNVT_elapsed_tm

```
TYPE ST_LON_SNVT_elapsed_tm :
STRUCT
  uiDay : UINT;
```

```

uiHour      : UINT;
uiMinute    : UINT;
uiSecond    : UINT;
uiMillisecond : UINT;
END_STRUCT
END_TYPE

```

uiDay: Min: 0 / Max: 65535 / Days (days). The value 65535 represents NULL or unknown elapsed time.

uiHour: Min: 0 / Max: 23 / Hours (hours). This field uses a 24-hour value.

uiMinute: Min: 0 / Max: 59 / Minutes (minutes).

uiSecond: Min: 0 / Max: 59 / Seconds (seconds).

uiMillisecond: Min: 0 / Max: 999 / Milliseconds (milliseconds).

7.3.96 ST_LON_SNVT_ent_status

Used by: SNVT_ent_status

```

TYPE ST_LON_SNVT_ent_status :
STRUCT
  bUnlocked      : BOOL;
  bLocked        : BOOL;
  bSecurity_locked : BOOL;
  bClosed        : BOOL;
  bOpen          : BOOL;
  bIn_alarm      : BOOL;
  bIn_error_cond : BOOL;
  bOpen_pre_alarm : BOOL;
  bOpen_alarm    : BOOL;
  bService_alarm : BOOL;
  bTamper        : BOOL;
  bEntry_req     : BOOL;
  bExit_req      : BOOL;
  bKey_req       : BOOL;
  bSafety_ext_req : BOOL;
  bEmergency_req : BOOL;
  bUnable_lock   : BOOL;
  bUnable_unlock : BOOL;
  bStuck         : BOOL;
  bForced_open   : BOOL;
  bForced_close  : BOOL;
  bOpening       : BOOL;
  bClosing       : BOOL;
  bMoving        : BOOL;
  bStopped       : BOOL;
  bSafety_alarm  : BOOL;
  bUnknown_state : BOOL;
  eCmd_fb       : E_LON_ent_opmode_cmd_t;
END_STRUCT
END_TYPE

```

bUnlocked: Unlocked device (boolean). Device is in unlocked position.

bLocked: Locked device (boolean). Device is in locked position.

bSecurity_locked: Security locked (boolean). Device is in a security-driven locked position.

bClosed: Closed device (boolean). Device is in a closed position

bOpen: Open device (boolean). Device is in an open position.

bIn_alarm: In alarm state (boolean). The device is in the alarm state.

bIn_error_cond: In error condition (boolean). Device has an error condition

bOpen_pre_alarm: Open device, pre-alarm (boolean). Device is open, and in warning state.

bOpen_alarm: Open Device, alarm state (boolean). Device is open, and in not-closed alarm state.

bService_alarm: Service alarm (boolean). Device needs service.

bTamper: Tamper mode (boolean). Device has detected tamper.

bEntry_req: Entry request pending (boolean). Device has a pending entry request.

bExit_req: Exit request pending (boolean). Device has a pending exit request.

bKey_req: Key request pending (boolean). Device has a pending key request.

bSafety_ext_req: Safety-exit request pending (boolean). Device has a pending safety-exit request.

bEmergency_req: Emergency-exit request pending (boolean). Device has a pending emergency-exit request.

bUnable_lock: Unable to lock (boolean). Device is unable to close and/or lock

bUnable_unlock: Unable to unlock (boolean). Device is unable to open and/or unlock

bStuck: Device is stuck (boolean). Device is unable to move.

bForced_open: Forced-open Device (boolean). Device is/was forced to go to an open position.

bForced_close: Forced-closed Device (boolean). Device is/was forced to go to a closed position.

bOpening: Device is opening (boolean). Device is currently opening from a closed position.

bClosing: Device is closing (boolean). Device is currently closing from an open position.

bMoving: Device is in motion (boolean). Device is currently changing position.

bStopped: Device Stopped (boolean). The device is stopped and can be moved manually.

bSafety_alarm: Safety-alarm (boolean). Device is in a safety-alarm state.

bUnknown_state: Unknown state (boolean). The state of the device is currently unknown

eCmd_fb: Command feedback (entry command names, see [E_LON_ent_opmode_cmd_t \[► 501\]](#)). Feedback of requested-operation-mode of device

7.3.97 ST_LON_SNVT_environment

Used by: SNVT_environment

```

TYPE ST_LON_SNVT_environment :
STRUCT
  uiLampCurrent      : UINT;
  uiLampVoltage      : UINT;
  uiSupplyVoltage    : UINT;
  uiSupplyCurrent    : UINT;
  rBallastTemp       : REAL;
  rPower             : REAL;
  rPowerFactor       : REAL;
  udiRunHours        : UDINT;
  lrEnergy           : LREAL;
END_STRUCT
END_TYPE

```

uiLampCurrent: Min: 0 / Max: 65534 / Invalid: 65535 / Lamp current (milliAmperes). This is the current the lamp consumes.

uiLampVoltage: Min: 0 / Max: 65535 / Lamp Voltage (Volts). This is the lamp voltage.

uiSupplyVoltage: Min: 0 / Max: 65535 / Supply Voltage (Volts). This is the luminaire supply voltage.

uiSupplyCurrent: Min: 0 / Max: 65534 / Invalid: 65535 / Supply Current (milliAmperes). This is the luminaire supply current.

rBallastTemp: Min: -273.17 / Max: 327.67 / Ballast temperature (degrees Celsius). This is the temperature at the ballast.

rPower: Min: 0 / Max: 6553.5 / Power (Watts). The value shows the at this moment consumed power of the ballast and the luminaire.

rPowerFactor: Min: -1 / Max: 1 / Power factor. This is the luminaire power-factor.

udiRunHours: Min: 0 / Max: 4294967294 / Run Hours (hours). This are the run hours since the last maintenance.

IrEnergy: Min: -214748364.8 / Max: 214748364.7 / Energy (kiloWatt-hours). This is the energy the luminair has consumt since the last maintenance.

7.3.98 ST_LON_SNVT_ex_control

Used by: SNVT_ex_control

```
TYPE ST_LON_SNVT_ex_control :
STRUCT
  eControl_status      : E_LON_ex_control_t;
  stControl_device_addr : ST_LON_Control_device_addr;
END_STRUCT
END_TYPE
```

eControl_status: Control type (control type names) (see [E_LON_ex_control_t \[► 502\]](#)).

stControl_device_addr: Control device address (LonWorks subnet-node address) (see [ST_LON_Control_device_addr \[► 544\]](#)).

7.3.99 ST_LON_SNVT_file_pos

Used by: SNVT_file_pos

```
TYPE ST_LON_SNVT_file_pos :
STRUCT
  diRw_ptr      : DINT;
  uiRw_length   : UINT;
END_STRUCT
END_TYPE
```

diRw_ptr: Min: 0 / Max: 2147483647 / Read/Write pointer (file byte address).

uiRw_length: Min: 0 / Max: 65535 / Read/Write length (number of bytes).

7.3.100 ST_LON_SNVT_file_req

Used by: SNVT_file_req

```
TYPE ST_LON_SNVT_file_req :
STRUCT
  eRequest      : E_LON_file_request_t;
  uiIndex       : UINT;
  uiReceive_timeout : UINT;
  stDest_address : ST_LON_dest_address;
  byAuth_on    : BYTE;
  byPrio_on    : BYTE;
END_STRUCT
END_TYPE
```

eRequest: Request (file request names) (see [E_LON_file_request_t \[► 503\]](#)).

uiIndex: Min: 0 / Max: 65535 / Index (file index).

uiReceive_timeout: Min: 0 / Max: 65535 / Receive timeout (milliseconds).

stDest_address: Destination address (LonWorks address) (see [ST_LON_dest_address \[► 545\]](#)).

byAuth_on: Min: 0 / Max: 1 / Authentication on (boolean). This field specifies whether the message requires authentication.

byPrio_on: Min: 0 / Max: 1 / Priority on (boolean). This field specifies whether the message is to be sent with priority.

7.3.101 ST_LON_SNVT_file_status

Used by: SNVT_file_status

```
TYPE ST_LON_SNVT_file_status :
STRUCT
  eStatus      : E_LON_file_status_t;
  uiNumber_of_files : UINT;
  uiSelected_file  : UINT;
END_STRUCT
END_TYPE
```

eStatus: Status (file status names, see [E_LON_file_status_t](#) [► 503]).

uiNumber_of_files: Min: 0 / Max: 65535 / Number of files (count).

uiSelected_file: Min: 0 / Max: 65535 / Selected file (file index).

7.3.102 ST_LON_SNVT_geo_loc

Used by: SNVT_geo_loc

```
TYPE ST_LON_SNVT_geo_loc :
STRUCT
  lrLongitude  : LREAL;
  lrLatitude  : LREAL;
  rElevation  : REAL;
  sName       : STRING(19);
END_STRUCT
END_TYPE
```

lrLongitude: Min: -180.0 / Max: 180.0 / Longitude. Longitude is given as an angular measurement ranging from 0° at the prime meridian to +180° eastward and -180° westward

lrLatitude: Min: -90.0 / Max: 90.0 / Latitude. Latitude is given as an angular measurement ranging from 0° at the equator to +90° northward and -90° southward

rElevation: Min: -3.40E+51 / Max: 3..40+51 / Elevation (meters).

sName:

7.3.103 ST_LON_SNVT_hvac_overid

Used by: SNVT_hvac_overid

```
TYPE ST_LON_SNVT_hvac_overid :
STRUCT
  eState      : E_LON_hvac_overid_t;
  rPercent    : REAL;
  uiFlow      : UINT;
END_STRUCT
END_TYPE
```

eState: HVAC override state (override state names, see [E_LON_hvac_overid_t](#) [► 507]).

rPercent: Min: -163,84 / Max: 163,835 / Percent (% of full scale). Position or flow override value.

uiFlow: Min: 0 / Max: 65535 / Flow (liters/second).

7.3.104 ST_LON_SNVT_hvac_satsts

Used by: SNVT_hvac_satsts

```
TYPE ST_LON_SNVT_hvac_satsts :
STRUCT
  bPri_heat      : BOOL;
  bSec_heat      : BOOL;
  bPri_cool      : BOOL;
  bSec_cool      : BOOL;
  bPri_duct_starved : BOOL;
  bSec_duct_starved : BOOL;
```

```

bReserved1      : BOOL;
bReserved2      : BOOL;
byReserved1     : BYTE;
byManufacturer_defined : BYTE;
END_STRUCT
END_TYPE

```

bPri_heat: Primary heating saturation status (boolean). A value of 0 indicates primary heating is not saturated. A value of 1 indicates primary heating is saturated.

bSec_heat: Secondary heating saturation status (boolean). A value of 0 indicates secondary heating is not saturated. A value of 1 indicates secondary heating is saturated.

bPri_cool: Primary cooling saturation status (boolean). A value of 0 indicates primary cooling is not saturated. A value of 1 indicates primary cooling is saturated.

bSec_cool: Secondary cooling saturation status (boolean). A value of 0 indicates secondary cooling is not saturated. A value of 1 indicates secondary cooling is saturated.

bPri_duct_starved: Primary duct saturation status (boolean). A value of 0 indicates primary duct is not saturated (starved). A value of 1 indicates primary duct is saturated (starved).

bSec_duct_starved: Secondary duct saturation status (boolean). A value of 0 indicates secondary duct is not saturated (starved). A value of 1 indicates secondary duct is saturated (starved).

bReserved1:

bReserved2:

byReserved1: Min: 0 / Max: 15

byManufacturer_defined: Min: 0 / Max: 15 / Manufacturer defined (boolean). Four manufacturer-defined bits -- please see product documentation for proper interpretation of these bits

7.3.105 ST_LON_SNVT_hvac_status

Used by: SNVT_hvac_status

```

TYPE ST_LON_SNVT_hvac_status :
STRUCT
  eMode          : E_LON_hvac_t;
  rHeat_output_primary : REAL;
  rHeat_output_secondary : REAL;
  rCool_output   : REAL;
  rEcon_output   : REAL;
  rFan_output    : REAL;
  byIn_alarm     : BYTE;
END_STRUCT
END_TYPE

```

eMode: HVAC status mode (HVAC mode names, see [E_LON_hvac_t \[► 509\]](#)).

rHeat_output_primary: Min: -163.8400 / Max: 163.8350 / Primary heat output (% of full scale).

rHeat_output_secondary: Min: -163.8400 / Max: 163.8350 / Secondary heat output (% of full scale).

rCool_output: Min: -163.8400 / Max: 163.8350 / Cooling output (% of full scale).

rEcon_output: Min: -163.8400 / Max: 163.8350 / Economizer output (% of full scale).

rFan_output: Min: -163.8400 / Max: 163.8350 / Fan output (% of full scale).

byIn_alarm: Min: 0 / Max: 255 / in_alarm

7.3.106 ST_LON_SNVT_lamp_status

Used by: SNVT_lamp_status

```

TYPE ST_LON_SNVT_lamp_status :
STRUCT
  stTime_actual : TIMESTRUCT;

```

```

stAlarm_actual      : ST_LON_Alarm_actual;
stTime_previous    : TIMESTRUCT;
stAlarm_previous   : ST_LON_alarm_previous;
END_STRUCT
END_TYPE

```

stTime_actual: Actual alarm message. This is the time stamp for the actual alarm message (see TIMESTRUCT).

stAlarm_actual: (see [ST_LON_Alarm_actual](#) [▶ 547])

stTime_previous: Time Stamp Old. This is the time stamp for an old alarm message (see TIMESTRUCT).

stAlarm_previous: (see [ST_LON_alarm_previous](#) [▶ 549])

7.3.107 ST_LON_SNVT_log_fx_request

Used by: SNVT_log_fx_request

```

TYPE ST_LON_SNVT_log_fx_request :
STRUCT
  uiRequested_log   : UINT;
  udiRecord_count  : UDINT;
  stStart_time     : TIMESTRUCT;
  stEnd_time       : TIMESTRUCT;
END_STRUCT
END_TYPE

```

uiRequested_log: Min: 1 / Max: 65535 / Requested log number. The log number of the data log to be transferred. Logs are numbered from 1 to number_of_logs.

udiRecord_count: Min: 0 / Max: 4294967295 / Record count. The maximum number of log records to be transferred.

stStart_time: Start time. (seconds). Timestamp of first record to be transferred. If no records exist with this timestamp, the first record with a timestamp after this timestamp is the starting record.

stEnd_time: End time. (seconds). Timestamp of last record to be transferred. If no records exist with this timestamp, the last record with a timestamp before this timestamp is the ending record.

7.3.108 ST_LON_SNVT_log_fx_status

Used by: SNVT_log_fx_status

```

TYPE ST_LON_SNVT_log_fx_status :
STRUCT
  byRequestor_subnet : BYTE;
  byRequestor_node   : BYTE;
  uiLog_number       : UINT;
  rComplete          : REAL;
END_STRUCT
END_TYPE

```

byRequestor_subnet: Min: 1 / Max: 255 / Requestor subnet ID. Subnet ID of the device that requested the current log file transfer. Invalid if a file transfer is not active.

byRequestor_node: Min: 1 / Max: 255 / Requestor node ID. Node ID of the device that requested the current log file transfer. Invalid if a file transfer is not active.

uiLog_number: Min: 1 / Max: 65535 / Data log number. Log number for the log file currently being transferred via FTP. Invalid if none.

rComplete: Min: 0 / Max: 100 / Data log file transfer percent complete. Percent of the current data log file transfer that has been completed. Invalid if none.

7.3.109 ST_LON_SNVT_log_status

Used by: SNVT_log_status

```

TYPE ST_LON_SNVT_log_status :
STRUCT
  eStatus          : E_LON_log_status_t;
  uiLog_number     : UINT;
  rLevel           : REAL;
  diRecord_count   : DINT;
  diByte_count     : DINT;
  diTotal_record_count : DINT;
  diRecords_since_notification : DINT;
  stCurrent_notify_time : TIMESTRUCT;
  stPrevious_notify_time : TIMESTRUCT;
END_STRUCT
END_TYPE

```

eStatus: Log state [► 510]. State of the selected data log.

uiLog_number: Min: 1 / Max: 65535 / Selected log number. The log number of the reported data log. Logs are numbered from 1 to number_of_logs.

rLevel: Min: 0 / Max: 100 / Log level (Percent). The percent of maximum records in the selected data log.

diRecord_count: Min: -2147483648 / Max: 2147483647 / Record count. (records). Number of records in the selected data log. A record is a logged value and any associated data such as a timestamp.

diByte_count: Min: -2147483648 / Max: 2147483647 / Byte count. (bytes). Number of bytes in the selected data log.

diTotal_record_count: Min: -2147483648 / Max: 2147483647 / Total record count. (records). Total records collected in the selected data log since the data log was created. Wraps to 0 on overflow.

diRecords_since_notification: Min: -2147483648 / Max: 2147483647 / Records since notification. (records). The number of log records collected since the last notification.

stCurrent_notify_time: Current notify time. Timestamp of the most recently collected data point.

stPrevious_notify_time: Previous notify time. (seconds). Timestamp of the most recently collected data point in the previous update to the log status.

7.3.110 ST_LON_SNVT_muldiv

Used by: SNVT_muldiv

```

TYPE ST_LON_SNVT_muldiv :
STRUCT
  uiMultiplier : UINT;
  uiDivisor     : UINT;
END_STRUCT
END_TYPE

```

uiMultiplier: Min: 0 / Max: 65535 / Multiplier (16-bit unsigned value).

uiDivisor: Min: 0 / Max: 65535 / Divisor (16-bit unsigned value).

7.3.111 ST_LON_SNVT_nv_type

Used by: SNVT_nv_type

```

TYPE ST_LON_SNVT_nv_type :
STRUCT
  arrType_program_ID : ARRAY [0..7] OF BYTE;
  byType_scope       : BYTE;
  uiType_index       : UINT;
  eType_category     : E_LON_nv_type_category_t;
  byType_length      : BYTE;
  iScaling_factor_a  : INT;
  iScaling_factor_b  : INT;
  iScaling_factor_c  : INT;
END_STRUCT
END_TYPE

```

arrType_program_ID: Min: 0 / Max: 255 / Type program ID. Program ID template of the resource file containing the network variable type definition.

byType_scope: Min: 0 / Max: 6 / Type scope (file scope). Scope of the resource file containing the network variable type definition.

uiType_index: Min: 1 / Max: 65535 / Type index (type index). Index within the specified resource file of the network variable type definition.

eType_category: Type category (type category names, see [E_LON_nv_type_category_t](#) [▶ 511]). Type category of the network variable type

byType_length: Min: 1 / Max: 31 / Type length (bytes). Length of the network variable type

iScaling_factor_a: Min: -32768 / Max: 32767 / Scaling factor a. Scaling multiplier 'a' where $ScaledValue = a \cdot (10^b) \cdot (RawValue + c)$

iScaling_factor_b: Min: -32768 / Max: 32767 / Scaling factor b. Exponent 'b' where $ScaledValue = a \cdot (10^b) \cdot (RawValue + c)$

iScaling_factor_c: Min: -32768 / Max: 32767 / Scaling Factor c. Offset 'c' where $ScaledValue = a \cdot (10^b) \cdot (RawValue + c)$

7.3.112 ST_LON_SNVT_obj_request

Used by: SNVT_obj_request

```
TYPE ST_LON_SNVT_obj_request :
STRUCT
  uiObject_id      : UINT;
  eObject_request  : E_LON_object_request_t;
END_STRUCT
END_TYPE
```

uiObject_id: Min: 0 / Max: 65535 / Object ID (object index).

eObject_request: Object request (object request names, see [E_LON_object_request_t](#) [▶ 512]).

7.3.113 ST_LON_SNVT_obj_status

Used by: SNVT_obj_status

```
TYPE ST_LON_SNVT_obj_status :
STRUCT
  uiObject_id      : UINT;
  bInvalid_id      : BOOL;
  bInvalid_request : BOOL;
  bDisabled        : BOOL;
  bOut_of_limits   : BOOL;
  bOpen_circuit    : BOOL;
  bOut_of_service  : BOOL;
  bMechanical_fault : BOOL;
  bFeedback_failure : BOOL;
  bOver_range      : BOOL;
  bUnder_range     : BOOL;
  bElectrical_fault : BOOL;
  bUnable_to_measure : BOOL;
  bComm_failure    : BOOL;
  bFail_self_test  : BOOL;
  bSelf_test_in_progress : BOOL;
  bLocked_out      : BOOL;
  bManual_control  : BOOL;
  bIn_alarm        : BOOL;
  bIn_override     : BOOL;
  bReport_mask     : BOOL;
  bProgramming_mode : BOOL;
  bProgramming_fail : BOOL;
  bAlarm_notify_disabled : BOOL;
  bReset_complete  : BOOL;
  byReserved2      : BYTE;
END_STRUCT
END_TYPE
```

uiObject_id: Min: 0 / Max: 65535 / Object ID (object index).

bInvalid_id: Invalid-ID flag (boolean).

bInvalid_request: Invalid-request flag (boolean).

bDisabled: Disabled flag (boolean).

bOut_of_limits: Out-of-limits flag (boolean).

bOpen_circuit: Open-circuit flag (boolean).

bOut_of_service: Out-of-service flag (boolean).

bMechanical_fault: Mechanical-fault flag (boolean).

bFeedback_failure: Feedback-failure flag (boolean).

bOver_range: Over-range flag (boolean).

bUnder_range: Under-range flag (boolean).

bElectrical_fault: Electrical-fault flag (boolean).

bUnable_to_measure: Unable-to-measure flag (boolean).

bComm_failure: Communications-failure flag (boolean).

bFail_self_test: Failed-self-test flag (boolean).

bSelf_test_in_progress: Self-test-in-progress flag (boolean).

bLocked_out: Locked-out flag (boolean).

bManual_control: Manual-control flag (boolean).

bln_alarm: Input-alarm flag (boolean).

bln_override: Input-override flag (boolean).

bReport_mask: Report-mask flag (boolean).

bProgramming_mode: Programming-mode flag (boolean).

bProgramming_fail: Programming-fail flag (boolean).

bAlarm_notify_disabled: Alarm-notify-disabled flag (boolean).

bReset_complete: Reset (boolean).

byReserved2: This field is reserved.

7.3.114 ST_LON_SNVT_pos_ctrl

Used by: SNVT_pos_ctrl

```

TYPE ST_LON_SNVT_pos_ctrl :
STRUCT
  uiReceiver_id      : UINT;
  uiController_id   : UINT;
  byController_prio  : BYTE;
  eFunction          : E_LON_cam_func_t;
  eAction           : E_LON_cam_act_t;
  stValue           : ST_LON_Value;
END_STRUCT
END_TYPE

```

uiReceiver_id: Min: 0 / Max: 65535 / Receiver ID (ID number).

uiController_id: Min: 0 / Max: 65535 / Controller ID (ID number).

byController_prio: Min: 0 / Max: 100 / Controller priority (priority value).

eFunction: Camera function (camera function names) (see [E_LON_cam_func_t](#) [▶ 493]).

eAction: Camera action (camera action names) (see [E_LON_cam_act_t](#) [▶ 492]).

stValue: Function value (see [ST_LON_Value](#) [▶ 551]).

7.3.115 ST_LON_SNVT_preset

Used by: SNVT_preset

```

TYPE ST_LON_SNVT_preset :
STRUCT
  eLearn      : E_LON_learn_mode_t;
  uiSelector  : UINT;
  arrValue    : ARRAY[0..3] OF BYTE;
  uiDay       : UINT;
  uiHour      : UINT;
  uiMinute    : UINT;
  uiSecond    : UINT;
  uiMillisecond : UINT;
END_STRUCT
END_TYPE

```

eLearn: Learn mode (learn mode names, see [E_LON_learn_mode_t](#) [▶ 510]).

uiSelector: Min: 0 / Max: 65535 / Selector (16-bit unsigned value). The selector is used to choose which preset.

arrValue: Value (array of 4 bytes).

uiDay: Min: 0 / Max: 65535 / Days (days). The value 65535 represents NULL or unknown elapsed time.

uiHour: Min: 0 / Max: 23 / Hours (hours). This field uses a 24-hour value.

uiMinute: Min: 0 / Max: 59 / Minutes (minutes).

uiSecond: Min: 0 / Max: 59 / Seconds (seconds).

uiMillisecond: Min: 0 / Max: 999 / Milliseconds (milliseconds).

7.3.116 ST_LON_SNVT_privacyzone

Used by: SNVT_privacyzone

```

TYPE ST_LON_SNVT_privacyzone :
STRUCT
  eAction     : E_LON_privacyzone_t;
  byNumber    : BYTE;
  uiCamera_id : UINT;
END_STRUCT
END_TYPE

```

eAction: Privacy zone action type (privacy zone action type names, see [E_LON_privacyzone_t](#) [▶ 515]).

byNumber: Min: 0 / Max: 255 / Zone number (zone number).

uiCamera_id: Min: 0 / Max: 65535 / Camera ID (ID number).

7.3.117 ST_LON_SNVT_ptz

Used by: SNVT_ptz

```

TYPE ST_LON_SNVT_ptz :
STRUCT
  ePan_dir    : E_LON_pan_dir_t;
  rPan_speed  : REAL;
  eTilt_dir   : E_LON_tilt_dir_t;
  rTilt_speed : REAL;
  eZoom       : E_LON_zoom_t;
  rZoom_speed : REAL;
END_STRUCT
END_TYPE

```


ePan_dir: Pan direction (pan direction names, see [E_LON_pan_dir_t \[► 514\]](#)).

rPan_speed: Min: 0 / Max: 100 / Pan speed (% of full level).

eTilt_dir: Tilt direction (tilt direction names, see [E_LON_tilt_dir_t \[► 529\]](#)).

rTilt_speed: Min: 0 / Max: 100 / Tilt speed (% of full level).

eZoom: Zoom direction (zoom direction names, see [E_LON_zoom_t \[► 530\]](#)).

rZoom_speed: Min: 0 / Max: 100 / Zoom speed (% of full level).

7.3.118 ST_LON_SNVT_pump_sensor

Used by: SNVT_pump_sensor

```

TYPE ST_LON_SNVT_pump_sensor :
STRUCT
  rRotational_speed      : REAL;
  rBody_temperature     : REAL;
  rMotor_external_temperature : REAL;
  rMotor_internal_temperature : REAL;
  eMotor_overloaded     : E_LON_boolean_t;
  eOil_level_low        : E_LON_boolean_t;
  ePhase_imbalance_detected : E_LON_boolean_t;
  rCurrent_usage        : REAL;
  rPower_usage          : REAL;
  eTemperature_control  : E_LON_unit_temp_t;
  eElectromagnetic_brake_active : E_LON_boolean_t;
  eFriction_brake_active : E_LON_boolean_t;
  eGas_brake_active     : E_LON_boolean_t;
END_STRUCT
END_TYPE

```

rRotational_speed: Min: 0 / Max: 6553.5 / Rotational speed.

rBody_temperature: Min: -274 / Max: 6279.5 / Body temperature.

rMotor_external_temperature: Min: -274 / Max: 6279.5 / Motor external temp.

rMotor_internal_temperature: Min: -274 / Max: 6279.5 / Motor internal temp.

eMotor_overloaded: Motor overloaded (Boolean, see [E_LON_boolean_t \[► 492\]](#)).

eOil_level_low: Oil level low (boolean).

ePhase_imbalance_detected: Phase imbalance (boolean).

rCurrent_usage: Min: -3276.8 / Max: 3276.7 / Current usage.

rPower_usage: Min: 0 / Max: 6553.5 / Power usage.

eTemperature_control: Pump body temp control status (temperature control status names, see [E_LON_unit_temp_t \[► 529\]](#)).

eElectromagnetic_brake_active: Electromagnetic brake active (boolean).

eFriction_brake_active: Friction brake active (boolean).

eGas_brake_active: Gas brake active (boolean).

7.3.119 ST_LON_SNVT_pumpset_mn

Used by: SNVT_pumpset_mn

```

TYPE ST_LON_SNVT_pumpset_mn :
STRUCT
  eMain_pump           : E_LON_motor_state_t;
  eBooster_pump        : E_LON_motor_state_t;
  ePriority_level       : E_LON_Priority_level_t;
  eProcess_ready       : E_LON_boolean_t;

```

```

eEmergency_stop_activated      : E_LON_boolean_t;
eMain_pump_drive_enabled      : E_LON_boolean_t;
eBooster_pump_drive_enabled   : E_LON_boolean_t;
eMaintenance_required        : E_LON_boolean_t;
END_STRUCT
END_TYPE

```

eMain_pump: Main pump state (motor state names, see [E_LON_motor_state_t](#) [► 511]).

eBooster_pump: Booster pump state (motor state names).

ePriority_level: Priority level (priority level names, see [E_LON_priority_level_t](#) [► 514]).

eProcess_ready: Process ready (boolean).

eEmergency_stop_activated: Emergency stop (Boolean, see [E_LON_boolean_t](#) [► 492]).

eMain_pump_drive_enabled: Main pump enabled (boolean).

eBooster_pump_drive_enabled: Booster pump enabled (boolean).

eMaintenance_required: Maintenance required (boolean).

7.3.120 ST_LON_SNVT_pumpset_sn

Used by: SNVT_pumpset_sn

```

TYPE ST_LON_SNVT_pumpset_sn :
STRUCT
  uiTotal_dilution_flow      : UINT;
  rExhaust_temperature       : REAL;
  rExhaust_pressure          : REAL;
  rShaft_seal_purge_pressure  : REAL;
  rInlet_vacuum              : REAL;
  rSupply_voltage            : REAL;
  uiCoolant_flow             : UINT;
  eCoolant_flow_low          : E_LON_boolean_t;
  eDilution_active          : E_LON_boolean_t;
  eBallast_dilution_active   : E_LON_boolean_t;
  eInlet_purge_dilution_active : E_LON_boolean_t;
  eExhaust_dilution_active   : E_LON_boolean_t;
  eDilution_flow_Out_of_range : E_LON_boolean_t;
  ePower_supply_on           : E_LON_boolean_t;
END_STRUCT
END_TYPE

```

uiTotal_dilution_flow: Min: 0 / Max: 65535 / Dilution gas flow.

rExhaust_temperature: Min: 274 / Max: 6279.5 / Exhaust line external temperature.

rExhaust_pressure: Min: -3276.8 / Max: 3276.7 / Exhaust line pressure.

rShaft_seal_purge_pressure: Min: -3276.8 / Max: 3276.7 / Shaft seal purge pressure.

rInlet_vacuum: Min: -3.40E+38 / Max: 3.40E+38 / Process gas inlet pressure.

rSupply_voltage: Min: -3276.8 / Max: 3276.7 / Pumpset power supply voltage.

uiCoolant_flow: Min: 0 / Max: 65535 / Total coolant flow.

eCoolant_flow_low: Coolant flow too low (Boolean, see [E_LON_boolean_t](#) [► 492]).

eDilution_active: Coolant flow too low (boolean).

eBallast_dilution_active: Dilution gas being used as ballast (boolean).

eInlet_purge_dilution_active: Dilution gas being used to purge process gas (boolean).

eExhaust_dilution_active: Dilution gas being used to dilute exhaust (boolean).

eDilution_flow_Out_of_range: Dilution gas flow outside normal range (boolean).

ePower_supply_on: Dilution gas flow outside normal range (boolean).

7.3.121 ST_LON_SNVT_rac_ctrl

Used by: SNVT_rac_ctrl

```

TYPE ST_LON_SNVT_rac_ctrl :
STRUCT
  byAudio_line   : BYTE;
  bDuplex_full   : BOOL;
  bDest_p2p      : BOOL;
  byReserved     : BYTE;
  eAudio_type    : E_LON_rail_audio_type_t;
  stAddr_init    : ST_LON_addr_init;
  stAddr_talk    : ST_LON_addr_talk;
  stAddr_dest    : ST_LON_addr_dest;
END_STRUCT
END_TYPE

```

byAudio_line: Min: 0 / Max: 7

bDuplex_full:

bDest_p2p:

byReserved:

eAudio_type: (see [E_LON_rail_audio_type_t](#) [► 516])

stAddr_init: (see [ST_LON_addr_init](#) [► 551])

stAddr_talk: (see [ST_LON_addr_talk](#) [► 552])

stAddr_dest: (see [ST_LON_addr_dest](#) [► 551])

7.3.122 ST_LON_SNVT_rac_req

Used by: SNVT_rac_req

```

TYPE ST_LON_SNVT_rac_req :
STRUCT
  bDest_def      : BOOL;
  bDest_p2p      : BOOL;
  byReserved     : BYTE;
  eAudio_type    : E_LON_rail_audio_type_t;
  stAddr_init    : ST_LON_addr_init;
  stAddr_dest    : ST_LON_addr_dest;
END_STRUCT
END_TYPE

```

bDest_def:

bDest_p2p:

byReserved:

eAudio_type: (see [E_LON_rail_audio_type_t](#) [► 516])

stAddr_init: (see [ST_LON_addr_init](#) [► 551])

stAddr_dest: (see [ST_LON_addr_dest](#) [► 551])

7.3.123 ST_LON_SNVT_reg_val

Used by: SNVT_rac_val

```

TYPE ST_LON_SNVT_reg_val :
STRUCT
  diRaw          : DINT;
  eUnit          : E_LON_reg_val_unit_t;
  byNr_decimals  : BYTE;
END_STRUCT
END_TYPE

```

diRaw: Raw value.

eUnit: Unit code (defines unit of measure, see [E_LON_reg_val_unit_t](#) [► 517]).

byNr_decimals: Number of decimals (digits to right of decimal point).

7.3.124 ST_LON_SNVT_reg_val_ts

Used by: SNVT_rac_val_ts

```

TYPE ST_LON_SNVT_reg_val_ts :
STRUCT
  diRaw      : DINT;
  eUnit      : E_LON_reg_val_unit_t;
  byNr_decimals : BYTE;
  byStatus   : BYTE;
  bReg_state : BOOL;
  uiYear     : UINT;
  uiMonth    : UINT;
  uiDay      : UINT;
  uiHour     : UINT;
  uiMinute   : UINT;
  uiSecond   : UINT;
END_STRUCT
END_TYPE

```

diRaw: Min: -2147483648 / Max: 2147483647 / Raw value.

eUnit: Unit code (unit names, see [E_LON_reg_val_unit_t](#) [► 517]).

byNr_decimals: Min: 0 / Max: 7 / Number of decimals (digits to right of decimal point).

byStatus: Min: 0 / Max: 15 / Status (status or error during measuring period).

bReg_state: Activation state (activation state of register).

uiYear: Min: -1 / Max: 3000 / Year (years). Zero (0) means year not specified. Minus one (-1) represents NULL date.

uiMonth: Min: 0 / Max: 12 / Month (months). Zero (0) means month not specified.

uiDay: Min: 0 / Max: 31 / Day (days). Zero (0) means day not specified.

uiHour: Min: 0 / Max: 23 / Hour (hours). This field uses a 24-hour value.

uiMinute: Min: 0 / Max: 59 / Minute (minutes).

uiSecond: Min: 0 / Max: 59 / Second (seconds).

7.3.125 ST_LON_SNVT_sbIn_d_state

Used by: SNVT_sbIn_d_state

```

TYPE ST_LON_SNVT_sbIn_d_state :
STRUCT
  stPos      : ST_LON_SNVT_setting;
  eCmd_source : E_LON_sbIn_d_cmd_source_t;
  eError_code : E_LON_sbIn_d_error_t;
END_STRUCT
END_TYPE

```

stPos: (see [ST_LON_SNVT_setting](#) [► 581])

eCmd_source: (see [E_LON_sbIn_d_cmd_source_t](#) [► 519])

eError_code: (see [E_LON_sbIn_d_error_t](#) [► 520])

7.3.126 ST_LON_SNVT_scene

Used by: SNVT_scene

```

TYPE ST_LON_SNVT_scene :
STRUCT
  eFunction      : E_LON_Scene_t;
  byScene_number : BYTE;
END_STRUCT
END_TYPE

```

eFunction: Scene control function (scene control function names, see [E_LON_scene_t](#) [► 522]).

byScene_number: Min: 0 / Max: 255 / Scene number.

7.3.127 ST_LON_SNVT_scene_cfg

Used by: SNVT_scene_cfg

```

TYPE ST_LON_SNVT_scene_cfg :
STRUCT
  eFunction      : E_LON_Scene_config_t;
  byScene_number : BYTE;
  rSetting       : REAL;
  rRotation      : REAL;
  rFade_time     : REAL;
  rDelay_time    : REAL;
  scene_priority : BYTE;
END_STRUCT
END_TYPE

```

eFunction: Scene configuration function (scene configuration function names, see [E_LON_scene_config_t](#) [► 521]).

byScene_number: Min: 0 / Max: 255 / Scene number.

rSetting: Min: 0 / Max: 100.0 Invalid: 255 / Scene setting level (% of full level).

rRotation: Min: -359,98 / Max: 360,00 / Scene rotation angle (degrees).

rFade_time: Min: 0 / Max: 6553.5 / Scene fade time (seconds).

rDelay_time: Min: 0 / Max: 6553.5 / Scene delay time (seconds).

scene_priority: Min: 0 / Max: 255 / scene_priority

7.3.128 ST_LON_SNVT_setting

Used by: SNVT_setting

```

TYPE ST_LON_SNVT_setting :
STRUCT
  eFunction      : E_LON_setting_t;
  rSetting       : REAL;
  rRotation      : REAL;
END_STRUCT
END_TYPE

```

eFunction: Setting control function (setting control function names, see [E_LON_setting_t](#) [► 525]).

rSetting: Min: 0 / Max: 100 / Scene setting level (% of full level).

rRotation: Min: -359.98 / Max: 360.00 / Rotation angle (degrees).

7.3.129 ST_LON_SNVT_str_int

Used by: SNVT_str_int

```

TYPE ST_LON_SNVT_str_int :
STRUCT
  byChar_set     : BYTE;
  arrWide_char   : ARRAY [0..14] OF UINT;
END_STRUCT
END_TYPE

```

byChar_set: Min: 0 / Max: 255 / Locale code (code value).

arrWide_char: Min: 0 / Max: 65535 / Wide character string (array of 15 wide characters).

7.3.130 ST_LON_SNVT_switch

Used by: SNVT_switch

```
TYPE ST_LON_SNVT_switch :
STRUCT
  rValue      : REAL;
  siState     : SINT;
END_STRUCT
END_TYPE
```

rValue: Min: 0 / Max: 100 / Value (% of full level).

siState: Min: -1 / Max: 1 / State (state code). This field can either be -1 (NULL), 0 (OFF), or 1 (ON).

7.3.131 ST_LON_SNVT_switch_2

Used by: SNVT_switch_2

```
TYPE ST_LON_SNVT_switch_2 :
STRUCT
  eState      : E_LON_switch_state_t;
  stSetting   : ST_LON_setting;
  byScene_number : BYTE;
END_STRUCT
END_TYPE
```

eState: Switch state. Switch state; maybe a state of the switch or other switch properties such as scene, occupancy state, and level multiplier (see [E_LON_switch_state_t](#) [▶ 525]).

stSetting: Switch setting. Sets or reports the level, change, or angle for a switch (see [ST_LON_setting](#) [▶ 553]).

byScene_number: Min: 1 / Max: 255 / Scene number. Scene number that is applied based on the function specified in the state field.

7.3.132 ST_LON_SNVT_temp_setpt

Used by: SNVT_temp_setpt

```
TYPE ST_LON_SNVT_temp_setpt :
STRUCT
  rOccupied_cool      : REAL;
  rStandby_cool       : REAL;
  rUnoccupied_cool    : REAL;
  rOccupied_heat      : REAL;
  rStandby_heat       : REAL;
  rUnoccupied_heat    : REAL;
END_STRUCT
END_TYPE
```

rOccupied_cool: Min: -273,17 / Max: 237,67 / Occupied cooling setpoint (degrees Celsius).

rStandby_cool: Min: -273,17 / Max: 237,67 / Standby cooling setpoint (degrees Celsius).

rUnoccupied_cool: Min: -273,17 / Max: 237,67 / Unoccupied cooling setpoint (degrees Celsius).

rOccupied_heat: Min: -273,17 / Max: 237,67 / Occupied heating setpoint (degrees Celsius).

rStandby_heat: Min: -273,17 / Max: 237,67 / Standby heating setpoint (degrees Celsius).

rUnoccupied_heat: Min: -273,17 / Max: 237,67 / Unoccupied heating setpoint (degrees Celsius).

7.3.133 ST_LON_SNVT_time_zone

Used by: SNVT_time_zone

```

TYPE ST_LON_SNVT_time_zone :
STRUCT
  diSecond_time_offset      : DINT;
  eType_of_description      : E_LON_calendar_type_t;
  byHour_of_start_DST      : BYTE;
  byMinute_of_start_DST    : BYTE;
  bySecond_of_start_DST    : BYTE;
  stStart_DST               : ST_LON_start_DST;
  byHour_of_end_DST        : BYTE;
  byMinute_of_end_DST      : BYTE;
  bySecond_of_end_DST      : BYTE;
  stEnd_DST                 : ST_LON_end_DST;
END_STRUCT
END_TYPE

```

diSecond_time_offset: Min: -86400 / Max: 86400 / Offset from GMT (seconds). West direction is negative offset

eType_of_description: Calendar type (calendar type names, see [E_LON_calendar_type_t](#) [▶ 492]).

byHour_of_start_DST: Min: 0 / Max: 23 / DST start hour (hours).

byMinute_of_start_DST: Min: 0 / Max: 59 / DST start minute (minutes).

bySecond_of_start_DST: Min: 0 / Max: 59 / DST start second (seconds).

stStart_DST: DST start day (day descriptor). Daylight savings time start day (see [ST_LON_start_DST](#) [▶ 555]).

byHour_of_end_DST: Min: 0 / Max: 23 / DST end hour (hours).

byMinute_of_end_DST: Min: 0 / Max: 59 / DST end minute (minutes).

bySecond_of_end_DST: Min: 0 / Max: 59 / DST end second (seconds).

stEnd_DST: DST end day (day descriptor). Daylight savings time end day (see [ST_LON_end_DST](#) [▶ 554]).

7.3.134 ST_LON_SNVT_tod_event

Used by: SNVT_tod_event

```

TYPE ST_LON_SNVT_tod_event :
STRUCT
  eCurrent_state           : E_LON_occup_t;
  eNext_state              : E_LON_occup_t;
  uiTime_to_next_state     : UINT;
END_STRUCT
END_TYPE

```

eCurrent_state: Occupancy, current (occupancy code names, see [E_LON_occup_t](#) [▶ 513]).

eNext_state: Occupancy, next (occupancy code names).

uiTime_to_next_state: Min: 0 / Max: 65535 / Time to next state (minutes).

7.3.135 ST_LON_SNVT_trans_table

Used by: SNVT_trans_table

```

TYPE ST_LON_SNVT_trans_table :
STRUCT
  arrPoint                 : ARRAY [0..6] OF REAL;
  byInterp_pts_0_to_1     : BYTE;
  byInterp_pts_1_to_2     : BYTE;
  byInterp_pts_2_to_3     : BYTE;
  byInterp_pts_3_to_4     : BYTE;
  byInterp_pts_4_to_5     : BYTE;
  byInterp_pts_5_to_6     : BYTE;

```

```

    byInterp_pts_6_to_0 : BYTE;
END_STRUCT
END_TYPE

```

arrPoint: Points (array of 7 points).

byInterp_pts_0_to_1: Min: 0 / Max: 1 / Interpolation for point 0 to point 1 (interpolation method code).

byInterp_pts_1_to_2: Min: 0 / Max: 1 / Interpolation for point 1 to point 2 (interpolation method code).

byInterp_pts_2_to_3: Min: 0 / Max: 1 / Interpolation for point 2 to point 3 (interpolation method code).

byInterp_pts_3_to_4: Min: 0 / Max: 1 / Interpolation for point 3 to point 4 (interpolation method code).

byInterp_pts_4_to_5: Min: 0 / Max: 1 / Interpolation for point 4 to point 5 (interpolation method code).

byInterp_pts_5_to_6: Min: 0 / Max: 1 / Interpolation for point 5 to point 6 (interpolation method code).

byInterp_pts_6_to_0: Min: 0 / Max: 1 / Interpolation for point 6 to point 0 (interpolation method code). This field is used when multiple interpolation tables are linked.

7.3.136 ST_LON_SNVZ_zerospan

Used by: SNVT_zerospan

```

TYPE ST_LON_SNVZ_zerospan :
STRUCT
    rZero : REAL;
    rSpan : REAL;
END_STRUCT
END_TYPE

```

rZero: Min: -163.840 / Max: 163.835 / Zero-term (16-bit signed value).

rSpan: Min: 0.0 / Max: 32.7675 / Span-factor (16-bit unsigned value).

7.3.137 str_AddressTable

address table

```

TYPE str_AddressTable :
STRUCT
    bType      : BOOL;
    Node       : USINT;
    bDomain    : BOOL;
    Member     : USINT;
    RPT_Timer  : USINT;
    Retry      : USINT;
    RCV_Timer  : USINT;
    Tx_Timer   : USINT;
    Group      : BYTE;
END_STRUCT
END_TYPE

```

bType: Type

Node: Node

bDomain: Domain

Member: Member

RPT_Timer: RPT Timer

Retry: Retry

RCV_Timer: RCV Timer

Tx_Timer: Tx Timer

Group: Group

7.4 Globale_Variablen_LON

Default value for all send function blocks.

```
VAR_GLOBAL CONSTANT
  tMinSendTimeDefault := t#1000ms,
  tMaxSendTimeDefault := t#0s,
  bAutoDefault        := FALSE,
  bSendInitDefault    := FALSE,
END_VAR
```

tMinSendTimeDefault: Default value for all send function blocks. Applies to the [automatic mode \[► 604\]](#). Minimal time distance to resend a changed value in automatic sending.

tMaxSendTimeDefault: Default value for all send function blocks. Applies to the [automatic mode \[► 604\]](#). Maximal time distance to resend a value in automatic sending.

bAutoDefault: Default value for all send function blocks. Turn on the [automatic \[► 604\]](#).

bSendInitDefault: Default value for all send function blocks. Send initial values.

7.5 Error codes

Library error messages



The NV index in the PLC is not compared with the NV index (column Id) in the KS2000 during sending. Wrong/invalid values can be sent if the indices do not match.

Sending without binding does no result in an error message.

Value (hex)	Value (dec)	Value (enum)	Description
0x0000	0	eLON_no_Error	No error is pending.
0x0001	1	eLON_Value_out_of_range	The input variable "Value" is outside the permitted range. The value was not sent. "Value" can have different formats with corresponding prefix (e.g. LREAL = IrValue).
0x0002	2	eLON_Terminal_not_ready	The function block "FB_LON_KL6401" passes through an initialization step chain (query terminal type, query firmware etc.) when the PLC is started. This message is issued as long as the initialization is in progress. If an error is pending after a PLC reset, the controller should be de-energized once.
0x0003	3	eLON_Wrong_SNVT_Typ	The received SNVT type does not match the SNVT type of the addressed NV index (input variable "wId").
0x0004	4	eLON_Wrong_wNVIndex	Incorrect NV index.
0x0005	5	eKL6401_Wrong_Terminal	No KL6401 was detected.
0x0006	6	eKL6401_Error	The function block "FB_LON_KL6401" has an error. The error code is shown at output "dwErrorKL".
0x0007	7	eKL6401_Terminal_is_not_initialized	The terminal is not initialized. This message usually means that there is no connection to the terminal. Terminal linked to the variables in the System Manager? Terminal plugged in incorrectly? Clean all, rebuild all and read again in the System Manager?

Value (hex)	Value (dec)	Value (enum)	Description
0x0032	50	eLON_L_star_Out_of_range	SNVT 70 / The input variable "stValue.L_star" is outside the permitted range. The value was not sent.
0x0033	51	eLON_A_star_Out_of_range	SNVT 70 / The input variable "stValue.A_star" is outside the permitted range. The value was not sent.
0x0034	52	eLON_B_star_Out_of_range	SNVT 70 / The input variable "stValue.B_star" is outside the permitted range. The value was not sent.
0x0037	55	eLON_eRequest_Out_of_range	SNVT 73 / The input variable "stValue.eRequest" is outside the permitted range. The value was not sent.
0x0042	66	eLON_wYear_Out_of_range	SNVT 084 / 088 / The input variable "stValue.wYear" is outside the permitted range. The value was not sent.
0x0043	67	eLON_wMonth_Out_of_range	SNVT 084 / 088 / The input variable "stValue.wMonth" is outside the permitted range. The value was not sent.
0x0044	68	eLON_wDay_Out_of_range	eLON_wDay_Out_of_range: SNVT 084 / 088 / The input variable "stValue.wDay" is outside the permitted range. The value was not sent.
0x0045	69	eLON_wHour_Out_of_range	SNVT 084 / 088 / The input variable "stValue.wHour" is outside the permitted range. The value was not sent.
0x0046	70	eLON_wMinute_Out_of_range	SNVT 084 / 088 / The input variable "stValue.wMinute" is outside the permitted range. The value was not sent.
0x0047	71	eLON_wSecond_Out_of_range	SNVT 084 / 088 / The input variable "stValue.wSecond" is outside the permitted range. The value was not sent.
0x0048	72	eLON_wMillisecond_Out_of_range	SNVT 088 / The input variable "stValue.wMillisecond" is outside the permitted range. The value was not sent.
0x0050	80	eLON_rZero_Out_of_range	SNVT 085 / The input variable "stValue.rZero" is outside the permitted range. The value was not sent.
0x0051	81	eLON_rSpan_Out_of_range	SNVT 085 / The input variable "stValue.rSpan" is outside the permitted range. The value was not sent.
0x0055	85	eLON_arrValue01_Out_of_range	SNVT 086 / The input variable "arrValue[1]" is outside the permitted range. The value was not sent.
0x0056	86	eLON_arrValue02_Out_of_range	SNVT 086 / The input variable "arrValue[2]" is outside the permitted range. The value was not sent.
0x0057	87	eLON_arrValue03_Out_of_range	SNVT 086 / The input variable "arrValue[3]" is outside the permitted range. The value was not sent.
0x0058	88	eLON_arrValue04_Out_of_range	SNVT 086 / The input variable "arrValue[4]" is outside the permitted range. The value was not sent.
0x0059	89	eLON_arrValue05_Out_of_range	SNVT 086 / The input variable "arrValue[5]" is outside the permitted range. The value was not sent.

Value (hex)	Value (dec)	Value (enum)	Description
0x005A	90	eLON_arrValue06_Out_of_range	SNVT 086 / The input variable "arrValue[6]" is outside the permitted range. The value was not sent.
0x005B	91	eLON_arrValue07_Out_of_range	SNVT 086 / The input variable "arrValue[7]" is outside the permitted range. The value was not sent.
0x005C	92	eLON_arrValue08_Out_of_range	SNVT 086 / The input variable "arrValue[8]" is outside the permitted range. The value was not sent.
0x005D	93	eLON_arrValue09_Out_of_range	SNVT 086 / The input variable "arrValue[9]" is outside the permitted range. The value was not sent.
0x0064	100	eLON_arrValue10_Out_of_range	SNVT 086 / The input variable "arrValue[10]" is outside the permitted range. The value was not sent.
0x0065	101	eLON_arrValue11_Out_of_range	SNVT 086 / The input variable "arrValue[11]" is outside the permitted range. The value was not sent.
0x0066	102	eLON_arrValue12_Out_of_range	SNVT 086 / The input variable "arrValue[12]" is outside the permitted range. The value was not sent.
0x0067	103	eLON_arrValue13_Out_of_range	SNVT 086 / The input variable "arrValue[13]" is outside the permitted range. The value was not sent.
0x0068	104	eLON_arrValue14_Out_of_range	SNVT 086 / The input variable "arrValue[14]" is outside the permitted range. The value was not sent.
0x0069	105	eLON_arrValue15_Out_of_range	SNVT 086 / The input variable "arrValue[15]" is outside the permitted range. The value was not sent.
0x006A	106	eLON_arrValue16_Out_of_range	SNVT 086 / The input variable "arrValue[16]" is outside the permitted range. The value was not sent.
0x006B	107	eLON_arrValue17_Out_of_range	SNVT 086 / The input variable "arrValue[17]" is outside the permitted range. The value was not sent.
0x006C	108	eLON_arrValue18_Out_of_range	SNVT 086 / The input variable "arrValue[18]" is outside the permitted range. The value was not sent.
0x006D	109	eLON_arrValue19_Out_of_range	SNVT 086 / The input variable "arrValue[19]" is outside the permitted range. The value was not sent.
0x0073	115	eLON_arrValue20_Out_of_range	SNVT 086 / The input variable "arrValue[20]" is outside the permitted range. The value was not sent.
0x0074	116	eLON_arrValue21_Out_of_range	SNVT 086 / The input variable "arrValue[21]" is outside the permitted range. The value was not sent.
0x0075	117	eLON_arrValue22_Out_of_range	SNVT 086 / The input variable "arrValue[22]" is outside the permitted range. The value was not sent.
0x0076	118	eLON_arrValue23_Out_of_range	SNVT 086 / The input variable "arrValue[23]" is outside the permitted range. The value was not sent.

Value (hex)	Value (dec)	Value (enum)	Description
0x0077	119	eLON_arrValue24_Out_of_range	SNVT 086 / The input variable "arrValue[24]" is outside the permitted range. The value was not sent.
0x0078	120	eLON_arrValue25_Out_of_range	SNVT 086 / The input variable "arrValue[25]" is outside the permitted range. The value was not sent.
0x0079	121	eLON_arrValue26_Out_of_range	SNVT 086 / The input variable "arrValue[26]" is outside the permitted range. The value was not sent.
0x007A	122	eLON_arrValue27_Out_of_range	SNVT 086 / The input variable "arrValue[27]" is outside the permitted range. The value was not sent.
0x007B	123	eLON_arrValue28_Out_of_range	SNVT 086 / The input variable "arrValue[28]" is outside the permitted range. The value was not sent.
0x007C	124	eLON_arrValue29_Out_of_range	SNVT 086 / The input variable "arrValue[29]" is outside the permitted range. The value was not sent.
0x0082	130	eLON_arrValue30_Out_of_range	SNVT 086 / The input variable "arrValue[30]" is outside the permitted range. The value was not sent.
0x0083	131	eLON_arrValue31_Out_of_range	SNVT 086 / The input variable "arrValue[31]" is outside the permitted range. The value was not sent.
0x0084	132	eLON_arrValue32_Out_of_range	SNVT 086 / The input variable "arrValue[32]" is outside the permitted range. The value was not sent.
0x0085	133	eLON_arrValue33_Out_of_range	SNVT 086 / The input variable "arrValue[33]" is outside the permitted range. The value was not sent.
0x0086	134	eLON_arrValue34_Out_of_range	SNVT 086 / The input variable "arrValue[34]" is outside the permitted range. The value was not sent.
0x0087	135	eLON_arrValue35_Out_of_range	SNVT 086 / The input variable "arrValue[35]" is outside the permitted range. The value was not sent.
0x0088	136	eLON_arrValue36_Out_of_range	SNVT 086 / The input variable "arrValue[36]" is outside the permitted range. The value was not sent.
0x0089	137	eLON_arrValue37_Out_of_range	SNVT 086 / The input variable "arrValue[37]" is outside the permitted range. The value was not sent.
0x008A	138	eLON_arrValue38_Out_of_range	SNVT 086 / The input variable "arrValue[38]" is outside the permitted range. The value was not sent.
0x008B	139	eLON_arrValue39_Out_of_range	SNVT 086 / The input variable "arrValue[39]" is outside the permitted range. The value was not sent.
0x008C	140	eLON_arrValue40_Out_of_range	SNVT 086 / The input variable "arrValue[40]" is outside the permitted range. The value was not sent.
0x0091	145	eLON_087uiDay_Out_of_range	SNVT 087 / The input variable "stValue.uiDay" is outside the permitted range. The value was not sent.

Value (hex)	Value (dec)	Value (enum)	Description
0x0092	146	eLON_087uiHour_Out_of_range	SNVT 087 / The input variable "stValue.uiHour" is outside the permitted range. The value was not sent.
0x0093	147	eLON_087uiMinute_Out_of_range	SNVT 087 / The input variable "stValue.uiMinute" is outside the permitted range. The value was not sent.
0x0094	148	eLON_087uiSecond_Out_of_range	SNVT 087 / The input variable "stValue.uiSecond" is outside the permitted range. The value was not sent.
0x0095	149	eLON_087uiMillisecond_Out_of_range	SNVT 087 / The input variable "stValue.uiMillisecond" is outside the permitted range. The value was not sent.
0x009B	155	eLON_ePriority_level_Out_of_range	SNVT 088 / The input variable "stValue.ePriority_level" is outside the permitted range. The value was not sent.
0x009C	156	eLON_eAlarm_type_Out_of_range	SNVT 088 / The input variable "stValue.eAlarm_type" is outside the permitted range. The value was not sent.
0x00A0	160	eLON_Currency_Out_of_range	SNVT 089 / The input variable "stValue.Currency" is outside the permitted range. The value was not sent.
0x00A5	165	eLON_diRw_ptr_Out_of_range	SNVT 090 / The input variable "stValue.diRw_ptr" is outside the permitted range. The value was not sent.
0x00AA	170	eLON_Object_request_Out_of_range	SNVT 092 / The input variable "stValue.Object_request" is outside the permitted range. The value was not sent.
0x00AF	175	eLON_094eLearn_Out_of_range	SNVT 094 / The input variable "stValue.eLearn" is outside the permitted range. The value was not sent.
0x00B0	176	eLON_094uiHour_Out_of_range	SNVT 094 / The input variable "stValue.uiHour" is outside the permitted range. The value was not sent.
0x00B1	177	eLON_094uiMinute_Out_of_range	SNVT 094 / The input variable "stValue.uiMinute" is outside the permitted range. The value was not sent.
0x00B2	178	eLON_094uiSecond_Out_of_range	SNVT 094 / The input variable "stValue.uiSecond" is outside the permitted range. The value was not sent.
0x00B3	179	eLON_094uiMillisecond_Out_of_range	SNVT 094 / The input variable "stValue.uiMillisecond" is outside the permitted range. The value was not sent.
0x00B9	185	eLON_095rValue_Out_of_range	SNVT 095 / The input variable "stValue.rValue" is outside the permitted range. The value was not sent.
0x00BA	186	eLON_095siState_Out_of_range	SNVT 095 / The input variable "stValue.siState" is outside the permitted range. The value was not sent.
0x00BE	190	eLON_byInterp_pts_0_to_1_Out_of_range	SNVT 096 / The input variable "stValue.byInterp_pts_0_to_1" is outside the permitted range. The value was not sent.
0x00BF	191	eLON_byInterp_pts_1_to_2_Out_of_range	SNVT 096 / The input variable "stValue.byInterp_pts_1_to_2" is outside the permitted range. The value was not sent.

Value (hex)	Value (dec)	Value (enum)	Description
0x00C0	192	eLON_byInterp_pts_2_to_3_Out_of_range	SNVT 096 / The input variable "stValue.byInterp_pts_2_to_3" is outside the permitted range. The value was not sent.
0x00C1	193	eLON_byInterp_pts_3_to_4_Out_of_range	SNVT 096 / The input variable "stValue.byInterp_pts_3_to_4" is outside the permitted range. The value was not sent.
0x00C2	194	eLON_byInterp_pts_4_to_5_Out_of_range	SNVT 096 / The input variable "stValue.byInterp_pts_4_to_5" is outside the permitted range. The value was not sent.
0x00C3	195	eLON_byInterp_pts_5_to_6_Out_of_range	SNVT 096 / The input variable "stValue.byInterp_pts_5_to_6" is outside the permitted range. The value was not sent.
0x00C4	196	eLON_byInterp_pts_6_to_0_Out_of_range	SNVT 096 / The input variable "stValue.byInterp_pts_6_to_0" is outside the permitted range. The value was not sent.
0x00C8	200	eLON_rOccupied_cool_Out_of_range	SNVT 106 / The input variable "stValue.rOccupied_cool" is outside the permitted range. The value was not sent.
0x00C9	201	eLON_rStandby_cool_Out_of_range	SNVT 106 / The input variable "stValue.rStandby_cool" is outside the permitted range. The value was not sent.
0x00CA	202	eLON_rUnoccupied_cool_Out_of_range	SNVT 106 / The input variable "stValue.rUnoccupied_cool" is outside the permitted range. The value was not sent.
0x00CB	203	eLON_rOccupied_heat_Out_of_range	SNVT 106 / The input variable "stValue.rOccupied_heat" is outside the permitted range. The value was not sent.
0x00CC	204	eLON_rStandby_heat_Out_of_range	SNVT 106 / The input variable "stValue.rStandby_heat" is outside the permitted range. The value was not sent.
0x00CD	205	eLON_rUnoccupied_heat_Out_of_range	SNVT 106 / The input variable "stValue.rUnoccupied_heat" is outside the permitted range. The value was not sent.
0x00D2	210	eLON_111rPercent_Out_of_range	SNVT 111 / The input variable "stValue.rPercent" is outside the permitted range. The value was not sent.
0x00D3	211	eLON_111eState_Out_of_range	SNVT 111 / The input variable "stValue.eState" is outside the permitted range. The value was not sent.
0x00D7	215	eLON_eMode_Out_of_range	SNVT 112 / The input variable "stValue.eMode" is outside the permitted range. The value was not sent.
0x00D8	216	eLON_rHeat_output_primary_Out_of_range	SNVT 112 / The input variable "stValue.rHeat_output_primary" is outside the permitted range. The value was not sent.
0x00D9	217	eLON_rHeat_output_secondary_Out_of_range	SNVT 112 / The input variable "stValue.rHeat_output_secondary" is outside the permitted range. The value was not sent.

Value (hex)	Value (dec)	Value (enum)	Description
0x00DA	218	eLON_rCool_output_Out_of_range	SNVT 112 / The input variable "stValue.rCool_output" is outside the permitted range. The value was not sent.
0x00DB	219	eLON_rEcon_output_Out_of_range	SNVT 112 / The input variable "stValue.rEcon_output" is outside the permitted range. The value was not sent.
0x00DC	220	eLON_rFan_output_Out_of_range	SNVT 112 / The input variable "stValue.rFan_output" is outside the permitted range. The value was not sent.
0x00E1	225	eLON_115eFunction_Out_of_range	SNVT 115 / The input variable "stValue.eFunction" is outside the permitted range. The value was not sent.
0x00E2	226	eLON_eFunction_Out_of_range	SNVT 116 / 117 / The input variable "stValue.eFunction" is outside the permitted range. The value was not sent.
0x00E3	227	eLON_rSetting_Out_of_range	SNVT 116 / 117 / The input variable "stValue.rSetting" is outside the permitted range. The value was not sent.
0x00E4	228	eLON_rRotation_Out_of_range	SNVT 116 / 117 / The input variable "stValue.rRotation" is outside the permitted range. The value was not sent.
0x00E5	229	eLON_rFade_time_Out_of_range	SNVT 116 / The input variable "stValue.rFade_time" is outside the permitted range. The value was not sent.
0x00E6	230	eLON_rDelay_time_Out_of_range	SNVT 116 / The input variable "stValue.rDelay_time" is outside the permitted range. The value was not sent.
0x00EB	235	eLON_eChlr_run_mode_Out_of_range	SNVT 127 / The input variable "stValue.eChlr_run_mode" is outside the permitted range. The value was not sent.
0x00EC	236	eLON_echlr_op_mode_Out_of_range	SNVT 127 / The input variable "stValue.echlr_op_mode" is outside the permitted range. The value was not sent.
0x00F0	240	eLON_eNext_state_Out_of_range	SNVT 128 / The input variable "stValue.eNext_state" is outside the permitted range. The value was not sent.
0x00F1	241	eLON_eCurrent_state_Out_of_range	SNVT 128 / The input variable "stValue.eCurrent_state" is outside the permitted range. The value was not sent.
0x00F5	245	eLON_diSecond_time_offset_Out_of_range	SNVT 134 / The input variable "stValue.diSecond_time_offset" is outside the permitted range. The value was not sent.
0x00F6	246	eLON_eType_of_description_Out_of_range	SNVT 134 / The input variable "stValue.eType_of_description" is outside the permitted range. The value was not sent.
0x00FA	250	eLON_byHour_of_start_DST_Out_of_range	SNVT 134 / The input variable "stValue.byHour_of_start_DST" is outside the permitted range. The value was not sent.
0x00FB	251	eLON_byMinute_of_start_DST_Out_of_range	SNVT 134 / The input variable "stValue.byMinute_of_start_DST" is outside the permitted range. The value was not sent.

Value (hex)	Value (dec)	Value (enum)	Description
0x00FC	252	eLON_bySecond_of_start_DST_Out_of_range	SNVT 134 / The input variable "stValue.bySecond_of_start_DST" is outside the permitted range. The value was not sent.
0x0104	260	eLON_byHour_of_end_DST_Out_of_range	SNVT 134 / The input variable "stValue.byHour_of_end_DST" is outside the permitted range. The value was not sent.
0x0105	261	eLON_byMinute_of_end_DST_Out_of_range	SNVT 134 / The input variable "stValue.byMinute_of_end_DST" is outside the permitted range. The value was not sent.
0x0106	262	eLON_bySecond_of_end_DST_Out_of_range	SNVT 134 / The input variable "stValue.bySecond_of_end_DST" is outside the permitted range. The value was not sent.
0x0107	263	eLON_stStart_DST_uiG_day_of_start_DST_Out_of_range	SNVT 134 / The input variable "stValue.stStart_DST.uiG_day_of_start_DST" is outside the permitted range. The value was not sent.
0x0108	264	eLON_stStart_DST_uiJ_day_of_start_DST_Out_of_range	SNVT 134 / The input variable "stValue.stStart_DST.uiJ_day_of_start_DST" is outside the permitted range. The value was not sent.
0x0109	265	eLON_stStart_DST_stM_start_DST_byMonth_of_start_DST_Out_of_range	SNVT 134 / The input variable "stValue.stStart_DST.stM_start_DST.byMonth_of_start_DST" is outside the permitted range. The value was not sent.
0x010A	266	eLON_stStart_DST_stM_start_DST_byWeek_of_start_DST_Out_of_range	SNVT 134 / The input variable "stValue.stStart_DST.stM_start_DST.byWeek_of_start_DST" is outside the permitted range. The value was not sent.
0x010B	267	eLON_stStart_DST_stM_start_DST_eDateday_of_start_DST_Out_of_range	SNVT 134 / The input variable "stValue.stStart_DST.stM_start_DST.eDateday_of_start_DST" is outside the permitted range. The value was not sent.
0x010C	268	eLON_stEnd_DST_uiG_day_of_end_DST_Out_of_range	SNVT 134 / The input variable "stValue.stEnd_DST.uiG_day_of_end_DST" is outside the permitted range. The value was not sent.
0x010D	269	eLON_stEnd_DST_uiJ_day_of_end_DST_Out_of_range	SNVT 134 / The input variable "stValue.stEnd_DST.uiJ_day_of_end_DST" is outside the permitted range. The value was not sent.
0x010E	270	eLON_stEnd_DST_stM_end_DST_byMonth_of_end_DST_Out_of_range	SNVT 134 / The input variable "stValue.stEnd_DST.stM_end_DST.byMonth_of_end_DST" is outside the permitted range. The value was not sent.
0x010F	271	eLON_stEnd_DST_stM_end_DST_byWeek_of_end_DST_Out_of_range	SNVT 134 / The input variable "stValue.stEnd_DST.stM_end_DST.byWeek_of_end_DST" is outside the permitted range. The value was not sent.
0x0110	272	eLON_stEnd_DST_stM_end_DST_eDateday_of_end_DST_Out_of_range	SNVT 134 / The input variable "stValue.stEnd_DST.stM_end_DST.eDateday_of_end_DST" is outside the permitted range. The value was not sent.

Value (hex)	Value (dec)	Value (enum)	Description
0x0118	280	eLON_byLatitude_deg_Out_of_range	SNVT 135 / The input variable "stValue.byLatitude" is outside the permitted range. The value was not sent.
0x0119	281	eLON_rLatitude_min_Out_of_range	SNVT 135 / The input variable "stValue.rLatitude" is outside the permitted range. The value was not sent.
0x011A	282	eLON_bylongitude_deg_Out_of_range	SNVT 135 / The input variable "stValue.bylongitude_deg" is outside the permitted range. The value was not sent.
0x011B	283	eLON_rLongitude_min_Out_of_range	SNVT 135 / The input variable "stValue.rLongitude_min" is outside the permitted range. The value was not sent.
0x0122	290	eLON_byNr_decimals_Out_of_range	SNVT 136 / The input variable "stValue.byNr_decimals" is outside the permitted range. The value was not sent.
0x0123	291	eLON_eUnit_Out_of_range	SNVT 136 / The input variable "stValue.eUnit" is outside the permitted range. The value was not sent.
0x0127	295	eLON_137eUnit_Out_of_range	SNVT 137 / The input variable "stValue.eUnit" is outside the permitted range. The value was not sent.
0x0128	296	eLON_137byNr_decimals_Out_of_range	SNVT 137 / The input variable "stValue.byNr_decimals" is outside the permitted range. The value was not sent.
0x0129	297	eLON_137byStatus_Out_of_range	SNVT 137 / The input variable "stValue.byStatus" is outside the permitted range. The value was not sent.
0x012A	298	eLON_137uiYear_Out_of_range	SNVT 137 / The input variable "stValue.uiYear" is outside the permitted range. The value was not sent.
0x012B	299	eLON_137uiMonth_Out_of_range	SNVT 137 / The input variable "stValue.uiMonth" is outside the permitted range. The value was not sent.
0x012C	300	eLON_137uiDay_Out_of_range	SNVT 137 / The input variable "stValue.uiDay" is outside the permitted range. The value was not sent.
0x012D	301	eLON_137uiHour_Out_of_range	SNVT 137 / The input variable "stValue.uiHour" is outside the permitted range. The value was not sent.
0x012E	302	eLON_137uiMinute_Out_of_range	SNVT 137 / The input variable "stValue.uiMinute" is outside the permitted range. The value was not sent.
0x012F	303	eLON_137uiSecond_Out_of_range	SNVT 137 / The input variable "stValue.uiSecond" is outside the permitted range. The value was not sent.
0x0136	310	eLON_bySender_prio_Out_of_range	SNVT 148 / The input variable "stValue.bySender_prio" is outside the permitted range. The value was not sent.
0x013B	315	eLON_eStatus_Out_of_range	SNVT 149 / The input variable "stValue.eStatus" is outside the permitted range. The value was not sent.
0x013C	316	eLON_stSender_uid_Out_of_range	SNVT 149 / The input variable "stValue.stSender.uid" is outside the permitted range. The value was not sent.

Value (hex)	Value (dec)	Value (enum)	Description
0x013D	317	eLON_stSender_stRange_uiLower_Out_of_range	SNVT 149 / The input variable "stValue.stSender.stRange.uiLower" is outside the permitted range. The value was not sent.
0x013E	318	eLON_stSender_stRange_uiUpper_Out_of_range	SNVT 149 / The input variable "stValue.stSender.stRange.uiUpper" is outside the permitted range. The value was not sent.
0x013F	319	eLON_uiController_id_Out_of_range	SNVT 149 / The input variable "stValue.uiController" is outside the permitted range. The value was not sent.
0x0145	325	eLON_ePan_dir_Out_of_range	SNVT 150 / The input variable "stValue." is outside the permitted range. The value was not sent.
0x0146	326	eLON_rPan_speed_Out_of_range	SNVT 150 / The input variable "stValue." is outside the permitted range. The value was not sent.
0x0147	327	eLON_eTilt_dir_Out_of_range	SNVT 150 / The input variable "stValue." is outside the permitted range. The value was not sent.
0x0148	328	eLON_rTilt_speed_Out_of_range	SNVT 150 / The input variable "stValue." is outside the permitted range. The value was not sent.
0x0149	329	eLON_eZoom_Out_of_range	SNVT 150 / The input variable "stValue." is outside the permitted range. The value was not sent.
0x014A	330	eLON_rZoom_speed_Out_of_range	SNVT 150 / The input variable "stValue." is outside the permitted range. The value was not sent.
0x014F	335	eLON_eAction_Out_of_range	SNVT 151 / The input variable "stValue.eAction" is outside the permitted range. The value was not sent.
0x0154	340	eLON_byController_prio_Out_of_range	SNVT 152 / The input variable "stValue.byController" is outside the permitted range. The value was not sent.
0x0155	341	eLON_152eFunction_Out_of_range	SNVT 152 / The input variable "stValue.eFunction" is outside the permitted range. The value was not sent.
0x0156	342	eLON_152eAction_Out_of_range	SNVT 152 / The input variable "stValue.eAction" is outside the permitted range. The value was not sent.
0x0157	343	eLON_stValue_stAbspos_rZoom_Out_of_range	SNVT 152 / The input variable "stValue.stValue.stAbspos.rZoom" is outside the permitted range. The value was not sent.
0x0158	344	eLON_stValue_stAbspos_rTilt_Out_of_range	SNVT 152 / The input variable "stValue.stValue.stAbspos.rTilt" is outside the permitted range. The value was not sent.
0x0159	345	eLON_stValue_stAbspos_rPan_Out_of_range	SNVT 152 / The input variable "stValue.stValue.stAbspos.rPan" is outside the permitted range. The value was not sent.
0x015E	350	eLON_eMain_pump_Out_of_range	SNVT 156 / The input variable "stValue.eMain_pump" is outside the permitted range. The value was not sent.

Value (hex)	Value (dec)	Value (enum)	Description
0x015F	351	eLON_eBooster_pump_Out_of_range	SNVT 156 / The input variable "stValue.eBooster_pump" is outside the permitted range. The value was not sent.
0x0160	352	eLON_ePriority_level_Out_of_range	SNVT 156 / The input variable "stValue.ePriority_level" is outside the permitted range. The value was not sent.
0x0161	353	eLON_eProcess_ready_Out_of_range	SNVT 156 / The input variable "stValue.eProcess_ready" is outside the permitted range. The value was not sent.
0x0162	354	eLON_eEmergency_stop_activated_Out_of_range	SNVT 156 / The input variable "stValue.eEmergency_stop_activated" is outside the permitted range. The value was not sent.
0x0163	355	eLON_eMain_pump_drive_enabled_Out_of_range	SNVT 156 / The input variable "stValue.eMain_pump_drive_enabled" is outside the permitted range. The value was not sent.
0x0164	356	eLON_eBooster_pump_drive_enabled_Out_of_range	SNVT 156 / The input variable "stValue.eBooster_pump_drive_enabled" is outside the permitted range. The value was not sent.
0x0165	357	eLON_eMaintenance_required_Out_of_range	SNVT 156 / The input variable "stValue.eMaintenance_required" is outside the permitted range. The value was not sent.
0x016D	365	eLON_eControl_status_Out_of_range	SNVT 157 / The input variable "stValue.eControl_status" is outside the permitted range. The value was not sent.
0x016E	366	eLON_stControl_device_addr_byDomain_length_Out_of_range	SNVT 157 / The input variable "stValue.stControl_device_addr.byDomain_length" is outside the permitted range. The value was not sent.
0x016F	367	eLON_stControl_device_addr_bySubnet_Out_of_range	SNVT 157 / The input variable "stValue.stControl_device_addr.bySubnet" is outside the permitted range. The value was not sent.
0x0170	368	eLON_stControl_device_addr_byNode_Out_of_range	SNVT 157 / The input variable "stValue.stControl_device_addr.byNode" is outside the permitted range. The value was not sent.
0x0177	375	eLON_rExhaust_temperature_Out_of_range	SNVT 158 / The input variable "stValue.rExhaust_temperature" is outside the permitted range. The value was not sent.
0x0178	376	eLON_rExhaust_pressure_Out_of_range	SNVT 158 / The input variable "stValue.rExhaust_pressure" is outside the permitted range. The value was not sent.
0x0179	377	eLON_rShaft_seal_purge_pressure_Out_of_range	SNVT 158 / The input variable "stValue.rShaft_seal_purge_pressure" is outside the permitted range. The value was not sent.
0x017A	378	eLON_rSupply_voltage_Out_of_range	SNVT 158 / The input variable "stValue.rSupply_voltage" is outside the permitted range. The value was not sent.

Value (hex)	Value (dec)	Value (enum)	Description
0x017B	379	eLON_eCoolant_flow_low_Out_of_range	SNVT 158 / The input variable "stValue.eCoolant_flow_low" is outside the permitted range. The value was not sent.
0x017C	380	eLON_eDilution_active_Out_of_range	SNVT 158 / The input variable "stValue.eDilution_active" is outside the permitted range. The value was not sent.
0x017D	381	eLON_eBallast_dilution_active_Out_of_range	SNVT 158 / The input variable "stValue.eBallast_dilution_active" is outside the permitted range. The value was not sent.
0x017E	382	eLON_eInlet_purge_dilution_active_Out_of_range	SNVT 158 / The input variable "stValue.eInlet_purge_dilution_active" is outside the permitted range. The value was not sent.
0x017F	383	eLON_eExhaust_dilution_active_Out_of_range	SNVT 158 / The input variable "stValue.eExhaust_dilution_active" is outside the permitted range. The value was not sent.
0x0180	384	eLON_eDilution_flow_Out_of_range	SNVT 158 / The input variable "stValue.eDilution_flow" is outside the permitted range. The value was not sent.
0x0181	385	eLON_ePower_supply_on_Out_of_range	SNVT 158 / The input variable "stValue.ePower_supply_on" is outside the permitted range. The value was not sent.
0x0186	390	eLON_rRotational_speed_Out_of_range	SNVT 159 / The input variable "stValue.rRotational_speed" is outside the permitted range. The value was not sent.
0x0187	391	eLON_rBody_temperature_Out_of_range	SNVT 159 / The input variable "stValue.rBody" is outside the permitted range. The value was not sent.
0x0188	392	eLON_rMotor_external_temperature_Out_of_range	SNVT 159 / The input variable "stValue.rMotor_external_temperature" is outside the permitted range. The value was not sent.
0x0189	393	eLON_rMotor_internal_temperature_Out_of_range	SNVT 159 / The input variable "stValue.eMotor_overloaded" is outside the permitted range. The value was not sent.
0x018A	394	eLON_eMotor_overloaded_Out_of_range	SNVT 159 / The input variable "stValue." is outside the permitted range. The value was not sent.
0x018B	395	eLON_eOil_level_low_Out_of_range	SNVT 159 / The input variable "stValue.ePhase_imbalance_detected" is outside the permitted range. The value was not sent.
0x018C	396	eLON_ePhase_imbalance_detected_Out_of_range	SNVT 159 / The input variable "stValue." is outside the permitted range. The value was not sent.
0x018D	397	eLON_rCurrent_usage_Out_of_range	SNVT 159 / The input variable "stValue.rCurrent_usage" is outside the permitted range. The value was not sent.
0x018E	398	eLON_rPower_usage_Out_of_range	SNVT 159 / The input variable "stValue.Power_usage" is outside the permitted range. The value was not sent.

Value (hex)	Value (dec)	Value (enum)	Description
0x018F	399	eLON_eTemperature_control_Out_of_range	SNVT 159 / The input variable "stValue.eElectromagnetic_brake_active" is outside the permitted range. The value was not sent.
0x0190	400	eLON_eElectromagnetic_brake_active_Out_of_range	SNVT 159 / The input variable "stValue." is outside the permitted range. The value was not sent.
0x0191	401	eLON_eFriction_brake_active_Out_of_range	SNVT 159 / The input variable "stValue.eFriction_brake_active" is outside the permitted range. The value was not sent.
0x0192	402	eLON_eGas_brake_active_Out_of_range	SNVT 159 / The input variable "stValue.eGas_brake_active" is outside the permitted range. The value was not sent.
0x019A	410	eLON_164iMilliseconds_Out_of_range	SNVT 164 / The input variable "stValue.iMilliseconds" is outside the permitted range. The value was not sent.
0x019B	411	eLON_164ePriority_level_Out_of_range	SNVT 164 / The input variable "stValue.ePriority_level" is outside the permitted range. The value was not sent.
0x019C	412	eLON_164eAlarm_type_Out_of_range	SNVT 164 / The input variable "stValue.eAlarm" is outside the permitted range. The value was not sent.
0x01A4	420	eLON_byType_scope_Out_of_range	SNVT 166 / The input variable "stValue.byType_scope" is outside the permitted range. The value was not sent.
0x01A5	421	eLON_uiType_index_Out_of_range	SNVT 166 / The input variable "stValue.uiType_index" is outside the permitted range. The value was not sent.
0x01A6	422	eLON_eType_category_Out_of_range	SNVT 166 / The input variable "stValue.eType_category" is outside the permitted range. The value was not sent.
0x01A7	423	eLON_byType_length_Out_of_range	SNVT 166 / The input variable "stValue.byType" is outside the permitted range. The value was not sent.
0x01AE	430	eLON_eCmd_fb_Out_of_range	SNVT 170 / The input variable "stValue.eCmd_fb" is outside the permitted range. The value was not sent.
0x01B3	435	eLON_byManufacturer_Out_of_range	SNVT 172 / The input variable "stValue.byManufacturer" is outside the permitted range. The value was not sent.
0x01B8	440	eLON_eDevice_select_Out_of_range	SNVT 175 / The input variable "stValue.eDevice_select" is outside the permitted range. The value was not sent.
0x01BD	445	eLON_stPos_eFunction_Out_of_range	SNVT 180 / The input variable "stValue.stPos_eFunction" is outside the permitted range. The value was not sent.
0x01BE	446	eLON_stPos_rSetting_Out_of_range	SNVT 180 / The input variable "stValue.stPos.rSetting" is outside the permitted range. The value was not sent.
0x01BF	447	eLON_stPos_rRotation_Out_of_range	SNVT 180 / The input variable "stValue.stPos.rRotation" is outside the permitted range. The value was not sent.
0x01C0	448	eLON_eCmd_source_Out_of_range	SNVT 180 / The input variable "stValue.eCmd_source" is outside the permitted range. The value was not sent.

Value (hex)	Value (dec)	Value (enum)	Description
0x01C1	449	eLON_eError_code_Out_of_range	SNVT 180 / The input variable "stValue._eError_code" is outside the permitted range. The value was not sent.
0x01C7	455	eLON_181stAddr_talk_eAudio_sensor_type_Out_of_range	SNVT 181 / The input variable "stValue.stAddr_talk.eAudio_sensor" is outside the permitted range. The value was not sent.
0x01C8	456	eLON_181stAddr_talk_byCar_id_Out_of_range	SNVT 181 / The input variable "stValue.stAddr_talk.byCar_id" is outside the permitted range. The value was not sent.
0x01C9	457	eLON_181stAddr_talk_byLocation_Out_of_range	SNVT 181 / The input variable "stValue.stAddr_talk.byLocation" is outside the permitted range. The value was not sent.
0x01CA	458	eLON_181stAddr_talk_byUnit_id_Out_of_range	SNVT 181 / The input variable "stValue.stAddr_talk.byUnit" is outside the permitted range. The value was not sent.
0x01CB	459	eLON_181stAddr_init_eAudio_sensor_type_Out_of_range	SNVT 181 / The input variable "stValue.stAddr_init.eAudio_sensor_type" is outside the permitted range. The value was not sent.
0x01C2	450	eLON_181stAddr_init_byCar_id_Out_of_range	SNVT 181 / The input variable "stValue.stAddr_init.byCar" is outside the permitted range. The value was not sent.
0x01CD	461	eLON_181stAddr_init_byLocation_Out_of_range	SNVT 181 / The input variable "stValue.stAddr_init.byLocation" is outside the permitted range. The value was not sent.
0x01CE	462	eLON_181stAddr_init_byUnit_id_Out_of_range	SNVT 181 / The input variable "stValue.stAddr_init.byUnit" is outside the permitted range. The value was not sent.
0x01CF	463	eLON_181eAudio_type_Out_of_range	SNVT 181 / The input variable "stValue.eAudio_type" is outside the permitted range. The value was not sent.
0x01D0	464	eLON_181byAudio_line_Out_of_range	SNVT 181 / The input variable "stValue.byAudio_line" is outside the permitted range. The value was not sent.
0x01D1	465	eLON_181stAddr_dest_stP2p_eAudio_sensor_type_Out_of_range	SNVT 181 / The input variable "stValue.stAddr_dest.stP2p.eAudio_sensor_type" is outside the permitted range. The value was not sent.
0x01D2	466	eLON_181stAddr_dest_stP2p_byCar_id_Out_of_range	SNVT 181 / The input variable "stValue.stAddr_dest.stP2p.byLocation" is outside the permitted range. The value was not sent.
0x01D3	467	eLON_181stAddr_dest_stP2p_byLocation_Out_of_range	SNVT 181 / The input variable "stValue." is outside the permitted range. The value was not sent.
0x01D4	468	eLON_181stAddr_dest_stP2p_byUnit_id_Out_of_range	SNVT 181 / The input variable "stValue.stAddr_dest.stP2p.byUnit_id" is outside the permitted range. The value was not sent.

Value (hex)	Value (dec)	Value (enum)	Description
0x01DB	475	eLON_stAddr_dest_stP2p_eAudio_sens or_type_Out_of_range	SNVT 182 / The input variable "stValue.stAddr_dest.stP2p.eAudio_sensor_type" is outside the permitted range. The value was not sent.
0x01DC	476	eLON_stAddr_dest_stP2p_byCar_id_Ou t_of_range	SNVT 182 / The input variable "stValue.stAddr_dest.stP2p.byCar" is outside the permitted range. The value was not sent.
0x01DD	477	eLON_stAddr_dest_stP2p_byLocation_ Out_of_range	SNVT 182 / The input variable "stValue.stAddr_dest.stP2p.byLocation" is outside the permitted range. The value was not sent.
0x01DE	478	eLON_stAddr_dest_stP2p_byUnit_id_O ut_of_range	SNVT 182 / The input variable "stValue.stAddr_init.eAudio_sensor_type" is outside the permitted range. The value was not sent.
0x01DF	479	eLON_stAddr_init_eAudio_sensor_type_ Out_of_range	SNVT 182 / The input variable "stValue.stAddr_init.byCar_id" is outside the permitted range. The value was not sent.
0x01E0	480	eLON_stAddr_init_byCar_id_Out_of_ran ge	SNVT 182 / The input variable "stValue." is outside the permitted range. The value was not sent.
0x01E1	481	eLON_stAddr_init_byLocation_Out_of_r ange	SNVT 182 / The input variable "stValue.stAddr_init.byLocation" is outside the permitted range. The value was not sent.
0x01E2	482	eLON_stAddr_init_byUnit_id_Out_of_ran ge	SNVT 182 / The input variable "stValue.stAddr_init.byUnit_id" is outside the permitted range. The value was not sent.
0x01E3	483	eLON_eAudio_type_Out_of_range	SNVT 182 / The input variable "stValue.eAudio_type" is outside the permitted range. The value was not sent.
0x01EA	490	eLON_eCycle_Out_of_range	SNVT 184 / The input variable "stValue.eCycle" is outside the permitted range. The value was not sent.
0x01EB	491	eLON_eSubcycle_Out_of_range	SNVT 184 / The input variable "stValue.eSubcycle" is outside the permitted range. The value was not sent.
0x01EC	492	eLON_stFunction_eProgram_Out_of_ra nge	SNVT 184 / The input variable "stValue.stFunction.eProgram" is outside the permitted range. The value was not sent.
0x01ED	493	eLON_stFunction_stWash_eLoad_level_ Out_of_range	SNVT 184 / The input variable "stValue.stFunction.stWash_eLoad_level" is outside the permitted range. The value was not sent.
0x01EE	494	eLON_stFunction_stWash_ePrewash_O ut_of_range	SNVT 184 / The input variable "stValue.stFunction.stWash.ePrewash" is outside the permitted range. The value was not sent.
0x01EF	495	eLON_stFunction_stRinse_eOption_Out _of_range	SNVT 184 / The input variable "stValue.stFunction.stRinse.eOption" is outside the permitted range. The value was not sent.

Value (hex)	Value (dec)	Value (enum)	Description
0x01F0	496	eLON_stFunction_stRinse_byRepeat_Out_of_range	SNVT 184 / The input variable "stValue.stFunction.stRinse.byRepeat" is outside the permitted range. The value was not sent.
0x01F1	497	eLON_stFunction_stSpin_eHold_Out_of_range	SNVT 184 / The input variable "stValue.stFunction.stSpin.eHold" is outside the permitted range. The value was not sent.
0x01F2	498	eLON_stFunction_stDry_byTemp_Out_of_range	SNVT 184 / The input variable "stValue.stFunction.stDry.byTemp" is outside the permitted range. The value was not sent.
0x01F3	499	eLON_stFunction_stDry_stDuration_eDryness_Out_of_range	SNVT 184 / The input variable "stValue.stFunction.stDry.stDuration.eDryness" is outside the permitted range. The value was not sent.
0x01F9	505	eLON_186eCycle_Out_of_range	SNVT 186 / The input variable "stValue.eCycle" is outside the permitted range. The value was not sent.
0x01FA	506	eLON_186eSubcycle_Out_of_range	SNVT 186 / The input variable "stValue.eSubcycle" is outside the permitted range. The value was not sent.
0x01FB	507	eLON_stWasher_command_data_eCycle_Out_of_range	SNVT 186 / The input variable "stValue.stWasher_command_data.eCycle" is outside the permitted range. The value was not sent.
0x01FC	508	eLON_stWasher_command_data_eSubcycle_Out_of_range	SNVT 186 / The input variable "stValue.stWasher_command_data.eSubcycle" is outside the permitted range. The value was not sent.
0x01FD	509	eLON_stWasher_command_data_stFunction_eProgram_Out_of_range	SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.eProgram" is outside the permitted range. The value was not sent.
0x01FE	510	eLON_stWasher_command_data_stFunction_stWash_eLoad_level_Out_of_range	SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.stWash.eLoad" is outside the permitted range. The value was not sent.
0x01FF	511	eLON_stWasher_command_data_stFunction_stWash_ePrewash_Out_of_range	SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.stWash.ePrewash" is outside the permitted range. The value was not sent.
0x0200	512	eLON_stWasher_command_data_stFunction_stRinse_eOption_Out_of_range	SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.stRinse.eOption" is outside the permitted range. The value was not sent.
0x0201	513	eLON_stWasher_command_data_stFunction_stRinse_byRepeat_Out_of_range	SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.stRinse.byRepeat" is outside the permitted range. The value was not sent.
0x0202	514	eLON_stWasher_command_data_stFunction_stSpin_eHold_Out_of_range	SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.stSpin.eHold" is outside the permitted range. The value was not sent.

Value (hex)	Value (dec)	Value (enum)	Description
0x0203	515	eLON_stWasher_command_data_stFunction_stDry_byTemp_Out_of_range	SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.stDry.byTemp" is outside the permitted range. The value was not sent.
0x0204	516	eLON_stWasher_command_data_stFunction_stDry_stDuration_eDryness_Out_of_range	SNVT 186 / The input variable "stValue.stWasher_command_data.stFunction.stDry.stDuration.eDryness" is outside the permitted range. The value was not sent.
0x0206	518	eLON_eState_Out_of_range	SNVT 189 / The input variable "stValue.eState" is outside the permitted range. The value was not sent.
0x0207	519	eLON_stSetting_rValue_Out_of_range	SNVT 189 / The input variable "stValue.stSettings.rValue" is outside the permitted range. The value was not sent.
0x0208	520	eLON_stSetting_rChange_Out_of_range	SNVT 189 / The input variable "stValue.stSettings.rChange" is outside the permitted range. The value was not sent.
0x0209	521	eLON_stSetting_rMultiplier_Out_of_range	SNVT 189 / The input variable "stValue.stSettings.rMultiplier" is outside the permitted range. The value was not sent.
0x020A	522	eLON_stSetting_iAngle_Out_of_range	SNVT 189 / The input variable "stValue.stSettings.iAngle" is outside the permitted range. The value was not sent.
0x020B	523	eLON_stSetting_byGroup_number_Out_of_range	SNVT 189 / The input variable "stValue.stSettings.byGroup_number" is outside the permitted range. The value was not sent.
0x020C	524	eLON_stSetting_siFan_level_Out_of_range	SNVT 189 / The input variable "stValue.stSettings.siFan_level" is outside the permitted range. The value was not sent.
0x020D	525	eLON_stColor_value_stCIE1931_lumen_rX_Out_of_range	SNVT 190 / The input variable "stValue.stColor_value.stCIE1931_lumen.rX" is outside the permitted range. The value was not sent.
0x020E	526	eLON_stColor_value_stCIE1931_lumen_rY_Out_of_range	SNVT 190 / The input variable "stValue.stColor_value.stCIE1931_lumen.rY" is outside the permitted range. The value was not sent.
0x020F	527	eLON_stColor_value_stCIE1931_lumen_udiAbsolute_Y_Out_of_range	SNVT 190 / The input variable "stValue.stColor_value.stCIE1931_lumen.udiAbsolute_Y" is outside the permitted range. The value was not sent.
0x0210	528	eLON_stColor_value_stCIE1931_percent_rX_Out_of_range	SNVT 190 / The input variable "stValue.stColor_value.stCIE1931_percent.rX" is outside the permitted range. The value was not sent.
0x0211	529	eLON_stColor_value_stCIE1931_percent_rY_Out_of_range	SNVT 190 / The input variable "stValue.stColor_value.stCIE1931_percent.rY" is outside the permitted range. The value was not sent.
0x0212	530	eLON_stColor_value_stCIE1931_percent_rPercent_Y_Out_of_range	SNVT 190 / The input variable "stValue.stColor_value.stCIE1931_percent.rPercent_Y" is outside the permitted range. The value was not sent.

Value (hex)	Value (dec)	Value (enum)	Description
0x0213	531	eLON_stColor_value_uiColor_temperature_Out_of_range	SNVT 190 / The input variable "stValue.stColor_value_uiColor_temperature" is outside the permitted range. The value was not sent.
0x0217	535	eLON_191eStatus_Out_of_range	SNVT 191 / The input variable "stValue.Status" is outside the permitted range. The value was not sent.
0x0218	536	eLON_uiLog_number_Out_of_range	SNVT 191 / The input variable "stValue.uiLog_number" is outside the permitted range. The value was not sent.
0x0219	537	eLON_rLevel_Out_of_range	SNVT 191 / The input variable "stValue.rLevel" is outside the permitted range. The value was not sent.
0x021A	538	eLON_stCurrent_notify_time_rHundredths_Out_of_range	SNVT 191 / The input variable "stValue.stCurrent_notify_time_rHundredths" is outside the permitted range. The value was not sent.
0x021B	539	eLON_stPrevious_notify_time_rHundredths_Out_of_range	SNVT 191 / The input variable "stValue.stPrevious_notify_time_rHundredths" is outside the permitted range. The value was not sent.
0x0221	545	eLON_rHundredths_Out_of_range	SNVT 192 / The input variable "stValue.rHundredths" is outside the permitted range. The value was not sent.
0x0226	550	eLON_stStart_time_rHundredths_Out_of_range	SNVT 193 / The input variable "stValue.stStart_time_rHundredths" is outside the permitted range. The value was not sent.
0x0227	551	eLON_stEnd_time_rHundredths_Out_of_range	SNVT 193 / The input variable "stValue.stEnd_time_rHundredths" is outside the permitted range. The value was not sent.
0x0235	565	eLON_rComplete_Out_of_range	SNVT 194 / The input variable "stValue.rComplete" is outside the permitted range. The value was not sent.
0x023A	570	eLON_stTime_actual_rHundredths_Out_of_range	SNVT 199 / The input variable "stValue.stTime_actual_rHundredths" is outside the permitted range. The value was not sent.
0x023B	571	eLON_stTime_previous_rHundredths_Out_of_range	SNVT 199 / The input variable "stValue.stTime_previous_rHundredths" is outside the permitted range. The value was not sent.
0x0249	585	eLON_lrEnergy_Out_of_range	SNVT 200 / The input variable "stValue.lrEnergy" is outside the permitted range. The value was not sent.
0x024A	586	eLON_rPowerFactor_Out_of_range	SNVT 200 / The input variable "stValue.rPowerFactor" is outside the permitted range. The value was not sent.
0x024B	587	eLON_rPower_Out_of_range	SNVT 200 / The input variable "stValue.rPower" is outside the permitted range. The value was not sent.
0x024C	588	eLON_rBallastTemp_Out_of_range	SNVT 200 / The input variable "stValue.rBallastTemp" is outside the permitted range. The value was not sent.

Value (hex)	Value (dec)	Value (enum)	Description
0x0253	595	eLON_lrLongitude_Out_of_range	SNVT 201 / The input variable "stValue.lrLongitude" is outside the permitted range. The value was not sent.
0x0254	596	eLON_lrLatitude_Out_of_range	SNVT 201 / The input variable "stValue.lrLatitude" is outside the permitted range. The value was not sent.

8 Appendix

8.1 Automatic sending

Automatic sending is enabled with the input variable **bAuto**. The variable must be TRUE during the whole interval over which the block is to send independently.

The following three parameters (VAR_INPUT) can be used to influence automatic sending.

MaxSendTime: TIME;

This value enables transfers of values at regular intervals. The value is sent once the time has elapsed, irrespective of any change in value.

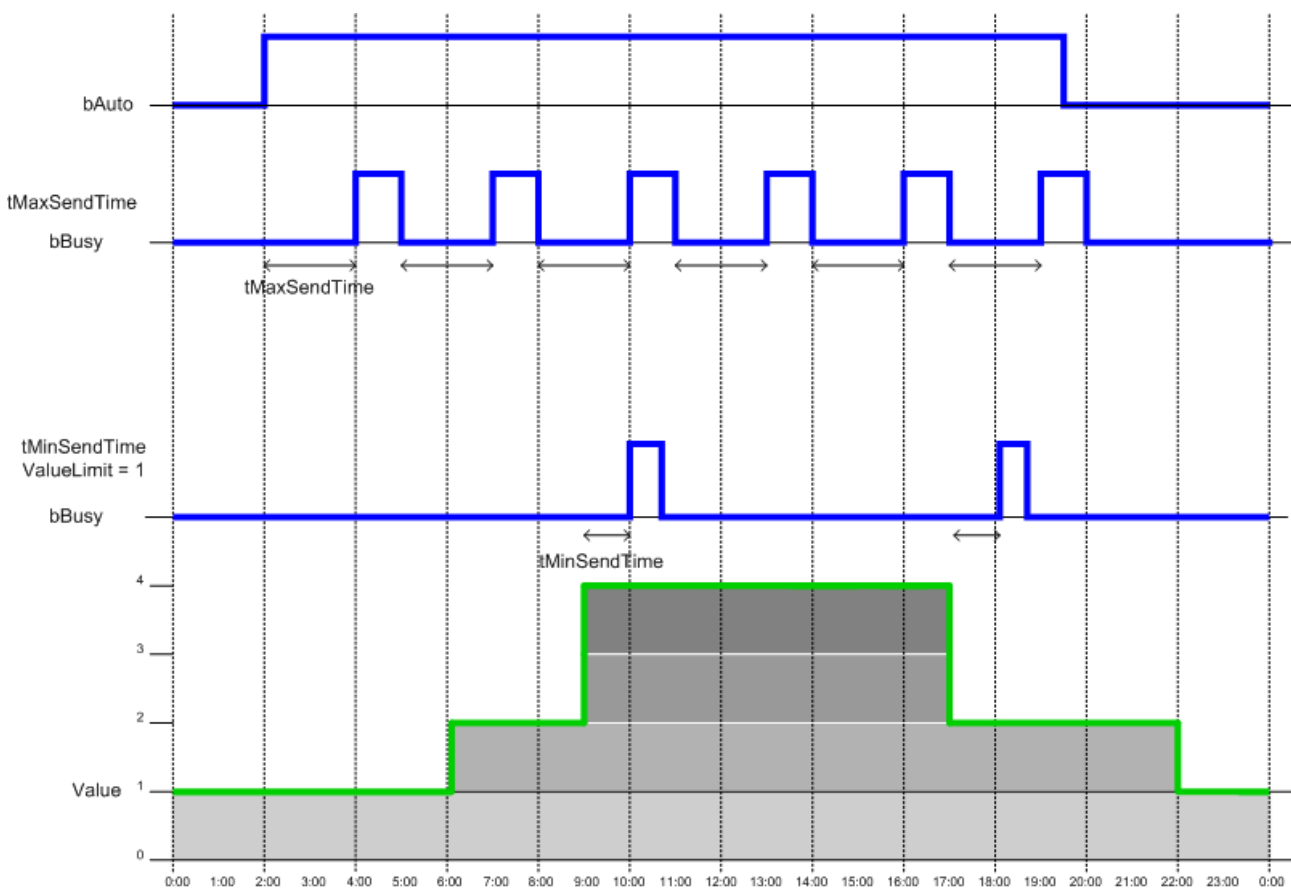
ValueLimit: This value can have the following formats: (r/lr/ui/i) **ValueLimit:** REAL / LREAL / UINT / INT.

The value is only sent if the absolute value of the change since the last transfer is greater than this parameter. If this value is 0, sending takes place after each change in value (even very small changes). Please note: The variable **ValueLimit** is not used for enums and structures. In this case sending takes place after each change in value.

tMinSendTime: TIME;

The block starts sending after **tMinSendTime** at the earliest. This parameter can be used to limit the number of telegrams in situations where the values change very quickly. (Reduction of network load). If the value is t#0s, sending takes place after each change in value (see **ValueLimit**).

Fig. 1



8.2 dwErrorKL


Value (hex)	Firmware	Description
0x0000 0000		No error.

Value (hex)	Firmware	Description
0x0000 0001		Write access to a read parameter.
0x0000 0004		Undefined parameter.
0x0000 0005		Illegal value for parameter.
0x0000 0007		Undefined slot.
0x0000 0008		Error when reading the NV parameter.
0x0000 0009		Checksum error when downloading the LON configuration.
0x0000 0010	from FW5	OUTGOING_MSG_FAILED
0x0000 0011	from FW5	OUTGOING_MSG_LATE_ACK
0x0000 0012	from FW5	OUTGOING_MSG_MALFORMED
0x0000 0013	from FW5	NEURON_QUERY_FAILED
0x0000 0014	from FW5	NEURON_UPDATE_FAILED
0x0000 000A		Fault with the upload of a LON configuration.
0x0000 0Cxx		An SNVT variable that is to be written has not arrived. xx corresponds to the SNVT index number.
0x0000 0Exx		NV index is not an output. The PLC attempts to write to an NV index that was not defined as an output with the KS2000 (NVO). xx corresponds to the SNVT index number.
0x0000 0Fxx		Confirmation of the LON telegram to be written has arrived too late (> 1500 ms). xx corresponds to the SNVT index number.
0x0000 400x	from FW5	NEURON_MGMT_ERROR. x corresponds to the API code.

8.3 Examples

Example	Description
https://infosys.beckhoff.com/content/1033/tcplcliblon/Resources/11997704331/.zip	All SNVTs be called once each (7 x KL6401)
https://infosys.beckhoff.com/content/1033/tcplcliblon/Resources/11997705739/.zip	All SNVTs be called once each (7 x KL6401)
https://infosys.beckhoff.com/content/1033/tcplcliblon/Resources/11997707147/.zip	Configuration files for KS2000 for parameterization of the terminals.

8.4 SNVT-variables (OFF)

XIF-File: <https://infosys.beckhoff.com/content/1033/tcplcliblon/Resources/11997698699/.zip> 

KS2000 BLC-File: <https://infosys.beckhoff.com/content/1033/tcplcliblon/Resources/11997700107/.zip> 

Implemented SNVT variables (KL6401_OFF)

Description	INPUT/OUTPUT	SNVT ID	Length	NV ID
nviSwitch00	INPUT	95	2	0
nviSwitch01	INPUT	95	2	1
nviSwitch02	INPUT	95	2	2
nviSwitch03	INPUT	95	2	3
nviSwitch04	INPUT	95	2	4
nviSwitch05	INPUT	95	2	5
nviSwitch06	INPUT	95	2	6
nviSwitch07	INPUT	95	2	7
nviSwitch08	INPUT	95	2	8
nviSwitch09	INPUT	95	2	9

Description	INPUT/OUTPUT	SNVT ID	Length	NV ID
nviSwitch10	INPUT	95	2	10
nviSwitch11	INPUT	95	2	11
nviSetting0	INPUT	117	4	12
nviSetting1	INPUT	117	4	13
nviSetting2	INPUT	117	4	14
nviSetting3	INPUT	117	4	15
nviTemp0	INPUT	105	2	16
nviTemp1	INPUT	105	2	17
nviTemp2	INPUT	105	2	18
nviTemp3	INPUT	105	2	19
nviTemp4	INPUT	105	2	20
nviHvacStatus0	INPUT	112	12	21
nviHvacStatus1	INPUT	112	12	22
nviHvacStatus2	INPUT	112	12	23
nviHvacStatus3	INPUT	112	12	24
nviHvacStatus4	INPUT	112	12	25
nvoSwitch00	OUTPUT	95	2	26
nvoSwitch01	OUTPUT	95	2	27
nvoSwitch02	OUTPUT	95	2	28
nvoSwitch03	OUTPUT	95	2	29
nvoSwitch04	OUTPUT	95	2	30
nvoSwitch05	OUTPUT	95	2	31
nvoSwitch06	OUTPUT	95	2	32
nvoSwitch07	OUTPUT	95	2	33
nvoSwitch08	OUTPUT	95	2	34
nvoSwitch09	OUTPUT	95	2	35
nvoSwitch10	OUTPUT	95	2	36
nvoSwitch11	OUTPUT	95	2	37
nvoSwitch12	OUTPUT	95	2	38
nvoSwitch13	OUTPUT	95	2	39
nvoSwitch14	OUTPUT	95	2	40
nvoSwitch15	OUTPUT	95	2	41
nvoSwitch16	OUTPUT	95	2	42
nvoSwitch17	OUTPUT	95	2	43
nvoSetting0	OUTPUT	117	4	44
nvoSetting1	OUTPUT	117	4	45
nvoSetting2	OUTPUT	117	4	46
nvoSetting3	OUTPUT	117	4	47
nvoLevP0	OUTPUT	81	2	48
nvoLevP1	OUTPUT	81	2	49
nvoLevP2	OUTPUT	81	2	50
nvoLevP3	OUTPUT	81	2	51
nvoLevP4	OUTPUT	81	2	52
nvoHvacMode0	OUTPUT	108	1	53
nvoHvacMode1	OUTPUT	108	1	54
nvoHvacMode2	OUTPUT	108	1	55
nvoHvacMode3	OUTPUT	108	1	56
nvoHvacMode4	OUTPUT	108	1	57

8.5 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Download finder

Our [download finder](#) contains all the files that we offer you for downloading. You will find application reports, technical documentation, technical drawings, configuration files and much more.

The downloads are available in various formats.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for [local support and service](#) on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on our internet page: www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline: +49 5246 963-157
e-mail: support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline: +49 5246 963-460
e-mail: service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone: +49 5246 963-0
e-mail: info@beckhoff.com
web: www.beckhoff.com

More Information:
www.beckhoff.com/tx1200

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com

