BECKHOFF New Automation Technology

Documentation | EN

AX2000

ASCII object description

2023-10-26 | Version: 2.1

1 Foreword

1.1 Notes on the documentation

This description is intended exclusively for trained specialists in control and automation technology who are familiar with the applicable national standards.
For installation and commissioning of the components, it is absolutely necessary to observe the documentation and the following notes and explanations.
The qualified personnel is obliged to always use the currently valid documentation.

The responsible staff must ensure that the application or use of the products described satisfies all 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 notice.
No claims to modify 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 and licensed trademarks of Beckhoff Automation GmbH.
If third parties make use of designations or trademarks used in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.

Patents

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
and similar applications and registrations in several other countries.

EtherCAT®

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

Copyright

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.

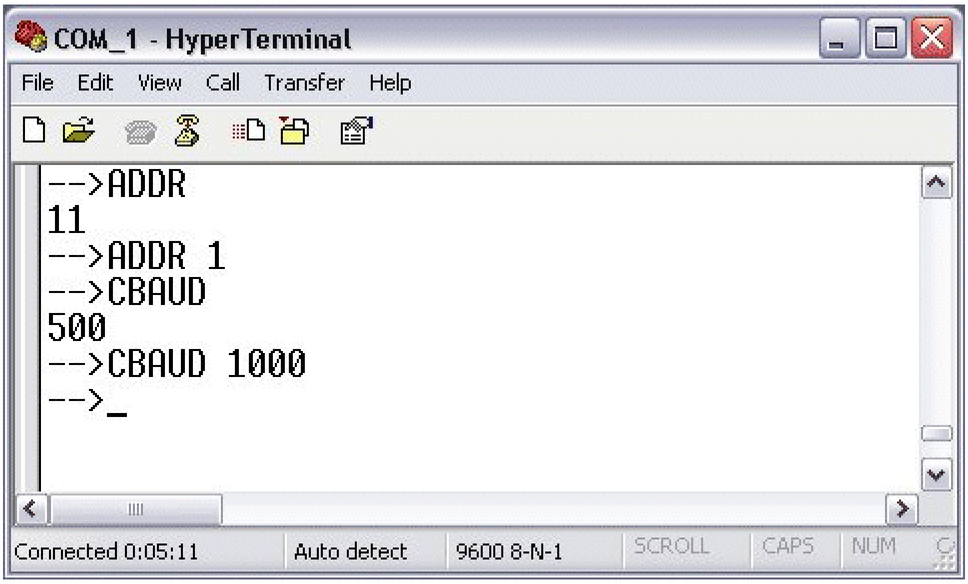
In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

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.

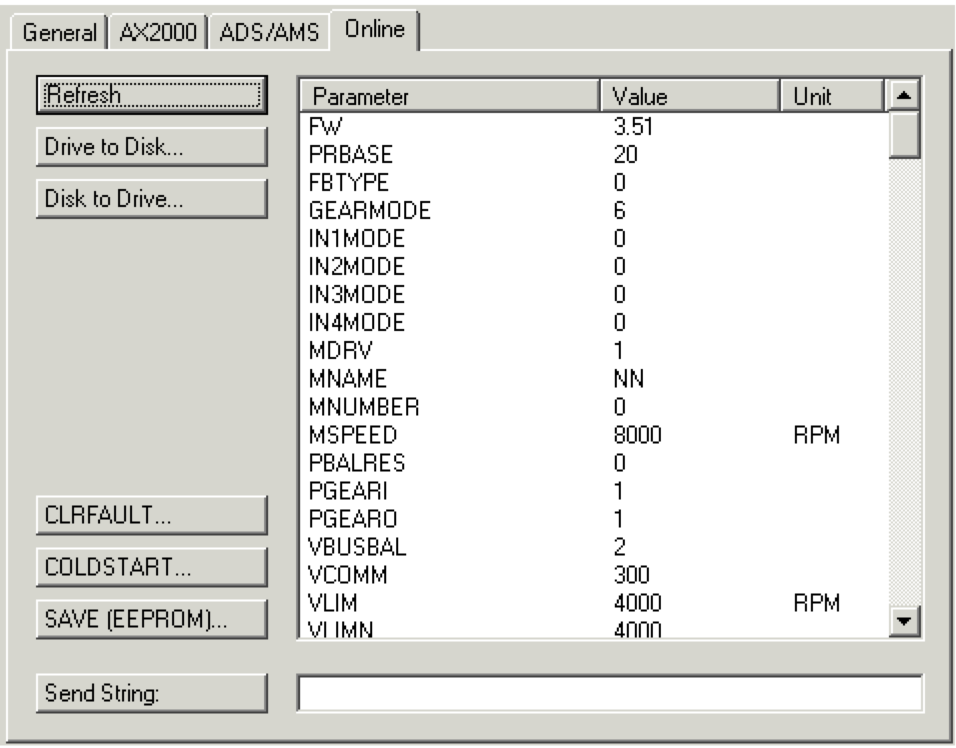
2 Introduction

The Beckhoff **AX2000** and **AX2500** drive contains internal parameters which can be manipulated fieldbus-independently via the serial drive interface (**X6**) through ASCII commands. For this procedure, e.g. the Microsoft Windows program *HyperTerminal* could be used. To start it, use *Start* | *Run...* | *hypertrm.exe.* The following picture shows a communication between **Microsoft HyperTerminal** and AX2xxx drive.



If an ASCII command without a value next to it is issued, it means a read request for a parameter value (see first line in above picture = '--> ADDR'). After entering the command and a pushing the Enter key, the next line shows the register content of this object (in above example it is e.g. '11'). In difference to that, a write-request to the drive is issued if the command is given together with the wanted value (in above example e.g. in the third line where the drive (CAN-)address is set to '1'.

In addition to the ASCII communication through the COM interface, the TwinCAT System Manager supports also the execution of ASCII commands via the **Beckhoff Lightbus** or Beckhoff Real-Time Ethernet. How to do so, is described in depth under "TwinCAT System Manager | Reference | Boxes | Beckhoff Lightbus | AX2xxx-B200 Axis" resp. **"**TwinCAT System Manager | Reference | Boxes | Beckhoff Real-Time Ethernet | AX2xxx-B900 Axis**"**. The following picture shall only give an idea how the AX2xxx ASCII communication dialog pops up in the TwinCAT System Manager:



For a detailed list of currently supported drive ASCII commands (resp. ASCII objects), please see... [▶ 8]

3 Index

Overview about the ASCII objects/commands of the AX2000 resp. AX2500 drive, effective from dedicated firmware revisions on (VER [▶ 89]).

ASCII Object	Short Description
ACC [▶ 325]	Acceleration Ramp
ACCR [▶ 247]	Acceleration Ramp for homing/jog modes
ACCUNIT [▶ 33]	Type of acceleration setpoint for the system
ACTFAULT [▶ 34]	Active Fault Mode
ACTIVE [▶ 168]	Output stage active/inhibited
ACTRS232 [▶ 89]	Activate RS232 Watchdog
ADDR [▶ 70]	Multidrop Address
ADDRFB [▶ 90]	Fieldbus address at AX2500 Slave
AENA [▶ 71]	Software Auto-Enable
ALIAS [▶ 72]	Drive Name
AN10TX [▶ 57]	
AN11NR [▶ 57]	No. Of INxTRIG variable, that is changed analog
AN11RANGE [▶ 58]	Range of the analog change of INxTRIG
AN1TRIG [▶ 59]	Scaling of the analog output 1
AN2TRIG [▶ 59]	Scaling of the analog output 2
ANCNFG [▶ 60]	Configuration of Analog Input
ANDB [▶ 62]	Dead Band of the Analog Velocity Input Signal
ANIN1 [▶ 20]	Voltage at Analog Input SW1
ANIN2 [▶ 20]	Voltage at Analog Input SW2
ANOFF1 [▶ 63]	Analog Offset for input SW1
ANOFF2 [▶ 63]	Analog Offset for input SW2
ANOUT1 [▶ 64]	Configuration of the Analog Output 1
ANOUT2 [▶ 64]	Source of the Analog Output 2
ANZERO1 [▶ 66]	Zero Analog Input SW1
ANZERO2 [▶ 66]	Zero Analog Input SW2
AUTOHOME [▶ 248]	
AVZ1 [▶ 67]	Filter Time Constant Input SW1
\ [▶ 91]	Selection of Remote Address
BCC [▶ 35]	EEPROM check sum
BOOT [▶ 72]	Type of Boot Initialization
BQDC [▶ 325]	Defines the Center Damping of the Bi-quad Filter
BQDR [▶ 326]	Defines the Damping Ratio of the Bi-quad Filter
BQFC [▶ 326]	Center Frequency of the Bi-Quad Filter
BQFR [▶ 327]	Frequency Ratio of the Bi-quad Filter
BQMODE [▶ 327]	Select Compensation Filter Mode for the Velocity Control
CALCCOG [▶ 186]	Determing the Cogging Table
CALCHP [▶ 187]	Determining the Hiperface Parameters
CALCRK [▶ 188]	Calculate resolver parameters
CALCRP [▶ 188]	Calculate resolver phase

ASCII Object	Short Description
WPOSP [▶ 313]	Polarity of Fast Position Registers 1 ... 16
WPOSX [▶ 314]	Mode of Fast Position Registers 1 ... 16
WSAMPL [▶ 206]	Minimum Move of W&S Mode
WSTIME [▶ 206]	Action Time of the W&S - Funktion

4 Commands

4.1 Actual Values

4.1.1 ANIN1

ASCII - Command	**ANIN1**			
Syntax Transmit	ANIN1			
Syntax Receive	ANIN1 <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Integer32		**CANBus Object Number**	3509 (hex)
DIM	Millivolts		**PROFIBUS PNU**	1609 (dec) IND = 0000xxxx (bin)
Range	-10000 .. 10000		**DPR**	9 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Actual Values		**EEPROM**	No
Short Description	Voltage at Analog Input SW1			

Description

The ANIN1 command returns the present value of the voltage at the analog input SW1.

4.1.2 ANIN2

ASCII - Command	**ANIN2**			
Syntax Transmit	ANIN2			
Syntax Receive	ANIN2 <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Integer32		**CANBus Object Number**	350A (hex)
DIM	Millivolts		**PROFIBUS PNU**	1610 (dec) IND = 0000xxxx (bin)
Range	-10000 .. 10000		**DPR**	10 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Actual Values		**EEPROM**	No
Short Description	Voltage at Analog Input SW2			

Description

The ANIN2 command returns the present value of the voltage at the analog input SW2.

4.1.3 I

ASCII - Command	**I**			
Syntax Transmit	I			
Syntax Receive	I <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Float		**CANBus Object Number**	3558 (hex)
DIM	Amperes		**PROFIBUS PNU**	1688 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	88 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	1.20			
Configuration	No		**Revision**	1.8
Function Group	Actual Values		**EEPROM**	No
Short Description	Current Monitor			

Description

This variable returns the actual current value in amperes. This value is always positive.

4.1.4 I2T

ASCII - Command	**I2T**			
Syntax Transmit	I2T			
Syntax Receive	I2T <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Integer32		**CANBus Object Number**	3559 (hex)
DIM	%		**PROFIBUS PNU**	1689 (dec) IND = 0000xxxx (bin)
Range	0 .. 100		**DPR**	89 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.8
Function Group	Actual Values		**EEPROM**	No
Short Description	Average (rms) current [[I2T Loading]			

Description

This variable returns the actual effective load as a percentage of the preset effective current (see ICONT [▸ 108]).

4.1.5 ID

ASCII - Command	**ID**		
Syntax Transmit	ID		
Syntax Receive	ID <Data>	**Available in**	
Type	Variable ro	**MMI**	Yes
ASCII Format	Float	**CANBus Object Number**	355D (hex)
DIM	Amperes	**PROFIBUS PNU**	1693 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	93 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Actual Values	**EEPROM**	No
Short Description	D-component of Current Monitor		

Description

The D-axes component of the actual current value.

4.1.6 IN

ASCII - Command	**IN**		
Syntax Transmit	IN		
Syntax Receive	IN <Data>	**Available in**	
Type	Multi-line Return Command	**MMI**	Yes
ASCII Format	String	**CANBus Object Number**	3560 (hex)
DIM	-	**PROFIBUS PNU**	1696 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	96 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Actual Values	**EEPROM**	-
Short Description	List Analog Voltage Values		

Description

- The IN command returns the input voltages for the 8 A/D channels as counts (-4096 ... +4096).
- Channel 0: Heat sink temperature
- Channel 1: Ambient temperature
- Channel 2: Regen power
- Channel 3: I_U
- Channel 4: Motor temperature
- Channel 5: DC-link/DC-bus voltage [4096 counts = 1015 V]
- Channel 6: Supply voltage [4096 counts = 800 V]
- Channel 7: I_W

4.1.7 IN1

ASCII - Command	**IN1**			
Syntax Transmit	IN1			
Syntax Receive	IN1 <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Integer8		**CANBus Object Number**	3561 (hex)
DIM	-		**PROFIBUS PNU**	1697 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	97 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Actual Values		**EEPROM**	-
Short Description	Status of Digital Input 1			

Description

The status of the digital input INPUT1.

4.1.8 IQ

ASCII - Command	**IQ**			
Syntax Transmit	IQ			
Syntax Receive	IQ <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Float		**CANBus Object Number**	3570 (hex)
DIM	Amperes		**PROFIBUS PNU**	1712 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	112 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Actual Values		**EEPROM**	No
Short Description	Q-Component of Current Monitor			

Description

The Q-axes component of the actual current value.

4.1.9 LED1

ASCII - Command	**LED1**		
Syntax Transmit	LED1 [Data]		
Syntax Receive	LED1 <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	357E (hex)
DIM	-	**PROFIBUS PNU**	1726 (dec) IND = 0000xxxx (bin)
Range	0 .. 127	**DPR**	126 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Actual Values	**EEPROM**	-
Short Description	State of Display 1 Segment		

Description

The command LED1 returns the present status (7-segment code) of the segment in LED1 (left).

Bit-assignment for a 7-segment display:

- Bit 0 (0x01, 1) segment A (top)
- Bit 1 (0x02, 2) segment B (top right)
- Bit 2 (0x04, 4) segment C (bottom right)
- Bit 3 (0x08, 8) segment D (bottom)
- Bit 4 (0x10, 16) segment E (bottom left)
- Bit 5 (0x20, 32) segment F (top left)
- Bit 6 (0x40, 64) segment G (center)

A write action LED1 <code> produces the defined code on the display.
This only makes sense if the internal display output has been switched off (LEDSTAT 0).
It is not possible to output a decimal point.

4.1.10 LED2

ASCII - Command	**LED2**		
Syntax Transmit	LED2 [Data]		
Syntax Receive	LED2 <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	357F (hex)
DIM	-	**PROFIBUS PNU**	1727 (dec) IND = 0000xxxx (bin)
Range	0 .. 127	**DPR**	127 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Actual Values	**EEPROM**	-
Short Description	State of Display 2 Segment		

Description

The command LED2 returns the present status (7-segment code) of the segment in LED2 (center).

Bit-assignment for a 7-segment display:

- Bit 0 (0x01, 1) segment A (top)
- Bit 1 (0x02, 2) segment B (top right)
- Bit 2 (0x04, 4) segment C (bottom right)
- Bit 3 (0x08, 8) segment D (bottom)
- Bit 4 (0x10, 16) segment E (bottom left)
- Bit 5 (0x20, 32) segment F (top left)
- Bit 6 (0x40, 64) segment G (center)

A write action LED2 <code> produces the defined code on the display.

This only makes sense if the internal display output has been switched off (LEDSTAT 0).

It is not possible to output a decimal point.

4.1.11 LED3

ASCII - Command	**LED3**			
Syntax Transmit	LED3 [Data]			
Syntax Receive	LED3 <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer8		**CANBus Object Number**	3580 (hex)
DIM	-		**PROFIBUS PNU**	1728 (dec) IND = 0000xxxx (bin)
Range	0 ..127		**DPR**	128 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Actual Values		**EEPROM**	-
Short Description	State of Display 3 Segment			

Description

The command LED3 returns the present status (7-segment code) of the segment in LED2 (right).

Bit-assignment for a 7-segment display:

- Bit 0 (0x01, 1) segment A (top)
- Bit 1 (0x02, 2) segment B (top right)
- Bit 2 (0x04, 4) segment C (bottom right)
- Bit 3 (0x08, 8) segment D (bottom)
- Bit 4 (0x10, 16) segment E (bottom left)
- Bit 5 (0x20, 32) segment F (top left)
- Bit 6 (0x40, 64) segment G (center)

A write action LED3 <code> produces the defined code on the display.

This only makes sense if the internal display output has been switched off (LEDSTAT 0).

It is not possible to output a decimal point.

4.1.12 PBAL

ASCII - Command	**PBAL**			
Syntax Transmit	PBAL			
Syntax Receive	PBAL <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Integer32		**CANBus Object Number**	35C0 (hex)
DIM	W		**PROFIBUS PNU**	1792 (dec) IND = 0000xxxx (bin)
Range	0 .. 1500		**DPR**	192 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Actual Values		**EEPROM**	No
Short Description	Actual Regen Power			

Description

The actual value of average regen power.

4.1.13 PE

ASCII - Command	**PE**			
Syntax Transmit	PE			
Syntax Receive	PE <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Integer32		**CANBus Object Number**	35C5 (hex)
DIM	µm		**PROFIBUS PNU**	1797 (dec) IND = 0000xxxx (bin)
Range	long int		**DPR**	197 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Actual Values		**EEPROM**	No
Short Description	Actual Following Error			

Description

The following error (sometimes called lag error or contouring error) is the momentary difference between the position setpoint and the actual position, and is displayed in the same units as the position control loop (PGEARI [▶ 283] / PGEARO [▶ 284]). See description of PFB [▶ 27]

4.1.14 PFB

ASCII - Command	**PFB**		
Syntax Transmit	PFB		
Syntax Receive	PFB <Data>	**Available in**	
Type	Variable ro	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	35C8 (hex)
DIM	µm	**PROFIBUS PNU**	1800 (dec) IND = 0000xxxx (bin)
Range	long int	**DPR**	200 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Actual Values	**EEPROM**	No
Short Description	Actual Position from Feedback Device		

Description

The PFB command returns the actual value of the position (from the position control loop feedback). The unit for the position value depends on the PGEARI [▸ 283], PGEARO [▸ 284] and PRBASE [▸ 286] settings.

PFB = Position * PGEARI [▸ 283] / PGEARO [▸ 284]
whereby:

Position = position value in increments, 1048576/turn for PRBASE [▸ 286]=20, 65536/turn for PRBASE [▸ 286]=16 PGEARI [▸ 283], PGEARO [▸ 284] - resolution of position control loop

Note

If the resolution is set to 1 (PGEARI [▸ 283]=PGEARO [▸ 284]) then the PFB command provides internal units (counts).

If the position information of an external encoder is evaluated (EXTPOS [▸ 253]=1,2,3), then this information can be displayed by using the PFB0 [▸ 27] command.

4.1.15 PFB0

ASCII - Command	**PFB0**		
Syntax Transmit	PFB0		
Syntax Receive	PFB0 <Data>	**Available in**	
Type	Variable ro	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	35C9 (hex)
DIM	Counts	**PROFIBUS PNU**	1801 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	201 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Actual Values	**EEPROM**	No
Short Description	Position from External Encoder		

Description

The PFB0 command returns the actual position, calculated from the position information provided by an external encoder. The position is only derived from an external encoder if the configuration variable EXTPOS [▶ 253] is set to 1,2,3. The unit for the position value depends on the PGEARI [▶ 283], PGEARO [▶ 284], ENCIN [▶ 212] and EXTMUL [▶ 252] settings.

PFB0 = Position * PGEARI [▶ 283] / PGEARO [▶ 284]

whereby: Position = position value in increments (resolution: EXTMUL [▶ 252] * ENCIN [▶ 212] per turn)
PGEARI [▶ 283], PGEARO [▶ 284] - resolution of position control loop

Note

If the resolution is set to 1 (PGEARI [▶ 283]=PGEARO [▶ 284]) then the PFB0 command provides internal units (counts).

4.1.16 PRD

ASCII - Command	**PRD**		
Syntax Transmit	PRD		
Syntax Receive	PRD <Data>	**Available in**	
Type	Variable ro	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	35D2 (hex)
DIM	Counts	**PROFIBUS PNU**	1810 (dec) IND = 0000xxxx (bin)
Range	0 .. 1048575	**DPR**	210 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Actual Values	**EEPROM**	No
Short Description	20-bit Position Feedback		

Description

The PRD command returns a 20-bit position (absolute within one turn) that is derived from the signals of the feedback device (FBTYPE [▶ 190]). Unlike the position from the position control loop, PFB, this position cannot be altered.

PRD is not related to PRBASE [▶ 286]

4.1.17 PV

ASCII - Command	**PV**			
Syntax Transmit	PV			
Syntax Receive	PV <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Integer32		**CANBus Object Number**	35D7 (hex)
DIM	VUNIT		**PROFIBUS PNU**	1815 (dec) IND = 0000xxxx (bin)
Range	long int		**DPR**	215 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Actual Values		**EEPROM**	No
Short Description	Actual Velocity (Position Control Loop)			

Description

The actual velocity (position control loop) can be requested by using the PV command.
The scaling of the velocity depends on the PGEARI [▸ 283], and PGEARO [▸ 284] parameters.

4.1.18 TASK

ASCII - Command	**TASK**			
Syntax Transmit	TASK			
Syntax Receive	TASK <Data>		**Available in**	
Type	Variable ro		**MMI**	No
ASCII Format	String		**CANBus Object Number**	360F (hex)
DIM	-		**PROFIBUS PNU**	1871 (dec) IND = 0000xxxx (bin)
Range	max 80 ASCII Characters		**DPR**	271 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Actual Values		**EEPROM**	No
Short Description	Task Workload			

Description

The TASK command shows the loading for the individual firmware tasks.

The figures signify the number of functions performed per millisecond.

4.1.19 TEMPE

ASCII - Command	**TEMPE**			
Syntax Transmit	TEMPE			
Syntax Receive	TEMPE <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Integer32		**CANBus Object Number**	3610 (hex)
DIM	Centigrade Degrees		**PROFIBUS PNU**	1872 (dec) IND = 0000xxxx (bin)
Range	-20 .. 90		**DPR**	272 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Actual Values		**EEPROM**	No
Short Description	Ambient Temperature			

Description

Displays the present internal temperature in °C.

4.1.20 TEMPH

ASCII - Command	**TEMPH**			
Syntax Transmit	TEMPH			
Syntax Receive	TEMPH <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Integer32		**CANBus Object Number**	3611 (hex)
DIM	Centigrade Degrees		**PROFIBUS PNU**	1873 (dec) IND = 0000xxxx (bin)
Range	-20 .. 90		**DPR**	273 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Actual Values		**EEPROM**	No
Short Description	Heat Sink Temperature			

Description

Displays the present heat sink temperature in °C.

4.1.21 TEMPM

ASCII - Command	**TEMPM**			
Syntax Transmit	TEMPM			
Syntax Receive	TEMPM <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Integer32		**CANBus Object Number**	3612 (hex)
DIM	Ohm		**PROFIBUS PNU**	1874 (dec) IND = 0000xxxx (bin)
Range	0 .. 10000		**DPR**	274 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Actual Values		**EEPROM**	No
Short Description	Motor Temperature			

Description

Indicates the motor temperature, in the form of the resistance of the temperature sensor (in ohms).

4.1.22 V

ASCII - Command	**V**			
Syntax Transmit	V			
Syntax Receive	V <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Integer32		**CANBus Object Number**	3618 (hex)
DIM	MUNIT		**PROFIBUS PNU**	1880 (dec) IND = 0000xxxx (bin)
Range	-15000 .. 15000		**DPR**	280 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Actual Values		**EEPROM**	No
Short Description	Actual Velocity			

Description

The present velocity of the motor.

4.1.23 VBUS

ASCII - Command	**VBUS**			
Syntax Transmit	VBUS			
Syntax Receive	VBUS <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Integer32		**CANBus Object Number**	361A (hex)
DIM	Volts		**PROFIBUS PNU**	1882 (dec) IND = 0000xxxx (bin)
Range	0 .. 900		**DPR**	282 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Actual Values		**EEPROM**	No
Short Description	DC-bus voltage			

Description

The present voltage of the DC-bus.

4.1.24 VCMD

ASCII - Command	**VCMD**			
Syntax Transmit	VCMD			
Syntax Receive	VCMD <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Float		**CANBus Object Number**	361E (hex)
DIM	MUNIT		**PROFIBUS PNU**	1886 (dec) IND = 0000xxxx (bin)
Range	-VMAX .. VMAX		**DPR**	286 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Actual Values		**EEPROM**	No
Short Description	Internal Velocity Setpoint in RPM			

Description

The VCMD variable contains the internal velocity setpoint (after the ramp generator) in RPM.

Depending on the operating mode that is set (OPMODE [▸ 50]=0), this value is either provided directly and digitally (fieldbus, slot card) or derived from the analog velocity setpoint (OPMODE [▸ 50]=1).

For operating modes that do not use a velocity control loop (OPMODE [▸ 50]=2,3) the VCMD variable has the value V of the actual velocity.

4.1.25 VF

ASCII - Command	**VF**		
Syntax Transmit	VF		
Syntax Receive	VF <Data>	**Available in**	
Type	Variable ro	**MMI**	Yes
ASCII Format	Float	**CANBus Object Number**	3661 (hex)
DIM	MUNIT	**PROFIBUS PNU**	1953 (dec) IND = 0000xxxx (bin)
Range	-15000 .. 15000	**DPR**	353 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	4.00		
Configuration	No	**Revision**	1.5
Function Group	Actual Values	**EEPROM**	No
Short Description	Actual Velocity in Floating Point Format		

Description

The present velocity of the motor in floating point format.

4.2 Amplifier

4.2.1 ACCUNIT

ASCII - Command	**ACCUNIT**		
Syntax Transmit	ACCUNIT [Data]		
Syntax Receive	ACCUNIT <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer32	**CANBus Object Number**	3659 (hex)
DIM	-	**PROFIBUS PNU**	1945 (dec) IND = 0000xxxx (bin)
Range	0, 1, .. , 5	**DPR**	345 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	3.41		
Configuration	No	**Revision**	1.7
Function Group	Amplifier	**EEPROM**	Yes
Short Description	Type of acceleration setpoint for the system		

Description

Using this command, the systemwide acceleration type is defined. This function is used for ramps of the trajectory generator (internal motion tasks OPMODE [▸ 50] 8) and for the ramps of the speed controller.

- ACCUNIT = 0 Acceleration is defined as acc time in msec
- ACCUNIT = 1 Acceleration is defined in rad/sec²
- ACCUNIT = 2 Acceleration is defined in rpm/sec
- ACCUNIT = 3 Acceleration is defined in PUNIT [▸ 289]/sec² (starting Version 4.00)
- ACCUNIT = 4 Acceleration is defined in 1000*PUNIT [▸ 289]/sec² (starting Version 4.00)

- ACCUNIT = 5 Acceleration is defined 1000000*PUNIT [▶ 289]/sec² (starting Version 4.00)

If ACCUNIT=0 is selected, the motion task acceleration can be given in mm/sec² (Bit 12 of the type of the motion task=1).

If ACCUNIT=1 is selected, this Bit is ignored, this means the ramps are calculated in rad/sec².

If ACCUNIT is changed, all acc/dec parameters are calculated in a different way to get the right unit. Affected are ACC [▶ 325], ACCR [▶ 247], DEC [▶ 329], DECR [▶ 249], DECSTOP [▶ 330], DECDIS [▶ 330].

The motion tasks are not affected. So, before defining a motion task ACCUNIT has to be set in right manner. If ACCUNIT is changed later, all motion tasks have to be proofed or changed !

The accdec-ramps of the motion tasks are limited bei PTMIN [▶ 289]. This setting is done with ACCUNIT starting with Firmware version 4.02. Before that, the ramps were calculated in msec.

High acceleration corresponds to small values of PTMIN [▶ 289] at ACCUNIT=0. If ACCUNIT is > 0, PTMIN [▶ 289] is small if the acceleration is high.

4.2.2 ACTFAULT

ASCII - Command	**ACTFAULT**		
Syntax Transmit	ACTFAULT [Data]		
Syntax Receive	ACTFAULT <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	3503 (hex)
DIM	-	**PROFIBUS PNU**	1603 (dec) IND = 0000xxxx (bin)
Range	0, 1	**DPR**	3 (dec)
Default	1		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	Disabled + Reset (Coldstart)	**Weightning**	
Start Firmware	1.20		
Configuration	Yes	**Revision**	1.4
Function Group	Amplifier	**EEPROM**	Yes
Short Description	Active Fault Mode		

Description

The ACTFAULT command is used to specify the response of the drive if a fault occurs.

ACTFAULT=0: If a fault occurs, the output stage is immediately inhibited, the drive coasts down.

ACTFAULT=1: If a fault occurs, an Emergency Stop procedure is initiated, that consists of the following steps.

1. Switch over the controller mode to velocity control (OPMODE [▶ 50]=0)
2. Change the braking ramp for the velocity control loop (DEC [▶ 329]) to the emergency stop ramp (DECSTOP [▶ 330])
3. Set the internal velocity setpoint to 0 (before the ramp generator).
4. Start a timer (with time-out = 5 seconds)

As soon as the internal velocity setpoint (after the ramp generator) has reached 0, the output stage is inhibited, and the original controller mode is re-activated. This will also happen if the time-out occurs before the velocity setpoint has reached 0.

4.2.3 BCC

ASCII - Command	**BCC**		
Syntax Transmit	BCC		
Syntax Receive	BCC <Data>	**Available in**	
Type	Variable ro	**MMI**	No
ASCII Format	Integer16	**CANBus Object Number**	363A (hex)
DIM	-	**PROFIBUS PNU**	1914 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	314 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	2.49		
Configuration	No	**Revision**	1.3
Function Group	Amplifier	**EEPROM**	No
Short Description	EEPROM check sum		

Description

The BCC variable returns a checksum for the parameter area of the serial EEPROM.
When a SAVE [▸ 51] command is carried out, all the internal parameters of the amplifier are saved in this area, in ASCII format. The checksum is obtained by summing all the stored bytes, and is recalculated with every LOAD [▸ 41] or SAVE [▸ 51] command. It is only intended for the detection of EEPROM errors.

But it can also be used to detect whether the data set that is present in the controls matches the data set that is stored in the servo amplifier.

4.2.4 CLRFAULT

ASCII - Command	**CLRFAULT**		
Syntax Transmit	CLRFAULT		
Syntax Receive	CLRFAULT	**Available in**	
Type	Command	**MMI**	Yes
ASCII Format	-	**CANBus Object Number**	3518 (hex)
DIM	-	**PROFIBUS PNU**	1624 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	24 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Amplifier	**EEPROM**	-
Short Description	Clear Drive Fault		

Description

The CLRFAULT command cancels the fault status of an amplifier. A hardware or software reset of the amplifier is carried out, depending on the type of fault that is present.

After a software reset the amplifier is immediately ready for operation, after a hardware reset the complete initialization phase must be gone through first (as for power-on).

As well as amplifier faults (display Fxx), the following warnings are also deleted.

- contouring/following error
- threshold monitoring

With the selection CLRWARN [▸ 169]=1 (separate cancellation of warnings) this command will delete all warnings that are present.

A listing of all possible fault/error messages, with information on the hardware/software reset required, can be found in the description of the ERRCODE [▸ 75] command.

The CLRFAULT command can either be implemented through the ASCII channel (CLRFAULT command) or via the CAN/PROFIBUS (with the □cancel fault□ bit in the control word), or through a digital input (□Controller reset□ function).

4.2.5 CUPDATE

ASCII - Command	**CUPDATE**			
Syntax Transmit	CUPDATE			
Syntax Receive	CUPDATE		**Available in**	
Type	Command		**MMI**	No
ASCII Format	-		**CANBus Object Number**	351F (hex)
DIM	-		**PROFIBUS PNU**	1631 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	31 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	Disabled		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Amplifier		**EEPROM**	-
Short Description	Program Update (CAN Bus)			

Description

The CUPDATE command activates a function that can receive data through a CAN bus interface and save them in the program memory of the amplifier. After this function has been activated, no more commands will be accepted through the serial interface.

The program PRGDOWN.EXE should be used for downloading data on the PC side. This program operates with the hardware in a handshaking procedure and prepares the data for CAN transmission.

4.2.6 DEVICE

ASCII - Command	**DEVICE**			
Syntax Transmit	DEVICE			
Syntax Receive	DEVICE <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	String		**CANBus Object Number**	3526 (hex)
DIM	-		**PROFIBUS PNU**	1638 (dec) IND = 0000xxxx (bin)
Range	max 50 ASCII Characters		**DPR**	38 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Amplifier		**EEPROM**	No
Short Description	Device ID			

Description

The command returns the amplifier ID in the following format:

Drive 6xx @ yyyV whereby xx = current rating

yyy = DC bus voltage
e.g. Drive 601 @ 700V

4.2.7 DICONT

ASCII - Command	**DICONT**			
Syntax Transmit	DICONT			
Syntax Receive	DICONT <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Float		**CANBus Object Number**	3527 (hex)
DIM	Amperes		**PROFIBUS PNU**	1639 (dec) IND = 0000xxxx (bin)
Range	1.5 .. 20.0		**DPR**	39 (dec)
Default	Hardware Defined			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Amplifier		**EEPROM**	No
Short Description	Drive Continuous Current			

Description

The continuous current rating of the drive. DICONT is depending on VBUSBAL [▶ 88] for drive 403, 406, 614 and 670.

4.2.8 DIPEAK

ASCII - Command	**DIPEAK**			
Syntax Transmit	DIPEAK			
Syntax Receive	DIPEAK <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Float		**CANBus Object Number**	3529 (hex)
DIM	Amperes		**PROFIBUS PNU**	1641 (dec) IND = 0000xxxx (bin)
Range	3.0 .. 70.0		**DPR**	41 (dec)
Default	Hardware Defined			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Amplifier		**EEPROM**	No
Short Description	Drive Peak Rated Current			

Description

The peak rated current of the drive. DIPEAK is depending on VBUSBAL [▸ 88] for drive 403, 406, 614 and 670.

4.2.9 DIS

ASCII - Command	**DIS**			
Syntax Transmit	DIS			
Syntax Receive	DIS		**Available in**	
Type	Command		**MMI**	Yes
ASCII Format	-		**CANBus Object Number**	352B (hex)
DIM	-		**PROFIBUS PNU**	1643 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	43 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	Enabled		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Amplifier		**EEPROM**	-
Short Description	Software-Disable			

Description

The DIS command sets the software enable for the output stage to 0. Depending on the configuration (see MBRAKE [▸ 226], STOPMODE [▸ 85]), the drive will coast down, or be run down under control.

4.2.10 DR_TYPE

ASCII - Command	**DR_TYPE**			
Syntax Transmit	DR_TYPE			
Syntax Receive	DR_TYPE <Data>		**Available in**	
Type	Variable ro		**MMI**	No
ASCII Format	Integer16		**CANBus Object Number**	352E (hex)
DIM	-		**PROFIBUS PNU**	1646 (dec) IND = 0000xxxx (bin)
Range	1 .. 8		**DPR**	46 (dec)
Default	-			
Opmode	-		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	1.27			
Configuration	No		**Revision**	1.8
Function Group	Amplifier		**EEPROM**	No
Short Description	Gives the Output Stage Indentification			

Description

This command can be used to read the drive type.

DR_TYPE Drive

- 1 SR601
- 2 SR603
- 3 SR606
- 4 SR610
- 5 SR614
- 6 SR620
- 7 SR640
- 8 SR670
- 9 SR610/30
- 18 SR403
- 19 SR406

4.2.11 EN

ASCII - Command	**EN**			
Syntax Transmit	EN			
Syntax Receive	EN		**Available in**	
Type	Command		**MMI**	Yes
ASCII Format	-		**CANBus Object Number**	3530 (hex)
DIM	-		**PROFIBUS PNU**	1648 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	48 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	Disabled		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Amplifier		**EEPROM**	-

Short Description	Software-Enable

Description

The EN command sets the software enable for the output stage.

If the software enable and the hardware enable are set and no fault is present (the BTB contact is closed), then the output stage is enabled.

If the MAINS BTB function is activated (OxMODE [▸ 150]=3), then the output stage will only be enabled when the supply power has been switched on and the charging circuit has charged up the DC bus. If the supply power is removed from an enabled instrument, then it remains enabled until the DC bus voltage has fallen below the undervoltage limit (VBUSMIN [▸ 54]).

4.2.12 FW

ASCII - Command	**FW**			
Syntax Transmit	FW			
Syntax Receive	FW <Data>		**Available in**	
Type	Variable ro		**MMI**	No
ASCII Format	Float		**CANBus Object Number**	3657 (hex)
DIM	-		**PROFIBUS PNU**	1943 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	343 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	3.30			
Configuration	No		**Revision**	1.8
Function Group	Amplifier		**EEPROM**	No
Short Description	Displays the Version Number of the Firmware			

Description

The command FW displays the versionnumber of the firmware. The command is also appearing in the DUMP [▸ 94] list and is part of the parameter settings of the drive.

4.2.13 K

ASCII - Command	**K**			
Syntax Transmit	K			
Syntax Receive	K		**Available in**	
Type	Command		**MMI**	Yes
ASCII Format	-		**CANBus Object Number**	3573 (hex)
DIM	-		**PROFIBUS PNU**	1715 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	115 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	Enabled		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Amplifier		**EEPROM**	No
Short Description	Kill (=Disable)			

Description

The K (Kill) command is a short form command of the DIS [▸ 38] command.

4.2.14 LASTWMASK

ASCII - Command	**LASTWMASK**		
Syntax Transmit	LASTWMASK		
Syntax Receive	LASTWMASK <Data>	**Available in**	
Type	Variable ro	**MMI**	No
ASCII Format	Integer32	**CANBus Object Number**	36CE (hex)
DIM	-	**PROFIBUS PNU**	1662 (dec) IND = 0001xxxx (bin)
Range	long int	**DPR**	462 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	5.41		
Configuration	No	**Revision**	1.9
Function Group	Amplifier	**EEPROM**	-
Short Description	Fault history of WMASK		

Description

WMASK [▸ 56] gives the possibility to create a mask to change warnings to errors. If the F24 occurs, LASTWMASK displays the warnings that caused the error.

4.2.15 LOAD

ASCII - Command	**LOAD**		
Syntax Transmit	LOAD		
Syntax Receive	LOAD	**Available in**	
Type	Command	**MMI**	Yes
ASCII Format	-	**CANBus Object Number**	3583 (hex)
DIM	-	**PROFIBUS PNU**	1731 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	131 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Amplifier	**EEPROM**	No
Short Description	Load parameters from serial EEPROM		

Description

The LOAD command loads the parameters from the serial EEPROM. All parameter changes that have been made since the last SAVE [▸ 51] command (save in the serial EEPROM) will be lost.

4.2.16 M

ASCII - Command	**M**		**For Manufacturer Use only**	
Syntax Transmit	M [Data]			
Syntax Receive	M <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	String		**CANBus Object Number**	No
DIM	-		**PROFIBUS PNU**	No
Range	-		**DPR**	No
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Amplifier		**EEPROM**	-
Short Description	Read/write Macro Variable			

Description

The M command can be used to access any internal macro variable.

Macro variables are variables which can be used by the macro routines. They are called by name within a macro routine. The complete variable information (name, address) is managed in the form of a table. There are two types of macro variable.

1. System variables □ these variables are determined by the firmware and are fixed. Both the names and the addresses for such variables are constant within a given firmware version. The table with the data for the variables is within the programming area, so that the system variables can be accessed at any time.
2. User variables □ these variables are set up in the macro routines during the initialization phase of the amplifier. The availability of a specific variable, or its physical address, depends on the amplifier configuration. The table with the information for the variables is set up in the RAM and is normally only required during the compilation of the macro programs (initialization phase). After the initialization has been concluded, the table is removed from the memory. In this case, it is not possible to access the user variables through the M command.

If the parameter setting MSG=2 is found when the amplifier is switched on, then the table for the user variables is kept in the memory. In this case, it will also be possible to access the user variables through the M command.

The M command can be used in one of three forms:

1. □M□ □ a list of all the system and user variables is generated
2. □M name□ □ an information line is generated for the variable <name>, in the following form: □name [TYPE] address FORMAT=value□
   The individual elements are interpreted as follows:
   - name = name of the macro variable
   - TYPE = variable type (SYSTEM or USER)
   - address = physical address of the variable (hexadecimal format)
   - FORMAT = variable type (BYTE,WORD,LONG,STRING), the suffix FAST means that the variable is stored in the internal (fast) RAM
     Value = variable contents (in hexadecimal format, or as ASCII string, depending on FORMAT)
3. □M name value□ □ the number □value□ is entered in the variable □name□
   The entry for □value□ must be made as a decimal number. If the character sequence □0x□ is added as a prefix, the number can be entered in hexadecimal format.

4.2.17 M_1000

ASCII - Command	**M_1000**		For Manufacturer Use only	
Syntax Transmit	M_1000			
Syntax Receive	M_1000 <Data>		Available in	
Type	Variable rw		MMI	No
ASCII Format	String		CANBus Object Number	No
DIM	-		PROFIBUS PNU	No
Range	-		DPR	No
Default	-			
Opmode	All		Data Type Bus/DPR	-
Drive State	-		Weightning	
Start Firmware	1.20			
Configuration	No		Revision	1.3
Function Group	Amplifier		EEPROM	-
Short Description	Display 1 msec Macro Program			

Description

The source code of the 1 msec macro function is displayed on the screen. The setting PROMPT [▸ 102]=2 makes the display appear page-by-page. Pressing a key steps, the display on to show the next side, <ESC> cancels the output to the screen.

4.2.18 M_125

ASCII - Command	**M_125**		For Manufacturer Use only	
Syntax Transmit	M_125			
Syntax Receive	M_125 <Data>		Available in	
Type	Variable rw		MMI	No
ASCII Format	String		CANBus Object Number	No
DIM	-		PROFIBUS PNU	No
Range	-		DPR	No
Default	-			
Opmode	All		Data Type Bus/DPR	-
Drive State	-		Weightning	
Start Firmware	1.20			
Configuration	No		Revision	1.3
Function Group	Amplifier		EEPROM	-
Short Description	Display 125 microsecond Macro Program			

Description

The source code of the 125 microsecond macro function (current control loop) is displayed on the screen. The setting PROMPT [▸ 102]=2 makes the display appear page-by-page. Pressing a key steps the display on to show the next side, <ESC> cancels the output to the screen.

4.2.19 M_1600

ASCII - Command	M_1600	For Manufacturer Use only	
Syntax Transmit	M_1600		
Syntax Receive	M_1600 <Data>	Available in	
Type	Variable rw	MMI	No
ASCII Format	String	CANBus Object Number	No
DIM	-	PROFIBUS PNU	No
Range	-	DPR	No
Default	-		
Opmode	All	Data Type Bus/DPR	-
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.3
Function Group	Amplifier	EEPROM	-
Short Description	Display 16 msec Macro Program		

Description

The source code of the 16 msec macro function is displayed on the screen. The setting PROMPT [▸ 102]=2 makes the display appear page-by-page. Pressing a key steps, the display on to show the next side, <ESC> cancels the output to the screen.

4.2.20 M_250

ASCII - Command	M_250	For Manufacturer Use only	
Syntax Transmit	M_250		
Syntax Receive	M_250 <Data>	Available in	
Type	Variable rw	MMI	No
ASCII Format	String	CANBus Object Number	No
DIM	-	PROFIBUS PNU	No
Range	-	DPR	No
Default	-		
Opmode	All	Data Type Bus/DPR	-
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.3
Function Group	Amplifier	EEPROM	-
Short Description	Display 250 microsecond Macro Program		

Description

The source code of the 250 microsecond macro function (current control loop) is displayed on the screen. The setting PROMPT [▸ 102]=2 makes the display appear page-by-page. Pressing a key steps, the display on to show the next side, <ESC> cancels the output to the screen.

4.2.21 M_250P

ASCII - Command	M_250p		For Manufacturer Use only	
Syntax Transmit	M_250p			
Syntax Receive	M_250p <Data>		Available in	
Type	Variable rw		MMI	No
ASCII Format	String		CANBus Object Number	No
DIM	-		PROFIBUS PNU	No
Range	-		DPR	No
Default	-			
Opmode	All		Data Type Bus/DPR	-
Drive State	-		Weightning	
Start Firmware	1.20			
Configuration	No		Revision	1.3
Function Group	Amplifier		EEPROM	-
Short Description	Display 250 microsecond Macro Program			

Description

The source code of the 250 microsecond macro function (current control loop) is displayed on the screen. The setting PROMPT [▶ 102]=2 makes the display appear page-by-page. Pressing a key steps, the display on to show the next side, <ESC> cancels the output to the screen.

4.2.22 M_4000

ASCII - Command	M_4000		For Manufacturer Use only	
Syntax Transmit	M_4000			
Syntax Receive	M_4000 <Data>		Available in	
Type	Variable rw		MMI	No
ASCII Format	String		CANBus Object Number	No
DIM	-		PROFIBUS PNU	No
Range	-		DPR	No
Default	-			
Opmode	All		Data Type Bus/DPR	-
Drive State	-		Weightning	
Start Firmware	1.20			
Configuration	No		Revision	1.3
Function Group	Amplifier		EEPROM	-
Short Description	Display 4 msec Macro Program			

Description

The source code of the 4 msec macro function is displayed on the screen. The setting PROMPT [▶ 102]=2 makes the display appear page-by-page. Pressing a key steps, the display on to show the next side, <ESC> cancels the output to the screen.

4.2.23 M_DISABLE

ASCII - Command	M_DISABLE	For Manufacturer Use only	
Syntax Transmit	M_DISABLE		
Syntax Receive	M_DISABLE <Data>	Available in	
Type	Variable rw	MMI	No
ASCII Format	String	CANBus Object Number	No
DIM	-	PROFIBUS PNU	No
Range	-	DPR	No
Default	-		
Opmode	All	Data Type Bus/DPR	-
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.3
Function Group	Amplifier	EEPROM	-
Short Description	Display the "Disable" Macro Program		

Description

The source code of the macro function "Disable" is displayed on the screen. It is only run once when the amplifier is disabled. The setting PROMPT [▸ 102]=2 makes the display appear page-by-page. Pressing a key steps the display on to show the next side, <ESC> cancels the output to the screen.

4.2.24 M_INIT

ASCII - Command	M_INIT	For Manufacturer Use only	
Syntax Transmit	M_INIT		
Syntax Receive	M_INIT <Data>	Available in	
Type	Variable rw	MMI	No
ASCII Format	String	CANBus Object Number	No
DIM	-	PROFIBUS PNU	No
Range	-	DPR	No
Default	-		
Opmode	All	Data Type Bus/DPR	-
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.3
Function Group	Amplifier	EEPROM	-
Short Description	Display the "Init" Macro Program		

Description

The source code of the macro function "Initialization" is displayed on the screen. It is only run once when the amplifier is started up. The setting PROMPT [▸ 102]=2 makes the display appear page-by-page. Pressing a key step, the display on to show the next side, <ESC> cancels the output to the screen.

4.2.25 M_IRQ

ASCII - Command	**M_IRQ**	**For Manufacturer Use only**	
Syntax Transmit	M_IRQ		
Syntax Receive	M_IRQ <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	String	**CANBus Object Number**	No
DIM	-	**PROFIBUS PNU**	No
Range	-	**DPR**	No
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Amplifier	**EEPROM**	-
Short Description	Display the □Interrupt□ Macro Program		

Description

The source code of the macro function □Interrupt□ is displayed on the screen. It is run when a macro-interrupt is called. The setting PROMPT [▸ 102]=2 makes the display appear page-by-page. Pressing a key steps the display on to show the next side, <ESC> cancels the output to the screen.

4.2.26 M_RESET

ASCII - Command	**M_RESET**	**For Manufacturer Use only**	
Syntax Transmit	M_RESET		
Syntax Receive	M_RESET	**Available in**	
Type	Command	**MMI**	No
ASCII Format	-	**CANBus Object Number**	35A9 (hex)
DIM	-	**PROFIBUS PNU**	1769 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	169 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	Disable	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Amplifier	**EEPROM**	-
Short Description	Recompile Macro Programs		

Description

All macro programs are compiled and started when the amplifier is switched on. The compilation of the macro programs is managed by configuration variables. The values for these configuration variables must be fixed before the compilation procedure is started. If the value of a configuration variable is altered later, this change will only take effect with the next compilation of the macro programs. This means, that after changing a configuration variable, this change should first be stored in the EEPROM (see SAVE [▸ 51] command) and the amplifier should then be switched off and on again.

The M_RESET command offers an alternative. This command is used to force a new compilation of the macro programs, without having to switch the amplifier off and on again. Since this function, unlike that performed during the initialization phase, is carried out while the interrupts are enabled, it takes longer to complete (about 5 min).

4.2.27 M_SMACRO

ASCII - Command	**M_SMACRO**		**For Manufacturer Use only**	
Syntax Transmit	M_SMACRO [*]			
Syntax Receive	M_SMACRO <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	String		**CANBus Object Number**	No
DIM	-		**PROFIBUS PNU**	No
Range	-		**DPR**	No
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Amplifier		**EEPROM**	-
Short Description	Display System Macros			

Description

A list of all the available system macros is displayed on the screen. The setting PROMPT [▸ 102]=2 makes the display appear page-by-page. Pressing a key steps the display on to show the next side, <ESC> cancels the output to the screen.

A line on the screen has the following format: NAME (parameter) info

NAME - name of the macro

(parameter) - transfer parameter for macro

info - short description of macro

The command M_SMACRO * generates an additional line for each macro, in which the formats for the transfer parameters are shown. The abbreviations are as follows:

- f - fast: the parameter is a variable that must be held in the fast processor RAM.
- G - global: the parameter is a variable, to be held in fast or slow RAM
- v - variable: the parameter is a variable
- c- constant: the parameter is a constant (number)
- b - byte: 8-bit parameter
- w - word: 16-bit parameter
- l - long: 32-bit parameter

4.2.28 M_TASK

ASCII - Command	M_TASK		For Manufacturer Use only	
Syntax Transmit	M_TASK			
Syntax Receive	M_TASK <Data>		Available in	
Type	Variable rw		MMI	No
ASCII Format	String		CANBus Object Number	No
DIM	-		PROFIBUS PNU	No
Range	-		DPR	No
Default	-			
Opmode	All		Data Type Bus/DPR	-
Drive State	-		Weightning	
Start Firmware	1.20			
Configuration	No		Revision	1.3
Function Group	Amplifier		EEPROM	-
Short Description	Display the Main Macro Program			

Description

The source code of the main macro program is displayed on the screen. This program section is always run if no other routine is being performed (idle). The setting PROMPT [▸ 102]=2 makes the display appear page-by-page. Pressing a key steps the display on to show the next side, <ESC> cancels the output to the screen.

4.2.29 M_UMACRO

ASCII - Command	M_UMACRO		For Manufacturer Use only	
Syntax Transmit	M_UMACRO [*]			
Syntax Receive	M_UMACRO <Data>		Available in	
Type	Variable rw		MMI	No
ASCII Format	String		CANBus Object Number	No
DIM	-		PROFIBUS PNU	No
Range	-		DPR	No
Default	-			
Opmode	All		Data Type Bus/DPR	-
Drive State	-		Weightning	
Start Firmware	1.20			
Configuration	No		Revision	1.3
Function Group	Amplifier		EEPROM	-
Short Description	Display User Macros			

Description

A list of all the available user macros is displayed on the screen. The setting PROMPT [▸ 102]=2 makes the display appear page-by-page. Pressing a key steps the display on to show the next side, <ESC> cancels the output to the screen.

The output format can be seen in the M_SMACRO [▸ 48] description.

Since only SYSTEM macros are used at present, an empty user macro list will be output.

4.2.30 OPMODE

ASCII - Command	OPMODE		
Syntax Transmit	OPMODE [Data]		
Syntax Receive	OPMODE <Data>	Available in	
Type	Variable rw	MMI	Yes
ASCII Format	Integer8	CANBus Object Number	35B4 (hex)
DIM	-	PROFIBUS PNU	1780 (dec) IND = 0000xxxx (bin)
Range	0, 1, .. , 8	DPR	180 (dec)
Default	1		
Opmode	All	Data Type Bus/DPR	Integer8
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.3
Function Group	Amplifier	EEPROM	Yes
Short Description	Operating Mode		

Description

The OPMODE command is used to set the operating mode (basic function) for the amplifier. This operating mode can be changed over at any time, through the fieldbus interface or the digital I/O (INxMODE [▶ 116]=24).

The following settings are possible:

Status	Short Description	Description
OPMODE=0	Velocity Control Digital	Digital (rotational) velocity There are different possibilities to generate a velocity setpoint: - RS232 Interface ("J [▶ 241]" command) - Fieldbus Interface (PROFIBUS, CANopen, SERCOS, DPR Slot boards) - INxMODE [▶ 116] 35 (VCT entry started by I/O)
OPMODE=1	Velocity Control Analog	Analog (rotational) velocity The velocity setpoint is generated by the analog inputs SW1/ SW2. The configuration is done with ANCNFG [▶ 60] and INxMODE [▶ 116]=8.
OPMODE=2	Current Control Digital	Digital torque The current setpoint can be generated by: - RS232 Interface ("T [▶ 247]" command) - Fieldbus Interface (PROFIBUS, CANopen, SERCOS, DPR Slot boards)
OPMODE=3	Current Control Analog	Analog torque The current setpoint is generated by the analog inputs SW1/ SW2. The configuration is done with ANCNFG [▶ 60] and INxMODE [▶ 116]=8.
OPMODE=4	Electronic Gearing (Master/Slave)	Position: electr. gearing The target position is generated by an extrenal encoder. The type of the activated interface is selected by GEARMODE [▶ 213]. If a SinCos type is selectzed by FBTYPE [▶ 190] the FPGA [▶ 77]has to be set to 3 (FW >4.56)
OPMODE=5	External Trajectory	Position: ext. position nodes The target position is generated by fieldbus (PROFIBUS, CANopen or DRP Slot board). The cycle time for writing the new position can be selected with the command PTBASE

Status	Short Description	Description
		[▶ 287] in 250µs steps. The position controller brings the actual position to the new target postion in the selected time. When using ANCNFG=8, the target position is given by the analog input SW1. The analog voltage is read every 250µs and is used as target position for the position controller. The scaling of the anlog input voltage is done with SRND [▶ 296] and ERND [▶ 251]. Before this function is active, a homing move has to started.
OPMODE=6	Sercos Position	SERCOS position control
OPMODE=7	Reserved	
OPMODE=8	Motion Tasks	Position: motion blocks This setting allows the swtarting of motion tasks and also the homing moves.

4.2.31 SAVE

ASCII - Command	**SAVE**		
Syntax Transmit	SAVE		
Syntax Receive	SAVE	**Available in**	
Type	Command	**MMI**	Yes
ASCII Format	-	**CANBus Object Number**	35EB (hex)
DIM	-	**PROFIBUS PNU**	1835 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	235 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Amplifier	**EEPROM**	-
Short Description	Save Data in EEPROM		

Description

The SAVE command stores the present settings of the amplifier parameters in the serial EEPROM. At the same time, the checksum for the parameter field is updated and saved in the serial EEPROM. The save process takes about 2 seconds. During this time, the 24V supply for the amplifier must not be switched off. If this supply voltage is switched off during the save process, this may result in invalid data (or none) being saved in the serial EEPROM. A checksum error will be detected at the next power-on of the equipment, and the fault message F09 will be generated.

Furthermore, all the amplifier parameters will be reset to the default values. To reset the F09 fault, the SAVE command must be used once more, and the amplifier must be switched off and on again.

4.2.32 UID

ASCII - Command	**UID**			
Syntax Transmit	UID [Data]			
Syntax Receive	UID <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer16		**CANBus Object Number**	3616 (hex)
DIM	-		**PROFIBUS PNU**	1878 (dec) IND = 0000xxxx (bin)
Range	Int		**DPR**	278 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.6
Function Group	Amplifier		**EEPROM**	Yes
Short Description	User-ID			

Description

The UID variable is used to distinguish customers. Many customers already have numbers assigned.

4.2.33 UID1

ASCII - Command	**UID1**			
Syntax Transmit	UID1 [Data]			
Syntax Receive	UID1 <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer32		**CANBus Object Number**	3634 (hex)
DIM	-		**PROFIBUS PNU**	1908 (dec) IND = 0000xxxx (bin)
Range	long int		**DPR**	308 (dec)
Default	0			
Opmode	-		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	2.49			
Configuration	No		**Revision**	1.3
Function Group	Amplifier		**EEPROM**	Yes
Short Description	Unused Variable for Customer use			

Description

The UID1 variable is used to store customer information. This variable is not affected and does not affect the firmware.

4.2.34 UPDATE

ASCII - Command	**UPDATE**			
Syntax Transmit	UPDATE [Data]			
Syntax Receive	UPDATE		**Available in**	
Type	Command		**MMI**	No
ASCII Format	-		**CANBus Object Number**	No
DIM	Name		**PROFIBUS PNU**	No
Range	ALL,USER,TABLE,PROG,PORDER,MBASE,Lookup		**DPR**	No
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	Disabled		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.6
Function Group	Amplifier		**EEPROM**	-
Short Description	Program Update via RS232			

Description

The UPDATE command makes it possible to program the internal Flash EEPROM via the serial interface.

Procedure for a software update:

1. Connect the amplifier to the serial interface of a PC.
2. Start the terminal program HINT2.EXE COM2: 9600 (the computer should be booted in DOS mode).This establishes the connection between the PPC and the command interpreter of the Drive. To check the connection, enter the LIST command. The response should be a list of all the available commands, shown on the screen.
3. Enter the command UPDATE [▶ 53] ALL xxxx in the command line.
   xxxx - Name of the firmware file that is to be programmed. Take care that the file xxx for programming is in the active directory. If no file name is entered, then the name ALL is used as a default.

When the drive is programmed with ALL, the loaded motion tasks are deleted. Save motion tasks before programming the firmware.

The programming takes about 40 minutes.

The addresses that are programmed are displayed during the programming procedure. If the display remains static, but the download has not been completed (this problem has been observed on several PCs), then operate the ENTER key. The programming will then carry on.
If the programming procedure is interrupted, then the amplifier will report this in the monitor program at the next power-on (a □-□ sign in the first position of the display).
In this case, the download can be restarted by the following command sequence:
X: xxxx

xxxx - Name of the firmware file to be programmed.

Procedure for loading a motor database:

A standard motor database is included as a subset of the firmware. If a customer-specific motor database is required, then it can be loaded after the firmware has been programmed.
This requires the following command:

UPDATE [▶ 53] MBASE xxxx

xxxx - Name of the motor database file that is to be programmed. Take care that the file xxx for programming is in the active directory. If no file name is entered, then the name MBASE is used as a default.

4.2.35 VBUSMAX

ASCII - Command	**VBUSMAX**			
Syntax Transmit	VBUSMAX			
Syntax Receive	VBUSMAX <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Integer32		**CANBus Object Number**	361C (hex)
DIM	Volts		**PROFIBUS PNU**	1884 (dec) IND = 0000xxxx (bin)
Range	450, 800, 900		**DPR**	284 (dec)
Default				
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Amplifier		**EEPROM**	Yes
Short Description	Maximum DC-bus Voltage			

Description

The VBUSMAX parameter shows the value for the monitoring threshold for the fault message F02 (overvoltage).

This fault message is generated as soon as the DC-bus voltage goes above the value of VBUSMAX. The VBUSMAX threshold depends on the setting for VBUSBAL [▶ 88].

- VBUSBAL=0 (230 V) VBUSMAX=450V
- VBUSBAL=1 (400 V) VBUSMAX=800V
- VBSBAL=2 (480 V) VBUSMAX=900V

4.2.36 VBUSMIN

ASCII - Command	**VBUSMIN**			
Syntax Transmit	VBUSMIN [Data]			
Syntax Receive	VBUSMIN <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer16		**CANBus Object Number**	361D (hex)
DIM	Volts		**PROFIBUS PNU**	1885 (dec) IND = 0000xxxx (bin)
Range	30 .. 800		**DPR**	285 (dec)
Default	100			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Amplifier		**EEPROM**	Yes
Short Description	Minimum DC-bus Voltage			

Description

VBUSMIN defines the lower threshold for monitoring the DC-bus voltage. The fault message F05 (undervoltage) is generated as soon as the DC-bus voltage goes below this threshold. Undervoltage monitoring is only active under the following conditions.

1. Output stage is enabled.

2. Monitoring is activated UVLTMODE [▸ 87]=1
3. MAINSBTB function is not active (OxMODE [▸ 150]<>3)
4. The switch-off of the monitoring function by a digital input (INxMODE [▸ 116]=21) is not active.

4.2.37 VBW

ASCII - Command	VBW		
Syntax Transmit	VBW		
Syntax Receive	VBW	**Available in**	
Type	Command	**MMI**	No
ASCII Format	-	**CANBus Object Number**	No
DIM	-	**PROFIBUS PNU**	No
Range	0	**DPR**	No
Default	0		
Opmode	0	**Data Type Bus/DPR**	-
Drive State	Enabled	**Weightning**	
Start Firmware	2.44		
Configuration	No	**Revision**	1.3
Function Group	Amplifier	**EEPROM**	-
Short Description	Generate Bode Diagram		

Description

VBW [fmin] [fmax] [sample]

Velocity BandWidth

Using the function VBW the Drive calculates a velocity loop Bode plot. Default are 50 samples (sample) between 20 Hz (fmin) and 500 Hz (fmax). Output is the gain in db and the phase shift in degree of the open loop and the closed loop of the velocity controller. It is necessary to set GVFR [▸ 335] to 1 to get the open loop result. Using MSG [▸ 99] 2 the results are immediately shown. Standard is to query the data with the GET [▸ 240] command. To use the function VBW the drive should be in OPMODE [▸ 50] 0 and enabled. The shaft will move only a few degrees. In case of resonance load velocity overshoot can cause a Fault, which can result in an uncontrolled coasting of the motor.

4.2.38 VCOMM

ASCII - Command	VCOMM		
Syntax Transmit	VCOMM [Data]		
Syntax Receive	VCOMM <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Float	**CANBus Object Number**	365A (hex)
DIM	rpm	**PROFIBUS PNU**	1946 (dec) IND = 0000xxxx (bin)
Range	0 .. 1.2 * MSPEED	**DPR**	346 (dec)
Default	1500		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	3.35		
Configuration	No	**Revision**	1.4
Function Group	Amplifier	**EEPROM**	Yes
Short Description	Velocity Threshold for Commutation error		

Description

The command VCOMM defines the threshold for the commutation error function.

The definition of a commutation error is that the sign of the actual current has the right relationship to the sign of the change of the velocity of the motor. This indicates a run-away of the motor and causes a disable of the output stage.

This commutation error is supervised if the actual velocity is above the VCOMM threshold. To disable the function, VCOMM must be set to VLIM [▸ 338].

4.2.39 VTUNE

ASCII - Command	**VTUNE**		
Syntax Transmit	VTUNE		
Syntax Receive	VTUNE	**Available in**	
Type	Command	**MMI**	No
ASCII Format	-	**CANBus Object Number**	No
DIM	-	**PROFIBUS PNU**	No
Range	0	**DPR**	No
Default	0		
Opmode	0	**Data Type Bus/DPR**	-
Drive State	Enabled	**Weightning**	
Start Firmware	2.44		
Configuration	No	**Revision**	1.3
Function Group	Amplifier	**EEPROM**	-
Short Description	Calculate Velocity Control Loop Parameters		

4.2.40 WMASK

ASCII - Command	**WMASK**		
Syntax Transmit	WMASK [Data]		
Syntax Receive	WMASK <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	363E (hex)
DIM	-	**PROFIBUS PNU**	1918 (dec) IND = 0000xxxx (bin)
Range	0 .. 4294967295 (1 Bit is 1 warning)	**DPR**	318 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	2.49		
Configuration	No	**Revision**	1.7
Function Group	Amplifier	**EEPROM**	No
Short Description	Warning as Fault Mask		

Description

The WMASK parameter can be used to reconfigure a warning as the fault message F24. The WMASK parameter is a bit-variable, with bit assignments that correspond to the STATCODE * [▸ 182] status variable. The bit that is set within the WMAKS variable means that the corresponding warning bit in the STATCODE * [▸ 182] variable should generate an F24 fault message, as well as a warning. Unlike warnings, a fault message results in the disabling of the output stage, and the opening of the BTB contact. A reconfiguration of a warning to a fault message can be especially relevant for the following warnings: contouring/following error, threshold detection, hardware limit switch. The value must be entered in decimal.

4.3 Analog I/O

4.3.1 AN10TX

ASCII - Command	AN10TX			
Syntax Transmit	AN10TX [Data]			
Syntax Receive	AN10TX <Data>		Available in	
Type	rw		MMI	No
ASCII Format	Decimal16		CANBus Object Number	No
DIM	counts		PROFIBUS PNU	No
Range	1000 ... 30000		DPR	No
Default	5000			
Opmode	2		Data Type Bus/DPR	Decimal16
Drive State	-		Weightning	
Start Firmware	4.91			
Configuration	No		Revision	2.0
Function Group	analog I/O		EEPROM	-
Short Description				

Description

With Servostar FW greater than 4.91 it will be possible to have an additional torque/ current loop in the drive with ANCNFG [▶ 60] = 10. The drive will read the analog input 1 and use it as torque/ current feedback to adjust the digital current command given by MMI command or fieldbus. With the parameter AN10TX it is possible to tune this additional torque loop. A higher value in this paramter will increase the dynamic of this loop and can cause ringing of this loop. A smaller value decreases the dynamic of this loop an cause higher response time.

- ANCNFG 10
- OPMODE 2
- ISCALE in A/Volt according the analog torque feedback
- AN10TX x x=Time constant of this new loop (default = 5000; Min = 1000; Max = 30000)

4.3.2 AN11NR

ASCII - Command	AN11NR			
Syntax Transmit	AN11NR [Data]			
Syntax Receive	AN11NR <Data>		Available in	
Type	Variable rw		MMI	No
ASCII Format	Integer8		CANBus Object Number	3699 (hex)
DIM	-		PROFIBUS PNU	2009 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2, 3, 4		DPR	409 (dec)
Default	0			
Opmode	All		Data Type Bus/DPR	Integer8
Drive State	Disable		Weightning	
Start Firmware	4.78			
Configuration	No		Revision	1.5
Function Group	Analog I/O		EEPROM	Yes

Short Description	No. Of INxTRIG variable, that is changed analog

Description

The parameter AN11NR defines the number (x) of the auxiliary variable IN1TRIG [▸ 122], IN2TRIG [▸ 128], IN3TRIG [▸ 134] or IN4TRIG [▸ 141], which can be changed by the analog input 2: This parameter has effect only with ANCNFG [▸ 60]=11 and ANCNFG [▸ 60]=12.

4.3.3 AN11RANGE

ASCII - Command	**AN11RANGE**		
Syntax Transmit	AN11RANGE [Data]		
Syntax Receive	AN11RANGE <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer32	**CANBus Object Number**	369A (hex)
DIM	-	**PROFIBUS PNU**	2010 (dec) IND = 0000xxxx (bin)
Range	-262144 .. 262143	**DPR**	410 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	4.78		
Configuration	No	**Revision**	1.6
Function Group	Analog I/O	**EEPROM**	Yes
Short Description	Range of the analog change of INxTRIG		

Description

The parameter AN11RANGE gives the change of IN1TRIG [▸ 122], IN2TRIG [▸ 128], IN3TRIG [▸ 134] or IN4TRIG [▸ 141], that is caused by an analog input 2 step from 0V to 10V. The function is supported using ANCNFG [▸ 60]=11 and ANCNFG [▸ 60]=12.

Example:

ANCNFG [▸ 60]=11
AN11NR [▸ 57]=1
IN1TRIG [▸ 122]=1000
AN11RANGE=500

at Analog input2 = 0V IN1TRIG [▸ 122] = 1000
at Analog input2 = 10V IN1TRIG [▸ 122] = 1500
at Analog input2 = -10V IN1TRIG [▸ 122] = 500

4.3.4 AN1TRIG

ASCII - Command	AN1TRIG		
Syntax Transmit	AN1TRIG [Data]		
Syntax Receive	AN1TRIG <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer32	**CANBus Object Number**	36A1 (hex)
DIM	-	**PROFIBUS PNU**	2017 (dec) IND = 0000xxxx (bin)
Range	Long Int	**DPR**	417 (dec)
Default	100		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	4.93		
Configuration	No	**Revision**	1.6
Function Group	Analog I/O	**EEPROM**	Yes
Short Description	Scaling of the analog output 1		

Description

Gives the possibility to scale the analog output. The scaling is done in %.

Example:

ANOUT1 [▸ 64] = 1 Actual velocity
AN1TRIG = 100 10V at the output at actual velocity=VLIM [▸ 338]
AN1TRIG = 50 5V at the output at actual velocity=VLIM [▸ 338]
AN1TRIG = 200 10V at the output at actual velocity=VLIM [▸ 338]/2

If ANOUT1 [▸ 64]=8 is selected (constant voltage), AN1TRIG gives the voltage in mV.

4.3.5 AN2TRIG

ASCII - Command	AN2TRIG		
Syntax Transmit	AN2TRIG [Data]		
Syntax Receive	AN2TRIG <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer32	**CANBus Object Number**	36A2 (hex)
DIM	-	**PROFIBUS PNU**	2018 (dec) IND = 0000xxxx (bin)
Range	Long Int	**DPR**	418 (dec)
Default	100		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	4.93		
Configuration	No	**Revision**	1.7
Function Group	Analog I/O	**EEPROM**	Yes
Short Description	Scaling of the analog output 2		

Description

Gives the possibility to scale the analog output. The scaling is done in %.

Example:

ANOUT2 [▸ 64] = 1 Actual velocity
AN2TRIG = 100 10V at the output at actual velocity=VLIM [▸ 338]
AN2TRIG = 50 5V at the output at actual velocity=VLIM [▸ 338]
AN2TRIG = 200 10V at the output at actual velocity=VLIM [▸ 338]/2

If ANOUT2 [▸ 64]=8 is selected (constant voltage), AN2TRIG gives the voltage in mV.

4.3.6 ANCNFG

ASCII - Command	**ANCNFG**		
Syntax Transmit	ANCNFG [Data]		
Syntax Receive	ANCNFG <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	3507 (hex)
DIM	-	**PROFIBUS PNU**	1607 (dec) IND = 0000xxxx (bin)
Range	0 .. 14	**DPR**	7 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	Disabled + Reset (Coldstart)	**Weightning**	
Start Firmware	1.20		
Configuration	Yes	**Revision**	1.6
Function Group	Analog I/O	**EEPROM**	Yes
Short Description	Configuration of Analog Input		

Description

The ANCNFG command is used to configure the function of the analog inputs.

Since the ANCNFG variable is used for the configuration of the instrument, the controller must be switched off and then on again after the variable has been changed (use SAVE [▸ 51] first).

Status		**Description**
ANCNFG=0	(Xcmd=Setp.1)	SW1 is used as velocity setpoint or current setpoint depending on the status of OPMODE [▸ 50] (velocity in OPMODE [▸ 50] =1; current in OPMODE [▸ 50]=3). If one of the digital inputs is selected for IN1TRIG [▸ 122], IN2TRIG [▸ 128], IN3TRIG [▸ 134] or IN4TRIG [▸ 141]=8 (switch-over between SW1 and SW2), SW1 (input=low) or SW2 (input=high) have the functionality. Scaling : SW1 velocity Setpoint VSCALE1 [▸ 69] (OPMODE [▸ 50]=1) SW1 Current Setpoint ISCALE1 [▸ 67] (OPMODE [▸ 50]=3) SW2 velocity Setpoint VSCALE2 [▸ 70] (OPMODE [▸ 50]=1) SW2 Current Setpoint ISCALE2 [▸ 68] (OPMODE [▸ 50]=3)
ANCNFG=1	v_cmd=Setp.1, Icmd=Setp.2	SW1 is used as velocity setpoint if OPMODE [▸ 50] = 1 (scaling factor VSCALE1 [▸ 69]) SW2 is used as current setpoint if OPMODE [▸ 50] = 3 (scaling factor ISCALE2 [▸ 68])

Status		Description
ANCNFG=2	Setp.1 = nsoll, Setp.2 = Isoll	SW1 velocity setpoint SW2 current feedforward (OPMODE [▶ 50]=0,1) scaling factor ISCALE2 [▶ 68]
ANCNFG=3	Xcmd=Setp.1, Ipeak1=Setp.2	SW1 depending on OPMODE [▶ 50] velocity or current setpoint (scaling VSCALE1 [▶ 69] or ISCALE1 [▶ 67]) The absolute of SW2 limits the current of the drive 10V 100% of IPEAK [▶ 110] 5V 50% of IPEAK [▶ 110]
ANCNFG=4	Xcmd=Setp.1+Setp.2	The sum of SW1 and SW2 is used for velocity or current setpoint, depending on OPMODE [▶ 50]. OPMODE [▶ 50] 1 velocity setpoint OPMODE [▶ 50] 3 current setpoint
ANCNFG=5	Xcmd=Setp.1*Setp.2	The multiplikation of SW1 and SW2 is used for velocity or current setpoint, depending on OPMODE [▶ 50]. OPMODE [▶ 50] 1 velocity setpoint OPMODE [▶ 50] 3 current setpoint SW1 VSCALE1 [▶ 69]/ISCALE1 [▶ 67] SW2 10V means 100% -10V means -100%
ANCNFG=6	Electronic Gearing	SW1 is used as velocity or current setpoint, depending on OPMODE [▶ 50] SW2 is used as scaling factor for electronic gearing (OPMODE [▶ 50]=4). VSCALE2 [▶ 70] is used to define a correction factor in %. e.g. VSCALE2 [▶ 70]=20 (means 20%) SW2= +10V GEAROeff=GEARO [▶ 218] * 1.2 SW2= -10V GEAROeff=GEARO [▶ 218] * 0.8 SW2= 0V GEAROeff=GEARO [▶ 218]
ANCNFG=7	Setp.1 = Isoll, Setp.2 = Nmax	SW1 is used as current setpoint (ISCALE1) (OPMODE [▶ 50] has to be set to 3). SW2 limits the velocity of the motor SW2=10V, Nmax=(VSCALE2 [▶ 70]) If the velocity of the motor is greater than Nmax, the velocity is limited.
ANCNFG=8	Setp.1 = Psoll	SW1 is used as a analog position setpoint (only available in OPMODE [▶ 50]=5). The working distance is defined by SRND [▶ 296] and ERND [▶ 251]. SW1 = 0V Position = SRND [▶ 296] SW1= +/-10V Position = ERND [▶ 251] When the drive is switched on, the reference point is not set and the drive does not move. The OPMODE [▶ 50] can be set to 5. The a reference move can be starte bs digital input. After that, when the homing move is finished, the input can be set to zero again and then the drive moves automatically to the given analog position. POSCNFG [▶ 285] has to be "0" (linear axes type).
ANCNFG=9		Analog input 1: velocity or current setpoint (same as ANCNFG=0) Analog input 2: Ferraris sensor
ANCNFG=10	Reserved	

Status		Description
ANCNFG=11		Change of an IN1TRIG [▸ 122], IN2TRIG [▸ 128], IN3TRIG [▸ 134] or IN4TRIG [▸ 141] variable via the analog input 2. The corresponding Number (x) of the trigger variable is set by AN11NR [▸ 57]. The range of the parameter change is defined by AN11RANGE [▸ 58]. The change of the analog in 2 directly changes the INxTRIG variable (update time 1 to 10ms), see also AN11NR [▸ 57] and AN11RANGE [▸ 58].
ANCNFG=12		Change of an IN1TRIG [▸ 122], IN2TRIG [▸ 128], IN3TRIG [▸ 134] or IN4TRIG [▸ 141] variable via the analog input 2. The corresponding Number (x) of the trigger variable is set by AN11NR [▸ 57]. The range of the parameter change is defined by AN11RANGE [▸ 58]. The change of the analog in 2 changes the INxTRIG variable after a rising edge at digital inputx, see also AN11NR [▸ 57] and AN11RANGE [▸ 58].
ANCNFG=13	Xcmd=Setp.1, Ipeak1=Setp.2	SW1 depending on OPMODE [▸ 50] velocity or current setpoint (scaling VSCALE1 [▸ 69] or ISCALE1 [▸ 67]) The absolute of SW2 limits the positive current of the drive 10V 100% of IPEAK [▸ 110] 5V 50% of IPEAK [▸ 110] The negative current is not effected. In the positive direction, the acceleration current is limited and in the negative direction the deceleration current.
ANCNFG=14	Xcmd=Setp.1, Ipeak1=Setp.2	SW1 depending on OPMODE [▸ 50] velocity or current setpoint (scaling VSCALE1 [▸ 69] or ISCALE1 [▸ 67]) The absolute of SW2 limits the negative current of the drive 10V 100% of IPEAK [▸ 110] 5V 50% of IPEAK [▸ 110] The positive current is not effected. In the negative direction, the acceleration current is limited and in the negative direction the deceleration current.

4.3.7 ANDB

ASCII - Command	**ANDB**			
Syntax Transmit	ANDB [Data]			
Syntax Receive	ANDB <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Float		**CANBus Object Number**	3508 (hex)
DIM	Millivolts		**PROFIBUS PNU**	1608 (dec) IND = 0000xxxx (bin)
Range	0.0 .. 10000.0		**DPR**	8 (dec)
Default	0			
Opmode	1, 3		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	1.20			
Configuration	No		**Revision**	1.8
Function Group	Analog I/O		**EEPROM**	Yes
Short Description	Dead Band of the Analog Velocity Input Signal			

Description

This variable suppresses small analog input signals by setting a dead band zone in which signals are ignored. This function is useful with OPMODE [▸ 50]=1 (without higher-level position control). Depending on the operating mode, this parameter applies to SW1 or SW2 (depending on which setpoint input is used as the source for the velocity value). See ANCNFG [▸ 60] for additional information.

4.3.8 ANOFF1

ASCII - Command	**ANOFF1**			
Syntax Transmit	ANOFF1 [Data]			
Syntax Receive	ANOFF1 <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer16		**CANBus Object Number**	350B (hex)
DIM	Millivolts		**PROFIBUS PNU**	1611 (dec) IND = 0000xxxx (bin)
Range	-10000 .. 10000		**DPR**	11 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.8
Function Group	Analog I/O		**EEPROM**	Yes
Short Description	Analog Offset for input SW1			

Description

This variable compensates for the offset voltages of CNC controls and the analog input, ANIN1 [▸ 20] (SW1). It can also correct an analog offset from external controls.

4.3.9 ANOFF2

ASCII - Command	**ANOFF2**			
Syntax Transmit	ANOFF2 [Data]			
Syntax Receive	ANOFF2 <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer16		**CANBus Object Number**	350C (hex)
DIM	Millivolts		**PROFIBUS PNU**	1612 (dec) IND = 0000xxxx (bin)
Range	-10000 .. 10000		**DPR**	12 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.8
Function Group	Analog I/O		**EEPROM**	Yes
Short Description	Analog Offset for input SW2			

Description

This variable compensates for the offset voltages of CNC controls and the analog input, ANIN2 [▸ 20] (SW2). It can also correct an analog offset from external controls.

4.3.10 ANOUT1

ASCII - Command	ANOUT1			
Syntax Transmit	ANOUT1 [Data]			
Syntax Receive	ANOUT1 <Data>		Available in	
Type	Variable rw		MMI	Yes
ASCII Format	Integer8		CANBus Object Number	350D (hex)
DIM	-		PROFIBUS PNU	1613 (dec) IND = 0000xxxx (bin)
Range	0 .. 8		DPR	13 (dec)
Default	1			
Opmode	All		Data Type Bus/DPR	Integer8
Drive State	Disabled + Reset (Coldstart)		Weightning	
Start Firmware	1.20			
Configuration	Yes		Revision	1.8
Function Group	Analog I/O		EEPROM	Yes
Short Description	Configuration of the Analog Output 1			

Description

Configuration of analog output 1. The actual value is read via MONITOR1 [▸ 68]. The output provides various analog setpoint values or actual values, depending on the selection in the operator software. Output resistor 2.2kOhm. Resolution 10 bit.

Status	Description
ANOUT1=0	No output voltage at Analog Output 1.
ANOUT1=1	Outputs the actual velocity (10V = VLIM [▸ 338]).
ANOUT1=2	Outputs the actual current (10V = IPEAK [▸ 110]).
ANOUT1=3	Outputs the velocity setpoint (10V = VLIM [▸ 338]).
ANOUT1=4	Outputs the current setpoint (10V = IPEAK [▸ 110]).
ANOUT1=5	Outputs the actual contouring error (10V = PEMAX [▸ 282]).
ANOUT1=6	Outputs a value given by a option DPR-slotboard. If a Device-Net option board is plugged in the drive, this setting enables access of Device-Net to analog output 1
ANOUT1=7	The actual position is at the analog output. The scaling is reffered to the Modulo axes defined by SRND [▸ 296] and ERND [▸ 251]. This output makes sense, using the modulo axes type (POSCNFG [▸ 285]=2).
ANOUT1=8	The analog output gives a constant voltage. The voltage can be set by using the help variable AN1TRIG [▸ 59] in mV (starting with firmware 4.91)

4.3.11 ANOUT2

ASCII - Command	ANOUT2		

Syntax Transmit	ANOUT2 [Data]			
Syntax Receive	ANOUT2 <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer8		**CANBus Object Number**	350E (hex)
DIM	-		**PROFIBUS PNU**	1614 (dec) IND = 0000xxxx (bin)
Range	0 .. 8		**DPR**	14 (dec)
Default	2			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	Disabled + Reset (Coldstart)		**Weightning**	
Start Firmware	1.20			
Configuration	Yes		**Revision**	1.8
Function Group	Analog I/O		**EEPROM**	Yes
Short Description	Source of the Analog Output 2			

Description

Configuration of analog output 2. The actual value is read via MONITOR2 [▸ 69]. The output provides various analog setpoint values or actual values, depending on the selection in the operator software. Output resistor 2.2kOhm. Resolution 10 bit.

Status	Description
ANOUT2=0	No output voltage at Analog Output 2.
ANOUT2=1	Outputs the actual velocity (10V = VLIM [▸ 338]).
ANOUT2=2	Outputs the actual current (10V = IPEAK [▸ 110]).
ANOUT2=3	Outputs the velocity setpoint (10V = VLIM [▸ 338]).
ANOUT2=4	Outputs the current setpoint (10V = IPEAK [▸ 110]).
ANOUT2=5	Outputs the actual contouring error (10V = PEMAX [▸ 282]).
ANOUT2=6	Outputs a value given by a option DPR-slotboard. If a Device-Net option board is plugged in the drive, this setting enables access of Device-Net to analog output 2
ANOUT2=7	The actual position is at the analog output. The scaling is reffered to the Modulo axes defined by SRND [▸ 296] and ERND [▸ 251]. This output makes sense, using the modulo axes type (POSCNFG [▸ 285]=2).
ANOUT2=8	The analog output gives a constant voltage. The voltage can be set by using the help variable AN2TRIG [▸ 59] in mV (starting with firmware 4.91)

4.3.12 ANZERO1

ASCII - Command	**ANZERO1**			
Syntax Transmit	ANZERO1			
Syntax Receive	ANZERO1		**Available in**	
Type	Command		**MMI**	Yes
ASCII Format	-		**CANBus Object Number**	350F (hex)
DIM	-		**PROFIBUS PNU**	1615 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	15 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Analog I/O		**EEPROM**	-
Short Description	Zero Analog Input SW1			

Description

This command can be used to start the automatic offset correction for the analog input 1. The setpoint at analog input 1 should be short-circuited before using this command. After the command has been carried out, the offset value that was determined is available in the ANOFF1 [▸ 63] parameter. To save this value permanently in the EEPROM, you should use the SAVE [▸ 51] (save to EEPROM) command.

4.3.13 ANZERO2

ASCII - Command	**ANZERO2**			
Syntax Transmit	ANZERO2			
Syntax Receive	ANZERO2		**Available in**	
Type	Command		**MMI**	Yes
ASCII Format	-		**CANBus Object Number**	3510 (hex)
DIM	-		**PROFIBUS PNU**	1616 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	16 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Analog I/O		**EEPROM**	-
Short Description	Zero Analog Input SW2			

Description

This command can be used to start the automatic offset correction for the analog input 2. The setpoint at analog input 2 should be short-circuited before using this command. After the command has been carried out, the offset value that was determined is available in the ANOFF2 [▸ 63] parameter. To save this value permanently in the EEPROM, you should use the SAVE [▸ 51] (save to EEPROM) command.

4.3.14 AVZ1

ASCII - Command	AVZ1			
Syntax Transmit	AVZ1 [Data]			
Syntax Receive	AVZ1 <Data>		Available in	
Type	Variable rw		MMI	Yes
ASCII Format	Float		CANBus Object Number	3511 (hex)
DIM	Milliseconds		PROFIBUS PNU	1617 (dec) IND = 0000xxxx (bin)
Range	0.2 .. 100.0		DPR	17 (dec)
Default	1			
Opmode	1		Data Type Bus/DPR	Integer32
Drive State	-		Weightning	1000
Start Firmware	1.20			
Configuration	No		Revision	1.3
Function Group	Analog I/O		EEPROM	Yes
Short Description	Filter Time Constant Input SW1			

Description

Filter time constant for analog input SW1. (250µs Update Rate)

4.3.15 ISCALE1

ASCII - Command	ISCALE1			
Syntax Transmit	ISCALE1 [Data]			
Syntax Receive	ISCALE1 <Data>		Available in	
Type	Variable rw		MMI	Yes
ASCII Format	Float		CANBus Object Number	3571 (hex)
DIM	A/10Volts		PROFIBUS PNU	1713 (dec) IND = 0000xxxx (bin)
Range	0.0 .. 100.0		DPR	113 (dec)
Default	DIPEAK			
Opmode	3		Data Type Bus/DPR	Integer32
Drive State	-		Weightning	1000
Start Firmware	1.20			
Configuration	No		Revision	1.3
Function Group	Analog I/O		EEPROM	Yes
Short Description	Scaling of Analog Current Setpoint 1			

Description

Defines the scaling for the analog setpoint input SW1 (if it is a current setpoint in OPMODE [▸ 50] = 3). The current value that is set here corresponds to the maximum input voltage (10V).

4.3.16 ISCALE2

ASCII - Command	**ISCALE2**			
Syntax Transmit	ISCALE2 [Data]			
Syntax Receive	ISCALE2 <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Float		**CANBus Object Number**	3572 (hex)
DIM	A/10Volts		**PROFIBUS PNU**	1714 (dec) IND = 0000xxxx (bin)
Range	0.0 .. 100.0		**DPR**	114 (dec)
Default	DIPEAK			
Opmode	3		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Analog I/O		**EEPROM**	Yes
Short Description	Scaling of Analog Current Setpoint 2			

Description

Defines the scaling for the analog setpoint input SW2 (if it is a current setpoint in OPMODE [▶ 50] = 3). The current value that is set here corresponds to the maximum input voltage (10V).

4.3.17 MONITOR1

ASCII - Command	**MONITOR1**			
Syntax Transmit	MONITOR1			
Syntax Receive	MONITOR1 <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Integer16		**CANBus Object Number**	359A (hex)
DIM	mV		**PROFIBUS PNU**	1754 (dec) IND = 0000xxxx (bin)
Range	-10000 ..10000		**DPR**	154 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Analog I/O		**EEPROM**	No
Short Description	Monitor 1 Output voltage			

Description

The actual value of the output voltage from Monitor 1.

4.3.18 MONITOR2

ASCII - Command	**MONITOR2**			
Syntax Transmit	MONITOR2			
Syntax Receive	MONITOR2 <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Integer16		**CANBus Object Number**	359B (hex)
DIM	mV		**PROFIBUS PNU**	1755 (dec) IND = 0000xxxx (bin)
Range	-10000 ..10000		**DPR**	155 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Analog I/O		**EEPROM**	No
Short Description	Monitor 2 Output Voltage			

Description

The actual value of the output voltage from Monitor 2.

4.3.19 VSCALE1

ASCII - Command	**VSCALE1**			
Syntax Transmit	VSCALE1 [Data]			
Syntax Receive	VSCALE1 <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer16		**CANBus Object Number**	3629 (hex)
DIM	rpm / 10 Volts		**PROFIBUS PNU**	1897 (dec) IND = 0000xxxx (bin)
Range	-15000 .. 15000		**DPR**	297 (dec)
Default	3000			
Opmode	1		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.8
Function Group	Analog I/O		**EEPROM**	Yes
Short Description	SW1 Velocity Scaling Factor			

Description

If the analog input SW1 is used as the setpoint input for velocity control, then the VSCALE1 parameter can be used to set the scaling of the input voltage.
A 10V velocity setpoint input at input SW1 produces a velocity of VSCALE1.

4.3.20 VSCALE2

ASCII - Command	VSCALE2		
Syntax Transmit	VSCALE2 [Data]		
Syntax Receive	VSCALE2 <Data>	Available in	
Type	Variable rw	MMI	Yes
ASCII Format	Integer16	CANBus Object Number	362A (hex)
DIM	rpm / 10 Volts	PROFIBUS PNU	1898 (dec) IND = 0000xxxx (bin)
Range	-15000 .. 15000	DPR	298 (dec)
Default	3000		
Opmode	1	Data Type Bus/DPR	Integer16
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.8
Function Group	Analog I/O	EEPROM	Yes
Short Description	SW2 Velocity Scaling Factor		

Description

If the analog input SW2 is used as the setpoint input for velocity control, then the VSCALE2 parameter can be used to set the scaling of the input voltage.
A 10V velocity setpoint input at input SW2 produces a velocity of VSCALE2.

4.4 Basic Setup

4.4.1 ADDR

ASCII - Command	ADDR		
Syntax Transmit	ADDR [Data]		
Syntax Receive	ADDR <Data>	Available in	
Type	Variable rw	MMI	Yes
ASCII Format	Unsigned8	CANBus Object Number	3505 (hex)
DIM	-	PROFIBUS PNU	1605 (dec) IND = 0000xxxx (bin)
Range	0 .. 63	DPR	5 (dec)
Default	0		
Opmode	All	Data Type Bus/DPR	Unsigned8
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.8
Function Group	Basic Setup	EEPROM	Yes
Short Description	Multidrop Address		

Description

This variable defines the station address (0 to 63) for the amplifier. This address is required by the fieldbus (CANBUS, PROFIBUS, SERCOS, etc.) and for the parameter setting of the servo amplifier in a multi-axis system for an unambiguous identification of the servo amplifier within the system. You can use the keys on the front panel of the servo panel to set the station address (refer to the Installation Manual). After changing the address, all parameters should be stored in the EEPROM (see SAVE [▶ 51]) and the amplifier should be switched off and on again.

If MDRV [▶ 98] = 0, the address range is changed to 0 .. 127.

With drive 400, the address of the master has to be set, that the first slave address is >=1.

Example: master with four slaves, minimal value for ADDR of the master is 5.

Using ADDRFB [▸ 90] gives the possibility to select the fieldbus address different from ADDR.

4.4.2 AENA

ASCII - Command	**AENA**		
Syntax Transmit	AENA [Data]		
Syntax Receive	AENA <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	3506 (hex)
DIM	-	**PROFIBUS PNU**	1606 (dec) IND = 0000xxxx (bin)
Range	0 ,1	**DPR**	6 (dec)
Default	1		
Opmode	0, 2, 4, 5, 8	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	1.37		
Configuration	No	**Revision**	2.0
Function Group	Basic Setup	**EEPROM**	Yes
Short Description	Software Auto-Enable		

Description

This variable defines the state of the software enable when the amplifier is switched on. To enable the output stage, both the hardware enable, and the software enable must be set (series AND configuration). The software enable gives an external control the option of enabling or disabling the output stage by software control, via a bus interface (CANBUS, PROFIBUS, SERCOS, RS232) or an expansion card in a slot.

- 0 = inactive
- 1 = active

When using an analog setpoint (OPMODE [▸ 50]=1, 3), the software enable is automatically set when the amplifier is switched on, so that these instruments are instantly ready for operation (provided that the hardware enable is already present). When using a digital setpoint (OPMODE [▸ 50]=0, 2, 4 through 8), the software enable is set to the same state as AENA at power-on.

For faults that can be reset in software after the fault has been cleared (digital input 1 or CLRFAULT [▸ 35]), the software enable is set to the state of AENA. In this way, the response of the amplifier to a software reset is analogous to the power-on behavior.

4.4.3 ALIAS

ASCII - Command	**ALIAS**		
Syntax Transmit	ALIAS [Data]		
Syntax Receive	ALIAS <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	String	**CANBus Object Number**	No
DIM	-	**PROFIBUS PNU**	No
Range	max 8 ASCII Characters	**DPR**	No
Default	DRIVE0		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Basic Setup	**EEPROM**	Yes
Short Description	Drive Name		

Description

The ALIAS command is used to assign a symbolic name to an amplifier. If the PC setup software is used, this name appears in the title bar of all open parameter windows.

In Multi-Drive mode (parameterizing several amplifiers that are grouped through the CAN bus) the ALIAS name can be used to give a clear assignment of the parameter window to the corresponding amplifier.

4.4.4 BOOT

ASCII - Command	**BOOT**		
Syntax Transmit	BOOT [Data]		
Syntax Receive	BOOT <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	365E (hex)
DIM	-	**PROFIBUS PNU**	1950 (dec) IND = 0000xxxx (bin)
Range	0, 1	**DPR**	350 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	3.43		
Configuration	No	**Revision**	1.3
Function Group	Basic Setup	**EEPROM**	Yes
Short Description	Type of Boot Initialization		

Description

The BOOT command selects the type of boot initialization.

BOOT = 0 The internal MACRO program is compiled every time the drive is switched on (24V auxillary supply). This takes about 12...15 seconds and depends only on the selected software configuration.

BOOT = 1 The internal MACRO program is only compiled one time, when the software configuration has changed. The first boot initialization takes about 12...15 seconds, the next times, the initialization time is reduced to 1.5 to 2 seconds.

The reduced boot initialization is available form hardware version 4.

The reduction of the boot initialization is realized, storing the compiled code in one FLASH segment. When the software configuration has not changed, the compiled code is executed directly.

4.4.5 CBAUD

ASCII - Command	**CBAUD**		
Syntax Transmit	CBAUD [Data]		
Syntax Receive	CBAUD <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	3515 (hex)
DIM	kBaud	**PROFIBUS PNU**	1621 (dec) IND = 0000xxxx (bin)
Range	10,20,50,100,125,250, 333,500,666,800,1000	**DPR**	21 (dec)
Default	500		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	Disabled + Reset (Coldstart)	**Weightning**	
Start Firmware	1.20		
Configuration	Yes	**Revision**	1.8
Function Group	Basic Setup	**EEPROM**	Yes
Short Description	Baud Rate CAN Bus		

Description

The transmission rate is required by the fieldbus (CANopen) and for the parameter setting of the servo amplifier in multi-axis systems (see the Installation Manual). You can also use the keys on the front panel of the servo amplifier to set the baud rate (see the Installation Manual).

4.4.6 DENA

ASCII - Command	**DENA**		
Syntax Transmit	DENA [Data]		
Syntax Receive	DENA <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	362D (hex)
DIM	-	**PROFIBUS PNU**	1901 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2	**DPR**	301 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	2.08		
Configuration	No	**Revision**	1.3
Function Group	Basic Setup	**EEPROM**	Yes
Short Description	DPR software disable reset mode		

Description

With external DPR-SLOT cards, it is possible to cancel existing instrument faults by removing the DPR software enable. This function can be activated or inhibited by using the DENA variable.

DENA=0 Removing the software enable causes a hardware/software to reset of the amplifier. The reset only takes place when a fault occurs, or the warning □contouring error□ or □threshold monitoring activated□ is present. (customer-specific protocol: Beckhoff).

DENA=1 Removing the software enable causes a hardware/software to reset of the amplifier. The reset only takes place when a fault occurs, or the warning □contouring error□ or □threshold monitoring activated□ is present.

DENA=2 No reset if the software enable is removed.

4.4.7 DILIM

ASCII - Command	**DILIM**		
Syntax Transmit	DILIM [Data]		
Syntax Receive	DILIM <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	362C (hex)
DIM	-	**PROFIBUS PNU**	1900 (dec) IND = 0000xxxx (bin)
Range	0, 1	**DPR**	300 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	Disabled + Reset (Coldstart)	**Weightning**	
Start Firmware	2.08		
Configuration	Yes	**Revision**	1.3
Function Group	Basic Setup	**EEPROM**	Yes
Short Description	DPR current limit		

Description

With external DPR-SLOT cards it is possible to limit the drive current through the DPR (RAM interface to the SLOT card). This function must be enabled through the DILIM configuration variable.

4.4.8 DPWM

ASCII - Command	**DPWM**		
Syntax Transmit	DPWM [Data]		
Syntax Receive	DPWM <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	3676 (hex)
DIM	-	**PROFIBUS PNU**	1974 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2	**DPR**	374 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	4.02		
Configuration	No	**Revision**	1.8
Function Group	Basic Setup	**EEPROM**	Yes
Short Description	Output Frequency of the Power Stage		

Description

The switching frequency of drive is most usually set to 8 kHz. For some special cases, the switching frequency can be increased to 16 kHz.

The change of the switching frequency is only possible for the 230 power supply (VBUSBAL [▸ 88] = 0).

The set parameter is the variable DPWM.

- DPWM = 0: 8kHz
- DPWM = 1: 16 kHz
- DPWM = 2: 8 kHz without power loss reduction (starting 4.32)

Starting with firmware 4.94 the 16kHz mode can also be used at 400 and 480V.
The restrictions (lower currents) are:

601 IPEAK [▸ 110] and ICONT [▸ 108] have the maximum rating

603-614 400V (VBUSBAL [▸ 88] = 1): IPEAK [▸ 110] = 55% peak current of the drive, ICONT [▸ 108] = 55% of rated current of the drive

603-614 480V (VBUSBAL [▸ 88] = 2): IPEAK [▸ 110] = 45% peak current of the drive, ICONT [▸ 108] = 45% of rated current of the drive

620 400V (VBUSBAL [▸ 88] = 1): IPEAK [▸ 110] max = 26A and ICONT [▸ 108] max = 14A
480V (VBUSBAL [▸ 88] = 2): IPEAK [▸ 110] max = 22A and ICONT [▸ 108] max = 12A

4.4.9 ERRCODE

ASCII - Command	**ERRCODE**		
Syntax Transmit	ERRCODE		
Syntax Receive	ERRCODE <Data>	**Available in**	
Type	Command	**MMI**	Yes
ASCII Format	String	**CANBus Object Number**	No
DIM	-	**PROFIBUS PNU**	No
Range	-	**DPR**	No
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Basic Setup	**EEPROM**	-
Short Description	Activated Fault Messages		

Description

The ERRCODE command returns the clear text information about any existing faults.

4.4.10 FLTCNT

ASCII - Command	**FLTCNT**			
Syntax Transmit	FLTCNT			
Syntax Receive	FLTCNT <Data>		**Available in**	
Type	Command		**MMI**	Yes
ASCII Format	String		**CANBus Object Number**	No
DIM	-		**PROFIBUS PNU**	No
Range	0, 65535 per Fault Message		**DPR**	No
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Drive Status		**EEPROM**	-
Short Description	Fault Frequency			

Description

The FLTCNT command provides a listing of all possible error messages, with the number of occurrences of each type of fault in clear text. The total number of faults (sum of the individual faults) is given out before the fault list.

4.4.11 FLTHIST

ASCII - Command	**FLTHIST**			
Syntax Transmit	FLTHIST			
Syntax Receive	FLTHIST <Data>		**Available in**	
Type	Command		**MMI**	Yes
ASCII Format	String		**CANBus Object Number**	No
DIM	Number and TRUN		**PROFIBUS PNU**	No
Range	10 No. of Last Messages+Times		**DPR**	No
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Drive Status		**EEPROM**	-
Short Description	Fault History: Display last 10 faults			

Description

The FLTHIST command produces a list of the last 10 faults that occurred, together with the corresponding number of operating hours at the time of occurrence, in clear text.

4.4.12 FOLDMODE

ASCII - Command	FOLDMODE		
Syntax Transmit	FOLDMODE [Data]		
Syntax Receive	FOLDMODE <Data>	Available in	
Type	Variable rw	MMI	No
ASCII Format	Integer8	CANBus Object Number	353D (hex)
DIM	-	PROFIBUS PNU	1661 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2	DPR	61 (dec)
Default	0		
Opmode	All	Data Type Bus/DPR	Integer8
Drive State	Disabled + Reset (Coldstart)	Weightning	
Start Firmware	1.20		
Configuration	Yes	Revision	1.3
Function Group	Basic Setup	EEPROM	Yes
Short Description	Foldback Mode		

Description

This command affects the behavior of the amplifier when it reaches the current limit.

- FOLDMODE=0 The instrument delivers the peak current (IPEAK [▸ 110]) for up to 5 seconds, after that the current is limited to the preset rated current (ICONT [▸ 108]).
- FOLDMODE=1 reserved
- FOLDMODE=2 The current limiting to the rated current does not happen. If the I2t value exceeds the threshold of 105%, the output stage is disabled and the "I2T-MAX" fault is generated

4.4.13 FPGA

ASCII - Command	FPGA		
Syntax Transmit	FPGA [Data]		
Syntax Receive	FPGA <Data>	Available in	
Type	Variable rw	MMI	No
ASCII Format	Integer8	CANBus Object Number	363B (hex)
DIM	-	PROFIBUS PNU	1915 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2, 3, 4	DPR	315 (dec)
Default	0		
Opmode	All	Data Type Bus/DPR	Integer8
Drive State	Disabled + Reset (Coldstart)	Weightning	
Start Firmware	2.49		
Configuration	Yes	Revision	2.0
Function Group	Basic Setup	EEPROM	Yes
Short Description	Select different FPGA functionalities		

Description

The command FPGA selects the FPGA program, which is downloaded to the FPGA in the initialization phase.

- FPGA=0 Programm with tracking counter in the encoder simulation output X5 (Drive 400 X4)

- FPGA=1 Program with Up/Down counter (this allows the usage of a high resolution feedback and Master/Slave functionality)
- FPGA=3 Program, which enables an synchronisation of several drives via Can (FW>=4.56)
- FPGA=4 If this program is selected, the SSI output is expanded to 12 Bit resolution and 15 Bit in the turn.

It exists another FPGA program, for the reading of external SSI encoder as a second encoder. This program is automatically selected if GEARMODE [▸ 213]=7 is selected.

4.4.14 HVER

ASCII - Command	**HVER**		
Syntax Transmit	HVER		
Syntax Receive	HVER <Data>	**Available in**	
Type	Variable ro	**MMI**	Yes
ASCII Format	String	**CANBus Object Number**	3557 (hex)
DIM	-	**PROFIBUS PNU**	1687 (dec) IND = 0000xxxx (bin)
Range	max 50 ASCII Characters	**DPR**	87 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Basic Setup	**EEPROM**	No
Short Description	Output the Hardware Version		

Description

The HVER command returns the designation for the hardware version, in the following form:
□Drive 6xx Hardware Version (yy) zzzz□

- xx - designation of the output stage (current rating)
- yy - designation of the hardware version
- zzzz - date of the first hardware revision

4.4.15 KEYLOCK

ASCII - Command	**KEYLOCK**		
Syntax Transmit	KEYLOCK [Data]		
Syntax Receive	KEYLOCK <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	3575 (hex)
DIM	-	**PROFIBUS PNU**	1717 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2	**DPR**	117 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.6
Function Group	Basic Setup	**EEPROM**	Yes
Short Description	Locks the push buttons		

Description

If KEYLOCK=1, operation of the amplifier from the keys on the front panel is inhibited.
The display functions of the instrument (error messages, warnings) remain active.

Keylock=2 is only for drive 4xx. The push button behavior of the master is changed to the behaivior of the standard drive. The slave axes (needs option -DISP) is working also in the same way. In the master module, the variable ADDR [▸ 70] and in the Slave the variable ADDRFB [▸ 90] is changed.

4.4.16 MAXTEMPE

ASCII - Command	**MAXTEMPE**		
Syntax Transmit	MAXTEMPE [Data]		
Syntax Receive	MAXTEMPE <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	3584 (hex)
DIM	Centigrade Degrees	**PROFIBUS PNU**	1732 (dec) IND = 0000xxxx (bin)
Range	10 .. 80	**DPR**	132 (dec)
Default	70		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Basic Setup	**EEPROM**	Yes
Short Description	Ambient Temperature Switch off Threshold		

Description

The ambient temperature value for switching off. If the drive temperature (as given by TEMPE [▸ 30]) exceeds this value, the drive faults.

4.4.17 MAXTEMPH

ASCII - Command	**MAXTEMPH**		
Syntax Transmit	MAXTEMPH [Data]		
Syntax Receive	MAXTEMPH <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	3585 (hex)
DIM	Centigrade Degrees	**PROFIBUS PNU**	1733 (dec) IND = 0000xxxx (bin)
Range	20 .. 85 (90 ;SR640,SR670)	**DPR**	133 (dec)
Default	80		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Basic Setup	**EEPROM**	Yes
Short Description	Heat Sink Temperature Switch off Threshold		

Description

The heat sink temperature for switching off. If the heat sink temperature (as given by TEMPH [▸ 30]) exceeds this value, the drive faults.

4.4.18 MAXTEMPM

ASCII - Command	**MAXTEMPM**		
Syntax Transmit	MAXTEMPM [Data]		
Syntax Receive	MAXTEMPM <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Float	**CANBus Object Number**	3586 (hex)
DIM	Ohm (KOhm)	**PROFIBUS PNU**	1734 (dec) IND = 0000xxxx (bin)
Range	0.0 .. 6000.0	**DPR**	134 (dec)
Default	291		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	2.49		
Configuration	No	**Revision**	1.6
Function Group	Basic Setup	**EEPROM**	Yes
Short Description	Motor Temperature Switch off Threshold		

Description

The motor temperature for switching off (defined by the resistance in ohms).

From version 2.49 the resistance value is given in kilohms, and no longer in ohms. If an amplifier is updated to version 2.49 or higher, the resistance value is automatically converted.

- Firmware <2.49
  Entry in ohms up to 6000 (6000 corresponds to about 800 ohms in reality).
- Firmware 2.49 or above
  Entry in kilohms up to 1.5 (1.5 corresponds to about 1500 ohms in reality).

4.4.19 MSLBRAKE

ASCII - Command	**MSLBRAKE**		
Syntax Transmit	MSLBRAKE [Data]		
Syntax Receive	MSLBRAKE <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer16	**CANBus Object Number**	3671 (hex)
DIM	-	**PROFIBUS PNU**	1969 (dec) IND = 0000xxxx (bin)
Range	1 .. 32	**DPR**	369 (dec)
Default	8		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	4.05		
Configuration	No	**Revision**	1.6
Function Group	Basic Setup	**EEPROM**	Yes
Short Description	DEC ramp at sensorless emergency stop		

Description

If the feedback unit trips (Fault F04, F08 or F25), it is impossible to stop the motor with the standard commutation. Therefore, a sensor less stop is implemented. It is not possible to define a ramp, because it depends on the friction and inertia of the system. MSLBRAKE offers the possibility to change the emergency ramp in this case. The lower MSLBRAKE is, the lower is the deceleration of the motor.

4.4.20 M_ENABLE

ASCII - Command	**M_ENABLE**	**For Manufacturer Use only**	
Syntax Transmit	M_ENABLE		
Syntax Receive	M_ENABLE <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	String	**CANBus Object Number**	No
DIM	-	**PROFIBUS PNU**	No
Range	-	**DPR**	No
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Basic Setup	**EEPROM**	-
Short Description	Display the □Enable□ Macro Program		

Description

The source code of the macro function □Enable□ is displayed on the screen. It is only run once when the amplifier is enabled. The setting PROMPT [▶ 102]=2 makes the display appear page-by-page. Pressing a key steps, the display on to show the next side, <ESC> cancels the output to the screen.

4.4.21 PASSCNFG

ASCII - Command	**PASSCNFG**		
Syntax Transmit	PASSCNFG [Data]		
Syntax Receive	PASSCNFG <Data>	**Available in**	
Type	rw	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	36D8 (hex)
DIM		**PROFIBUS PNU**	1672 (dec) IND = 0001xxxx (bin)
Range	0,1	**DPR**	472 (dec)
Default	0		
Opmode		**Data Type Bus/DPR**	Integer8
Drive State		**Weightning**	
Start Firmware	6.02		
Configuration	No	**Revision**	2.0
Function Group	Basic Setup	**EEPROM**	
Short Description	Password Function		

Description

The command PASSCNFG sets the passwort-function.(see also PASS, PASSX)

- PASSCNFG=0 all parameter writing procedures are locked
- PASSCNFG=1 only parameter SAVE is locked

4.4.22 PBALMAX

ASCII - Command	**PBALMAX**			
Syntax Transmit	PBALMAX [Data]			
Syntax Receive	PBALMAX <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer32		**CANBus Object Number**	35C1 (hex)
DIM	W		**PROFIBUS PNU**	1793 (dec) IND = 0000xxxx (bin)
Range	see Manual		**DPR**	193 (dec)
Default	80 / 200			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.6
Function Group	Basic Setup		**EEPROM**	Yes
Short Description	Maximum Regen Power			

Description

This parameter can be used to limit the continuous power dissipated in the regen resistor.

If the actual value of the power in the regen resistor exceeds the preset maximum value, then the regen resistor is switched off. This may trigger the fault message □Overvoltage□ as a result. If the maximum value is too high, the regen resistor may be overloaded.

4.4.23 PBALRES

ASCII - Command	**PBALRES**			
Syntax Transmit	PBALRES [Data]			
Syntax Receive	PBALRES <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer8		**CANBus Object Number**	35C2 (hex)
DIM	-		**PROFIBUS PNU**	1794 (dec) IND = 0000xxxx (bin)
Range	0, 1		**DPR**	194 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	Yes		**Revision**	1.3
Function Group	Basic Setup		**EEPROM**	Yes
Short Description	Select Regen Resistor			

Description

This parameter can be used to select whether the internal (0) or an external (1) regen resistor should be used. It affects the PBALMAX [▶ 82] parameter.

4.4.24 PMODE

ASCII - Command	PMODE		
Syntax Transmit	PMODE [Data]		
Syntax Receive	PMODE <Data>	Available in	
Type	Variable rw	MMI	Yes
ASCII Format	Integer8	CANBus Object Number	35CD (hex)
DIM	-	PROFIBUS PNU	1805 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2	DPR	205 (dec)
Default	1		
Opmode	All	Data Type Bus/DPR	Integer32
Drive State	Disabled + Reset (Coldstart)	Weightning	
Start Firmware	1.20		
Configuration	Yes	Revision	1.3
Function Group	Basic Setup	EEPROM	Yes
Short Description	Line Phase Error Mode		

Description

The PMODE parameter configures the response of the amplifier to the failure of a mains supply phase.

- PMODE=0 no warning, no fault message, current limited to max. 4A
- PMODE=1 warning n05, current limited to max. 4A
- PMODE=2 fault message F19, output stage is disabled

If the current limiting is activated, it only applies to periods of acceleration. A braking operation can still be carried out at full current.

4.4.25 RSTFW

ASCII - Command	RSTFW		
Syntax Transmit	RSTFW [Data]		
Syntax Receive	RSTFW <Data>	Available in	
Type	rw	MMI	No
ASCII Format	Integer8	CANBus Object Number	36DA (hex)
DIM		PROFIBUS PNU	1674 (dec) IND = 0001xxxx (bin)
Range	0 .. 1	DPR	474 (dec)
Default	0		
Opmode	All	Data Type Bus/DPR	Integer8
Drive State		Weightning	
Start Firmware	5.70		
Configuration	No	Revision	2.0
Function Group	Basic Setup	EEPROM	-
Short Description			

Description

The command RSTFW defines the configuration of function RSTVAR [▶ 84] .

- RSTFW=0 all parameters are reset to the actual default-data.
- RSTFW=345 all parameters are reset to the actual default-data of firmware -version 3.45.

4.4.26 RSTVAR

ASCII - Command	**RSTVAR**			
Syntax Transmit	RSTVAR			
Syntax Receive	RSTVAR		**Available in**	
Type	Command		**MMI**	Yes
ASCII Format	-		**CANBus Object Number**	35E9 (hex)
DIM	-		**PROFIBUS PNU**	1833 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	233 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	Disabled		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Basic Setup		**EEPROM**	No
Short Description	Restore Variables (Default Values)			

Description

The RSTVAR command resets all parameters/variables to the internal works (default) settings. The parameters which are stored in the EEPROM are not immediately affected by this. The default settings only become permanent when the SAVE [▸ 51] command is used (save parameters in the EEPORM).

4.4.27 SERIALNO

ASCII - Command	**SERIALNO**			
Syntax Transmit	SERIALNO			
Syntax Receive	SERIALNO <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Integer32		**CANBus Object Number**	35EF (hex)
DIM	-		**PROFIBUS PNU**	1839 (dec) IND = 0000xxxx (bin)
Range	10 ASCII characters		**DPR**	239 (dec)
Default	Factory default			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Basic Setup		**EEPROM**	No
Short Description	Drive Serial Number			

Description

The serial number of the drive amplifier.

4.4.28 STAGECODE

ASCII - Command	**STAGECODE**		
Syntax Transmit	-		
Syntax Receive	STAGECODE <Data>	**Available in**	
Type	Variable r	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	3682 (hex)
DIM	-	**PROFIBUS PNU**	1986 (dec) IND = 0000xxxx (bin)
Range	1, 2, .., 19	**DPR**	386 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	4.62		
Configuration	No	**Revision**	1.5
Function Group	Basic Setup	**EEPROM**	-
Short Description	Power Stage Identification		

Description

Give the identification of the power stage-

- STAGECODE=0 not allowed (Hardware error)
- STAGECODE=1 SR601
- STAGECODE=2 SR603
- STAGECODE=3 SR606
- STAGECODE=4 SR610
- STAGECODE=5 SR614
- STAGECODE=6 SR620
- STAGECODE=7 SR640
- STAGECODE=8 SR670
- STAGECODE=9 SR610/30
- STAGECODE=10 Reserve
- STAGECODE=11 Reserve
- STAGECODE=12 Reserve
- STAGECODE=13 Reserve
- STAGECODE=14 Reserve
- STAGECODE=15 Reserve
- STAGECODE=16 SR403
- STAGECODE=17 SR406
- STAGECODE=18 Reserve
- STAGECODE=19 Reserve

4.4.29 STOPMODE

ASCII - Command	**STOPMODE**		
Syntax Transmit	STOPMODE [Data]		
Syntax Receive	STOPMODE <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	35FF (hex)
DIM	-	**PROFIBUS PNU**	1855 (dec) IND = 0000xxxx (bin)
Range	0, 1	**DPR**	255 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	Disabled + Reset (Coldstart)	**Weightning**	
Start Firmware	1.20		
Configuration	Yes	**Revision**	1.8
Function Group	Basic Setup	**EEPROM**	Yes
Short Description	Brake Response for Disable		

Description

STOPMODE defines the response of the drive to a disabling of the output stage.
The following settings are possible:

- STOPMODE=0 the output stage is immediately disabled, and the drive coasts down.
- STOPMODE=1 the drive is run down under velocity control to velocity 0 (DECDIS [▶ 330] ramp). When the velocity falls below the standstill threshold VEL0 [▶ 338], the output stage is disabled.
  The output stage will also be disabled if the VEL0 [▶ 338] velocity is not reached within 5 seconds (a 5-second time-out).

4.4.30 TBRAKE

ASCII - Command	**TBRAKE**			
Syntax Transmit	TBRAKE [Data]			
Syntax Receive	TBRAKE <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer16		**CANBus Object Number**	366E (hex)
DIM	ms		**PROFIBUS PNU**	1966 (dec) IND = 0000xxxx (bin)
Range	10 .. 10000		**DPR**	366 (dec)
Default	100			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	4.00			
Configuration	No		**Revision**	1.3
Function Group	Basic Setup		**EEPROM**	Yes
Short Description	Disable Delaytime with Holding Brake			

Description

TBRAKE defines a disable delay time with holding brake.

If the drive is disabled, it controls the holding brake if selected. After the motor is stopped, the holding brake is switched off and a delay timer (value is TBRAKE) is started. When the time is gone, the drive is disabled.

4.4.31 TBRAKE0

ASCII - Command	**TBRAKE0**			
Syntax Transmit	TBRAKE0 [Data]			
Syntax Receive	TBRAKE0 <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer16		**CANBus Object Number**	366F (hex)
DIM	ms		**PROFIBUS PNU**	1967 (dec) IND = 0000xxxx (bin)
Range	-10 .. 10000		**DPR**	367 (dec)
Default	20			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	3.46/4.00			
Configuration	No		**Revision**	1.3
Function Group	Basic Setup		**EEPROM**	Yes
Short Description	Enable Delaytime with Holding Brake			

Description

The parameter TBRAKE0 defines a reaction time of the holding brake when the drive is enabled.

If the drive is enabled (hardware/software enable) thr drive controls the holding brake. During the selected time TBRAKE0, the internal velocity setpoint is set to 0. After the time when the brake is open, the setpoint is accepted internally and the motor can run.

If the TRBAKE0 is set to values <0, the internal setpoint is activated before the holding brake is open (to make it compatible to older firmware versions <3.46)

4.4.32 TRUN

ASCII - Command	**TRUN**		
Syntax Transmit	TRUN		
Syntax Receive	TRUN <Data>	**Available in**	
Type	Variable ro	**MMI**	Yes
ASCII Format	String	**CANBus Object Number**	3614 (hex)
DIM	hhhhh:mm	**PROFIBUS PNU**	1876 (dec) IND = 0000xxxx (bin)
Range	00000:00 to 99999:45	**DPR**	276 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.9
Function Group	Basic Setup	**EEPROM**	Yes
Short Description	Run-time counter		

Description

The run-time counter shows the operating life of the amplifier (if the 24V is applied) in minutes.

The internal resolution of the run-time counter is 1 second.

Since the run-time counter value is included in the serial EEPROM of the amplifier, it is only updated in the EEPROM every 8 minutes So switching off the 24V supply can cause a loss in the record of up to 8 minutes.

4.4.33 UVLTMODE

ASCII - Command	**UVLTMODE**		
Syntax Transmit	UVLTMODE [Data]		
Syntax Receive	UVLTMODE <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	3617 (hex)
DIM	-	**PROFIBUS PNU**	1879 (dec) IND = 0000xxxx (bin)
Range	0, 1	**DPR**	279 (dec)
Default	1		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	Disabled + Reset (Coldstart)	**Weightning**	
Start Firmware	1.20		
Configuration	Yes	**Revision**	1.3
Function Group	Basic Setup	**EEPROM**	Yes
Short Description	Undervoltage Mode		

Description

The configuration variable UVLTMODE activates or inhibits the undervoltage monitoring of the amplifier.

If the monitoring is activated (UVLTMODE=1), then the fault message F05 (undervoltage) is generated as soon a s the DC-bus voltage falls below the undervoltage threshold VBUSMIN [▸ 54].

4.4.34 VBUSBAL

ASCII - Command	VBUSBAL		
Syntax Transmit	VBUSBAL [Data]		
Syntax Receive	VBUSBAL <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	361B (hex)
DIM	-	**PROFIBUS PNU**	1883 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2	**DPR**	283 (dec)
Default	2		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.6
Function Group	Basic Setup	**EEPROM**	Yes
Short Description	Maximum Line Voltage		

Description

This setting is for the maximum permissible voltage for the motor. For instance, if a motor that is rated for a 400V supply is connected to the amplifier, then the setting must be VBUSBAL = 1 (400V). This sets regen and overvoltage thresholds in the amplifier to acceptable values for the motor. This ensures that the motor windings are not damaged.

If several amplifiers are connected to the supply with the DC-buses in parallel, then they must all have the same value for VBUSBAL. It must therefore be set to suit the motor with the lowest voltage rating.

- VBUSBAL=0 (230 V) VBUSMAX [▸ 54]=450V
- VBUSBAL=1 (400 V) VBUSMAX=800V
- VBUSBAL=2 (480 V) VBUSMAX=900V

The setting for the drive 40xM is limited to 0 (230V). The default setting is also 0.

The setting for the drive 44xM is limited to 0 (400V). The default setting is also 1.

VBUSBAL also changes DICONT [▸ 37] and DIPEAK [▸ 38] at following drive types (starting with firmware 4.94):

VBUSBAL = 0	VBUSBAL = 1	VBUSBAL = 2
443 DICONT = 3 DIPEAK = 9	443 DICONT = 2 DIPEAK = 6	443 not possible
446 DICONT = 6 DIPEAK = 12	446 DICONT =4 DIPEAK = 8	446 not possible
614 DICONT = 20 DIPEAK = 40	614 DICONT = 14 DIPEAK = 28	614 DICONT = 14 DIPEAK = 28
670 DICONT = 85 DIPEAK = 160	670 DICONT = 80 DIPEAK = 160	670 DICONT = 70 DIPEAK = 140

4.4.35 VER

ASCII - Command	**VER**			
Syntax Transmit	VER [*]			
Syntax Receive	VER <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	String		**CANBus Object Number**	No
DIM	-		**PROFIBUS PNU**	No
Range	max 50 ASCII Characters		**DPR**	No
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Basic Setup		**EEPROM**	No
Short Description	Firmware Version			

Description

The command VER returns the version designation and the date of creation for the firmware.

The expanded form of the command (VER *) returns a version list for the various firmware and hardware components.

- Version of the basic firmware
- Hardware revision
- CPLD version
- FPGA version (this version designation labels the type of FPGA program that is loaded, and can vary according to the equipment configuration), see FPGA [▶ 77], GEARMODE [▶ 213].
- CAN: firmware version
- Version of the motor database (MDB)
- Profibus/Sercos firmware version

4.5 Communication

4.5.1 ACTRS232

ASCII - Command	**ACTRS232**			
Syntax Transmit	ACTRS232 [Data]			
Syntax Receive	ACTRS232 <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer8		**CANBus Object Number**	3655 (hex)
DIM	-		**PROFIBUS PNU**	1941 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2		**DPR**	341 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	-		**Weightning**	
Start Firmware	2.40			
Configuration	No		**Revision**	1.3
Function Group	Communication		**EEPROM**	No

Short Description	Activate RS232 Watchdog

Description

The ACTRS232 command activates or deactivates the monitoring of the serial interface (RS232-watchdog).

- ACTRS232=0 no monitoring of serial communication
- ACTRS232=1 the RS232-watchdog is activated. The watchdog timer can be set in msec through the RS232T [▶ 104] command. The watchdog must be triggered by every serial command. When the timer runs out, all movement is stopped and the warning n04 is displayed. The warning must be cancelled by the □Acknowledge fault□ function.
- ACTRS232=2 the RS232-watchdog is activated. The watchdog timer can be set in msec through the RS232T [▶ 104] command. The watchdog must be triggered by every serial command. When the timer runs out, the present movement is stopped and ACTRS232 is set to 0. No warning is given out.

After switching on the amplifier, the RS232-watchdog is always deactivated (ACTRS232=0). When a service function is initiated via the serial interface, the PC program (or external controls) should ensure that the monitoring of the serial interface is switched on. In this way, you can be sure that if communication is interrupted or the PC crashes, the service function will be automatically terminated.

4.5.2 ADDRFB

ASCII - Command	**ADDRFB**			
Syntax Transmit	ADDRFB [Data]			
Syntax Receive	ADDRFB <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer16		**CANBus Object Number**	369C (hex)
DIM	-		**PROFIBUS PNU**	2012 (dec) IND = 0000xxxx (bin)
Range	0 .. 63		**DPR**	412 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	4.91			
Configuration	No		**Revision**	1.6
Function Group	Communication		**EEPROM**	Yes
Short Description	Fieldbus address at AX2500 Slave			

Description

ADDRFB defines the fieldbus address of the AX2500 for CAN / PROFIBUS and SERCOS. After changing the parameter, a parameter save (SAVE [▶ 51]) has to be initiated and the drive has to be reset.

This address is used only for the external Fieldbus communication. The internal AX2500 communication still uses the address given by ADDR [▶ 70].

If ADDRFB = 0, the internal address ADDR [▶ 70] is used.

4.5.3 \

ASCII - Command	\		
Syntax Transmit	\ [Data]		
Syntax Receive	\ <Data>	**Available in**	
Type	Command	**MMI**	Yes
ASCII Format	Unsigned8	**CANBus Object Number**	362B (hex)
DIM	-	**PROFIBUS PNU**	1899 (dec) IND = 0000xxxx (bin)
Range	0(=Master) .. 63	**DPR**	299 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Unsigned8
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Communication	**EEPROM**	-
Short Description	Selection of Remote Address		

Description

For a CAN network with several amplifiers, there is an option for using a serial connection to one of the devices (master) to communicate with all the other amplifiers. To do this, the SCAN [▶ 104] command is initiated on the master device, which performs an automatic detection of all the drivess that are connected. The response to the SCAN [▶ 104] command contains a list of the addresses of all the drives devices that have been detected.

Typing the backslash character followed by a drive address (\ addr) in the range of 0 to 63 selects the addressed drive for communications. Further commands sent via the serial interface are ignored by the master device and passed on directly across the CAN bus to the activated AX2xxx device. The response that this device outputs to the CAN bus is diverted to the serial interface. The command, \ 0 deselects the slave device and re-activate the master.

The AX2500 communicates not via CAN, but via an internal serial link.

The setting MDRV [▶ 98] = 0 disables the multi-link functionality.

4.5.4 CMDDLY

ASCII - Command	**CMDDLY**		
Syntax Transmit	CMDDLY [Data]		
Syntax Receive	CMDDLY <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer16	**CANBus Object Number**	3670 (hex)
DIM	ms	**PROFIBUS PNU**	1968 (dec) IND = 0000xxxx (bin)
Range	0 .. 100	**DPR**	368 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	4.00		
Configuration	No	**Revision**	1.3
Function Group	Communication	**EEPROM**	Yes
Short Description	Command Delay Time for RS232		

Description

The parameter CMDDLY defines a minimum delay time for answers from the drive via RS232 (ASCII). This enables the possibility for slower controller to communicate with the Drive.

The delay time CMDDLY defines the time between the last character of an ASCII string send to the drive to the first character of the answer.

The time between the characters cannot be changed, they are defined by the baud rate and the internal calculation times.

This time defines only the minimum delay time between the ASCII strings. The time can be longer, depending on the internal calculation time.

4.5.5 DIFVAR

ASCII - Command	**DIFVAR**		
Syntax Transmit	DIFVAR		
Syntax Receive	DIFVAR <Data>	**Available in**	
Type	Multi-line Return Command	**MMI**	No
ASCII Format	-	**CANBus Object Number**	3528 (hex)
DIM	-	**PROFIBUS PNU**	1640 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	40 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.46		
Configuration	No	**Revision**	1.3
Function Group	Communication	**EEPROM**	-
Short Description	List Variables with Values		

Description

This command produces a list of parameters with settings that differ from the default values. The list contains entries in the following form:

PARAMETER Value (Default) PARAMETER = Parameter name

- Value = the actual parameter setting
- Default = the default value for the parameter

4.5.6 DISDPR

ASCII - Command	DISDPR		
Syntax Transmit	DISDPR [Data]		
Syntax Receive	DISDPR <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	3673 (hex)
DIM	-	**PROFIBUS PNU**	1971 (dec) IND = 0000xxxx (bin)
Range	0, 1	**DPR**	371 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	Disable	**Weightning**	
Start Firmware	3.51		
Configuration	No	**Revision**	1.5
Function Group	Communication	**EEPROM**	-
Short Description	Disable DPR access		

Description

DISDPR=1 disables the write access of e.g. Lightbus option boards to the drive. Read access is still possible. This enables the service functionality via PC even if the Bus is running.

- DISDPR=0 Full access from the controller side.
- DISDPR=1 Only read access.

This parameter is not stored in the EEPROM.

4.5.7 DRVCNFG

ASCII - Command	DRVCNFG		
Syntax Transmit	DRVCNFG [Data]		
Syntax Receive	DRVCNFG <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer32	**CANBus Object Number**	3672 (hex)
DIM	-	**PROFIBUS PNU**	1970 (dec) IND = 0000xxxx (bin)
Range	long int	**DPR**	370 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	4.03		
Configuration	No	**Revision**	2.0
Function Group	Communication	**EEPROM**	Yes
Short Description	Configuration Variable for CAN-Bus		

Description

The configuration variable makes sure, that new or enhanced features do not create problems with excisting machines. If a function has changed, this can be changed back by setting the corresponding bit.

Bit	Description
Bit 0 (0x1)	=1 The CAN-open switch-on telegram is 0 byte. =0 The CAN-open switch-on telegram is 8 byte.

Bit	Description
Bit 1 (0x2)	=1 The state machine is effected by enable/disable. The CAN-open state machine is alligned to the real status of the drive. =0 The state machine is not updated by enable/disable of the drive.
Bit 2 (0x4)	=1 The size of the SDO-object is checked. If a wrong size is detected, an Emergency object is generated. =0 No check of the size of an SDO object
Bit 3 (0x8)	=1 Fieldbus mapping data is stored in the EEPROM =0 Data is not stored
Bit 4 (0x10)	Reserved
Bit 5 (0x20)	If the bit is set, the signal "Homing set" (see TRJSTAT [▸ 185]) is delayed by INPT [▸ 259]. This gives the possibility, equivalent to the signal "In Position", to get a low/high transition. This is important for homing mode "set homing position" and "homing in one revolution". If the homing move is startet again, the transition can be lost without this function. (4.96)
Bit 6 (0x40)	=1 the max input voltage at the sine/cosine input at X1 (Drive 400 X2) is supervised =0 the the max. input voltage at the sine/cosine input at X1 (Drive 400 X2) is not supervised
Bit 7 (0x80)	The overflow detection of a multi turn encoder is switches off. (4.94)
Bit 8 (0x100)	A single turn encoder is executed as a multi turn encoder. The actual position is set to the single turn absolute position at start-up. Homing is not required. (4.94)

4.5.8 DUMP

ASCII - Command	**DUMP**		
Syntax Transmit	DUMP		
Syntax Receive	DUMP <Data>	**Available in**	
Type	Multi-line Return Command	**MMI**	Yes
ASCII Format	-	**CANBus Object Number**	352F (hex)
DIM	-	**PROFIBUS PNU**	1647 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	47 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Communication	**EEPROM**	-
Short Description	List All EEPROM Variables with Values		

Description

This command produces a list of all the parameters that can be stored in the EEPROM, together with their present values. All the amplifier-specific parameters (e.g. A/D-offset values) start with a □;□(semicolon).

4.5.9 DUMPDIF

ASCII - Command	**DUMPDIF**			
Syntax Transmit	DUMPDIF			
Syntax Receive	DUMPDIF <Data>		**Available in**	
Type	Multi-line Return Command		**MMI**	No
ASCII Format	-		**CANBus Object Number**	No
DIM	-		**PROFIBUS PNU**	No
Range	-		**DPR**	No
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	2.49			
Configuration	No		**Revision**	1.9
Function Group	Communication		**EEPROM**	-
Short Description	List of Parameter unequal default setting			

Description

The command DUMPDIF displays a list of parameters, which have not the default value, stored in the firmware. In contrast to the command DIFVAR [▸ 92], DUMPDIF gives the list in the right order and format. The output of this command gives the possibility to set the servo drive in combination with the default settings. The order of the commands is in that way, that interdependencies are considered.

If RSTVAR [▸ 84] and then the stored DUMPDIF - list is send to the drive, the full setting of the drive is complete.

Also see about this

- DUMP [▸ 94]

4.5.10 EXTWD

ASCII - Command	**EXTWD**			
Syntax Transmit	EXTWD [Data]			
Syntax Receive	EXTWD <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer32		**CANBus Object Number**	353A (hex)
DIM	Milliseconds		**PROFIBUS PNU**	1658 (dec) IND = 0000xxxx (bin)
Range	1 .. 32000		**DPR**	58 (dec)
Default	100			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Communication		**EEPROM**	Yes
Short Description	external watch dog (Fieldbus)			

Description

The EXTWD parameter can be used to define the monitoring time (watchdog timer) for the fieldbus/slot communication. The monitoring is only active if the EXTWD parameter has a value greater than zero (EXTWD=0 means monitoring is switched off) and the output stage is enabled. If the preset time runs out, without the timer being retriggered, then the warning n04 (threshold monitoring) is generated and the drive is stopped. The amplifier remains ready for operation, and the output stage is still enabled. This warning must be cancelled (function CLRFAULT [▸ 35] or INxMODE [▸ 116]=14) before a new motion command (setpoint) can be accepted.

4.5.11 FLASH

ASCII - Command	**FLASH**		
Syntax Transmit	FLASH [Data]		
Syntax Receive	FLASH <Data>	**Available in**	
Type	wo	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	36D9 (hex)
DIM		**PROFIBUS PNU**	1673 (dec) IND = 0001xxxx (bin)
Range	0 ... 4	**DPR**	473 (dec)
Default	-		
Opmode	-	**Data Type Bus/DPR**	Integer8
Drive State	Disable	**Weightning**	
Start Firmware	5.70		
Configuration	No	**Revision**	2.0
Function Group	Communication	**EEPROM**	
Short Description			

Description

The command FLASH is used to transfer firmware/parameter-data between drive and external flash-card.

The external flash-card is used as an external memory for Sr600-firmware and also als for drive-parameter. This tool can be used to update the firmware without an additional hardware (computer, ...). THe upload/ download-commandos can be set by the ASCII-command FLASH or by using the keys S1/S2 at the flash -card.

The FLASH-command options:

- FLASH=0 Output of information text with version number of the save firware on the flash card. The second line includes the actual drive firmware version.
- FLASH=1 Saves all drive-data into the flash-card (UPLOAD). The actual firmware-version, drive-parameter, data bank, PLC-programm, cam profiles and flash-motion tasks are saved into the flash-card. The command is activated also,if the key S1 is pressed. The drive display-segment shows an □u□during the upload process. (UPLOAD). The procedure ends with an automatic reset of the drive.
- FLASH=2 Saves all drive-data into the flash-card (DOWNLOAD).The actual firmware-version, drive-parameter, data bank, PLC-programm, cam profiles and flash-motion tasks are saved into the flash-card. The command is activated also,if the key S2 is pressed. The drive display-segment shows an □u□during the upload process. (DOWNLOAD). The procedure ends with an automatic reset of the drive.
- FLASH=3 Erase the exteral flash-card.
- FLASH=4 Load parameter-data into the drive.Only data from the serial EEPROM (parameter-memory) are transmitted.

NOTICE

Data loss

The FLASH-command must be used only by disabled output stage and switched of main voltage. During this process the 24V DC Link has not to be switched off. If you ignore this, you may erase data.

Drives without a firmware will start the monitor program automatic. The flash-card can be used with monitor-program >= version 7.2 implements.

4.5.12 HELP

ASCII - Command	**HELP**		
Syntax Transmit	HELP		
Syntax Receive	HELP <Data>	**Available in**	
Type	Variable ro	**MMI**	Yes
ASCII Format	String	**CANBus Object Number**	No
DIM	-	**PROFIBUS PNU**	No
Range	-	**DPR**	No
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.46		
Configuration	No	**Revision**	1.3
Function Group	Communication	**EEPROM**	-
Short Description	Output Parameter Help Information		

Description

Using the parameter HELP <name> produces a display of help information for the ASCII parameter □name□. This help information includes input limits and both the actual and default values for the parameter.

e.g. HELP GV
GV act=6 min=0 max=1000 default=1

4.5.13 LIST

ASCII - Command	**LIST**		
Syntax Transmit	LIST		
Syntax Receive	LIST <Data>	**Available in**	
Type	Multi-line Return Command	**MMI**	No
ASCII Format	String	**CANBus Object Number**	3582 (hex)
DIM	-	**PROFIBUS PNU**	1730 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	130 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Communication	**EEPROM**	No
Short Description	List All ASCII Commands		

Description

All those commands are listed which can be used for to the present motor/amplifier configuration. ASCII commands that require specific hardware (e.g. Hiperface/Endat, Profibus,Sercos) will only be displayed if the corresponding hardware has been recognized correctly.

4.5.14 MAXSDO

ASCII - Command	**MAXSDO**			
Syntax Transmit	MAXSDO			
Syntax Receive	MAXSDO <Data>		**Available in**	
Type	Variable ro		**MMI**	No
ASCII Format	Integer32		**CANBus Object Number**	3500 (hex)
DIM	-		**PROFIBUS PNU**	1600 (dec) IND = 0000xxxx (bin)
Range	int		**DPR**	0 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	2.46			
Configuration	No		**Revision**	1.3
Function Group	Communication		**EEPROM**	-
Short Description	Number of Objects of the Parameter Channel			

Description

The command MAXSDO gives the number of objects of the parameter channel.

4.5.15 MDRV

ASCII - Command	**MDRV**			
Syntax Transmit	MDRV [Data]			
Syntax Receive	MDRV <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer8		**CANBus Object Number**	3639 (hex)
DIM	-		**PROFIBUS PNU**	1913 (dec) IND = 0000xxxx (bin)
Range	0, 1		**DPR**	313 (dec)
Default	1			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	-		**Weightning**	
Start Firmware	2.49			
Configuration	No		**Revision**	1.3
Function Group	Communication		**EEPROM**	Yes
Short Description	Selection of Multidrive Functionality			

Description

The command MDRV enables the multi drive functionality.

- MDRV=0 Multi drive functionality (address range ADDR [▸ 70] 0...127)
  The SCAN [▸ 104] command gives every time a "0" (no external drives recognized)
- MDRV=1 Multi drive active (address range ADDR [▸ 70] 0...63)
  The SCAN [▸ 104] command checks the CAN-bus if there are more drives.

4.5.16 MSG

ASCII - Command	MSG		
Syntax Transmit	MSG [Data]		
Syntax Receive	MSG <Data>	Available in	
Type	Variable rw	MMI	Yes
ASCII Format	Integer8	CANBus Object Number	35A2 (hex)
DIM	-	PROFIBUS PNU	1762 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2	DPR	162 (dec)
Default	0		
Opmode	All	Data Type Bus/DPR	Integer8
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.3
Function Group	Communication	EEPROM	Yes
Short Description	Enable / Disable All Messages via RS232		

Description

If □MSG 2□ is set, then the execution of the individual initialization steps will be signaled through the serial interface when the amplifier is switched on (initialization phase). This setting should only be used for test purposes (e.g., during commissioning). Since the PC operating program basically only works with the setting □MSG 1□, the □MSG 2□ setting can only be implemented with the help of a terminal program (not in the terminal window of the operating program).

4.5.17 OBJCO

ASCII - Command	OBJCO		
Syntax Transmit	OBJCO		
Syntax Receive	OBJCO <Data>	Available in	
Type	Variable ro	MMI	No
ASCII Format	-	CANBus Object Number	No
DIM	-	PROFIBUS PNU	No
Range	-	DPR	No
Default	-		
Opmode	All	Data Type Bus/DPR	-
Drive State	-	Weightning	
Start Firmware	3.20		
Configuration	No	Revision	1.3
Function Group	Communication	EEPROM	No
Short Description	Mirror CAN - Objects for debug		

Description

The ASCII - object mirrors the CAN - objects. The objects need up to two parameters. The first parameter represents the Index and the second the Subindex. The Subindex is optional. The subindex is internally set to zero if no Subindex is given.

4.5.18 PBAUD

ASCII - Command	PBAUD		
Syntax Transmit	PBAUD		
Syntax Receive	PBAUD <Data>	Available in	
Type	Variable ro	MMI	Yes
ASCII Format	Float	CANBus Object Number	35C3 (hex)
DIM	kBaud	PROFIBUS PNU	1795 (dec) IND = 0000xxxx (bin)
Range	1.0 .. 12000.0	DPR	195 (dec)
Default	-		
Opmode	-	Data Type Bus/DPR	Integer32
Drive State	-	Weightning	1000
Start Firmware	1.73		
Configuration	No	Revision	1.3
Function Group	Communication	EEPROM	No
Short Description	Profibus Baud Rate		

Description

The PBAUD command reads out the present PROFIBUS baud rate. The baud rate is provided by the master (control system). The Drive detects the baud rate automatically.

The following settings are possible (in kbaud/kbps):

- 12000
- 6000
- 3000
- 1500
- 500
- 187.5
- 93.75
- 45.45
- 19.2
- 9.6

4.5.19 PIOBUF

ASCII - Command	PIOBUF		
Syntax Transmit	PIOBUF		
Syntax Receive	PIOBUF <Data>	Available in	
Type	Variable rw	MMI	Yes
ASCII Format	String	CANBus Object Number	35CC (hex)
DIM	-	PROFIBUS PNU	1804 (dec) IND = 0000xxxx (bin)
Range	-	DPR	204 (dec)
Default	-		
Opmode	-	Data Type Bus/DPR	-
Drive State	-	Weightning	
Start Firmware	1.73		
Configuration	No	Revision	1.3
Function Group	Communication	EEPROM	No
Short Description	Profibus data		

Description

This command reads out the present PROFIBUS input and output buffers. The output buffer handles the data flow from the control system to the Drive, and the input buffer handles the data flow from the Drive to the control system. Each buffer is 20 bytes long (telegram length) and is put together from the PKW section (8 bytes. i.e., 4 words) and the PZD section (12 bytes, 6 words). The individual bytes are in hexadecimal format.

PIOBUF provides 20 bytes of output buffer in the first line, and 20 bytes of input buffer in the second line.

If the communication over the PROFIBUS is interrupted or faulty, then a fault message □ERR [PIOBUF] NO DATA EXCHANGE SPC3 - INTERRUPT□ is generated.

4.5.20 PNOID

ASCII - Command	**PNOID**		
Syntax Transmit	PNOID		
Syntax Receive	PNOID <Data>	**Available in**	
Type	Variable ro	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	35CE (hex)
DIM	-	**PROFIBUS PNU**	1806 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	206 (dec)
Default	-		
Opmode	-	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	1.73		
Configuration	No	**Revision**	1.3
Function Group	Communication	**EEPROM**	No
Short Description	PROFIBUS ID		

Description

The PNOID command reads out the identification number of the Drive. This number is used for the unique identification of the Drive as a participant in the PROFIBUS network. The ID is allocated and managed by the PROFIBUS User Organization.

This instrument ID is also part of the GSD (base data for the instrument).

PNOID returns the identification number 045D (hexadecimal).

4.5.21 PPOTYP

ASCII - Command	**PPOTYP**		
Syntax Transmit	PPOTYP		
Syntax Receive	PPOTYP <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	35D0 (hex)
DIM	-	**PROFIBUS PNU**	1808 (dec) IND = 0000xxxx (bin)
Range	2	**DPR**	208 (dec)
Default	2		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	1.73		
Configuration	No	**Revision**	1.8
Function Group	Communication	**EEPROM**	Yes

Short Description	Profibus PPO Type

Description

PPOTYP reads the PROFIBUS DP telegram type. The amplifier supports telegram type 2 (telegram consists of 10 words (20 bytes) and is divided into a PKW section (4 words) and a PZD section (6 words)). PPOTYP returns the value, 2.

4.5.22 PROMPT

ASCII - Command	**PROMPT**		
Syntax Transmit	PROMPT [Data]		
Syntax Receive	PROMPT <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	35D3 (hex)
DIM	-	**PROFIBUS PNU**	1811 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2, 3	**DPR**	211 (dec)
Default	1		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Communication	**EEPROM**	No
Short Description	Select RS232 Protocol		

Description

The PROMPT parameter can be used to set the protocol for the RS232 transmission.
The following settings are possible:

PROMPT=0	No Echo	The data that are received through the RS232 interface are not automatically echoed (transmitted). There is no output of the prompt (-->) symbol.
PROMPT=1	"-->" plus Echo	The data that are received through the RS232 interface are automatically echoed (transmitted). The prompt (-->) symbol is given for inputting data.
PROMPT=2	Terminal Mode	This setting is the same as PROMPT=1 except: 1. If a CR(Enter) command is typed in at the beginning of the line, the last command is repeated. 2. Some commands (like DUMP) output more than one monitor screen. In this cases, the output is automatically stopped after one page.
PROMPT=3	"-->" plus Echo plus Checksum	This setting is the same as PROMPT=1 except. Additional to that, a Checksum is transmitted and checked in both directions to prevent wrong data. All character of a command are summed (Modulo 256 without CR). e.g. Command string : "ADDR [▸ 70] 1<CR>" generate Checksum: "A" = 0x41 "D" = 0x44 "D" = 0x44 "R" = 0x52 " " = 0x20 "1" = 0x31

		The sum is: 0x16C Modulo 256: 0x6C = 108 (Dec) First Character: 108/16 + 0x30 =0x36 = "6" Sec. Character: 108%16 + 0x30 = 0x3C = "<" The command string is: "ADDR 16<" <CR> When the command string is received, that same calculation is done and the last two characters in front of the <CR> are compared with the received data. If the checksum is ok, the ACK (0x06) is send, if no NACK (0x15) is send.

4.5.23 PSTATE

ASCII - Command	**PSTATE**			
Syntax Transmit	PSTATE			
Syntax Receive	PSTATE <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	String		**CANBus Object Number**	35D4 (hex)
DIM	-		**PROFIBUS PNU**	1812 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	212 (dec)
Default	-			
Opmode	-		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.73			
Configuration	No		**Revision**	1.3
Function Group	Communication		**EEPROM**	No
Short Description	Profibus Status			

Description

The command PSTATE reads out the present status of the PROFIBUS communication. The first value that is shown provides the SPC3 WATCHDOG status, the second value provides the PROFIBUS DP-status.

- SPC3 WATCHDOG status
  - 0 = baud rate search
  - 1 = check baud rate
  - 2 = DP mode, i.e. the bus watchdog is active
- PROFIBUS-DP status
  - 0 = wait for parameterization, performed by the master
  - 1 = wait for configuration, performed by the master
  - 2 = data exchange
  - 3 = fault □ the cause could, for instance, have been a faulty parameterization telegram in the data transfer phase.

Productive data can only be received, i.e. data exchanged for the PKW and PZD sections of the Drive, when the SPC3 WATCHDOG status has the value 2, and the PROFIBUS-DP status has the value 2.

4.5.24 RS232T

ASCII - Command	**RS232T**			
Syntax Transmit	RS232T [Data]			
Syntax Receive	RS232T <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer16		**CANBus Object Number**	35E8 (hex)
DIM	Milliseconds		**PROFIBUS PNU**	1832 (dec) IND = 0000xxxx (bin)
Range	1 .. 5000		**DPR**	232 (dec)
Default	2500			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	2.40			
Configuration	No		**Revision**	1.3
Function Group	Communication		**EEPROM**	Yes
Short Description	RS232 Watch Dog			

Description

If the monitoring of the serial interface is activated (RS232 watchdog), then the RS232T command can be used to set the time for the watchdog timer.
See also ACTRS232 [▸ 89]

4.5.25 SCAN

ASCII - Command	**SCAN**			
Syntax Transmit	SCAN			
Syntax Receive	SCAN <Data>		**Available in**	
Type	Command		**MMI**	Yes
ASCII Format	[Integer8...Integer8]		**CANBus Object Number**	35ED (hex)
DIM	-		**PROFIBUS PNU**	1837 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	237 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.8
Function Group	Communication		**EEPROM**	-
Short Description	Detect CAN Stations			

Description

For a CAN network with several drives connected, there is an option for using a serial connection to one of the devices (master) to communicate with all the other amplifiers. To do this, the SCAN command is initiated on the master device, which performs an automatic detection of all the drivess that are connected. The response to the SCAN command contains the total number and a list of the addresses of all the drive devices that have been detected.

The time taken to carry out this command is strongly dependent on the baud rate (CBAUD [▸ 73]) that is has been set for CAN, and is in the range from 1 second (at 1 Mbaud/1 Mbps) to 37 seconds (at 10Kbaud/10 kbps).

With drive 400, the communication is not done via CAN, but via an internal serial link. The behavior is the same.

4.5.26 SCANX

ASCII - Command	SCANX			
Syntax Transmit	SCANX			
Syntax Receive			Available in	
Type	Command		MMI	Yes
ASCII Format	-		CANBus Object Number	3696 (hex)
DIM	-		PROFIBUS PNU	2006 (dec) IND = 0000xxxx (bin)
Range	-		DPR	406 (dec)
Default	-			
Opmode	-		Data Type Bus/DPR	-
Drive State	-		Weightning	
Start Firmware	4.74			
Configuration	No		Revision	1.5
Function Group	Communication		EEPROM	-
Short Description	Restart internal communication of AX2500			

Description

SCANX restarts the communication inside the AX2500 system and also starts the automatic address selection of the AX2500 Slaves. This command is accepted only from the AX2500 Master module (not from AX2000 nor from AX2500 Slave modules).

At the startup of the master, this command is automatically started to establish the address selection. If a slave is reset, this command can be used to restart the communication.

4.5.27 VMUL

ASCII - Command	VMUL			
Syntax Transmit	VMUL [Data]			
Syntax Receive	VMUL <Data>		Available in	
Type	Variable rw		MMI	Yes
ASCII Format	Integer32		CANBus Object Number	3626 (hex)
DIM	-		PROFIBUS PNU	1894 (dec) IND = 0000xxxx (bin)
Range	0 .. 65535		DPR	294 (dec)
Default	1			
Opmode	All		Data Type Bus/DPR	Integer32
Drive State	-		Weightning	
Start Firmware	1.73			
Configuration	No		Revision	1.3
Function Group	Communication		EEPROM	Yes
Short Description	Velocity Scale Factor			

Description

The VMUL parameter is used to scale the speed (position control loop) for jog mode and motion tasks. This scaling is required by many fieldbus systems (PROFIBUS,CANBUS), since some fieldbus protocols only permit speed values in 16-bit format. This scaling factor can then be used to expand the 16-bit speed value from the fieldbus to give the internal 32-bit value. See also: manual for PROFIBUS, CANBUS

4.6 Current Controller

4.6.1 CDUMP

ASCII - Command	CDUMP		
Syntax Transmit	CDUMP		
Syntax Receive	CDUMP	Available in	
Type	Multi-line Return Command	MMI	Yes
ASCII Format	-	CANBus Object Number	3517 (hex)
DIM	-	PROFIBUS PNU	1623 (dec) IND = 0000xxxx (bin)
Range	-	DPR	23 (dec)
Default	-		
Opmode	All	Data Type Bus/DPR	-
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.3
Function Group	Current Controller	EEPROM	-
Short Description	Current Loop Parameter Dump		

Description

Outputs the current control loop parameters as a listing in several lines.

4.6.2 CTUNE

ASCII - Command	CTUNE		
Syntax Transmit	CTUNE [Data]		
Syntax Receive	CTUNE	Available in	
Type	Command	MMI	No
ASCII Format	-	CANBus Object Number	351E (hex)
DIM	Hz	PROFIBUS PNU	1630 (dec) IND = 0000xxxx (bin)
Range	400 .. 3000	DPR	30 (dec)
Default	1200		
Opmode	All	Data Type Bus/DPR	-
Drive State	Enabled	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.8
Function Group	Current Controller	EEPROM	-
Short Description	Calculate current parameters		

Description

This command calculates current parameters. Set the OPMODE [▶ 50] = 2 before starting.

4.6.3 I2TLIM

ASCII - Command	**I2TLIM**		
Syntax Transmit	I2TLIM [Data]		
Syntax Receive	I2TLIM <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	355A (hex)
DIM	%	**PROFIBUS PNU**	1690 (dec) IND = 0000xxxx (bin)
Range	0 .. 100	**DPR**	90 (dec)
Default	80		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Current Controller	**EEPROM**	Yes
Short Description	I2T Warning		

Description

This variable defines a threshold for the I2T warning. As soon as the I2T [▶ 21] values goes above this threshold, the warning, n01, is generated. This warning is passed on to a control system via a digital output (OxMODE [▶ 150]=11). If the I2TLIM value is too low, the message appears too soon and the drive is not fully utilized. If the I2TLIM value is too high, limiting occurs at the same time as the message.

4.6.4 ICMD

ASCII - Command	**ICMD**		
Syntax Transmit	ICMD		
Syntax Receive	ICMD <Data>	**Available in**	
Type	Variable ro	**MMI**	Yes
ASCII Format	Float	**CANBus Object Number**	355B (hex)
DIM	Amperes	**PROFIBUS PNU**	1691 (dec) IND = 0000xxxx (bin)
Range	-DIPEAK .. DPEAK	**DPR**	91 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Current Controller	**EEPROM**	No
Short Description	Current Setpoint		

Description

Shows the internal current setpoint.

4.6.5 ICMDVLIM

ASCII - Command	**ICMDVLIM**			
Syntax Transmit	ICMDVLIM [Data]			
Syntax Receive	ICMDVLIM <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer32		**CANBus Object Number**	3685 (hex)
DIM	VUNIT		**PROFIBUS PNU**	1989 (dec) IND = 0000xxxx (bin)
Range	0 .. VLIM		**DPR**	389 (dec)
Default	0			
Opmode	2,3		**Data Type Bus/DPR**	Integer32
Drive State	Disabled + Reset (Coldstart)		**Weightning**	
Start Firmware	4.71			
Configuration	Yes		**Revision**	1.5
Function Group	Current Controller		**EEPROM**	Yes
Short Description	Velocity Limit in Current Control			

Description

ICMDVLIM defines the velocity limit (given in VUNIT [▸ 342]) at current control in OPMODE [▸ 50] = 2 and 3.

This function is enabled, if ICMDVLIM is >0. It is a configuration parameter, but if ICMDVLIM was >0 while startup of the drive, this function can be disabled online by setting ICMDVLIM afterwards.

4.6.6 ICONT

ASCII - Command	**ICONT**			
Syntax Transmit	ICONT [Data]			
Syntax Receive	ICONT <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Float		**CANBus Object Number**	355C (hex)
DIM	Amperes		**PROFIBUS PNU**	1692 (dec) IND = 0000xxxx (bin)
Range	10% of DICONT, max(DICONT,IPEAK)		**DPR**	92 (dec)
Default	Minimum of DICONT and MICONT			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	1.20			
Configuration	No		**Revision**	1.8
Function Group	Current Controller		**EEPROM**	Yes
Short Description	Rated Current			

Description

This variable sets the required rated output continuous current. The adjustment is usually made to the standstill current for the connected motor. The value entered is limited to the lower of the rated current of the motor (MICONT [▸ 231]) or the rated current of the amplifier (DICONT [▸ 37]). This variable is used in the monitoring of the actual RMS current that is drawn. If the ICONT value is too low, the drive shows following errors, and the torque is too low. If the ICONT value is too high, the motor can be thermally overloaded.

4.6.7 IDUMP

ASCII - Command	**IDUMP**			
Syntax Transmit	IDUMP			
Syntax Receive	IDUMP <Data>		**Available in**	
Type	Multi-line Return Command		**MMI**	Yes
ASCII Format	String		**CANBus Object Number**	355E (hex)
DIM	-		**PROFIBUS PNU**	1694 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	94 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.8
Function Group	Current Controller		**EEPROM**	-
Short Description	Output Current Limit List			

Description

This command returns a list of the current limit variables and their settings (see CDUMP [▶ 106]).

4.6.8 IMAX

ASCII - Command	**IMAX**			
Syntax Transmit	IMAX			
Syntax Receive	IMAX <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Float		**CANBus Object Number**	355F (hex)
DIM	Amperes		**PROFIBUS PNU**	1695 (dec) IND = 0000xxxx (bin)
Range	0.3 .. 40.0		**DPR**	95 (dec)
Default	Minimum of DIPEAK and MIPEAK			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Current Controller		**EEPROM**	No
Short Description	Current Limit for Drive/Motor Configuration			

Description

The IMAX command returns the larger value of the two parameters MIPEAK [▶ 232] and DIPEAK [▶ 38]. IMAX = max (MIPEAK [▶ 232], DIPEAK [▶ 38])

4.6.9 IPEAK

ASCII - Command	**IPEAK**			
Syntax Transmit	IPEAK [Data]			
Syntax Receive	IPEAK <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Float		**CANBus Object Number**	356E (hex)
DIM	Amperes		**PROFIBUS PNU**	1710 (dec) IND = 0000xxxx (bin)
Range	0.0 ... DIPEAK		**DPR**	110 (dec)
Default	IMAX			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	1.20			
Configuration	No		**Revision**	1.8
Function Group	Current Controller		**EEPROM**	Yes
Short Description	Application Peak Current			

Description

IPEAK sets the peak rated current of the application (RMS value). The value to be entered is limited to the lower of the peak rated current of the motor (MIPEAK [▶ 232]) or amplifier (DIPEAK [▶ 38]). If the IPEAK value is too low, the drive shows following errors, and the peak torque is too low. If the IPEAK value is too high, the motor is endangered.

4.6.10 IPEAKN

ASCII - Command	**IPEAKN**			
Syntax Transmit	IPEAKN [Data]			
Syntax Receive	IPEAKN <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Float		**CANBus Object Number**	356F (hex)
DIM	A		**PROFIBUS PNU**	1711 (dec) IND = 0000xxxx (bin)
Range	0.0 ... DIPEAK		**DPR**	111 (dec)
Default	IMAX			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	1.77			
Configuration	No		**Revision**	1.3
Function Group	Current Controller		**EEPROM**	Yes
Short Description	Negative Peak current Limit			

Description

Sets the intended pulse current (r.m.s. value) for the negative range.

4.6.11 KC

ASCII - Command	**KC**			
Syntax Transmit	KC [Data]			
Syntax Receive	KC <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Float		**CANBus Object Number**	3574 (hex)
DIM	-		**PROFIBUS PNU**	1716 (dec) IND = 0000xxxx (bin)
Range	0.0 .. 1.0		**DPR**	116 (dec)
Default	1.0			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Current Controller		**EEPROM**	Yes
Short Description	I-Controller Prediction Constant			

Description

KC is a tuning variable of the current loop. For compensation of time delay a predicted current value can be used in addition to the measured motor current. KC 1 switches the current prediction on, KC 0.5 sets it to 50% and KC 0 switches it off. Disabling the current prediction can cause an unstable current loop.

4.6.12 KTN

ASCII - Command	**KTN**			
Syntax Transmit	KTN [Data]			
Syntax Receive	KTN <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Float		**CANBus Object Number**	362F (hex)
DIM	Milliseconds		**PROFIBUS PNU**	1903 (dec) IND = 0000xxxx (bin)
Range	0.2 .. 2.0		**DPR**	303 (dec)
Default	0.6			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	1.20			
Configuration	No		**Revision**	1.8
Function Group	Current Controller		**EEPROM**	Yes
Short Description	Current Controller Integral-Action Time			

Description

The integral-action time (integration time constant) of the current control loop.

4.6.13 MLGC

ASCII - Command	MLGC			
Syntax Transmit	MLGC [Data]			
Syntax Receive	MLGC <Data>		Available in	
Type	Variable rw		MMI	Yes
ASCII Format	Float		CANBus Object Number	3595 (hex)
DIM	ratet to MLGQ		PROFIBUS PNU	1749 (dec) IND = 0000xxxx (bin)
Range	0.2 .. 1.0		DPR	149 (dec)
Default	0.7			
Opmode	All		Data Type Bus/DPR	Integer32
Drive State	-		Weightning	1000
Start Firmware	1.20			
Configuration	No		Revision	1.3
Function Group	Current Controller		EEPROM	Yes
Short Description	Current Control loop Adaptive Gain (Q-component at rated current)			

Description

The current control loop includes an adaptive alteration of the gain that depends on the current. The MLGC parameter defines the relative gain referred to MLGQ [▸ 113] for continuous current.

MLGC = 0.8 means that the gain of the current control loop for continuous current is 80% of MLGQ [▸ 113]. A linear interpolation is made for the gain from current = 0 up to current = ICONT [▸ 108]

4.6.14 MLGD

ASCII - Command	MLGD			
Syntax Transmit	MLGD [Data]			
Syntax Receive	MLGD <Data>		Available in	
Type	Variable rw		MMI	Yes
ASCII Format	Float		CANBus Object Number	3596 (hex)
DIM	ratet to MLGQ		PROFIBUS PNU	1750 (dec) IND = 0000xxxx (bin)
Range	0.4 .. 1.0		DPR	150 (dec)
Default	0.7			
Opmode	All		Data Type Bus/DPR	Integer32
Drive State	-		Weightning	1000
Start Firmware	1.20			
Configuration	No		Revision	1.6
Function Group	Current Controller		EEPROM	Yes
Short Description	Adaptive Gain for Current Control loop, D-component			

Description

The D-component of the current control loop (field component). The MLGD parameter defines the relative gain referred to MLGQ [▸ 113].

MLGC = 0.6 means that the gain of the current control loop D-component is 60% of MLGQ [▸ 113]

4.6.15 MLGP

ASCII - Command	**MLGP**		
Syntax Transmit	MLGP [Data]		
Syntax Receive	MLGP <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Float	**CANBus Object Number**	3597 (hex)
DIM	ratet to MLGQ	**PROFIBUS PNU**	1751 (dec) IND = 0000xxxx (bin)
Range	0.1 .. 1.0	**DPR**	151 (dec)
Default	0.4		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Current Controller	**EEPROM**	Yes
Short Description	Current Control loop Adaptive Gain (Q-component at peak current)		

Description

The current control loop includes an adaptive alteration of the gain that depends on the current. The MLGP parameter defines the relative gain referred to MLGQ [▸ 113] for peak current.

MLGP = 0.6 means that the gain of the current control loop for peak current is 60% of MLGQ [▸ 113]. A linear interpolation is made for the gain from current = ICONT [▸ 108] up to current = IPEAK [▸ 110].

4.6.16 MLGQ

ASCII - Command	**MLGQ**		
Syntax Transmit	MLGQ [Data]		
Syntax Receive	MLGQ <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Float	**CANBus Object Number**	3598 (hex)
DIM	-	**PROFIBUS PNU**	1752 (dec) IND = 0000xxxx (bin)
Range	0.01 .. 15.0	**DPR**	152 (dec)
Default	1		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	1.20		
Configuration	No	**Revision**	1.6
Function Group	Current Controller	**EEPROM**	Yes
Short Description	Absolute Gain of Current Control loop		

Description

MLGQ gives the absolute gain of the current control loop. This also affects MLGC [▸ 112], MLGP [▸ 113] and MLGD [▸ 112].

4.6.17 POP

ASCII - Command	POP	For Manufacturer Use only	
Syntax Transmit	POP time1(msec) time2(msec)		
Syntax Receive	POP <Data>	Available in	
Type	Multi Line Command	MMI	No
ASCII Format	-	CANBus Object Number	No
DIM	-	PROFIBUS PNU	No
Range	-	DPR	No
Default	-		
Opmode	All	Data Type Bus/DPR	-
Drive State	Enabled	Weightning	
Start Firmware	4.32		
Configuration	No	Revision	1.9
Function Group	Current Controller	EEPROM	-
Short Description	Generate Current Step		

Description

The POP command is used along with POPI [▶ 114], POPI2 [▶ 115] and POPV [▶ 115] for giving the drive a current step. The command is only available when MSG [▶ 99] = 2. POP is used by the Current Loop Tuning Wizard and other test programs; it is not normally used in applications.

POP time1(msec) time2(msec)

When a POP command is entered, the drive will switch the OPMODE [▶ 50] = 2 and step at current level POPI for time1 msec, then at POPI2 [▶ 115] for time2 msec. Time2 is optional. At the end of the full period, control will return to the prior OPMODE [▶ 50]. POPV [▶ 115] limits the maximum velocity during the current step - if velocity reaches the POPV [▶ 115] limit, the POP command will be canceled.

4.6.18 POPI

ASCII - Command	POPI	For Manufacturer Use only	
Syntax Transmit	POPI [Data]		
Syntax Receive	POPI <Data>	Available in	
Type	Variable rw	MMI	No
ASCII Format	Float	CANBus Object Number	No
DIM	A	PROFIBUS PNU	No
Range	-DIPEAK .. DIPEAK	DPR	No
Default	0		
Opmode	-	Data Type Bus/DPR	Integer32
Drive State	-	Weightning	1000
Start Firmware	4.32		
Configuration	No	Revision	1.9
Function Group	Current Controller	EEPROM	No
Short Description	Current Level for POP Command		

Description

POPI gives the current level for the current step command POP [▶ 114]. The value is in amps. POPI is used by the Current Loop Tuning Wizard and other test programs; it is not normally used in applications.

See also: POP [▶ 114], POPI2 [▶ 115], POPV [▶ 115]

4.6.19 POPI2

ASCII - Command	POPI2		For Manufacturer Use only	
Syntax Transmit	POPI2 [Data]			
Syntax Receive	POPI2 <Data>		Available in	
Type	Variable rw		MMI	No
ASCII Format	Float		CANBus Object Number	No
DIM	A		PROFIBUS PNU	No
Range	-DIPEAK .. DIPEAK		DPR	No
Default	0			
Opmode	-		Data Type Bus/DPR	Integer32
Drive State	-		Weightning	1000
Start Firmware	4.32			
Configuration	No		Revision	1.9
Function Group	Current Controller		EEPROM	No
Short Description	Current Level for POP Command			

Description

POPI2 gives the current level for the second stage of the current step command POP [▶ 114]. The value is in amps. POPI2 is used by the Current Loop Tuning Wizard and other test programs; it is not normally used in applications.

See also: POP [▶ 114], POPI [▶ 114], POPV [▶ 115]

4.6.20 POPV

ASCII - Command	POPV		For Manufacturer Use only	
Syntax Transmit	POPV [Data]			
Syntax Receive	POPV <Data>		Available in	
Type	Variable rw		MMI	No
ASCII Format	Float		CANBus Object Number	No
DIM	A		PROFIBUS PNU	No
Range	-VLIM .. VLIM		DPR	No
Default	0			
Opmode	-		Data Type Bus/DPR	Integer32
Drive State	-		Weightning	1000
Start Firmware	4.32			
Configuration	No		Revision	1.9
Function Group	Current Controller		EEPROM	No
Short Description	Max. Speed Level for POP Command			

Description

POPV limits the maximum velocity during a POP [▶ 114] current step - if velocity reaches the POPV limit, the POP [▶ 114] command will be canceled. POPV is used by the Current Loop Tuning Wizard and other test programs; it is not normally used in applications.

See also: POP [▶ 114], POPI [▶ 114], POPI2 [▶ 115]

4.6.21 REFIP

ASCII - Command	**REFIP**		
Syntax Transmit	REFIP [Data]		
Syntax Receive	REFIP <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Float	**CANBus Object Number**	35E2 (hex)
DIM	Amperes	**PROFIBUS PNU**	1826 (dec) IND = 0000xxxx (bin)
Range	0.0 .. min(IPEAK,IPEAKN)	**DPR**	226 (dec)
Default	min(IPEAK,IPEAKN,DICONT/2)		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	1.71		
Configuration	No	**Revision**	1.6
Function Group	Current Controller	**EEPROM**	Yes
Short Description	Peak Rated Current for Homing 7		

Description

The REFIP parameter can be used to set the peak current for homing to a stop. When Homing mode 7 is started (homing to a stop and searching for a zero mark), IPEAK [▶ 110], the normal value for peak current, is set to the value REFIP. When the homing movement is finished, the IPEAK [▶ 110] parameter is reset to the previous (normal) value.

This parameter is also used to reduce the current for Wake&Shake mode (FBTYPE [▶ 190]=7). If the wake&shake mode is started, IPEAK [▶ 110] is set to REFIP. After the mode is stopped, automatically the old IPEAK [▶ 110] value is used.

4.7 Digital I/O

4.7.1 IN1MODE

ASCII - Command	**IN1MODE**		
Syntax Transmit	IN1MODE [Data]		
Syntax Receive	IN1MODE <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	3562 (hex)
DIM	-	**PROFIBUS PNU**	1698 (dec) IND = 0000xxxx (bin)
Range	0 .. 50	**DPR**	98 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	Disabled + Reset (Coldstart)	**Weightning**	
Start Firmware	1.20		
Configuration	Yes	**Revision**	1.9
Function Group	Digital I/O	**EEPROM**	Yes
Short Description	Function of Digital Input 1		

Description

The IN1MODE command is used to configure the function of the digital input INPUT1. The amplifier must be switched off and then on again after an alteration of this parameter.
The following functions can be configured:

Status	Function	Description
IN1MODE=0	Off	The state of the input 1 is read and can be used via fieldbus or Slot card.
IN1MODE=1	Reset	Software reset of the servo amplifier in the event of a fault. The high input signal is ignored, if the drive has no fault. All the functions and displays are set to the initial status. Parameters that are not stored in the EEPROM are erased, the parameter set that is stored in the EEPROM is loaded. If any of the error messages F01, F02, F03, F05, F08, F13, F16 or F19 (p.52) are present, then no software-reset will be carried out, just the error message will be deleted. This means that, for example, the encoder output signals are stable and can continue to be evaluated by the controls. When the input is high, while the auxillary 24V supply is switched on, the drive waits, before the input is set to low. This state is symbolised in the display. The first of the three display positions displays a "A".
IN1MODE=2	Off	
IN1MODE=3	Off	
IN1MODE=4	Off	
IN1MODE=5	Off	
IN1MODE=6	Off	
IN1MODE=7	Off	
IN1MODE=8	SETP.1 / SETP.2	Switches over the setpoint inputs SW/SETP.1/2 at ANCNFG [▸ 60] = 0. This function is only effective if the analog set-point function 0,Xcmd=Setp.1 has been selected. High level at the input : Setpoint input 2 (terminals X3/6,7) is active Low level at the input : Setpoint input 1 (terminals X3/4,5) is active
IN1MODE=9	MT_No_Bit	Here you can select the motion tasks that are stored in the servo amplifier (numbers 1...7) or the reference traverse/homing (0). The motion task number is presented externally at the digital inputs as a logical word, with a width of max. 3 bits . An input is required to start the motion task (INxMODE [▸ 116] =17, Start_MT IO). If you wire up a reference/homing switch (INxMODE [▸ 116] =12, Reference) and (also) want to start a following task (INxMODE [▸ 116] =15, Start_MT Next) externally, the number of inputs that are available for selecting the motion tasks will be further reduced. This function can also be used for the VCT entry functionality, to select the adress of the VCT's. The start of the VCT entry is done by selecting one input with INxMODE [▸ 116]=35.
IN1MODE=10	Intg.Off	Switch off the integral component of the velocity controller, the P-gain remains at the set value, the actual- (rotational) velocity feedback remains in operation.
IN1MODE=11	V / Torq.Contr.	Bypasses the velocity controller. The analog setpoint is taken 1:1 as the setpoint for current control, i.e. change over from velocity control to current (torque) control. High-level at the input : torque control Low-level at the input : velocity control Depending on OPMODE [▸ 50], it changes between OPMODE [▸ 50]=0 (low) and OPMODE [▸ 50]=2 (high) or OPMODE [▸ 50]=1 (low) and OPMODE [▸ 50]=3 (high).
IN1MODE=12	Reference	Polls the reference switch.

Status	Function	Description
IN1MODE=13	ROD/SSI	Changeover of the encoder-emulation (position output) on connector X5. High level at the input : SSI-compatible position signals (ENCMODE [▸ 314] = 2) High level at the input : ROD-compatible position signals (ENCMODE [▸ 314] = 1)
IN1MODE=14	FError_clear	Clear the warning of a contouring error (display n03) or the response monitoring (display n04).
IN1MODE=15	Start_MT Next	The following task, that is defined in the motion task by Start with I/O is started. The target position of the present motion task must be reached before the following task can be started.
IN1MODE=16	Start_MT No x	Start a motion task that is stored in the servo amplifier, by giving the motion task number. After the function has been selected you can enter the motion task number as the auxiliary variable IN1TRIG [▸ 122]. Motion task number 0 (IN1TRIG [▸ 122]=0) initiates homing/reference traverse. A rising edge starts the motion task, a falling edge cancels the motion task.
IN1MODE=17	Start_MT IO	17, Start_MT IO Start of the motion task that has the number that is presented, bit-coded, at the digital inputs (PSTOP/NSTOP/DIGITAL-IN1/DIGITAL-IN2, see function 9, MT_No_Bit). A rising edge starts the motion task a falling edge cancels the motion task by a STOP [▸ 297] - command
IN1MODE=18	Ipeak2 x	Switch over to a second (lower) peak value of current. Scaled as x (0...100) % of the peak current of the instrument. After the function has been selected you can enter the percentage value as the auxiliary variable IN1TRIG [▸ 122]. Make the conversion according to the following equation: IN1TRIG [▸ 122] given in % of IPEAK [▸ 110]
IN1MODE=19	Off	
IN1MODE=20	Start_Jog v=x	Start of the setup mode "Constant velocity" with a defined speed. After selecting the function, you can enter the speed in IN1TRIG [▸ 122]. A rising edge starts the motion, a falling edge cancels the motion. This function works in position control, so OPMODE [▸ 50]=8 has to be selected. The speed is given in units of the position controller given by VUNIT [▸ 342].
IN1MODE=21	U_Mon.off	Turns off the undervoltage monitoring function of the servo amplifier.
IN1MODE=22	MT Restart	Continues the motion task that was previously interrupted by a STOP [▸ 297] - command.
IN1MODE=23	Start2_MT No x	Start of a motion task that is stored in the servo amplifier, with definition of the motion task number. After selecting the function, you can enter the motion task number in IN1TRIG [▸ 122] Motion task number 0 starts the homing run. A rising edge starts the motion task. Warning ! The motion task does not stop automatically if the start signal is removed ! The motion task must be stopped by a falling edge on another digital input (configured with 16, FStart_Nr x) the ASCII command STOP [▸ 297] the STOP function via Bus or digital input

Status	Function	Description
IN1MODE=24	Switch over OPMODE	The two different OPMODE [▶ 50]s, that can be selected for switching over via the digital input, are written in the IN1TRIG [▶ 122] help variable of the this input. The lower byte consists the OPMODE [▶ 50] that should be available when the input has a negative edge. The higher byte consists the OPMODE [▶ 50] that should be available when the input has a positive edge. When the drive is switched on, the OPMODE [▶ 50] is set automatically to the corresponding state of the input. The contents of the help variable must be in decimal !! e.g.: Input1=low OPMODE [▶ 50]=4 Input1=high OPMODE [▶ 50]=8 IN1MODE=24 (Activate Input) IN1TRIG [▶ 122]=2052 (Decimal 0804h) 2052 (Dec) = 0804 (Hex)
IN1MODE=25	Zero_latch	Sets the ROD zero pulse offset. The current position, depending on the ROD resolution that is set, is calculated at the rising edge and stored as NI-Offset in ENCZERO [▶ 316]. After that, an automatic SAVE [▶ 51] is generated. This function is used to perform an automatic setting of the zero pulse in one turn of the motor..
IN1MODE=26	Position Latch	A edge on this input latches the actual position. The position can then be read by LATCHX32 [▶ 263] (positive edge) or LATCHX32N [▶ 264] (negative edge). The actual 16-Bit position (absolute in one turn) can be read by LATCHX16 [▶ 262] (positive edge) and LATCHX16N [▶ 262] (negative edge). The status of the latching can be read by the equivalent bits of DRVSTAT [▶ 171]. The min. cycle time for a low/high to high/low transaction is 500µs. The min. time between two latch pulses is 8 msec. The Latch function does not work with POSCNFG [▶ 285]=1.
IN1MODE=27	Emergency Stop	Low state on the input starts an emergency stop function, that is executed with the ramp DECSTOP [▶ 330]. Independently of the selected OPMODE [▶ 50], in this phase, the drive stops in velocity control. When it has stopped, it switches over to the original mode.
IN1MODE=28	Reserved	
IN1MODE=29	Reserved	
IN1MODE=30	Command Buffer 1	A positive or negative edge on the input starts a command buffer. This command buffer contains separate ASCII objects, that are separated with semicolon (;). The command buffer for the positive edge is INHCMD [▶ 141], the command buffer for the negative edge is INLCMD [▶ 142]. The max. length of that buffers is 56 character for each. If a digital input is configured with INxMODE=30, this input will proceed in that way. When the drive is switched on, the Command buffer is set automatically started to the corresponding state of the input. Remark: Only one of the digital inputs can use the INxMODE=30 function.
IN1MODE=31	Command Buffer 2	A positive or negative edge on the input starts a command buffer. This command buffer contains separate ASCII objects, that are separated with semicolon (;). The command buffer for the positive edge is INHCMDX [▶ 142], the command buffer for the negative edge is INLCMDX [▶ 143].

Status	Function	Description
		The max. length of that buffers is 56 character for each. If a digital input is configured with INxMODE=31, this input will proceed in that way. When the drive is switched on, the Command buffer is set automatically started to the corresponding state of the input. Remark: Only one of the digital inputs can use the INxMODE=31 function.
IN1MODE=32	Brake	A rising edge at the input triggers the braking output of the servo amplifier. This function is only available while the amplifier is disabled. If an error message is active, the brake cannot be de-energized. Warning ! With suspended loads, this function will lead to slipping of the axes ! Starting with 4.78, this function also works if the drive has an error.
IN1MODE=33	see 30	Different from the functionality 30, the resulting answers of the commands are not suppressed, but are send to the seriell communication channel RS232.
IN1MODE=34	see 31	Different from the functionality 31, the resulting answers of the commands are not suppressed, but are send to the seriell communication channel RS232.
IN1MODE=35	Select Velocity/ Current Entry	A positive edge on the digital input causes a takeover of the corresponding VCT entry (see command VCT). The number of the VCT entry is defined by the digital inputs configured with mode 9.
IN1MODE=36	Give Offset to Gearing Function	Gearing mode OPMODE [▸ 50] =4. A high signal on the digital input configured with this INxMODE adds a difference velocity to the gearing. This allows a simple synchronisation of two axes. The difference velocity is given to IN1TRIG [▸ 122]. The scaling is in 20Bit per revolution every 250µs. The difference velocity (n) must be known, then the IN1TRIG [▸ 122] can be calculated: IN1TRIG [▸ 122]= n * 250 / (60 * rpm) e.g. n = 50 rpm IN1TRIG [▸ 122] = 208
IN1MODE=37	Change source of the actual position at EXTPOS=1.	= 0 Actual position is generated by the external encoder selected by GEARMODE [▸ 213] = 1 Actual position is generated by the first feedback device (resolver od high resolution feedback EnDAT or Hiperface)
IN1MODE=38	Enable signal for following motion task	Definition of a motion task with following motion tasks. If INxMODE=15 is used (start of an following motion task via I/O), IN1MODE=38 can be used, to have an additional enable for the start of the following motion tasks. Means, that the following motion task is started, if once a rising edge on digital input 1 was detected and then the INxMODE=15 input is enabled to start the following motion task.
IN1MODE=39	Constant velocity for defined time	This function starts a constant velocity for a defined time. The parameters for velocity and time are given by IN1TRIG [▸ 122]. The velocity is given by the lower 16 bit (scaling by VUNIT [▸ 342]) and the time by the upper 16 bit (given in msec) of the help variable IN1TRIG [▸ 122]. A rising edge at INPUT1 changes the OPMODE [▸ 50] to 0 (digital velocity) and gives the velocity that is given by IN1TRIG [▸ 122].

Status	Function	Description
		After the defined time or a falling edge at INPUT1 is detected, the digital velocity setpoint is set to "0". After the actual velocity has reached "0" the OPMODE [▸ 50] is automatically switched back to the old one. Example for defining the help variable IN1TRIG [▸ 122] 1. Velocity = 1000 rpm time = 10 sec = 10000 msec IN1TRIG [▸ 122] = 0x271003E8 = 655361000 2. Velocity = -500 rpm time = 10 msec IN1TRIG [▸ 122] = 0x000afe0c = 720396
IN1MODE=40	Additional hardware input	The digital input works as an additional hardware input. Only if this input has a high signal, the power stage is enabled. This Function can be used by several inputs. In this case, the inputs are configured in series. All inputs have to be high to enable the power stage. (Starting firmware 4.91)
IN1MODE=41	Fast emergency stop	If the input is going to low, the drive stops the motor using the DECSTOP [▸ 330] ramp. If zero velocity is reached (V [▸ 31]<VEL0 [▸ 338]), the power stage is disabled. While stopping the motor the bit 24 (0x01000000) in TRJSTAT [▸ 185] is set. The input is read in the 250µs task.
IN1MODE=42	Activate/ deactivate electronic gearing	Activate/deactivate electronic gearing in OPMODE [▸ 50] = 4. This function is practical only with slave axis. A rising edge on the digital input starts the motion from 0 to the master speed and a falling edge changes the speed from master speed to 0. The ramp times can be set by ACCR [▸ 247] for the acceleration and DECR [▸ 249] for deceleration time (stating with firmware 5.51).
IN1MODE=43	Activate/ deactivate electronic gearing with position latch	Activate/deactivate electronic gearing in OPMODE [▸ 50] = 4. This function is practical only with slave axis. A rising edge on the digital input starts the motion from 0 to the master speed and a falling edge changes the speed from master speed to 0. The ramp times can be set by ACCR [▸ 247] for the acceleration and DECR [▸ 249] for deceleration time In contrast to IN1MODE = 42, the master position is latched at the rising edge of the input and the position delay caused by the ramp is compensated. IN1TRIG [▸ 122] gives the possibility to add an position offset (in PGEARI [▸ 283] units) to the latched position(starting with Firmware 5.51).

4.7.2 IN1TRIG

ASCII - Command	IN1TRIG		
Syntax Transmit	IN1TRIG [Data]		
Syntax Receive	IN1TRIG <Data>	Available in	
Type	Variable rw	MMI	Yes
ASCII Format	Integer32	CANBus Object Number	3563 (hex)
DIM	-	PROFIBUS PNU	1699 (dec) IND = 0000xxxx (bin)
Range	long int	DPR	99 (dec)
Default	0		
Opmode	All	Data Type Bus/DPR	Integer32
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.8
Function Group	Digital I/O	EEPROM	Yes
Short Description	Variable for IN1MODE		

Description

Auxiliary trigger variable for IN1MODE [▶ 116]. Certain settings of IN1MODE [▶ 116] require you to specify an additional trigger level. See IN1MODE [▶ 116] for further details.

4.7.3 IN2

ASCII - Command	IN2		
Syntax Transmit	IN2		
Syntax Receive	IN2 <Data>	Available in	
Type	Variable ro	MMI	Yes
ASCII Format	Integer8	CANBus Object Number	3564 (hex)
DIM	-	PROFIBUS PNU	1700 (dec) IND = 0000xxxx (bin)
Range	-	DPR	100 (dec)
Default	-		
Opmode	All	Data Type Bus/DPR	Integer8
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.3
Function Group	Digital I/O	EEPROM	No
Short Description	Status of Digital Input 2		

Description

The status of the digital input INPUT2.

4.7.4 IN2MODE

ASCII - Command	**IN2MODE**		
Syntax Transmit	IN2MODE [Data]		
Syntax Receive	IN2MODE <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	3565 (hex)
DIM	-	**PROFIBUS PNU**	1701 (dec) IND = 0000xxxx (bin)
Range	0 .. 50	**DPR**	101 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	Disabled + Reset (Coldstart)	**Weightning**	
Start Firmware	1.20		
Configuration	Yes	**Revision**	1.9
Function Group	Digital I/O	**EEPROM**	Yes
Short Description	Function of Digital Input 2		

Description

The IN2MODE command is used to configure the function of the digital input INPUT2. The amplifier must be switched off and then on again after an alteration of this parameter.
The following functions can be configured:

Status	Function	Description
IN2MODE=0	Off	The state of the input 2 is read and can be used via fieldbus or Slot card.
IN2MODE=1	Off	
IN2MODE=2	Off	
IN2MODE=3	Off	
IN2MODE=4	Off	
IN2MODE=5	Off	
IN2MODE=6	Off	
IN2MODE=7	Off	
IN2MODE=8	SETP.1/ SETP.2	Switches over the setpoint inputs SW/SETP.1/2 . This function is only effective if the analog set-point function ANCNFG [▶ 60]=0,Xcmd=Setp.1 has been selected. High level at the input : Setpoint input 2 (terminals X3/6,7) is active Low level at the input : Setpoint input 1 (terminals X3/4,5) is active
IN2MODE=9	MT_No_Bit	Here you can select the motion tasks that are stored in the servo amplifier (numbers 1...7) or the reference traverse/homing (0). The motion task number is presented externally at the digital inputs as a logical word, with a width of max. 3 bits . An input is required to start the motion task (17, Start_MT IO). If you wire up a reference/homing switch (12, Reference) and (also) want to start a following task (15, Start_MT Next) externally, the number of inputs that are available for selecting the motion tasks will be further reduced. This function can also be used for the VCT entry functionality, to select the adress of the VCT's. The start of the VCT entry is done by selecting one input with INxMODE=35.
IN2MODE=10	Intg.Off	Switch off the integral component of the velocity controller, the P-gain remains at the set value, the actual- (rotational) velocity feedback remains in operation.

Status	Function	Description
IN2MODE=11	v/Torq.Contr.	Bypasses the velocity controller. The analog setpoint is taken 1:1 as the setpoint for current control, i.e. change over from velocity control to current (torque) control. High-level at the input : torque control Low-level at the input : velocity control Depending on OPMODE [▸ 50], it changes between OPMODE [▸ 50]=0 (low) and OPMODE [▸ 50]=2 (high) or OPMODE [▸ 50]=1 (low) and OPMODE [▸ 50]=3 (high).
IN2MODE=12	Reference	Polls the reference switch. A high means reference switch aktive.
IN2MODE=13	ROD/SSI	Changeover of the encoder-emulation (position output) on connector X5. High level at the input : SSI-compatible position signals (ENCMODE [▸ 314]=2) High level at the input : A/B encoder emulation-compatible position signals (ENCMODE [▸ 314]=1)
IN2MODE=14	FError_clear	Clear the warning of a contouring error (display n03) or the response monitoring (display n04).
IN2MODE=15	Start_MT Next	The following task, that is defined in the motion task by Start with I/O is started. The target position of the present motion task must be reached before the following task can be started.
IN2MODE=16	Start_MT No x	Start a motion task that is stored in the servo amplifier, by giving the motion task number. After the function has been selected you can enter the motion task number as the auxiliary variable IN2TRIG [▸ 128]. Motion task number 0 (IN2TRIG [▸ 128]=0) initiates homing/reference traverse. A rising edge starts the motion task, a falling edge cancels the motion task.
IN2MODE=17	Start_MT IO	17, Start_MT IO Start of the motion task that has the number that is presented, bit-coded, at the digital inputs (PSTOP/NSTOP/DIGITAL-IN1/DIGITAL-IN2, see function 9, MT_No_Bit). A rising edge starts the motion task a falling edge cancels the motion task by a STOP [▸ 297] - command
IN2MODE=18	Ipeak2 x	Switch over to a second (lower) peak value of current. Scaled as x (0...100) % of the peak current of the instrument. After the function has been selected you can enter the percentage value as the auxiliary variable IN2TRIG [▸ 128]. Make the conversion according to the following equation: IN2TRIG [▸ 128] given in % of IPEAK [▸ 110]
IN2MODE=19	Reserved	
IN2MODE=20	Start_Jog v=x	Start of the setup mode "Constant velocity" with a defined speed. After selecting the function, you can enter the speed in IN2TRIG [▸ 128]. A rising edge starts the motion, a falling edge cancels the motion. This function works in position control, so OPMODE [▸ 50]=8 has to be selected. The speed is given in units of the position controller given by VUNIT [▸ 342].
IN2MODE=21	U_Mon.off	Turns off the undervoltage monitoring function of the servo amplifier.
IN2MODE=22	MT Restart	Continues the motion task that was previously interrupted by a STOP [▸ 297] - command.
IN2MODE=23	Start2_MT No x	Start of a motion task that is stored in the servo amplifier, with definition of the motion task number. After selecting the function, you can enter the motion task number in IN2TRIG [▸ 128]

Status	Function	Description
		Motion task number 0 starts the homing run. A rising edge starts the motion task. Warning ! The motion task does not stop automatically if the start signal is removed ! The motion task must be stopped by a falling edge on another digital input (configured with 16, FStart_Nr x) the ASCII command STOP [▶ 297] the STOP function via Bus or digital input
IN2MODE=24	Switch over OPMODE	The two different OPMODE [▶ 50]s, that can be selected for switching over via the digital input, are written in the IN2TRIG [▶ 128] help variable of the this input. The lower byte consists the OPMODE [▶ 50] that should be available when the input has a negative edge. The higher byte consists the OPMODE [▶ 50] that should be available when the input has a positive edge. When the drive is switched on, the OPMODE [▶ 50] is set automatically to the corresponding state of the input. The contents of the help variable must be in decimal !! e.g.: Input2=low OPMODE [▶ 50]=4 Input2=high OPMODE [▶ 50]=8 IN2MODE=24 (Activate Input) IN2TRIG [▶ 128]=2052 (Decimal 0804h) 2052 (Dec) = 0804 (Hex)
IN2MODE=25	Zero_latch	Sets the ROD zero pulse offset. The current position, depending on the A/B encoder emulation resolution that is set, is calculated at the rising edge and stored as NI-Offset in ENCZERO [▶ 316]. After that, an automatic SAVE [▶ 51] is generated. This function is used to perform an automatic setting of the zero pulse in one turn of the motor..
IN2MODE=26	Position Latch	A edge on this input latches the actual position. The position can then be read by LATCH32 [▶ 261] (positive edge) or LATCH32N [▶ 261] (negative edge). The actual 16-Bit position (absolute in one turn) can be read by LATCH16 [▶ 260] (positive edge) and LATCH16N [▶ 260] (negative edge). The status of the latching can be read by the equivalent bits of DRVSTAT [▶ 171]. The min. cycle time for a low/high to high/low transaction is 500µs. The min. time between two latch pulses is 8 msec. The Latch function does not work with POSCNFG [▶ 285]=1.
IN2MODE=27	Emergency Stop	Low state on the input starts an emergency stop function, that is executed with the ramp DECSTOP [▶ 330]. Independently of the selected OPMODE [▶ 50], in this phase, the drive stops in velocity control. When it has stopped, it switches over to the original mode.
IN2MODE=28	Reserved	
IN2MODE=29	Reserved	
IN2MODE=30	Command Buffer 1	A positive or negative edge on the input starts a command buffer. This command buffer contains separate ASCII objects, that are separated with semicolon (;). The command buffer for the positive edge is INHCMD [▶ 141], the command buffer for the negative edge is INLCMD [▶ 142]. The max. length of that buffers is 56 character for each. If a digital input is configured with INxMODE=30, this input will proceed in that way. When the drive is switched on, the Command buffer is set automatically

Status	Function	Description
		started to the corresponding state of the input. Remark: Only one of the digital inputs can use the INxMODE=30 function.
IN2MODE=31	Command Buffer 2	A positive or negative edge on the input starts a command buffer. This command buffer contains separate ASCII objects, that are separated with semicolon (;). The command buffer for the positive edge is INHCMDX [▸ 142], the command buffer for the negative edge is INLCMDX [▸ 143]. The max. length of that buffers is 56 character for each. If a digital input is configured with INxMODE=31, this input will proceed in that way. When the drive is switched on, the Command buffer is set automatically started to the corresponding state of the input. Remark: Only one of the digital inputs can use the INxMODE=31 function.
IN2MODE=32	Brake	A rising edge at the input triggers the braking output of the servo amplifier. This function is only available while the amplifier is disabled. If an error message is active, the brake cannot be de-energized. Warning ! With suspended loads, this function will lead to slipping of the axes !
IN2MODE=33	see 30	Different from the functionality 30, the resulting answers of the commands are not suppressed, but are send to the seriell communication channel RS232.
IN2MODE=34	see 31	Different from the functionality 31, the resulting answers of the commands are not suppressed, but are send to the seriell communication channel RS232.
IN2MODE=35	Select Velocity/ Current Entry	A positive edge on the digital input causes a takeover of the corresponding VCT entry (see command VCT). The number of the VCT entry is defined by the digital inputs configured with mode 9.
IN2MODE=36	Give Offset to Gearing Function	Gearing mode 4. A high signal on the digital input configured with this INxMODE adds a difference velocity to the gearing. This allows a simple synchronisation of two axes. The difference velocity is given to IN1TRIG [▸ 122]. The scaling is in 20Bit per revolution every 250µs. The difference velocity (n) must be known, then the IN1TRIG [▸ 122] can be calculated: IN1TRIG [▸ 122]= n * 250 / (60 * rpm) e.g. n = 50 rpm IN1TRIG [▸ 122] = 208
IN2MODE=37	Change source of the actual position at EXTPOS [▸ 253]=1.	= 0 Actual position is generated by the external encoder selected by GEARMODE [▸ 213] = 1 Actual position is generated by the first feedback device (resolver od high resolution feedback EnDAT or Hiperface)
IN2MODE=38		Definition of a motion task with following motion tasks. If INxMODE=15 is used (start of an following motion task via I/O), IN1MODE=38 can be used, to have an additional enable for the start of the following motion tasks. Means, that the following motion task is started, if once a rising edge on digital input 1 was detected and then the INxMODE=15 input is enabled to start the following motion task.

Status	Function	Description
IN2MODE=39	Constant velocity for defined time	This function starts a constant velocity for a defined time. The parameters for velocity and time are given by IN2TRIG [▸ 128]. The velocity is given by the lower 16 bit (scaling by VUNIT [▸ 342]) and the time by the upper 16 bit (given in msec) of the help variable IN2TRIG [▸ 128]. A rising edge at INPUTx changes the OPMODE [▸ 50] to 0 (digital velocity) and gives the velocity that is given by IN2TRIG [▸ 128]. After the defined time or a falling edge at INPUTx is detected, the digital velocity setpoint is set to "0". After the actual velocity has reached "0" the OPMODE [▸ 50] is automatically switched back to the old one. Example for defining the help variable IN2TRIG [▸ 128] 1. Velocity = 1000 rpm time = 10 sec = 10000 msec IN2TRIG [▸ 128] = 0x271003E8 = 655361000 2. Velocity = -500 rpm time = 10 msec IN2TRIG [▸ 128] = 0x000afe0c = 720396
IN2MODE=40	Additional hardware enable	The digital input works as an additional hardware input. Only if this input has a high signal, the power stage is enabled. This Function can be used by several inputs. In this case, the inputs are configured in series. All inputs have to be high to enable the power stage. (Starting firmware 4.91)
IN2MODE=41	Fast emercency stop	If the input is going to low, the drive stops the motor using the DECSTOP [▸ 330] ramp. If zero velocity is reached (V<VEL0 [▸ 338]), the power stage is disabled. While stopping the motor the bit 24 (0x01000000) in TRJSTAT [▸ 185] is set. The input is read in the 250µs task.
IN2MODE=42	Activate/ deactivate electronic gearing	Activate/deactivate electronic gearing in OPMODE [▸ 50] = 4. This function is practical only with slave axis. A rising edge on the digital input starts the motion from 0 to the master speed and a falling edge changes the speed from master speed to 0. The ramp times can be set by ACCR [▸ 247] for the acceleration and DECR [▸ 249] for deceleration time (stating with firmware 5.51).
IN2MODE=43	Activate/ deactivate electronic gearing with position latch	Activate/deactivate electronic gearing in OPMODE [▸ 50] = 4. This function is practical only with slave axis. A rising edge on the digital input starts the motion from 0 to the master speed and a falling edge changes the speed from master speed to 0. The ramp times can be set by ACCR [▸ 247] for the acceleration and DECR [▸ 249] for deceleration time In contrast to IN2MODE = 42, the master position is latched at the rising edge of the input and the position delay caused by the ramp is compensated. IN2TRIG [▸ 128] gives the possibility to add an position offset (in PGEARI [▸ 283] units) to the latched position(starting with Firmware 5.51).

Also see about this

IN3TRIG [▸ 134]

4.7.5 IN2TRIG

ASCII - Command	**IN2TRIG**			
Syntax Transmit	IN2TRIG [Data]			
Syntax Receive	IN2TRIG <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer32		**CANBus Object Number**	3566 (hex)
DIM	-		**PROFIBUS PNU**	1702 (dec) IND = 0000xxxx (bin)
Range	long int		**DPR**	102 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.8
Function Group	Digital I/O		**EEPROM**	Yes
Short Description	Variable for IN2MODE			

Description

Auxiliary trigger variable for IN2MODE [▶ 123]. Certain settings of IN2MODE [▶ 123] require you to specify an additional trigger level. See IN2MODE [▶ 123] for further details.

4.7.6 IN3

ASCII - Command	**IN3**			
Syntax Transmit	IN3			
Syntax Receive	IN3 <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Integer8		**CANBus Object Number**	3567 (hex)
DIM	-		**PROFIBUS PNU**	1703 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	103 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Digital I/O		**EEPROM**	No
Short Description	Status of Digital Input 3			

Description

The status of the digital input INPUT3.

4.7.7 IN3MODE

ASCII - Command	**IN3MODE**		
Syntax Transmit	IN3MODE [Data]		
Syntax Receive	IN3MODE <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	3568 (hex)
DIM	-	**PROFIBUS PNU**	1704 (dec) IND = 0000xxxx (bin)
Range	0 .. 50	**DPR**	104 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	Disabled + Reset (Coldstart)	**Weightning**	
Start Firmware	1.20		
Configuration	Yes	**Revision**	1.9
Function Group	Digital I/O	**EEPROM**	Yes
Short Description	Function of Digital Input 3		

Description

The IN3MODE command is used to configure the function of the digital input INPUT3. The amplifier must be switched off and then on again after an alteration of this parameter.
The following functions can be configured:

Status	Function	Description
IN3MODE=0	Off	The state of the input 3 is read and can be used via fieldbus or Slot card.
IN3MODE=1	Off	
IN3MODE=2	PSTOP	A low on the input disables the positive direction (clockwise if DIR [▶ 331]=1, counterclockwise if DIR [▶ 331]=0). At the same time, a warning "n10" is displayed. If a negative edge is recognized while the motor is running, the drive stops the motor in OPMODE [▶ 50]=0 (velocity control with setpoint zero) using the DECSTOP [▶ 330] ramp. When the motor has stopped, the old OPMODE [▶ 50] is activated.
IN3MODE=3	NSTOP (4.78)	A low on the input disables the negative direction (clockwise if DIR [▶ 331]=0, counterclockwise if DIR [▶ 331]=1). At the same time, a warning "n11" is displayed. If a negative edge is recognized while the motor is running, the drive stops the motor in OPMODE [▶ 50]=0 (velocity control with setpoint zero) using the DECSTOP [▶ 330] ramp. When the motor has stopped, the old OPMODE [▶ 50] is activated.
IN3MODE=4	PSTOP + Intg.Off	A low on the input disables the positive direction (clockwise if DIR [▶ 331]=1, counterclockwise if DIR [▶ 331]=0). At the same time, a warning "n10" is displayed. If a negative edge is recognized while the motor is running, the drive stops the motor in OPMODE [▶ 50]=0 (velocity control with setpoint zero) using the DECSTOP [▶ 330] ramp. When the motor has stopped, the old OPMODE [▶ 50] is activated (without integral part in the velocity controller).
IN3MODE=5	Off	
IN3MODE=6	PSTOP+NSTOP	A low on the input disables the positive and the negative direction. At the same time, a warning "n10" and "n11" is displayed. If a negative edge is recognized while the motor is running, the drive stops the motor in OPMODE [▶ 50]=0 (velocity control with setpoint zero) using the DECSTOP [▶ 330] ramp. When the motor has stopped, the old OPMODE [▶ 50] is activated.

Status	Function	Description
IN3MODE=7	P/ Nstop+Intg.Off	A low on the input disables the positive and the negative direction. At the same time, a warning "n10" and "n11" is displayed. If a negative edge is recognized while the motor is running, the drive stops the motor in OPMODE [▸ 50]=0 (velocity control with setpoint zero) using the DECSTOP [▸ 330] ramp. When the motor has stopped, the old OPMODE [▸ 50] is activated (without integral part in the velocity controller).
IN3MODE=8	SETP.1/ SETP.2	Switches over the setpoint inputs SW/SETP.1/2 . This function is only effective if the analog set-point function ANCNFG [▸ 60] = 0,Xcmd=Setp.1 has been selected. High level at the input : Setpoint input 2 (terminals X3/6,7) is active Low level at the input : Setpoint input 1 (terminals X3/4,5) is active
IN3MODE=9	MT_No_Bit	Here you can select the motion tasks that are stored in the servo amplifier (numbers 1...7) or the reference traverse/homing (0). The motion task number is presented externally at the digital inputs as a logical word, with a width of max. 3 bits . An input is required to start the motion task (17, Start_MT IO). If you wire up a reference/homing switch (12, Reference) and (also) want to start a following task (15, Start_MT Next) externally, the number of inputs that are available for selecting the motion tasks will be further reduced. This function can also be used for the VCT entry functionality, to select the adress of the VCT's. The start of the VCT entry is done by selecting one input with INxMODE=35.
IN3MODE=10	Intg.Off	Switch off the integral component of the velocity controller, the P-gain remains at the set value, the actual- (rotational) velocity feedback remains in operation.
IN3MODE=11	v/Torq.Contr.	Bypasses the velocity controller. The analog setpoint is taken 1:1 as the setpoint for current control, i.e. change over from velocity control to current (torque) control. High-level at the input : torque control Low-level at the input : velocity control Depending on OPMODE [▸ 50], it changes between OPMODE [▸ 50]=0 (low) and OPMODE [▸ 50]=2 (high) or OPMODE [▸ 50]=1 (low) and OPMODE [▸ 50]=3 (high).
IN3MODE=12	Reference	Polls the reference switch.
IN3MODE=13	ROD/SSI	Changeover of the encoder-emulation (position output) on connector X5. High level at the input : SSI-compatible position signals (ENCMODE [▸ 314]=2 High level at the input : A/B encoder-compatible position signals (ENCMODE [▸ 314]=1)
IN3MODE=14	FError_clear	Clear the warning of a contouring error (display n03) or the response monitoring (display n04).
IN3MODE=15	Start_MT Next	The following task, that is defined in the motion task by “Start with I/O” is started. The target position of the present motion task must be reached before the following task can be started.
IN3MODE=16	Start_MT No x	Start a motion task that is stored in the servo amplifier, by giving the motion task number. After the function has been selected you can enter the motion task number as the auxiliary variable IN3TRIG [▸ 134]. Motion task number “0” (**IN3TRIG** =0) initiates homing/reference traverse. A rising edge starts the motion task, a falling edge cancels the motion task (STOP [▸ 297]).

Status	Function	Description
IN3MODE=17	Start_MT IO	17, Start_MT IO Start of the motion task that has the number that is presented, bit-coded, at the digital inputs (PSTOP/NSTOP/DIGITAL-IN1/DIGITAL-IN2, see function 9, MT_No_Bit). A rising edge starts the motion task a falling edge cancels the motion task by a STOP [▸ 297] - command
IN3MODE=18	Ipeak2 x	Switch over to a second (lower) peak value of current. Scaled as x (0...100) % of the peak current of the instrument. After the function has been selected you can enter the percentage value as the auxiliary variable IN3TRIG [▸ 134]. Make the conversion according to the following equation: IN3TRIG [▸ 134] given in % of IPEAK [▸ 110]
IN3MODE=19	Reserved	
IN3MODE=20	Start_Jog v=x	Start of the setup mode "Constant velocity" with a defined speed. After selecting the function, you can enter the speed in IN3TRIG [▸ 134]. A rising edge starts the motion, a falling edge cancels the motion. This function works in position control, so OPMODE [▸ 50]=8 has to be selected. The speed is given in units of the position controller given by VUNIT [▸ 342].
IN3MODE=21	U_Mon.off	Turns off the undervoltage monitoring function of the servo amplifier.
IN3MODE=22	MT Restart	Continues the motion task that was previously interrupted by a STOP [▸ 297] - command.
IN3MODE=23	Start2_MT No x	Start of a motion task that is stored in the servo amplifier, with definition of the motion task number. After selecting the function, you can enter the motion task number in IN3TRIG [▸ 134] Motion task number “0” starts the homing run. A rising edge starts the motion task. Warning ! The motion task does not stop automatically if the start signal is removed ! The motion task must be stopped by — a falling edge on another digital input (configured with 16, FStart_Nr x) — the ASCII command STOP [▸ 297] — the STOP function via Bus or digital input
IN3MODE=24	Switch over OPMODE	The two different OPMODE [▸ 50]s, that can be selected for switching over via the digital input, are written in the IN3TRIG [▸ 134] help variable of the this input. The lower byte consists the OPMODE [▸ 50] that should be available when the input has a negative edge. The higher byte consists the OPMODE [▸ 50] that should be available when the input has a positive edge. When the drive is switched on, the OPMODE [▸ 50] is set automatically to the corresponding state of the input. The contents of the help variable must be in decimal !! e.g.: Input3=low **OPMODE** =4 Input3=high **OPMODE** =8 IN3MODE=24 (Activate Input) IN3TRIG [▸ 134] =2052 (Decimal 0804h) 2052 (Dec) = 0804 (Hex)

Status	Function	Description
IN3MODE=25	Zero_latch	Sets the ROD zero pulse offset. The current position, depending on the A/B encoder emulation resolution that is set, is calculated at the rising edge and stored as NI-Offset in ENCZERO [▶ 316]. After that, an automatic SAVE [▶ 51] is generated. This function is used to perform an automatic setting of the zero pulse in one turn of the motor..
IN3MODE=26	Off	
IN3MODE=27	Emergency Stop	Low state on the input starts an emergency stop function, that is executed with the ramp DECSTOP [▶ 330]. Independently of the selected OPMODE [▶ 50], in this phase, the drive stops in velocity control. When it has stopped, it switches over to the original mode.
IN3MODE=28	Reserved	
IN3MODE=29	Reserved	
IN3MODE=30	Command Buffer 1	A positive or negative edge on the input starts a command buffer. This command buffer contains separate ASCII objects, that are separated with semicolon (;). The command buffer for the positive edge is INHCMD [▶ 141], the command buffer for the negative edge is INLCMD [▶ 142]. The max. length of that buffers is 56 character for each. If a digital input is configured with INxMODE=30, this input will proceed in that way. When the drive is switched on, the Command buffer is set automatically started to the corresponding state of the input. Remark: Only one of the digital inputs can use the INxMODE=30 function.
IN3MODE=31	Command Buffer 2	A positive or negative edge on the input starts a command buffer. This command buffer contains separate ASCII objects, that are separated with semicolon (;). The command buffer for the positive edge is INHCMDX [▶ 142], the command buffer for the negative edge is INLCMDX [▶ 143]. The max. length of that buffers is 56 character for each. If a digital input is configured with INxMODE=31, this input will proceed in that way. When the drive is switched on, the Command buffer is set automatically started to the corresponding state of the input. Remark: Only one of the digital inputs can use the INxMODE=31 function.
IN3MODE=32	Brake	A rising edge at the input triggers the braking output of the servo amplifier. This function is only available while the amplifier is disabled. If an error message is active, the brake cannot be de-energized. Warning ! With suspended loads, this function will lead to slipping of the axes !
IN3MODE=33	see 30	Different from the functionality 30, the resulting answers of the commands are not suppressed, but are send to the seriell communication channel RS232.
IN3MODE=34	see 31	Different from the functionality 31, the resulting answers of the commands are not suppressed, but are send to the seriell communication channel RS232.
IN3MODE=35	Select Velocity/ Current Entry	A positive edge on the digital input causes a takeover of the corresponding VCT entry (see command VCT). The number of the VCT entry is defined by the digital inputs configured with mode 9.

Status	Function	Description
IN3MODE=36	Give Offset to Gearing Function	Gearing mode 4. A high signal on the digital input configured with this INxMODE adds a difference velocity to the gearing. This allows a simple synchronisation of two axes. The difference velocity is given to IN3TRIG [▶ 134]. The scaling is in 20Bit per revolution every 250µs. The difference velocity (n) must be known, then the IN3TRIG [▶ 134] can be calculated: IN3TRIG [▶ 134]= n * 250 / (60 * rpm) e.g. n = 50 rpm IN3TRIG [▶ 134] = 208
IN3MODE=37	Change source of the actual position at EXTPOS [▶ 253]=1.	= 0 Actual position is generated by the external encoder selected by GEARMODE [▶ 213] = 1 Actual position is generated by the first feedback device (resolver od high resolution feedback EnDAT or Hiperface)
IN3MODE=38		Definition of a motion task with following motion tasks. If INxMODE=15 is used (start of an following motion task via I/O), IN3MODE=38 can be used, to have an additional enable for the start of the following motion tasks. Means, that the following motion task is started, if once a rising edge on digital input 1 was detected and then the INxMODE=15 input is enabled to start the following motion task.
IN3MODE=39	Constant velocity for defined time	This function starts a constant velocity for a defined time. The parameters for velocity and time are given by INxTRIG. The velocity is given by the lower 16 bit (scaling by VUNIT [▶ 342]) and the time by the upper 16 bit (given in msec) of the help variable INxTRIG. A rising edge at INPUTx changes the OPMODE [▶ 50] to 0 (digital velocity) and gives the velocity that is given by IN3TRIG [▶ 134]. After the defined time or a falling edge at INPUTx is detected, the digital velocity setpoint is set to "0". After the actual velocity has reached "0" the OPMODE [▶ 50] is automatically switched back to the old one. Example for defining the help variable IN3TRIG [▶ 134] 1. Velocity = 1000 rpm time = 10 sec = 10000 msec IN3TRIG [▶ 134] = 0x271003E8 = 655361000 2. Velocity = -500 rpm time = 10 msec IN3TRIG [▶ 134] = 0x000afe0c = 720396
IN3MODE=40	Additional hardware enable	The digital input works as an additional hardware input. Only if this input has a high signal, the power stage is enabled. This Function can be used by several inputs. In this case, the inputs are configured in series. All inputs have to be high to enable the power stage. (Starting firmware 4.91)
IN3MODE=41	Fast emergency stop	If the input is going to low, the drive stops the motor using the DECSTOP [▶ 330] ramp. If zero velocity is reached (V [▶ 31]<VEL0 [▶ 338]), the power stage is disabled. While stopping the motor the bit 24 (0x01000000) in TRJSTAT [▶ 185] is set. The input is read in the 250µs task.

Status	Function	Description
IN3MODE=42	Activate/ deactivate electronic gearing	Activate/deactivate electronic gearing in OPMODE [▸ 50] = 4. This function is practical only with slave axis. A rising edge on the digital input starts the motion from 0 to the master speed and a falling edge changes the speed from master speed to 0. The ramp times can be set by ACCR [▸ 247] for the acceleration and DECR [▸ 249] for deceleration time (stating with firmware 5.51).
IN3MODE=43	Activate/ deactivate electronic gearing with position latch	Activate/deactivate electronic gearing in OPMODE [▸ 50] = 4. This function is practical only with slave axis. A rising edge on the digital input starts the motion from 0 to the master speed and a falling edge changes the speed from master speed to 0. The ramp times can be set by ACCR [▸ 247] for the acceleration and DECR [▸ 249] for deceleration time In contrast to IN3MODE = 42, the master position is latched at the rising edge of the input and the position delay caused by the ramp is compensated. IN3TRIG [▸ 134] gives the possibility to add an position offset (in PGEARI [▸ 283] units) to the latched position(starting with Firmware 5.51).

Also see about this

- IN2TRIG [▸ 128]

4.7.8 IN3TRIG

ASCII - Command	IN3TRIG		
Syntax Transmit	IN3TRIG [Data]		
Syntax Receive	IN3TRIG <Data>	Available in	
Type	Variable rw	MMI	Yes
ASCII Format	Integer32	CANBus Object Number	3569 (hex)
DIM	-	PROFIBUS PNU	1705 (dec) IND = 0000xxxx (bin)
Range	long int	DPR	105 (dec)
Default	0		
Opmode	All	Data Type Bus/DPR	Integer32
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.8
Function Group	Digital I/O	EEPROM	Yes
Short Description	Variable for IN3MODE		

Description

Auxiliary trigger variable for IN3MODE [▸ 129]. Certain settings of IN3MODE [▸ 129] require you to specify an additional trigger level. See IN3MODE [▸ 129] for further details

4.7.9 IN4

ASCII - Command	**IN4**			
Syntax Transmit	IN4			
Syntax Receive	IN4 <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Integer8		**CANBus Object Number**	356A (hex)
DIM	-		**PROFIBUS PNU**	1706 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	106 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Digital I/O		**EEPROM**	No
Short Description	Status of Digital Input 4.			

Description

The status of the digital input INPUT4.

4.7.10 IN4MODE

ASCII - Command	**IN4MODE**			
Syntax Transmit	IN4MODE [Data]			
Syntax Receive	IN4MODE <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer8		**CANBus Object Number**	356B (hex)
DIM	-		**PROFIBUS PNU**	1707 (dec) IND = 0000xxxx (bin)
Range	0 .. 50		**DPR**	107 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	Disabled + Reset (Coldstart)		**Weightning**	
Start Firmware	1.20			
Configuration	Yes		**Revision**	1.9
Function Group	Digital I/O		**EEPROM**	Yes
Short Description	Function of Digital Input 4			

Description

The IN4MODE command is used to configure the function of the digital input INPUT4.
The amplifier must be switched off and then on again after an alteration of this parameter.
The following functions can be configured:

Status	Function	Description
IN4MODE=0	Off	The state of the input 4 is read and can be used via fieldbus or Slot card.
IN4MODE=1	Off	
IN4MODE=2	PSTOP (4.78)	A low on the input disables the positive direction (clockwise if DIR [▸ 331]=1, counterclockwise if DIR [▸ 331]=0). At the same time, a warning "n10" is displayed. If a negative edge is recognized while

Status	Function	Description
		the motor is running, the drive stops the motor in OPMODE [▶ 50]=0 (velocity control with setpoint zero) using the DECSTOP [▶ 330] ramp. When the motor has stopped, the old OPMODE [▶ 50] is activated.
IN4MODE=3	NSTOP	A low on the input disables the negative direction (clockwise if DIR [▶ 331]=0, counterclockwise if DIR [▶ 331]=1). At the same time, a warning "n11" is displayed. If a negative edge is recognized while the motor is running, the drive stops the motor in OPMODE [▶ 50]=0 (velocity control with setpoint zero) using the DECSTOP [▶ 330] ramp. When the motor has stopped, the old OPMODE [▶ 50] is activated.
IN4MODE=4	Off	
IN4MODE=5	NSTOP+Intg.Off	A low on the input disables the negative direction (clockwise if DIR [▶ 331]=0, counterclockwise if DIR [▶ 331]=1). At the same time, a warning "n11" is displayed. If a negative edge is recognized while the motor is running, the drive stops the motor in OPMODE [▶ 50]=0 (velocity control with setpoint zero) using the DECSTOP [▶ 330] ramp. When the motor has stopped, the old OPMODE [▶ 50] is activated (without integral part in the velocity controller).
IN4MODE=6	Off	
IN4MODE=7	Off	
IN4MODE=8	SETP.1/SETP.2	Switches over the setpoint inputs SW/SETP.1/2 . This function is only effective if the analog set-point function ANCNFG [▶ 60]=0,Xcmd=Setp.1 has been selected. High level at the input : Setpoint input 2 (terminals X3/6,7) is active Low level at the input : Setpoint input 1 (terminals X3/4,5) is active
IN4MODE=9	MT_No_Bit	Here you can select the motion tasks that are stored in the servo amplifier (numbers 1...7) or the reference traverse/homing (0). The motion task number is presented externally at the digital inputs as a logical word, with a width of max. 3 bits . An input is required to start the motion task (17, Start_MT IO). If you wire up a reference/homing switch (12, Reference) and (also) want to start a following task (15, Start_MT Next) externally, the number of inputs that are available for selecting the motion tasks will be further reduced. This function can also be used for the VCT entry functionality, to select the adress of the VCT's. The start of the VCT entry is done by selecting one input with INxMODE=35.
IN4MODE=10	Intg.Off	Switch off the integral component of the velocity controller, the P-gain remains at the set value, the actual- (rotational) velocity feedback remains in operation.
IN4MODE=11	v/Torq.Contr.	Bypasses the velocity controller. The analog setpoint is taken 1:1 as the setpoint for current control, i.e. change over from velocity control to current (torque) control. High-level at the input : torque control Low-level at the input : velocity control Depending on OPMODE [▶ 50], it changes between OPMODE [▶ 50]=0 (low) and OPMODE [▶ 50]=2 (high) or OPMODE [▶ 50]=1 (low) and OPMODE [▶ 50]=3 (high).
IN4MODE=12	Reference	Polls the reference switch.
IN4MODE=13	ROD/SSI	Changeover of the encoder-emulation (position output) on connector X5. High level at the input : SSI-compatible position signals (ENCMODE

Status	Function	Description
		[▸ 314]=2) High level at the input : A/B encoder compatible position signals (ENCMODE [▸ 314]=1)
IN4MODE=14	FError_clear	Clear the warning of a contouring error (display n03) or the response monitoring (display n04).
IN4MODE=15	Start_MT Next	The following task, that is defined in the motion task by “Start with I/O” is started. The target position of the present motion task must be reached before the following task can be started.
IN4MODE=16	Start_MT No x	Start a motion task that is stored in the servo amplifier, by giving the motion task number. After the function has been selected you can enter the motion task number as the auxiliary variable IN4TRIG [▸ 141]. Motion task number “0” (IN4TRIG [▸ 141]=0) initiates homing/reference traverse. A rising edge starts the motion task, a falling edge cancels the motion task.
IN4MODE=17	Start_MT IO	17, Start_MT IO Start of the motion task that has the number that is presented, bit-coded, at the digital inputs (PSTOP/NSTOP/DIGITAL-IN1/DIGITAL-IN2, see function 9, MT_No_Bit). A rising edge starts the motion task a falling edge cancels the motion task by a STOP [▸ 297] - command
IN4MODE=18	Ipeak2 x	Switch over to a second (lower) peak value of current. Scaled as x (0...100) % of the peak current of the instrument. After the function has been selected you can enter the percentage value as the auxiliary variable IN4TRIG [▸ 141]. Make the conversion according to the following equation: IN4TRIG [▸ 141] given in % of IPEAK [▸ 110]
IN4MODE=19	Reserved	
IN4MODE=20	Start_Jog v=x	Start of the setup mode "Constant velocity" with a defined speed. After selecting the function, you can enter the speed in IN4TRIG [▸ 141]. A rising edge starts the motion, a falling edge cancels the motion. This function works in position control, so OPMODE [▸ 50]=8 has to be selected. The speed is given in units of the position controller given by VUNIT [▸ 342].
IN4MODE=21	U_Mon.off	Turns off the undervoltage monitoring function of the servo amplifier.
IN4MODE=22	MT Restart	Continues the motion task that was previously interrupted by a STOP [▸ 297] - command.
IN4MODE=23	Start2_MT No x	Start of a motion task that is stored in the servo amplifier, with definition of the motion task number. After selecting the function, you can enter the motion task number in IN4TRIG [▸ 141] Motion task number “0” starts the homing run. A rising edge starts the motion task. Warning ! The motion task does not stop automatically if the start signal is removed ! The motion task must be stopped by — a falling edge on another digital input (configured with 16, FStart_Nr x) — the ASCII command STOP [▸ 297] — the STOP function via Bus or digital input

Status	Function	Description
IN4MODE=24	Switch over OPMODE	The two different OPMODE [▶ 50]s, that can be selected for switching over via the digital input, are written in the IN4TRIG [▶ 141] help variable of the this input. The lower byte consists the OPMODE [▶ 50] that should be available when the input has a negative edge. The higher byte consists the OPMODE [▶ 50] that should be available when the input has a positive edge. When the drive is switched on, the OPMODE [▶ 50] is set automatically to the corresponding state of the input. The contents of the help variable must be in decimal !! e.g.: Input4=low OPMODE [▶ 50]=4 Input4=high OPMODE [▶ 50]=8 IN4MODE=24 (Activate Input) **IN4TRIG** =2052 (Decimal 0804h) 2052 (Dec) = 0804 (Hex)
IN4MODE=25	Zero_latch	Sets the A/B encoder emulation zero pulse offset. The current position, depending on the A/B encoder emulation resolution that is set, is calculated at the rising edge and stored as NI-Offset in ENCZERO [▶ 316]. After that, an automatic SAVE [▶ 51] is generated. This function is used to perform an automatic setting of the zero pulse in one turn of the motor..
IN4MODE=26	Off	
IN4MODE=27	Emergency Stop	Low state on the input starts an emergency stop function, that is executed with the ramp DECSTOP [▶ 330]. Independently of the selected OPMODE [▶ 50], in this phase, the drive stops in velocity control. When it has stopped, it switches over to the original mode.
IN4MODE=28	Reserved	
IN4MODE=29	Reserved	
IN4MODE=30	Command Buffer 1	A positive or negative edge on the input starts a command buffer. This command buffer contains separate ASCII objects, that are separated with semicolon (;). The command buffer for the positive edge is INHCMD [▶ 141], the command buffer for the negative edge is INLCMD [▶ 142]. The max. length of that buffers is 56 character for each. If a digital input is configured with INxMODE=30, this input will proceed in that way. When the drive is switched on, the Command buffer is set automatically started to the corresponding state of the input. Remark: Only one of the digital inputs can use the INxMODE=30 function.
IN4MODE=31	Command Buffer 2	A positive or negative edge on the input starts a command buffer. This command buffer contains separate ASCII objects, that are separated with semicolon (;). The command buffer for the positive edge is INHCMDX [▶ 142], the command buffer for the negative edge is INLCMDX [▶ 143]. The max. length of that buffers is 56 character for each. If a digital input is configured with INxMODE=31, this input will proceed in that way. When the drive is switched on, the Command buffer is set automatically started to the corresponding state of the input.

Status	Function	Description
		Remark: Only one of the digital inputs can use the INxMODE=31 function.
IN4MODE=32	Brake	A rising edge at the input triggers the braking output of the servo amplifier. This function is only available while the amplifier is disabled. If an error message is active, the brake cannot be de-energized. Warning ! With suspended loads, this function will lead to slipping of the axes !
IN4MODE=33	see 30	Different from the functionality 30, the resulting answers of the commands are not suppressed, but are send to the seriell communication channel RS232.
IN4MODE=34	see 31	Different from the functionality 31, the resulting answers of the commands are not suppressed, but are send to the seriell communication channel RS232.
IN4MODE=35	Select Velocity/ Current Entry	A positive edge on the digital input causes a takeover of the corresponding VCT entry (see command VCT). The number of the VCT entry is defined by the digital inputs configured with mode 9.
IN4MODE=36	Give Offset to Gearing Function	Gearing mode 4. A high signal on the digital input configured with this INxMODE adds a difference velocity to the gearing. This allows a simple synchronisation of two axes. The difference velocity is given to IN4TRIG [▸ 141]. The scaling is in 20Bit per revolution every 250µs. The difference velocity (n) must be known, then the IN4TRIG [▸ 141] can be calculated: **IN4TRIG** = n * 250 / (60 * rpm) e.g. n = 50 rpm IN4TRIG [▸ 141] = 208
IN4MODE=37	Change source of the actual position at EXTPOS=1.	= 0 Actual position is generated by the external encoder selected by GEARMODE [▸ 213] = 1 Actual position is generated by the first feedback device (resolver od high resolution feedback EnDAT or Hiperface)
IN4MODE=38		Definition of a motion task with following motion tasks. If INxMODE=15 is used (start of an following motion task via I/O), IN4MODE=38 can be used, to have an additional enable for the start of the following motion tasks. Means, that the following motion task is started, if once a rising edge on digital input 1 was detected and then the INxMODE=15 input is enabled to start the following motion task.
IN4MODE=39	Constant velocity for defined time	This function starts a constant velocity for a defined time. The parameters for velocity and time are given by IN4TRIG [▸ 141]. The velocity is given by the lower 16 bit (scaling by VUNIT [▸ 342]) and the time by the upper 16 bit (given in msec) of the help variable IN4TRIG [▸ 141]. A rising edge at INPUTx changes the OPMODE [▸ 50] to 0 (digital velocity) and gives the velocity that is given by IN4TRIG [▸ 141]. After the defined time or a falling edge at INPUTx is detected, the digital velocity setpoint is set to "0". After the actual velocity has reached "0" the OPMODE [▸ 50] is automatically switched back to the old one.

Status	Function	Description
		Example for defining the help variable IN4TRIG [▶ 141] 1. Velocity = 1000 rpm time = 10 sec = 10000 msec IN4TRIG [▶ 141] = 0x271003E8 = 655361000 2. Velocity = -500 rpm time = 10 msec IN4TRIG [▶ 141] = 0x000afe0c = 720396
IN4MODE=40	Additional hardware enable	The digital input works as an additional hardware input. Only if this input has a high signal, the power stage is enabled. This Function can be used by several inputs. In this case, the inputs are configured in series. All inputs have to be high to enable the power stage. (Starting firmware 4.91)
IN4MODE=41	Fast emergency stop	If the input is going to low, the drive stops the motor using the DECSTOP [▶ 330] ramp. If zero velocity is reached (V [▶ 31]<VEL0 [▶ 338]), the power stage is disabled. While stopping the motor the bit 24 (0x01000000) in TRJSTAT [▶ 185] is set. The input is read in the 250µs task.
IN4MODE=42	Activate/deactivate electronic gearing	Activate/deactivate electronic gearing in OPMODE [▶ 50] = 4. This function is practical only with slave axis. A rising edge on the digital input starts the motion from 0 to the master speed and a falling edge changes the speed from master speed to 0. The ramp times can be set by ACCR [▶ 247] for the acceleration and DECR [▶ 249] for deceleration time (stating with firmware 5.51).
IN4MODE=43	Activate/deactivate electronic gearing with position latch	Activate/deactivate electronic gearing in OPMODE [▶ 50] = 4. This function is practical only with slave axis. A rising edge on the digital input starts the motion from 0 to the master speed and a falling edge changes the speed from master speed to 0. The ramp times can be set by ACCR [▶ 247] for the acceleration and DECR [▶ 249] for deceleration time In contrast to IN4MODE = 42, the master position is latched at the rising edge of the input and the position delay caused by the ramp is compensated. IN4TRIG [▶ 141] gives the possibility to add an position offset (in PGEARI [▶ 283] units) to the latched position(starting with Firmware 5.51).

4.7.11 IN4TRIG

ASCII - Command	**IN4TRIG**			
Syntax Transmit	IN4TRIG [Data]			
Syntax Receive	IN4TRIG <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer32		**CANBus Object Number**	356C (hex)
DIM	-		**PROFIBUS PNU**	1708 (dec) IND = 0000xxxx (bin)
Range	long int		**DPR**	108 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.8
Function Group	Digital I/O		**EEPROM**	Yes
Short Description	Variable for IN4MODE			

Description

Auxiliary trigger variable for IN4MODE [▸ 135]. Certain settings of IN4MODE [▸ 135] require you to specify an additional trigger level. See IN4MODE [▸ 135] for further details.

4.7.12 INHCMD

ASCII - Command	**INHCMD**			
Syntax Transmit	INHCMD [Data]			
Syntax Receive	INHCMD <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	String		**CANBus Object Number**	No
DIM	-		**PROFIBUS PNU**	No
Range	-		**DPR**	No
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.67			
Configuration	No		**Revision**	1.3
Function Group	Digital I/O		**EEPROM**	-
Short Description	Command buffer for high level			

Description

The command INHCMD can be used to define an ASCII command sequence. This command sequence will always be carried out when a rising edge is detected at the input that has been configured with the function INxMODE [▸ 116]=30,33

A command sequence consists of individual ASCII commands, separated by a semicolon (;) The maximum length of this command sequence is 56 characters.

Example:

INHCMD GV [▸ 332] 10; GVTN [▸ 336] 15

If a LOW/HIGH edge is detected, the gain of the velocity control loop is set to 10 and the integral action time is set to 15 msec.

4.7.13 INHCMDX

ASCII - Command	**INHCMDX**			
Syntax Transmit	INHCMDX [Data]			
Syntax Receive	INHCMDX <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	String		**CANBus Object Number**	No
DIM	-		**PROFIBUS PNU**	No
Range	-		**DPR**	No
Default	ECHO			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.67			
Configuration	No		**Revision**	1.3
Function Group	Digital I/O		**EEPROM**	Yes
Short Description	Command buffer for high level (INxMODE=31,34)			

Description

The command INHCMDX can be used to define an ASCII command sequence. This command sequence will always be carried out when a rising edge is detected at the input that has been configured with the function INxMODE [▸ 116]=31,34
A command sequence consists of individual ASCII commands, separated by a semicolon (;) The maximum length of this command sequence is 56 characters.

Example

INHCMDX GV [▸ 332] 10; GVTN [▸ 336] 15

If a LOW/HIGH edge is detected, the gain of the velocity control loop is set to 10 and the integral action time is set to 15 msec.

4.7.14 INLCMD

ASCII - Command	**INLCMD**			
Syntax Transmit	INLCMD [Data]			
Syntax Receive	INLCMD <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	String		**CANBus Object Number**	No
DIM	-		**PROFIBUS PNU**	No
Range	-		**DPR**	No
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.67			
Configuration	No		**Revision**	1.3
Function Group	Digital I/O		**EEPROM**	-
Short Description	Command buffer for low level			

Description

The command INLCMD can be used to define an ASCII command sequence. This command sequence will always be carried out when a falling edge is detected at the input that has been configured with the function INxMODE [▶ 116]=30,33 A command sequence consists of individual ASCII commands, separated by a semicolon (;) The maximum length of this command sequence is 56 characters.

Example

INLCMD GV [▶ 332] 5; GVTN [▶ 336] 10

If a HIGH/LOW edge is detected, the gain of the velocity control loop is set to 5 and the integral action time is set to 10 msec.

4.7.15 INLCMDX

ASCII - Command	**INLCMDX**			
Syntax Transmit	INLCMD [Data]			
Syntax Receive	INLCMDX <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	String		**CANBus Object Number**	No
DIM	-		**PROFIBUS PNU**	No
Range	-		**DPR**	No
Default	ECHO			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.67			
Configuration	No		**Revision**	1.3
Function Group	Digital I/O		**EEPROM**	Yes
Short Description	Command buffer for low level (INxMODE=31,34)			

Description

The command INLCMDX can be used to define an ASCII command sequence. This command sequence will always be carried out when a falling edge is detected at the input that has been configured with the function INxMODE [▶ 116]=31,34 A command sequence consists of individual ASCII commands, separated by a semicolon (;) The maximum length of this command sequence is 56 characters.

Example

INLCMDX GV [▶ 332] 5; GVTN [▶ 336] 10

If a HIGH/LOW edge is detected, the gain of the velocity control loop is set to 5 and the integral action time is set to 10 msec.

4.7.16 INS0

ASCII - Command	**INS0**			
Syntax Transmit	INS0			
Syntax Receive	INS0 <Data>		**Available in**	
Type	Variable ro		**MMI**	No
ASCII Format	Integer8		**CANBus Object Number**	36BE (hex)
DIM	-		**PROFIBUS PNU**	2046 (dec) IND = 0000xxxx (bin)
Range	0, 1		**DPR**	446 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	-		**Weightning**	
Start Firmware	5.41			
Configuration	No		**Revision**	1.9
Function Group	Digital I/O		**EEPROM**	No
Short Description	State of Input A0 of the I/O Option Card			

Description

INS0 is used to read input A0 (terminal 1) at the I/O option card. This input is normally used to select a motion task, but can be reassigned for general purpose mode by setting IO11IN [▸ 149] = 2.

4.7.17 INS1

ASCII - Command	**INS1**			
Syntax Transmit	INS1			
Syntax Receive	INS1 <Data>		**Available in**	
Type	Variable ro		**MMI**	No
ASCII Format	Integer8		**CANBus Object Number**	36BF (hex)
DIM	-		**PROFIBUS PNU**	1647 (dec) IND = 0001xxxx (bin)
Range	0, 1		**DPR**	447 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	-		**Weightning**	
Start Firmware	5.41			
Configuration	No		**Revision**	1.9
Function Group	Digital I/O		**EEPROM**	No
Short Description	State of Input A1 of the I/O Option Card			

Description

INS1 is used to read input A1 (terminal 2 of X11A) at the I/O option card. This input is normally used to select a motion task but can be reassigned for general purpose mode by setting IO11IN [▸ 149] = 2.

4.7.18 INS2

ASCII - Command	**INS2**		
Syntax Transmit	INS2		
Syntax Receive	INS2 <Data>	**Available in**	
Type	Variable ro	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	36C0 (hex)
DIM	-	**PROFIBUS PNU**	1648 (dec) IND = 0001xxxx (bin)
Range	0, 1	**DPR**	448 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	5.41		
Configuration	No	**Revision**	1.9
Function Group	Digital I/O	**EEPROM**	No
Short Description	State of Input A2 of the I/O Option Card		

Description

INS2 is used to read input A2 (terminal 3 of X11A) at the I/O option card. This input is normally used to select a motion task but can be reassigned for general purpose mode by setting IO11IN [▶ 149] = 2.

4.7.19 INS3

ASCII - Command	**INS3**		
Syntax Transmit	INS3		
Syntax Receive	INS3 <Data>	**Available in**	
Type	Variable ro	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	36C1 (hex)
DIM	-	**PROFIBUS PNU**	1649 (dec) IND = 0001xxxx (bin)
Range	0, 1	**DPR**	449 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	5.41		
Configuration	No	**Revision**	1.9
Function Group	Digital I/O	**EEPROM**	No
Short Description	State of Input A3 of the I/O Option Card		

Description

INS3 is used to read input A3 (terminal 4 of X11A) at the I/O option card. This input is normally used to select a motion task but can be reassigned for general purpose mode by setting IO11IN [▶ 149] = 2.

4.7.20 INS4

ASCII - Command	INS4			
Syntax Transmit	INS4			
Syntax Receive	INS4 <Data>		Available in	
Type	Variable ro		MMI	No
ASCII Format	Integer8		CANBus Object Number	36C2 (hex)
DIM	-		PROFIBUS PNU	1650 (dec) IND = 0001xxxx (bin)
Range	0, 1		DPR	450 (dec)
Default	-			
Opmode	All		Data Type Bus/DPR	Integer8
Drive State	-		Weightning	
Start Firmware	5.41			
Configuration	No		Revision	1.9
Function Group	Digital I/O		EEPROM	No
Short Description	State of Input A4 of the I/O Option Card			

Description

INS4 is used to read input A4 (terminal 5 of X11A) at the I/O option card. This input is normally used to select a motion task but can be reassigned for general purpose mode by setting IO11IN [▶ 149] = 2.

4.7.21 INS5

ASCII - Command	INS5			
Syntax Transmit	INS5			
Syntax Receive	INS5 <Data>		Available in	
Type	Variable ro		MMI	No
ASCII Format	Integer8		CANBus Object Number	36C3 (hex)
DIM	-		PROFIBUS PNU	1651 (dec) IND = 0001xxxx (bin)
Range	0, 1		DPR	451 (dec)
Default	-			
Opmode	All		Data Type Bus/DPR	Integer8
Drive State	-		Weightning	
Start Firmware	5.41			
Configuration	No		Revision	1.9
Function Group	Digital I/O		EEPROM	No
Short Description	State of Input A5 of the I/O Option Card			

Description

INS5 is used to read input A5 (terminal 6 of X11A) at the I/O option card. This input is normally used to select a motion task but can be reassigned for general purpose mode by setting IO11IN [▶ 149] = 2.

4.7.22 INS6

ASCII - Command	**INS6**		
Syntax Transmit	INS6		
Syntax Receive	INS6 <Data>	**Available in**	
Type	Variable ro	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	36C4 (hex)
DIM	-	**PROFIBUS PNU**	1652 (dec) IND = 0001xxxx (bin)
Range	0, 1	**DPR**	452 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	5.41		
Configuration	No	**Revision**	1.9
Function Group	Digital I/O	**EEPROM**	No
Short Description	State of Input A6 of the I/O Option Card		

Description

INS6 is used to read input A6 (terminal 7 of X11A) at the I/O option card. This input is normally used to select a motion task but can be reassigned for general purpose mode by setting IO11IN [▶ 149] = 2

4.7.23 INS7

ASCII - Command	**INS7**		
Syntax Transmit	INS7		
Syntax Receive	INS7 <Data>	**Available in**	
Type	Variable ro	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	36C5 (hex)
DIM	-	**PROFIBUS PNU**	1653 (dec) IND = 0001xxxx (bin)
Range	0, 1	**DPR**	453 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	5.41		
Configuration	No	**Revision**	1.9
Function Group	Digital I/O	**EEPROM**	No
Short Description	State of Input A7 of the I/O Option Card		

Description

INS7 is used to read input A7 (terminal 8 of X11A) at the I/O option card. This input is normally used to select a motion task but can be reassigned for general purpose mode by setting IO11IN [▶ 149] = 2.

4.7.24 INS8

ASCII - Command	**INS8**		
Syntax Transmit	INS8		
Syntax Receive	INS8 <Data>	**Available in**	
Type	Variable ro	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	36C6 (hex)
DIM	-	**PROFIBUS PNU**	1654 (dec) IND = 0001xxxx (bin)
Range	0, 1	**DPR**	454 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	5.41		
Configuration	No	**Revision**	1.9
Function Group	Digital I/O	**EEPROM**	No
Short Description	State of FSTART_IO of the I/O Option Card		

Description

INS8 is used to read input FSTART_IO (terminal 2 of X11B) at the I/O option card. This input is normally used to start a motion task but can be reassigned for general purpose mode by setting IO11IN [▶ 149] = 2.

4.7.25 IO11A

ASCII - Command	**IO11A**		
Syntax Transmit	IO11A [Data]		
Syntax Receive	IO11A <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	3677 (hex)
DIM	-	**PROFIBUS PNU**	1975 (dec) IND = 0000xxxx (bin)
Range	0,1	**DPR**	375 (dec)
Default	0		
Opmode	8	**Data Type Bus/DPR**	Integer8
Drive State	Disabled + Reset (Coldstart)	**Weightning**	
Start Firmware	3.42		
Configuration	Yes	**Revision**	1.8
Function Group	Digital I/O	**EEPROM**	Yes
Short Description	Behaivior of the start input at the I/O expansion		

Description

This variable defines the behaivior of the start input at the I/O expansion card (terminal X11B / 2).

IO11A=0 A negative edge generates a stop command. An active motion task is stopped.

IO11A=1 A positive edge generates no stop command. It is possible to switch over from one motion task to another without stop.

E.g.:

1. Select motion task 1 (connector X11A/1-8)
2. A rising edge at start input starts motion task 1.

3. Switch start input to low. Motion task 1 is executed.
4. Select motion task 2
5. A rising edge starts motion task 2 without stop.

Remark

This setting disables the stop command at the state low of the start. To be able to stop a motion task, you have to select another input of the standard I/O for emergency stop or start/stop (see command INxMODE [▶ 116])

Also see about this

- STOP [▶ 297]

4.7.26 IO11IN

ASCII - Command	IO11IN			
Syntax Transmit	IO11IN [Data]			
Syntax Receive	IO11IN <Data>		Available in	
Type	Variable rw		MMI	No
ASCII Format	Integer8		CANBus Object Number	36CC (hex)
DIM	-		PROFIBUS PNU	1660 (dec) IND = 0001xxxx (bin)
Range	0, 1, 2		DPR	460 (dec)
Default	0			
Opmode	All		Data Type Bus/DPR	Integer8
Drive State	Disbale + Reset (Coldstart)		Weightning	
Start Firmware	5.41			
Configuration	Yes		Revision	1.9
Function Group	Digital I/O		EEPROM	Yes
Short Description	Functionality of the Inputs of the I/O Option Board			

Description

The I/O option card has 9 input pins which are normally used to select and begin a motion task (terminal 1-8 of X11A select the task, terminal 2 of X11B starts the move). IO11IN is used with the I/O option card to reassign the Motion Block Number and StartMove Input for general purpose use. This allows for up to 13 general purpose inputs (IN1 [▶ 23]-4 and INS0 [▶ 144]-8) that can be used through standard communication channels or through motion tasking.

Usage: IO11IN <mode>

mode:(The discription is used for the contacts X11A of the I/O optioncard)

- 0 All contacts (A0 ... A7) are used to adress a motion task
- 1 Contacts A0 ... A3 are used for adressing motion tasks, contacts A4 ... A7 are used for graphical motion tasks
- 2 contacts A4 ... A7 are used for graphical motion tasks

4.7.27 O1

ASCII - Command	O1		
Syntax Transmit	O1 [data]		
Syntax Receive	O1	Available in	
Type	Variable rw	MMI	Yes
ASCII Format	Integer8	CANBus Object Number	35AE (hex)
DIM	-	PROFIBUS PNU	1774 (dec) IND = 0000xxxx (bin)
Range	-	DPR	174 (dec)
Default	-		
Opmode	All	Data Type Bus/DPR	Integer8
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.3
Function Group	Digital I/O	EEPROM	No
Short Description	State of Digital Output 1		

Description

The O1 command returns the state of the digital output 1 (0 = Low,1 = High).
If no function is assigned to digital output 1 (O1MODE [▶ 150]=0), then the High/Low state can be given out at output 1 by using the command □O1 1□ or □O1 0□.

4.7.28 O1MODE

ASCII - Command	O1MODE		
Syntax Transmit	O1MODE [Data]		
Syntax Receive	O1MODE <Data>	Available in	
Type	Variable rw	MMI	Yes
ASCII Format	Integer8	CANBus Object Number	35AF (hex)
DIM	-	PROFIBUS PNU	1775 (dec) IND = 0000xxxx (bin)
Range	0 .. 50	DPR	175 (dec)
Default	0		
Opmode	All	Data Type Bus/DPR	Integer8
Drive State	Disabled + Reset (Coldstart)	Weightning	
Start Firmware	1.20		
Configuration	Yes	Revision	2.0
Function Group	Digital I/O	EEPROM	Yes
Short Description	Function of Digital Output 1		

Description

The O1 command returns the state of the digital output 1 (0 = Low,1 = High).
If no function is assigned to digital output 1 (O1MODE=0), then the High/Low state can be given out at output 1 by using the command □O1 [▶ 150] 1□ or □O1 [▶ 150] 0□.

Status	Function	Description
O1MODE=0	Off	OFF The state of the digital output 1 can be set/cleared by RS232/Fieldbus or Slot board

Status	Function	Description
O1MODE=1	v_act<O1TRIG [▶ 155]	If the absolute value for the motor velocity is lower than a preset value (O1TRIG [▶ 155]), a HIGH-signal will be output. After the function has been selected you can enter the velocity in rpm in O1TRIG [▶ 155]. The ouput is high, if V [▶ 31] < O1TRIG [▶ 155] and becomes low, if V [▶ 31] > O1TRIG [▶ 155] + 0.01 * MSPEED [▶ 235].
O1MODE=2	v_act>O1TRIG [▶ 155]	If the absolute value for the motor velocity is higher than a preset value (O1TRIG [▶ 155]), a HIGH-signal will be output. After the function has been selected you can enter the velocity in rpm in O1TRIG [▶ 155]. The ouput is high, if V [▶ 31] > O1TRIG [▶ 155] and becomes low, if V [▶ 31] < O1TRIG [▶ 155] + 0.01 * MSPEED [▶ 235].
O1MODE=3	Mains-RTO	This signals the operational readiness of the SERVOSTAR□ 600 power output stage. After switching on the mains supply, a HIGH-signal is output until the DC-link circuit is fully charged up. A LOW-signal is output when the charging of the DC-link circuit is finished. If the DC-link voltage falls below 100V, then a HIGH-signal will be output. The □Undervoltage□ monitoring is inactive. O1MODE=3 only works, if NONBTB [▶ 179] = 0.
O1MODE=4	Regen off	Signals if the preset regen power (screen page □Basic Setup□) is exceeded.
O1MODE=5	Sw_limit	This produces a HIGH-signal if a software limit-switch is reached (a preset function of the corresponding position register, set to □SW limit-switch 1□ or □SW limit-switch 2□ □ the function is defined in the screen page □Position□).
O1MODE=6	Pos.>x	If the position (angular position of the motor shaft) exceeds a preset value (O1TRIG [▶ 155]), a HIGH-signal will be output. After the function has been selected, you can enter the signaling posi-tion in increment (a number or fraction of motor turns N) as O1TRIG [▶ 155]. Make the calculation according to the following equation: x = 1048576 *N *Inkr. Maximum possible entry value:x =2 31 = 2147483648, this corresponds to N = 2048
O1MODE=7	InPos	When the target position for a motion task has been reached (the InPosition window PEINPOS [▶ 282]), this is signaled by the output of a HIGH-signal. A cable break will not be detected. The width of the InPosition window for all the valid motion tasks is entered in the □Position data□ screen page. If a sequence of motion tasks is performed one after another, then the signal for reaching the final position of the motion-task sequence will be output (target position of the last motion task). Signaling that the target position of each motion task has been reached, in a sequence of motion tasks, can be achieved with the function □16, Next_InPos□.
O1MODE=8	I_act<O1TRIG [▶ 155]	The output is a HIGH-signal, as long as the absolute r.m.s. value of the actual current is lower than a defined value in mA (O1TRIG [▶ 155]). After the function has been selected, you can enter the current value as O1TRIG [▶ 155] in mA.
O1MODE=9	I_act>O1TRIG [▶ 155]	The output is a HIGH-signal, as long as the absolute r.m.s. value of the actual current is higher than a defined value in mA (O1TRIG [▶ 155]). After the function has been selected, you can enter the current value as O1TRIG [▶ 155] in mA.

Status	Function	Description
O1MODE=10	Ferror	If the position goes outside the preset contouring-error window, this is indicated by a LOW-signal. The width of the contouring-error window (PEMAX [▸ 282]) is entered in the screen page Position for all the valid motion tasks.
O1MODE=11	I2T	If the preset I2T monitoring threshold (I2TLIM [▸ 107]) is reached (screen page Current) this is indicated by a HIGH-signal.
O1MODE=12	Posreg.1	The preset function of the corresponding position register 1 (the function is defined in the screen page Position) is indicated by a HIGH-signal.
O1MODE=13	Posreg.2	The preset function of the corresponding position register 2 (the function is defined in the screen page Position) is indicated by a HIGH-signal.
O1MODE=14	Posreg.3	The preset function of the corresponding position register 3 (the function is defined in the screen page Position) is indicated by a HIGH-signal.
O1MODE=15	Posreg.4	The preset function of the corresponding position register 4 (the function is defined in the screen page Position) is indicated by a HIGH-signal.
O1MODE=16	Next-InPos	The start of each motion task in an automatically executed sequence of motion tasks is signalled by an inversion of the output signal. The output produces a Low signal at the start of the first motion task of the motion task sequence. The type of message can be set by IN2PM [▸ 258]
O1MODE=17	Error/Warn	The output produces a HIGH-signal if an error or a warning message is signaled by the servo amplifier. A list of the error messages can be found under ERRCODE [▸ 75].
O1MODE=18	Error	The output produces a HIGH-signal if an error is signaled by the servo amplifier. A list of the error messages can be found under ERRCODE [▸ 75]
O1MODE=19	DC_Link>O1TRIG [▸ 155]	A HIGH-signal is output if the actual value of the DC-link voltage is higher than a defined value in volts (O1TRIG [▸ 155]). After the function has been selected, you can enter the voltage value as O1TRIG [▸ 155] in Volt..
O1MODE=20	DC_Link > O1TRIG [▸ 155]	A HIGH-signal is output if the actual value of the DC-link voltage is lower than a defined value in volts (O1TRIG [▸ 155]). After the function has been selected, you can enter the voltage value in O1TRIG [▸ 155] in Volt.
O1MODE=21	ENABLE	A HIGH-signal is output if the servo amplifier is enabled. To obtain the enable, the external Hardware Enable signal must be present, the Enable status must be set in the setup software (or via the fieldbus interface) and no errors must be present that would cause an automatic internal disabling of the servo amplifier. If function OxMODE [▸ 150] = is selected, the enable signal is high, if the line voltage is applied and the charging of the link capacitors is finished. The drive is disabled, if the DC-link voltage wents under the threshold VBUSMIN [▸ 54].
O1MODE=22	Zero_pulse	The zero mark/pulse (HIGH-signal) is indicated by the encoder-emulation. This function is only use-ful at very low velocities.

Status	Function	Description
		Vmax = 15000 / ENCOUT [▶ 315] e.g. ENCOUT [▶ 315]=256 Pulses/Rev Vmax = 58 rpm
O1MODE=23	Slot-DPR	This configuration enables the possibility to output a state from the Slot board (mem DPR Slot Board Offset 0x3E4). If no Slot board is available, this configuration is equal to OxMODE=0. If a Device-Net option board is plugged in the drive, this setting enables access of Device-Net to digital output 1
O1MODE=24	Ref_OK	The output signals High, if a reference point is available. Reference traverse (homing) has been carried out, or a reference point has been set. (see NREF [▶ 267])
O1MODE=25	Reserved	
O1MODE=26	Reserved	
O1MODE=27	Reserved	
O1MODE=28	Posreg.0	The preset function of the corresponding position register 0 is indicated by a HIGH-signal. Valid only with expansion card -I/0-14/08- .
O1MODE=29	Posreg.5	The preset function of the corresponding position register 5 is indicated by a HIGH-signal. Valid only with expansion card -I/0-14/08- .
O1MODE=30	OR-Operation of all Posreg.	If one of the position registers 0...5 indicates high, the output is high, otherwise low.
O1MODE=31	Analog SW1< O1TRIG [▶ 155]	The output is high, if the analog input voltage at SW1 is lower than the threshold O1TRIG [▶ 155]. The auxiliary variable O1TRIG [▶ 155] is given in mV (with sign).
O1MODE=32	Analog SW1> O1TRIG [▶ 155]	The output is high, if the analog input voltage at SW1 is higher than the threshold O1TRIG [▶ 155]. The auxiliary variable O1TRIG [▶ 155] is given in mV (with sign).
O1MODE=33	Analog SW2 < O1TRIG [▶ 155]	The output is high, if the analog input voltage at SW2 is lower than the threshold O1TRIG [▶ 155]. The auxiliary variable O1TRIG [▶ 155] is given in mV (with sign).
O1MODE=34	Analog SW2 > O1TRIG [▶ 155]	The output is high, if the analog input voltage at SW2 is higher than the threshold O1TRIG [▶ 155]. The auxiliary variable O1TRIG [▶ 155] is given in mV (with sign).
O1MODE=35	Internal Enable	The state of the internal ENABLE signal is mirrored on the digital output. If the setiings are: MBRAKE [▶ 226]=0, STOPMODE [▶ 85]=0 and ACTFAULT [▶ 34]=0, the function is similar to O1MODE [▶ 150]=21. If one of the three variables is "1", the output changes to low, when the drives starts to dec to "0". If OPMODE [▶ 50]=21, the output is low, if the drive has stopped and has disabled the output stage.
O1MODE=36	Logical OR: DRVSTAT [▶ 171] - O1TRIG [▶ 155]	This function makes a OR operation between the Bit-variable DRVSTAT [▶ 171] and a Bit mask given by O1TRIG [▶ 155]. The result is present at output 1.
O1MODE=37	Logical AND: DRVSTAT [▶ 171] -O1TRIG [▶ 155]	This function makes a AND operation between the Bit-variable DRVSTAT [▶ 171] and a Bit mask given by O1TRIG [▶ 155]. The result is present at output 1.

Status	Function	Description
O1MODE=38	Logical OR: TRJSTAT [▶ 185] - O1TRIG [▶ 155]	This function makes a OR operation between the Bit-variable TRJSTAT [▶ 185] and a Bit mask given by O1TRIG [▶ 155]. The result is present at output 1.
O1MODE=39	Logical AND: TRJSTAT [▶ 185] - O1TRIG [▶ 155]	This function makes a AND operation between the Bit-variable TRJSTAT [▶ 185] and a Bit mask given by O1TRIG [▶ 155]. The result is present at output 1.
O1MODE=40	Logical OR: POSRSTAT [▶ 286] - O1TRIG [▶ 155]	This function makes a OR operation between the Bit-variable POSRSTAT [▶ 286] and a Bit mask given by O1TRIG [▶ 155]. The result is present at output 1.
O1MODE=41	Logical AND: POSRSTAT [▶ 286] - O1TRIG [▶ 155]	This function makes a AND operation between the Bit-variable POSRSTAT [▶ 286] and a Bit mask given by O1TRIG [▶ 155]. The result is present at output 1.
O1MODE=42		This function enables a temperature warning. If one of the three internal measured temperatures reaches the trip level, the digital output ist set to high. After the selected delay time given in O1TRIG [▶ 155] the drive generates a error message and disables the output stage. The delay time has the range 0...30000 msec and effects following temperatures: Motor temperature TEMPM [▶ 31], threshold MAXTEMPM [▶ 80] Heatsink temperature TEMPH [▶ 30], Threshold MAXTEMPH [▶ 79] Ambient temperature TEMPE [▶ 30], threshold MAXTEMPE [▶ 79] Start Firmware 3.41
O1MODE=43	The sign of the actual velocity	OUTPUT1 = 1 V < -VEL0 [▶ 338] OUTPUT1 = 0 V > -VEL0 [▶ 338] This function is available starting firmware version 4.01.
O1MODE=44	Velocity In-Position (active high)	The output 1 is set, if the absolute of the difference between the internal velocity command and the actual velocity is smaller than O1TRIG [▶ 155]. The size of the window (O1TRIG [▶ 155]) is given in valid velocity units (VUNIT [▶ 342]). (4.30)
O1MODE=45	Velocity In-Position (aktive low)	The output 1 is set, if the absolute of the difference between the internal velocity command and the actual velocity is bigger than O1TRIG [▶ 155]. The size of the window (O1TRIG [▶ 155]) is given in valid velocity units (VUNIT [▶ 342]). (4.30)
O1MODE=46	Current in Window (low active)	The digital output 1 is set, if the absolute of the difference between current command and actual value is smaller than the window, defined by O1TRIG [▶ 155]. The window is given in mA.
O1MODE=47	Current not in Window (low active)	The digital output 1 is set, if the absolute of the difference between current command and actual value is greater than the window, defined by O1TRIG [▶ 155]. The window is given in mA.

Status	Function	Description
O1MODE=48	Logical NOR: DRVSTAT [▸ 171] - O1TRIG [▸ 155]	This function makes a inverted OR operation between the Bit-variable DRVSTAT [▸ 171] and a Bit mask given by O1TRIG [▸ 155]. The result is present at output 1. (starting with 4.92)
O1MODE=49	Logical NAND: DRVSTAT [▸ 171] -O1TRIG [▸ 155]	This function makes a inverted AND operation between the Bit-variable DRVSTAT [▸ 171] and a Bit mask given by O1TRIG [▸ 155]. The result is present at output 1. (starting with 4.92)
O1MODE=50	Logical NOR: TRJSTAT [▸ 185] - O1TRIG [▸ 155]	This function makes a inverted OR operation between the Bit-variable TRJSTAT [▸ 185] and a Bit mask given by O1TRIG [▸ 155]. The result is present at output 1. (starting with 4.92)
O1MODE=51	Logical NAND: TRJSTAT [▸ 185] - O1TRIG [▸ 155]	This function makes a AND operation between the Bit-variable TRJSTAT [▸ 185] and a Bit mask given by O1TRIG [▸ 155]. The result is present at output 1. (starting with 4.92)

Also see about this

- PBALMAX [▸ 82]
- PBALRES [▸ 82]
- SWCNFG [▸ 297]
- PGEARI [▸ 283]
- PGEARO [▸ 284]

4.7.29 O1TRIG

ASCII - Command	O1TRIG		
Syntax Transmit	O1TRIG [Data]		
Syntax Receive	O1TRIG <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	35B0 (hex)
DIM	-	**PROFIBUS PNU**	1776 (dec) IND = 0000xxxx (bin)
Range	long int	**DPR**	176 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Digital I/O	**EEPROM**	Yes
Short Description	Auxiliary Variable for O1MODE		

Description

The function of the auxiliary variable O1TRIG depends on the configuration of O1MODE [▸ 150].

4.7.30 O2

ASCII - Command	O2		
Syntax Transmit	O2 [Data]		
Syntax Receive	O2 <Data>	Available in	
Type	Variable rw	MMI	Yes
ASCII Format	Integer8	CANBus Object Number	35B1 (hex)
DIM	-	PROFIBUS PNU	1777 (dec) IND = 0000xxxx (bin)
Range	-	DPR	177 (dec)
Default	-		
Opmode	All	Data Type Bus/DPR	Integer8
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.3
Function Group	Digital I/O	EEPROM	No
Short Description	State of Digital Output 2		

Description

The O2 command returns the state of the digital output 2 (0 = Low,1 = High).
If no function is assigned to digital output 2 (O2MODE [▶ 156]=0), then the High/Low state can be given out at output 2 by using the command □O2 [▶ 156] 1□ or □O2 [▶ 156] 0□.

4.7.31 O2MODE

ASCII - Command	O2MODE		
Syntax Transmit	O2MODE [Data]		
Syntax Receive	O2MODE <Data>	Available in	
Type	Variable rw	MMI	Yes
ASCII Format	Integer8	CANBus Object Number	35B2 (hex)
DIM	-	PROFIBUS PNU	1778 (dec) IND = 0000xxxx (bin)
Range	0 .. 50	DPR	178 (dec)
Default	0		
Opmode	All	Data Type Bus/DPR	Integer8
Drive State	Disabled + Reset (Coldstart)	Weightning	
Start Firmware	1.20		
Configuration	Yes	Revision	2.0
Function Group	Digital I/O	EEPROM	Yes
Short Description	Function of Digital Output 2		

Description

The O2MODE command can be used to configure the function of the digital output OUT2. The amplifier must be switched off and then on again after an alteration of this parameter.

The following functions can be configured

Status	Function	Description
O2MODE=0	Off	OFF The state of the digital output 2 can be set/cleared by RS232/Fieldbus or Slot board
O2MODE=1	v_act<O1TRIG [▶ 155]	If the absolute value for the motor velocity is lower than a preset value (O2TRIG [▶ 161]), a HIGH-signal will be output. After the function has been selected you can enter the velocity in rpm in O2TRIG [▶ 161]. The ouput is high, if V [▶ 31] < O2TRIG [▶ 161] and becomes low, if V [▶ 31] > O2TRIG [▶ 161] + 0.01 * MSPEED [▶ 235].
O2MODE=2	v_act>O1TRIG [▶ 155]	If the absolute value for the motor velocity is higher than a preset value (O2TRIG [▶ 161]), a HIGH-signal will be output. After the function has been selected you can enter the velocity in rpm in O2TRIG [▶ 161]. The ouput is high, if V [▶ 31] > O2TRIG [▶ 161] and becomes low, if V [▶ 31] < O2TRIG [▶ 161] + 0.01 * MSPEED [▶ 235].
O2MODE=3	Mains-RTO	This signals the operational readiness of the SERVOSTAR□ 600 power output stage. After switch-ing on the mains supply, a LOW-signal is output until the DC-link circuit is fully charged up. A HIGH-signal is output when the charging of the DC-link circuit is finished. If the DC-link voltage falls below 100V, then 0V will be output. The □Undervoltage□ monitoring is inactive. O1MODE=3 only works with NONBTB [▶ 179] = 0.
O2MODE=4	Regen off	Signals if the preset regen power (screen page □Basic Setup□) is exceeded.
O2MODE=5	Sw_limit	This produces a HIGH-signal if a software limit-switch is reached (a preset function of the corresponding position register, set to □SW limit-switch 1□ or □SW limit-switch 2□ □ the function is defined in the screen page □Position□).(see SWCNFG [▶ 297])
O2MODE=6	Pos.>O1TRIG [▶ 155]	If the position (angular position of the motor shaft) exceeds a preset value (O1TRIG [▶ 155]), a HIGH-signal will be output. After the function has been selected, you can enter the signaling posi-tion in increment (a number or fraction of motor turns N) as O1TRIG [▶ 155]. Make the calculation according to the following equation: O1TRIG [▶ 155] = 1048576 *N *Inkr. Maximum possible entry value:x =2 31 = 2147483648, this corresponds to N = 2048
O2MODE=7	InPos	When the target position for a motion task has been reached (the InPosition window), this is signaled by the output of a HIGH-signal. A cable break will not be detected. The width of the InPosition window PEINPOS [▶ 282] for all the valid motion tasks is entered in the □Position data□ screen page. If a sequence of motion tasks is performed one after another, then the signal for reaching the final position of the motion-task sequence will be output (target position of the last motion task). Signaling that the target position of each motion task has been reached, in a sequence of motion tasks, can be achieved with the function □16, Next_InPos□.
O2MODE=8	I_act< O2TRIG [▶ 161]	The output is a HIGH-signal, as long as the absolute r.m.s. value of the actual current is lower than a defined value in mA (O2TRIG [▶ 161]). After the function has been selected, you can enter the current value as O2TRIG [▶ 161].

Status	Function	Description
O2MODE=9	I_act > O2TRIG [▶ 161]	The output is a HIGH-signal, as long as the absolute r.m.s. value of the actual current is higher than a defined value in mA (O2TRIG [▶ 161]). After the function has been selected, you can enter the current value as O2TRIG [▶ 161].
O2MODE=10	Ferror	If the position goes outside the preset contouring-error window (PEMAX [▶ 282]) , this is indicated by a LOW-signal. The width of the contouring-error window is entered in the screen page Position for all the valid motion tasks.
O2MODE=11	I²t	If the preset I2t monitoring threshold I2TLIM [▶ 107] is reached (screen page Current) this is indicated by a HIGH-signal.
O2MODE=12	Posreg.1	The preset function of the corresponding position register 1 (the function is defined in the screen page Position) is indicated by a HIGH-signal.
O2MODE=13	Posreg.2	The preset function of the corresponding position register 2 (the function is defined in the screen page Position) is indicated by a HIGH-signal.
O2MODE=14	Posreg.3	The preset function of the corresponding position register 3 (the function is defined in the screen page Position) is indicated by a HIGH-signal.
O2MODE=15	Posreg.4	The preset function of the corresponding position register 4 (the function is defined in the screen page Position) is indicated by a HIGH-signal.
O2MODE=16	Next-InPos	The start of each motion task in an automatically executed sequence of motion tasks is signalled by an inversion of the output signal. The output produces a Low signal at the start of the first motion task of the motion task sequence. The type of message can be set by IN2PM [▶ 258]
O2MODE=17	Error/Warn	The output produces a HIGH-signal if an error or a warning message is signaled by the servo amplifier. A list of the error messages can be found under ERRCODE [▶ 75].
O2MODE=18	Error	The output produces a HIGH-signal if an error is signaled by the servo amplifier. A list of the error messages can be found under ERRCODE [▶ 75]
O2MODE=19	DC_Link > O2TRIG [▶ 161]	A HIGH-signal is output if the actual value of the DC-link voltage is higher than a defined value in volts (O2TRIG [▶ 161]). After the function has been selected, you can enter the voltage value as O2TRIG [▶ 161].
O2MODE=20	DC_Link > O2TRIG [▶ 161]	A HIGH-signal is output if the actual value of the DC-link voltage is lower than a defined value in volts (O2TRIG [▶ 161]). After the function has been selected, you can enter the voltage value as O2TRIG [▶ 161].
O2MODE=21	ENABLE	A HIGH-signal is output if the servo amplifier is enabled. To obtain the enable, the external Hardware Enable signal must be present, the Enable status must be set in the setup software (or via the fieldbus interface) and no errors must be present that would cause an automatic internal disabling of the servo amplifier. If function OxMODE [▶ 150] = is selected, the enable signal is high, if the line voltage is applied and the charging of the link capacitors is finished. The drive is disabled, if the DC-link voltage wents under the threshold VBUSMIN [▶ 54].
O2MODE=22	Zero_pulse	The zero mark/pulse (HIGH-signal) is indicated by the encoder-emulation. This function is only useful at very low velocities.

Status	Function	Description
		Vmax = 15000 / ENCOUT [▶ 315] e.g. ENCOUT [▶ 315]=256 Pulses/Rev Vmax = 58 rpm
O2MODE=23	Slot-DPR	This configuration enables the possibility to output a state from the Slot board (mem DPR Slot Board Offset 0x3E4). If no Slot board is available, this configuration is equal to O2MODE=0. If a Device-Net option board is plugged in the drive, this setting enables access of Device-Net to digital output 2+
O2MODE=24	Ref_OK	The output signals High, if a reference point is available. Reference traverse (homing) NREF [▶ 267] has been carried out, or a reference point has been set)
O2MODE=25	Reserved	
O2MODE=26	Reserved	
O2MODE=27	Reserved	
O2MODE=28	Posreg. 0	The preset function of the corresponding position register 0 is indicated by a HIGH-signal. Valid only with expansion card -I/0-14/08- .
O2MODE=29	Posreg. 5	The preset function of the corresponding position register 5 is indicated by a HIGH-signal. Valid only with expansion card -I/0-14/08- .
O2MODE=30	OR-Operation of all Posreg.	If one of the position registers 0...5 indicates high, the output is high, otherwise low.
O2MODE=31	Analog SW1 > O2TRIG [▶ 161]	The output is high, if the analog input voltage at SW1 is lower than the threshold O2TRIG [▶ 161]. The auxiliary variable O2TRIG [▶ 161] is given in mV (with sign).
O2MODE=32	Analog SW1 < O2TRIG [▶ 161]	The output is high, if the analog input voltage at SW1 is lower than the threshold O2TRIG [▶ 161]. The auxiliary variable O2TRIG [▶ 161] is given in mV (with sign).
O2MODE=33	Analog SW2 < O2TRIG [▶ 161]	The output is high, if the analog input voltage at SW1 is higher than the threshold O2TRIG [▶ 161]. The auxiliary variable O2TRIG [▶ 161] is given in mV (with sign).
O2MODE=34	Analog SW2 > O2TRIG [▶ 161]	The output is high, if the analog input voltage at SW2 is lower than the threshold O2TRIG [▶ 161]. The auxiliary variable O2TRIG [▶ 161] is given in mV (with sign).
O2MODE=35	Internal Enable	The state of the internal ENABLE signal is mirrored on the digital output. If the setiings are: MBRAKE [▶ 226]=0, STOPMODE [▶ 85]=0 and ACTFAULT [▶ 34]=0, the function is similar to O1MODE=21. If one of the three variables is "1", the output changes to low, when the drives starts to dec to "0". If OPMODE=21, the output is low, if the drive has stopped and has disabled the output stage.
O2MODE=36	Logical OR: DRVSTAT [▶ 171] - O2TRIG [▶ 161]	This function makes a OR operation between the Bit-variable DRVSTAT [▶ 171] and a Bit mask given by O2TRIG [▶ 161]. The result is present at output 2.
O2MODE=37	Logical AND: DRVSTAT [▶ 171] - O2TRIG [▶ 161]	This function makes a AND operation between the Bit-variable DRVSTAT [▶ 171] and a Bit mask given by O2TRIG [▶ 161]. The result is present at output 2.

Status	Function	Description
O2MODE=38	Logical OR: TRJSTAT [▶ 185] - O2TRIG [▶ 161]	This function makes a OR operation between the Bit-variable TRJSTAT [▶ 185] and a Bit mask given by O2TRIG [▶ 161]. The result is present at output 2.
O2MODE=39	Logical AND: TRJSTAT [▶ 185] - O2TRIG [▶ 161]	This function makes a AND operation between the Bit-variable TRJSTAT [▶ 185] and a Bit mask given by O2TRIG [▶ 161]. The result is present at output 2.
O2MODE=40	Logical OR: POSRSTAT [▶ 286] - O2TRIG [▶ 161]	This function makes a OR operation between the Bit-variable POSRSTAT [▶ 286] and a Bit mask given by O2TRIG [▶ 161]. The result is present at output 2.
O2MODE=41	Logical AND: POSRSTAT [▶ 286] - O2TRIG [▶ 161]	This function makes a AND operation between the Bit-variable POSRSTAT [▶ 286] and a Bit mask given by O2TRIG [▶ 161]. The result is present at output 2.
O2MODE=42		This function enables a temperature warning. If one of the three internal measured temperatures reaches the trip level, the digital output ist set to high. After the selected delay time given in O2TRIG [▶ 161] the drive generates a error message and disables the output stage. The delay time has the range 0...30000 msec and effects following temperatures: Motor temperature TEMPM [▶ 31], threshold MAXTEMPM [▶ 80] Heatsink temperature TEMPH [▶ 30], Threshold MAXTEMPH [▶ 79] Ambient temperature TEMPE [▶ 30], threshold MAXTEMPE [▶ 79] Start Firmware 3.41
O2MODE=43	The sign of the actual velocity	OUTPUT2 = 1 V < -VEL0 [▶ 338] OUTPUT2 = 0 V > -VEL0 [▶ 338] This function is available starting firmware version 4.01.
O2MODE=44	Velocity In-Position (active high)	The output 2 is set, if the absolute of the difference between the internal velocity command and the actual velocity is smaller than O2TRIG [▶ 161]. The size of the window (O2TRIG [▶ 161]) is given in valid velocity units (VUNIT [▶ 342]). (4.30)
O2MODE=45	Velocity In-Position (aktive low)	The output 2 is set, if the absolute of the difference between the internal velocity command and the actual velocity is bigger than O2TRIG [▶ 161]. The size of the window (O2TRIG [▶ 161]) is given in valid velocity units (VUNIT [▶ 342]). (4.30)
O2MODE=46	Current in Window (low active)	The digital output 2 is set, if the absolute of the difference between current command and actual value is smaller than the window, defined by O1TRIG [▶ 155]. The window is given in mA.
O2MODE=47	Current not in Window (low active)	The digital output 2 is set, if the absolute of the difference between current command and actual value is greater than the window, defined by O2TRIG [▶ 161]. The window is given in mA.

Status	Function	Description
O2MODE=48	Logical NOR: DRVSTAT [▶ 171] - O2TRIG [▶ 161]	This function makes a inverted OR operation between the Bit-variable DRVSTAT [▶ 171] and a Bit mask given by O2TRIG [▶ 161]. The result is present at output 2. (starting with 4.92)
O2MODE=49	Logical NAND: DRVSTAT [▶ 171] -O2TRIG [▶ 161]	This function makes a inverted AND operation between the Bit-variable DRVSTAT [▶ 171] and a Bit mask given by O2TRIG [▶ 161]. The result is present at output 2. (starting with 4.92)
O2MODE=50	Logical NOR: TRJSTAT [▶ 185] - O2TRIG [▶ 161]	This function makes a inverted OR operation between the Bit-variable TRJSTAT [▶ 185] and a Bit mask given by O2TRIG [▶ 161]. The result is present at output 2. (starting with 4.92)
O2MODE=51	Logical NAND: TRJSTAT [▶ 185] - O2TRIG [▶ 161]	This function makes a AND operation between the Bit-variable TRJSTAT [▶ 185] and a Bit mask given by O2TRIG [▶ 161]. The result is present at output 2. (starting with 4.92)

Also see about this

- PBALMAX [▶ 82]
- PBALRES [▶ 82]
- PGEARI [▶ 283]
- PGEARO [▶ 284]

4.7.32 O2TRIG

ASCII - Command	O2TRIG		
Syntax Transmit	O2TRIG [Data]		
Syntax Receive	O2TRIG <Data>	Available in	
Type	Variable rw	MMI	Yes
ASCII Format	Integer32	CANBus Object Number	35B3 (hex)
DIM	-	PROFIBUS PNU	1779 (dec) IND = 0000xxxx (bin)
Range	long int	DPR	179 (dec)
Default	0		
Opmode	All	Data Type Bus/DPR	Integer32
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.3
Function Group	Digital I/O	EEPROM	Yes
Short Description	Auxiliary Variable for O2MODE		

Description

The function of the auxiliary variable O2TRIG depends on the configuration of O2MODE [▶ 156]

4.7.33 OS1

ASCII - Command	**OS1**		
Syntax Transmit	OS1 [Data]		
Syntax Receive	OS1 <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	36C7 (hex)
DIM	-	**PROFIBUS PNU**	1655 (dec) IND = 0001xxxx (bin)
Range	0, 1	**DPR**	455 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	5.41		
Configuration	No	**Revision**	1.9
Function Group	Digital I/O	**EEPROM**	No
Short Description	Set/Reset of "Posreg1" of the I/O Option Card		

Description

The I/O option card has 5 output pins which are normally used to signal on position registers SWE0 [▸ 301]-4.

OS1 sets the digital output "PosReg1" (terminal 6 of X11B) of the I/O option card.

Set SWCNFG [▸ 297] Bit 0 to "0" to disable the appropriate register (SWE1 [▸ 302]) to reassign the output for use in general purpose mode.

This command allows using the output for general purposes through standard communication channels or through motion tasking.

4.7.34 OS2

ASCII - Command	**OS2**		
Syntax Transmit	OS2 [Data]		
Syntax Receive	OS2 <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	36C8 (hex)
DIM	-	**PROFIBUS PNU**	1656 (dec) IND = 0001xxxx (bin)
Range	0, 1	**DPR**	456 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	5.41		
Configuration	No	**Revision**	1.9
Function Group	Digital I/O	**EEPROM**	No
Short Description	Set/Reset of "Posreg2" of the I/O Option Card		

Description

The I/O option card has 5 output pins which are normally used to signal on position registers SWE0 [▸ 301]-4.

OS2 sets the digital output "PosReg2" (terminal 7 of X11B) of the I/O option card.

Set SWCNFG [▸ 297] Bit 4 to "0" to disable the appropriate register (SWE2 [▸ 303]) to reassign the output for use in general purpose mode.

This command allows using the output for general purposes through standard communication channels or through motion tasking.

4.7.35 OS3

ASCII - Command	**OS3**		
Syntax Transmit	OS3 [Data]		
Syntax Receive	OS3 <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	36C9 (hex)
DIM	-	**PROFIBUS PNU**	1657 (dec) IND = 0001xxxx (bin)
Range	0, 1	**DPR**	457 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	5.41		
Configuration	No	**Revision**	1.9
Function Group	Digital I/O	**EEPROM**	No
Short Description	Set/Reset of "Posreg3" of the I/O Option Card		

Description

The I/O option card has 5 output pins which are normally used to signal on position registers SWE0 [▶ 301]-4.

OS3 sets the digital output "PosReg3" (terminal 8 of X11B) of the I/O option card.

Set SWCNFG [▶ 297] Bit 8 to "0" to disable the appropriate register (SWE3 [▶ 304]) to reassign the output for use in general purpose mode.

This command allows using the output for general purposes through standard communication channels or through motion tasking.

4.7.36 OS4

ASCII - Command	**OS4**		
Syntax Transmit	OS4 [Data]		
Syntax Receive	OS4 <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	36CA (hex)
DIM	-	**PROFIBUS PNU**	1658 (dec) IND = 0001xxxx (bin)
Range	0, 1	**DPR**	458 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	5.41		
Configuration	No	**Revision**	1.9
Function Group	Digital I/O	**EEPROM**	No
Short Description	Set/Reset of "Posreg4" of the I/O Option Card		

Description

The I/O option card has 5 output pins which are normally used to signal on position registers SWE0 [▶ 301]-4.

OS4 sets the digital output "PosReg4" (terminal 9 of X11B) of the I/O option card.

Set SWCNFG [▶ 297] Bit 12 to "0" to disable the appropriate register (SWE4 [▶ 305]) to reassign the output for use in general purpose mode.

This command allows using the output for general purposes through standard communication channels or through motion tasking.

4.7.37 OS5

ASCII - Command	**OS5**			
Syntax Transmit	OS5 [Data]			
Syntax Receive	OS5 <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer8		**CANBus Object Number**	36CB (hex)
DIM	-		**PROFIBUS PNU**	1659 (dec) IND = 0001xxxx (bin)
Range	0, 1		**DPR**	459 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	-		**Weightning**	
Start Firmware	5.41			
Configuration	No		**Revision**	1.9
Function Group	Digital I/O		**EEPROM**	No
Short Description	Set/Reset of "Posreg5" of the I/O Option Card			

Description

The I/O option card has 5 output pins which are normally used to signal on position registers SWE0 [▶ 301]-4.

OS5 sets the digital output "PosReg5" (terminal 10 of X11B) of the I/O option card.

Set SWCNFG2 [▶ 300] Bit 0 to "0" to disable the appropriate register (SWE0 [▶ 301]) to reassign the output for use in general purpose mode.

This command allows using the output for general purposes through standard communication channels or through motion tasking.

4.7.38 SETVCT

ASCII - Command	**SETVCT**		
Syntax Transmit	SETVCT [Data]		
Syntax Receive	SETVCT <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer16	**CANBus Object Number**	3635 (hex)
DIM	-	**PROFIBUS PNU**	1909 (dec) IND = 0000xxxx (bin)
Range	0 .. 7	**DPR**	309 (dec)
Default	0		
Opmode	0, 1	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	2.42		
Configuration	No	**Revision**	1.3
Function Group	Digital I/O	**EEPROM**	No
Short Description	Select a VCT Entry		

Description

The SETVCT command can be used to activate an entry in the VC table (velocity setpoint / current limit).

The VC table contains 8 velocity/current pairs (VCT entries).

When a VCT entry is activated, the velocity value that has been entered is taken as a digital velocity setpoint (only possible with OPMODE [▶ 50]=0). The current value from the VCT entry is taken as the current limit.

With analog velocity setpoint provision (OPMODE [▶ 50]=1) the velocity setpoint is ignored, but the current value is still used as the current limit.

If the SETVCT command is entered without additional parameters, the number of the presently active VCT entry is shown.

See also description of the VCTAB [▶ 167] command.

4.7.39 SLOTIO

ASCII - Command	**SLOTIO**		
Syntax Transmit	SLOTIO		
Syntax Receive	SLOTIO <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	35F3 (hex)
DIM	-	**PROFIBUS PNU**	1843 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	243 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	1.67		
Configuration	No	**Revision**	1.8
Function Group	Digital I/O	**EEPROM**	No
Short Description	I/O-Expansion Card: I/O States		

Description

The SLOTIO command returns the states of the inputs and outputs on the I/O-expansion card (slot card) in the format Hxxxxxxxx

Bit number Bit combination Input/Output Description

- 0 0x00000001 Input Bit 0 Motion block number (A0)
- 1 0x00000002 Input Bit 1 Motion block number (A1)
- 2 0x00000004 Input Bit 2 Motion block number (A2)
- 3 0x00000008 Input Bit 3 Motion block number (A3)
- 4 0x00000010 Input Bit 4 Motion block number (A4)
- 5 0x00000020 Input Bit 5 Motion block number (A5)
- 6 0x00000040 Input Bit 6 Motion block number (A6)
- 7 0x00000080 Input Bit 7 Motion block number (A7)
- 8 0x00000100 Input Reference switch
- 9 0x00000200 Input Acknowledge Contouring error
- 10 0x00000400 Input Start next motion block
- 11 0x00000400 Input Start jog mode
- 12 0x00001000 Input Continue a motion block
- 13 0x00002000 Input Start motion block no. A0 ... A7
- 14 0x00004000 Output □In-Position□ signal
- 15 0x00008000 Output □In-Position2□ signal (next)
- 16 0x00010000 Output Contouring error
- 17 0x00020000 Output □Position register 1□ signal
- 18 0x00040000 Output □Position register 2□ signal
- 19 0x00080000 Output □Position register 3□ signal
- 20 0x00100000 Output □Position register 4□ signal
- 21 0x00200000 Output □Position register 5□ signal
- 22 0x00400000 Status 24Volt - On
- 23 0x00800000 Status Slot fault
- 24...31 Reserve

4.7.40 STATIO

ASCII - Command	**STATIO**			
Syntax Transmit	STATIO			
Syntax Receive	STATIO <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	7 x Integer8		**CANBus Object Number**	35FC (hex)
DIM	-		**PROFIBUS PNU**	1852 (dec) IND = 0000xxxx (bin)
Range	0,1 (=State)		**DPR**	252 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Digital I/O		**EEPROM**	No
Short Description	I/O Status			

Description

The STATIO command returns the actual state of the digital inputs and outputs of the servo amplifier, in the following sequence.

IN1 IN2 IN3 IN4 ENABLE OUT1 OUT2

A 0 at the appropriate position means that the corresponding input/output is in the Low state, a 1 signifies the High state.

4.7.41 VCTAB

ASCII - Command	**VCTAB**		
Syntax Transmit	VCTAB [Data]		
Syntax Receive	VCTAB <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer8 Integer16 Integer32	**CANBus Object Number**	No
DIM	-	**PROFIBUS PNU**	No
Range	0 .. 7, 0 .. +/- VLIM, +/- IPEAK	**DPR**	No
Default	0		
Opmode	0, 1	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	2.42		
Configuration	No	**Revision**	1.3
Function Group	Digital I/O	**EEPROM**	No
Short Description	Define a VCT Entry		

Description

The VCTAB command is used for the definition /display of the VC table (velocity/current table). The VC table contains 8 velocity/current pairs (VCT entries).

A VCT entry can be activated either from the digital inputs or by an ASCII or SDO command.

When a VCT entry is activated, the velocity value that has been entered is taken as a digital velocity setpoint (only possible with OPMODE [▸ 50]=0). The current value from the VCT entry is taken as the current limit.

With analog velocity setpoint provision (OPMODE [▸ 50]=1), the velocity setpoint is ignored, but the current value is still used as the current limit.

The VC table is not saved in the serial EEPROM, but in the Flash EEPROM (motion task segment). For this reason, changes to the table entries can only be made while the output stage is disabled.

The VCTAB command can be used in one of three different forms:

1. VCTAB nr vsetp ilimit
   This command initializes the VCT entry □nr□ with the velocity setpoint □vsetp□ and the current limit □ilimit□.
   nr <0 ... 7>
   vsetp <-16000 RPM ... 16000 RPM>
   ilimit <0 .... 100000 mA>

   The command can only be used in this form while the output stage is disabled.
2. VCTAB nr
   This command is used to output the contents of the VCT entry □nr□.
   The output is made in the format VCTAB nr vsetp ilimit.
3. VCTAB
   This command is used to output the contents of the VCT table via the RS232 interface.
   The output consists of 8 lines in the following form:
   VCTAB nr vsetp ilimit

See also description of the SETVCT [▶ 165] and INxMODE [▶ 116] commands.

4.8 Drive Status

4.8.1 ACTIVE

ASCII - Command	**ACTIVE**			
Syntax Transmit	ACTIVE			
Syntax Receive	ACTIVE <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Integer8		**CANBus Object Number**	3504 (hex)
DIM	-		**PROFIBUS PNU**	1604 (dec) IND = 0000xxxx (bin)
Range	0, 1		**DPR**	4 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Drive Status		**EEPROM**	No
Short Description	Output stage active/inhibited			

Description

The ACTIVE command returns the present status of the output stage.

- ACTIVE=1 output stage is active/enabled
- ACTIVE=0 output stage is inhibited/disabled

The following conditions must be met to enable the output stage, depending on the controller configuration:

1. Standard configuration (no active MainsBTB function)
   - software enable set
   - hardware enable set
   - BTB is present
2. With active MainsBTB function (O1MODE [▶ 150] or O2MODE [▶ 156]=3)
   - Software enable set
   - hardware enable set
   - BTB is present
   - MainsBTB (Mains supply BTB) is present
   - DC-link (DC-bus) voltage > undervoltage threshold

4.8.2 CLRHR

ASCII - Command	**CLRHR**		
Syntax Transmit	CLRHR		
Syntax Receive	CLRHR	**Available in**	
Type	Command	**MMI**	Yes
ASCII Format	-	**CANBus Object Number**	3519 (hex)
DIM	-	**PROFIBUS PNU**	1625 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	25 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.27		
Configuration	No	**Revision**	1.8
Function Group	Drive Status	**EEPROM**	-
Short Description	Bit 5 of status register STAT is cleared		

Description

After every start-up or hardware reset of the drive, BIT 5 (0x20) of the STAT [▶ 181] register is set to high. This bit is cleared by CLRHR.

Possible usage

The Software User Interface uploads all the data stored in the drive, if the hardware reset Bit5 in the STAT [▶ 181] register is set too high. When the parameter is completely uploaded, the Bit5 is set to low using the command CLRHR. The STAT [▶ 181] register is monitored form the Software User Interface all the time and is checked if it is low. When it becomes high again (drive had a reset) the Software User Interface uploads the data again.

4.8.3 CLRWARN

ASCII - Command	**CLRWARN**		
Syntax Transmit	CLRWARN [Data]		
Syntax Receive	CLRWARN <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Unsigned8	**CANBus Object Number**	351B (hex)
DIM	-	**PROFIBUS PNU**	1627 (dec) IND = 0000xxxx (bin)
Range	0, 1	**DPR**	27 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Unsigned8
Drive State	Disabled + Reset (Coldstart)	**Weightning**	
Start Firmware	1.71		
Configuration	Yes	**Revision**	1.3
Function Group	Drive Status	**EEPROM**	Yes
Short Description	Warning mode		

Description

The CLRWARN configuration variable can be used to control the response of the drive if a warning occurs.

- CLRWARN=0 Warnings will be displayed until the cause of the warning has been removed. Warnings cannot be cancelled (exceptions: - contouring/following error, threshold detection).
- CLRWARN=1 A warning is only displayed now when it occurs (transition).

All warnings can be cancelled by the CLRFAULT [▶ 35] command, or through the digital input (“Controller reset” function).

The listing of the possible warnings can be found in the description of the STATCODE * [▶ 182] command.

4.8.4 COLDSTART

ASCII - Command	COLDSTART		
Syntax Transmit	COLDSTART		
Syntax Receive	COLDSTART	Available in	
Type	Command	MMI	Yes
ASCII Format	-	CANBus Object Number	3632 (hex)
DIM	-	PROFIBUS PNU	1906 (dec) IND = 0000xxxx (bin)
Range	-	DPR	306 (dec)
Default	-		
Opmode	All	Data Type Bus/DPR	-
Drive State	Disabled	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.8
Function Group	Drive Status	EEPROM	-
Short Description	Drive Reset		

Description

Software reset (warm boot) of the servo amplifier. The servo amplifier must be disabled. The current faults are cancelled, the servo amplifier software is initialized, and communications are re-established. This command has the same effect as turning the drive power off and then back on.

4.8.5 CONFIG

ASCII - Command	CONFIG		
Syntax Transmit	CONFIG		
Syntax Receive	CONFIG	Available in	
Type	Command	MMI	Yes
ASCII Format	-	CANBus Object Number	351C (hex)
DIM	-	PROFIBUS PNU	1628 (dec) IND = 0000xxxx (bin)
Range	-	DPR	28 (dec)
Default	-		
Opmode	All	Data Type Bus/DPR	-
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.8
Function Group	Drive Status	EEPROM	-
Short Description	Adaption and Conversion of Entered Parameter		

Description

The CONFIG command has only been implemented for reasons of compatibility, but it is not necessary to use it.

4.8.6 DRVSTAT

ASCII - Command	DRVSTAT		
Syntax Transmit	DRVSTAT		
Syntax Receive	DRVSTAT <Data>	Available in	
Type	Variable ro	MMI	Yes
ASCII Format	Integer32	CANBus Object Number	352D (hex)
DIM	-	PROFIBUS PNU	1645 (dec) IND = 0000xxxx (bin)
Range	0 .. 0xFFFFFFFF	DPR	45 (dec)
Default	-		
Opmode	All	Data Type Bus/DPR	Integer32
Drive State	-	Weightning	
Start Firmware	1.77		
Configuration	No	Revision	2.0
Function Group	Drive Status	EEPROM	No
Short Description	internal Status information		

Description

The DRVSTAT command returns the internal status information in the form of a bit-variable.

Bit / Display / Level	Bit combination	Description
0 / n01 / 0	0x00000001	=1 I2T threshold exceed. Is high, if I2T [▶ 21] is greater than the adjusted threshold for I2TLIM [▶ 107], otherwise low.
1 / n02 / 0	0x00000002	=1 Regen message. Is high, if the actual regen power exceeds the adjusted threshold PBALMAX [▶ 82], otherwise low.
2 / n03 / 1	0x00000004	=1 Contouring error. Is set, if the distance between the actual position and the target position of the trajectory generator is greater than the contouring error window PEMAX [▶ 282]. Is cleared by the command CLRFAULT [▶ 35] or by an digital input with INxMODE [▶ 116]=14.
3 / n04 / 1	0x00000008	=1 Node guarding. Is set, if the Bus (PROFIBUS or CAN) or the Slotcard has no communication to the master for the adjusted time EXTWD [▶ 95]. Is cleared by the command CLRFAULT [▶ 35] or by an digital input with INxMODE [▶ 116]=14.
4 / n05 / 0	0x00000010	=1 Line phase missing. Is high, if one or all of the three input line phase are lost, otherwise low.
5 / n06 / 1	0x00000020	=1 Software limit switch 1 (SWE1 [▶ 302]) underrun. Is set if: - the position is lower than SWE1 [▶ 302] - a motion task is started which has a target position lower than SWE1 [▶ 302]. At the same time the bit "faulty motion task" is set. the bit is cleared if: - the actual position overruns SWE1 [▶ 302] and a positive velocity is

		given - a motion task is started which has a target position greater than SWE1 [▸ 302].
6 / n07 / 1	0x00000040	=1 Software limit switch 2 (SWE2 [▸ 303]) overrun. Is set if: - the position is higher than SWE2 [▸ 303] - a motion task is started which has a target position higher than SWE2 [▸ 303]. At the same time the bit "faulty motion task" is set. the bit is cleared if: - the actual position underruns SWE2 [▸ 303] and a negative velocity is given - a motion task is started which has a target position smaller than SWE2 [▸ 303].
7 / n08 / 0	0x00000080	=1 Faulty motion task was started Is set, if a faulty motion task (wrong checksum) is started. Is cleared, if a valid motion task is started.
8 / n09 / 0	0x00000100	=1 No reference point. Is set, is a motion task is started without starting a refernce move before. Is cleared, if a refernce move is started.
9 / n10 / 1	0x00000200	= 1 PSTOP active. Is high, if the hardware limit switch PSTOP is active, otherwise low.
10 / n11 / 1	0x00000400	= 1 NSTOP active. Is high, if the hardware limit switch NSTOP is active, otherwise low.
11 / n12 / 0	0x00000800	=1 Default motor data loaded. Is set, if the motor number stored in the EEPROM of the drive is not the same as the motor number stored in the feedback system (EnDAT or Hiperface). By saving the drive parameter (SAVE [▸ 51]) and restart the drive, the warning disappears.
12 / n13 / 1	0x00001000	=1 Slot warning (I/-expansion board). Is high, if the external 24V supply of the I/O expansion board is missing, otherwise low.
13 / n14 / 0	0x00002000	=1 Scanning for MPHASE [▸ 199] (FBTYPE [▸ 190]=7) Is set while start-up of the drive and is cleared after the drive was enabled and MPHASE [▸ 199] was determined.
14 / n15 / 0	0x00004000	=1 Faulty VCT entry was selected. Is set, if INxMODE [▸ 116]=35 is selected, and a faulty VCT entry is started.
15 / n16 / 0	0x00008000	Is active, if one or more of the warnings n17...n31 are active.
16	0x00010000	=1 Motion task active. Is set, if a motion task is started (motion task, jog or homing move). Is cleared, if the action is finished or a STOP [▸ 297] - command is executed.
17	0x00020000	=1 Reference point is set. Is set, if the homing move was done or a absolute encoder feedback device is used. Is cleared if a homing move is started.
18	0x00040000	=1 Home switch. Is high, if the homing switch is closed, otherwise low.
19	0x00080000	=1 In-Position. Is high, if the distance between the target position and the actual position is smaller than PEINPOS [▸ 282], otherwise low. When several motion tasks are tied together, only the last motion task enables this bit.

20	0x00100000	=1 Position latch occurred (positive edge). Is set, if a positive edge at the latch input (Input2 with IN2MODE [▶ 123]=26) was detected. Is cleared, if the lachted position was read (LATCH16 [▶ 260]/ LATCH32 [▶ 261]).
21	0x00200000	=1 Position register 0. Is high, if the configurated condition (SWCNFG2 [▶ 300], SWE0 [▶ 301], SWE0N [▶ 301]) is true, otherwise low (See SWCNFG2 [▶ 300]).
22	0x00400000	=1 Position register 1. Is high, if the configurated condition (SWCNFG [▶ 297], SWE1 [▶ 302], SWE1N [▶ 302]) is true, otherwise low (See SWCNFG [▶ 297]).
23	0x00800000	=1 Position register 2. Is high, if the configurated condition (SWCNFG [▶ 297], SWE2 [▶ 303], SWE2N [▶ 304]) is true, otherwise low (See SWCNFG [▶ 297]).
24	0x01000000	=1 Position register 3. Is high, if the configurated condition (SWCNFG [▶ 297] **,SWE3**, **SWE3N**) is true, otherwise low (See SWCNFG [▶ 297]).
25	0x02000000	=1 Position register 4. Is high, if the configurated condition (SWCNFG [▶ 297], SWE4 [▶ 305], SWE4N [▶ 306]) is true, otherwise low (See SWCNFG [▶ 297]).
26	0x04000000	=1 Initialization phase finished. Is set, if the initialization phase of the drive is finished (takes about 15s).
27	0x08000000	=1 Position register 5. Is high, if the configurated condition (SWCNFG2 [▶ 300], SWE5 [▶ 307], SWE5N [▶ 307]) is true, otherwise low (See SWCNFG2 [▶ 300]).
28	0x10000000	=1 Motor stand still message. Is high, if the actual motor velocity is lower than the threshold VEL0 [▶ 338], otherwise low.
29	0x20000000	=1 Safety relais selected. Is high, if the safety relay of the option -AS- is switched on, otherwise low.
30	0x40000000	= Output stage enabled. Is high, if the soft- and the hardware enable is present, otherwise low.
31	0x80000000	=1 Drive has an error state. Is set, if the drive has a fault (output stage is disabled, error number is displayed). The command ERRCODE [▶ 75] gives the error in plain text. The bit is cleared, if the drive is reset or the command CLRFAULT [▶ 35] is send.

Also see about this

- DECSTOP [▶ 330]
- SWE3 [▶ 304]
- SWE3N [▶ 305]

4.8.7 ERRCODES

ASCII - Command	**ERRCODE ***		
Syntax Transmit	ERRCODE *		
Syntax Receive	ERRCODE * <Data>	**Available in**	
Type	Command	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	No
DIM	-	**PROFIBUS PNU**	No
Range	0 .. 0xFFFFFFFF	**DPR**	No
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Drive Status	**EEPROM**	-
Short Description	Output Error Register		

Description

The ERRCODE* command returns the internal status information in the form of a bit-variable. A bit is set for as long as the corresponding error/fault is detected. The bit is deleted by the hardware reset of the amplifier. Faults that are designated by the SW label can also be deleted by a software reset (function CLRFAULT [▸ 35] □ clear fault).

Level gives an information about the error handling in the drive.

- Level 2: A fault causes an emergency stop. The stop of the motor is done in velocity control using the emergency stop ramp (DECSTOP [▸ 330]). When the motor reaches the zero velocity level (VEL0 [▸ 338]) (limited by max. 5 sec), the power stage is disabled. The Ready-to-Operate relay is switched off. The drive must be reset before it can be enabled again. The fault is displayed.
- Level 3: (starting with firmware 4.01) A fault causes an emergency stop. The stop of the motor is done without feedback device (sensorless). When the motor has stopped, the power stage is disabled. The Ready-to-Operate relay is switched off. The drive has to be reset before it can be enabled again. The fault is displayed.
- Level 4: A fault causes an directly disable of the power stage. The motor has no torque (coast). The Ready-to-Operate relay is switched off. The drive has to be reset before it can be enabled again. The fault is displayed.

Faults, that have different levels (2/3 and 4), the behaivior is controlled by ACTFAULT [▸ 34] and MBRAKE [▸ 226] or STOPMODE [▸ 85]

- ACTFAULT [▸ 34]=1 or MBRAKE [▸ 226]=1 LEVEL 2 or 3 (Default-Setting)
- ACTFAULT [▸ 34]=0 and MBRAKE [▸ 226]=0 LEVEL 4

Bit / Displ. / Reset/ Level	Bit	Description
00/F01/SW/2,4	0x00000001	=1 Heatsink overtemperature is set, if the heatsink temperature TEMPH exceeds the max allowed threshold MAXTEMPH.
01/F02/SW/2,4	0x00000002	=1 DC-link overvoltage Is set if the DC-link voltage exceeds the max threshold selected by VBUSBAL [▸ 88].
02/F03/SW/2	0x00000004	=1 Contouring error of the external trajectory (OPMODE [▸ 50]=6/ SERCOS) Is set, if the target speed which is given by the extrnal trajectory is higher than VLIM [▸ 338]/VLIMN [▸ 339].

Bit / Displ. / Reset/ Level	Bit	Description
03/F04/HW/3,4	0x00000008	=1 Feedback error Is set if a feedback error was detected.
04/F05/SW/2,4	0x00000010	=1 Undervoltage protection Is set, if the DC-link voltage is lower than VBUSMIN [▶ 54] (only if the drive is enabled).
05/F06/HW/2,4	0x00000020	=1 Motor overtemperature is set, if the heatsink temperature TEMPM [▶ 31] exceeds the max allowed threshold MAXTEMPM [▶ 80].
06/F07/HW/2,4	0x00000040	=1 if the internal electronic supply is faulty.
07/F08/SW/3,4	0x00000080	=1 Overspeed Is set if the velocity of the motor exceeds the overspeed threshold (VOSPD [▶ 341]).
08/F09/HW/4	0x00000100	=1 EEPROM Checksum error Is set, if the data read/written from the EEPROM is not valid. There are two possibilities, that can cause this error. First is a defect EEPROM and the second is a wrong checksum in the EEPROM. In the second case, a SAVE [▶ 51] can solve the problem.
09/F10/HW	0x00000200	Reserved
10/F11/HW/2,4	0x00000400	=1 Brake error Is set, if the brake switch detects a fault (e.g. Brake is selected, but no brake is connected).
11/F12/HW	0x00000800	=1 Missing motor connection. This error occurs, if a motor connection (not feedback device) is missing. The error can be disabled by CPHASE [▶ 222] = 0.
12/F13/SW/2,4	0x00001000	=1 Ambient overtemperature is set, if the ambient temperature TEMPE [▶ 30] exceeds the max allowed threshold MAXTEMPE [▶ 79].
13/F14/HW/2,4	0x00002000	=1 Output stage fault This fault can be caused by: Earth short circuit of the motor Short circuit of the motor phases Short circuit of the regen.
14/F15/SW/2,4	0x00004000	=1 I²tmax override Is set, if I²t exceeds 115% of ICONT (FOLDMODE [▶ 77]=0) or 105% of ICONT (FOLDMODE [▶ 77]=1).
15/F16/SW/2,4	0x00008000	=1 Mains BTB
16/F17/HW/2,4	0x00010000	=1 A/D converter error
17/F18/HW/2,4	0x00020000	=1 Regen error destroyed regen transistor regen resistor extern selected, but the internal one is used.
18/F19/SW/2,4	0x00040000	=1 line phase missing (PMODE [▶ 83]=2)
19/F20/HW/2,4	0x00080000	=1 Slot error Error depends on the type of Slot board: 1. I/O expansion board The error is caused by a missing 24V supply at the I/O board. 2. DPR Slot board (Beckhoff, L&B, Sigmatek) The error is generated, if the DPR interrupt fails to appear. The watch-dog time can be selected by EXTWD [▶ 95]. 3. PROFIBUS Error in the initialization time.

Bit / Displ. / Reset/ Level	Bit	Description
20/F21/HW/2,4	0x00100000	=1 PROFIBUS handling error If the OPMODE [▶ 50] is changed by another communication channel than PROFIBUS, when the drive is under control of the PROFIBUS, this error is generated. Exception: Working mode -126 for PROFIBUS. This is the safe opmode when the drive is switched on.
21/F22/HW/2,4	0x00200000	=1 Earth short circuit The earth short circuit supervisor of the 40/70 A units.
22/F23/HW/2,4	0x00400000	=1 CANopen Bus-Off Fault in CAN communication. The communication fault BUSOFF is generated by layer 2 (CAN controller). This fault can have several resons. Some examples are: Drive tries to establish communication, but there is no other node. CAN nodes have different baud rates, Bus cable defekt, reflections because of missing or wrong bus terminations, etc. A BUSOFF is displayed by the drive, if another CAN node is connected and minimum one errorfree object is generated. If the BUSOFF is generated and the drive is moving the motor, the motor is stopped using the emergency ramp and then the drive is disabled.
23/F24/SW/2,4	0x00800000	Warning generates a error message (defined by WMASK [▶ 56])
24/F25/HW/3,4	0x01000000	Commutation Error (Run-away of the motor)
25/F26/SW/2,4	0x02000000	Hardware limit switch error at homing move (defined by REFLS [▶ 291])
26/F27/HW/4	0x04000000	=1 "-AS-Option" error If the ENABLE signal of the drive is high and the -AS-option is activated, this error is generated (starting Version 3.44).
27/F28/SW/2	0x08000000	=1 error "external tarjectory" is generated, if an external position profile generator creates a step, that exceeds the maximum value.
28/F29/SW/2	0x10000000	=1 Sercos error
29/F30/SW	0x20000000	Sercos Emergency time out
30/F31/SW	0x40000000	Reserved
31/F32/HW/4	0x80000000	=1 System error Is set, if an error occurred in the system check of the initialization phase or a watch-dog error in the working phase. Following reasons are possible: 1. Wrong program data in the FLASH (e.g. interrupted program download) 2. Macro error (the macros could not be compiled) 3. Software watch-dog activated 4. Error with the EEPROM (read or write). 5. FPGA error (FPGA could not be loaded correctly) 6. Macro RAM (the compilation of the MACROs detect too less RAM) When the drive is switched on, a detailed message is send vis RS232.

4.8.8 FLTCNTS

ASCII - Command	**FLTCNT ***			
Syntax Transmit	FLTCNT *			
Syntax Receive	FLTCNT * <Data>		**Available in**	
Type	Command		**MMI**	Yes
ASCII Format	1 x Integer32 + 32 x Integer16		**CANBus Object Number**	No
DIM	-		**PROFIBUS PNU**	No
Range	-		**DPR**	No
Default	-			
Opmode	-		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Basic Setup		**EEPROM**	-
Short Description	Fault Frequency			

Description

The command returns a list of 33 numbers:

1st number: total number of faults (Integer32)

2nd number number of occurrences of fault F01

3rd number number of occurrences of fault F02

....

33rd number number of occurrences of fault F32

4.8.9 FLTHISTS

ASCII - Command	**FLTHIST ***			
Syntax Transmit	FLTHIST *			
Syntax Receive	FLTHIST * <Data>		**Available in**	
Type	Command		**MMI**	Yes
ASCII Format	20 x Integer32		**CANBus Object Number**	No
DIM	Number and TRUN		**PROFIBUS PNU**	No
Range	-		**DPR**	No
Default	-			
Opmode	-		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Basic Setup		**EEPROM**	No
Short Description	Fault History: Display last 10 faults			

Description

The FLTHIST command produces a list of the last 10 faults that occurred, together with the corresponding number of operating hours at the time of occurrence, in plain text.

The output looks like this:
n1 t1 n2 t2 n3 t3 ....n10 t10
n □ fault number
t □ time of the event (operating hours counter) [in 1024/60000 minutes]

4.8.10 LEDSTAT

ASCII - Command	LEDSTAT		
Syntax Transmit	LEDSTAT [Data]		
Syntax Receive	LEDSTAT <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	3581 (hex)
DIM	-	**PROFIBUS PNU**	1729 (dec) IND = 0000xxxx (bin)
Range	0 .. 16	**DPR**	129 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	1.30		
Configuration	No	**Revision**	1.3
Function Group	Drive Status	**EEPROM**	-
Short Description	Display page		

Description

The variable LEDSTAT shows the number of the present □display page□. Altering the LEDSTAT variable makes it possible the change the display via the serial interface.
The assignments are as follows:

- LEDSTAT=0 Display is switched off
- LEDSTAT=1 Status display
- LEDSTAT=2 Fieldbus address
- LEDSTAT=3 CAN Baud rate
- LEDSTAT=4 Parameter S01 (Kp velocity control loop)
- LEDSTAT=5 Parameter S02 (Tn velocity control loop)
- LEDSTAT=6 Parameter S03 (setpoint offset)
- LEDSTAT=7 Parameter S04 (motor number)
- LEDSTAT=8 Parameter S05 (encoder selection)
- LEDSTAT=9 Parameter S06 (brake selection)
- LEDSTAT=10 Parameter S07 (Multidrive selection, from software 3.00)
- LEDSTAT=11 Load data from the EEPROM
- LEDSTAT=12 Save data in the EEPROM
- LEDSTAT=13 Set default values (from software 3.00)
- LEDSTAT=14 New configuration of the amplifier (M_RESET, from software 3.00)
- LEDSTAT=15 Error messages
- LEDSTAT=16 Serial number

Also see about this

M_RESET [▸ 47]

4.8.11 NONBTB

ASCII - Command	**NONBTB**			
Syntax Transmit	NONBTB [Data]			
Syntax Receive	NONBTB <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer8		**CANBus Object Number**	35AA (hex)
DIM	-		**PROFIBUS PNU**	1770 (dec) IND = 0000xxxx (bin)
Range	0, 1		**DPR**	170 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	Yes		**Revision**	1.9
Function Group	Drive Status		**EEPROM**	Yes
Short Description	Mains-BTB Check On/Off			

Description

If the Mains-BTB signal (power on) is missing, then the output stage will produce a fault message F16 (Mains-BTB) when it is enabled. If this response is not wanted, then the monitoring of the Mains-BTB signal can be switched off (NONBTB 1). This function can be used to control a DC supply. See also UVLTMODE [▸ 87].

Starting with firmware 5.41, this parameter is changed to a configuration parameter

4.8.12 OPTION

ASCII - Command	**OPTION**			
Syntax Transmit	OPTION			
Syntax Receive	OPTION <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Integer16		**CANBus Object Number**	35B5 (hex)
DIM	-		**PROFIBUS PNU**	1781 (dec) IND = 0000xxxx (bin)
Range	int (=Word)		**DPR**	181 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Drive Status		**EEPROM**	No
Short Description	Option Slot ID			

Description

The OPTION command returns the identification for the slot card that has been detected. The following IDs are possible at present:

- H0000 no slot card detected
- H01xx I/O-expansion card
- H02xx PROFIBUS
- H03xx SERCOS

- H06xx DeviceNet
- H8100 Beckhoff-Lightbus
- H8200 Lenord&Bauer
- H8300 Sigmatek

The least significant 8 bits (xx) indicate the hardware revision of the corresponding card.

4.8.13 READY

ASCII - Command	**READY**		
Syntax Transmit	READY		
Syntax Receive	READY <Data>	**Available in**	
Type	Variable ro	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	35DD (hex)
DIM	-	**PROFIBUS PNU**	1821 (dec) IND = 0000xxxx (bin)
Range	0, 1	**DPR**	221 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Drive Status	**EEPROM**	No
Short Description	Status of the Software Enable		

Description

Requests the status of the internal software enable.

- READY = 0 Disabled
- READY = 1 Enabled

4.8.14 REMOTE

ASCII - Command	**REMOTE**		
Syntax Transmit	REMOTE		
Syntax Receive	REMOTE <Data>	**Available in**	
Type	Variable ro	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	35E4 (hex)
DIM	-	**PROFIBUS PNU**	1828 (dec) IND = 0000xxxx (bin)
Range	0 .. 1	**DPR**	228 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	2.0
Function Group	Drive Status	**EEPROM**	No
Short Description	Status of the Hardware Enable		

Description

The REMOTE command indicates the state of the hardware enable input.

A 1 indicates a high state of the inputs (hardware enable is set), a 0 indicates a Low state.

4.8.15 STAT

ASCII - Command	STAT		
Syntax Transmit	STAT		
Syntax Receive	STAT	Available in	
Type	Variable ro	MMI	Yes
ASCII Format	Integer16	CANBus Object Number	35FB (hex)
DIM	-	PROFIBUS PNU	1851 (dec) IND = 0000xxxx (bin)
Range	int (=Word)	DPR	251 (dec)
Default	-		
Opmode	All	Data Type Bus/DPR	Integer16
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.3
Function Group	Drive Status	EEPROM	No
Short Description	Drive Status Word		

Description

The STAT command provides a 16-bit status word in hexadecimal format (Hxxxx).

Bit Value Interpretation
0 0x0001 =0 if output stage is enabled
=1 output stage disabled
1 0x0002 =0 if controller is ready for operation (BTB)
=1 fault present
2 0x0004 reserve
3 0x0008 =1 if service function is active
=0 no service function active
4 0x0004 reserve
5 0x0020 =1 after a hardware reset, is cancelled by CLRHR [▸ 169]
6 0x0040 =1 configuration variable was altered (SAVE [▸ 51] and COLDSTART [▸ 170])
=0 no configuration variable altered
7 0x0080 =1 safety relay is active (AS-Option)
=0 safety relay is not active
8 0x0100 =1 discrepancy between RAM and EEPROM parameters (cancelled by SAVE [▸ 51] command).
=0 RAM and EEPROM parameters are the same
9 0x0200 =1 slot-expansion card is available
=0 slot-expansion card is not available
10 0x0400 =1 RAM parameter modified (cancelled by DUMP [▸ 94] command)
=0 no change in RAM parameters since the last DUMP [▸ 94].
11...15 Reserve

Bits 5, 6, 8 and 10 are used for an external signal that internal parameters have been changed

Bit 5 hardware reset
Bit 5 is set if the parameters are copied from the serial EEPROM to the RAM (this happens after a hardware reset of a LOAD command). If this bit is set, all the parameters should be requested by the parameterization software (DUMP [▸ 94] command) and bit 5 should be cancelled by the CLRHR [▸ 169] command.

Bit 6 configuration variable was altered
Any alteration of a configuration variable (a variable that makes it necessary to recompile the macro, i.e. to reset the amplifier) means that this bit will be set to 1. If this bit is set, the parameterization software should generate a SAVE [▸ 51] / COLDSTART [▸ 170] command (controller reset) at a suitable moment. Bit 6 is only

cancelled by a hardware reset (COLDSTART [▶ 170]).

Bit 8 discrepancy between EEPROM and RAM parameters
Any alteration of a RAM parameter means that this bit is set to 1. If this bit is set, the parameterization software should generate a SAVE [▶ 51] command (save the data in the EEPROM) at a suitable moment (e.g. on exiting the program). This bit is cancelled by a SAVE [▶ 51] command.

Bit 10 RAM parameters modified
Any alteration of a RAM parameter through a parameterization channel other than the RS232 means that this bit is set to 1. If this bit is set, the parameterization software should generate a DUMP [▶ 94] command (read all data) at a suitable moment. This bit is cancelled by a DUMP [▶ 94] command.

4.8.16 STATCODE

ASCII - Command	**STATCODE**		
Syntax Transmit	STATCODE		
Syntax Receive	STATCODE <Data>	**Available in**	
Type	Command	**MMI**	Yes
ASCII Format	String	**CANBus Object Number**	No
DIM	-	**PROFIBUS PNU**	No
Range	-	**DPR**	No
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Drive Status	**EEPROM**	-
Short Description	Plain Text Warnings		

Description

The warnings are displayed as plain text.

4.8.17 STATCODES

ASCII - Command	**STATCODE ***		
Syntax Transmit	STATCODE *		
Syntax Receive	STATCODE * <Data>	**Available in**	
Type	Command	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	No
DIM	-	**PROFIBUS PNU**	No
Range	0 .. 0xFFFFFFFF	**DPR**	No
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Drive Status	**EEPROM**	-
Short Description	Status Variable □Warnings□		

Description

The STATCODE * command returns the internal warnings in the form of a bit-variable.

The assignments for the individual bits can be seen in the following table.

Bit	Display	Meaning
0 / 0x00000001	n01	=1 I2T [▸ 21] Threshold override
1 / 0x00000002	n02	=1 Regen message
2 / 0x00000004	n03	=1 Contouring error
3 / 0x00000008	n04	=1 Node guarding active
4 / 0x00000010	n05	=1 Line phase missing
5 / 0x00000020	n06	=1 Software limit switch 1 active
6 / 0x00000040	n07	=1 Software limit switch 2 active
7 / 0x00000080	n08	=1 Wrong motion task started
8 / 0x00000100	n09	=1 Reference point not set
9 / 0x00000200	n10	=1 PSTOP active
10 / 0x00000400	n11	=1 NSTOP active
11 / 0x00000800	n12	=1 Default motor settings loaded
12 / 0x00001000	n13	=1 Slot warning (I/O extension board)
13 / 0x00002000	n14	=1 Calculation of MPHASE [▸ 199] active (FBTYPE [▸ 190]=7)
14 / 0x00004000	n15	=1 Wrong VCT entry started
15 / 0x00008000	n16	Is active, if one or more of the warnings n17...n31 are active.
16 / 0x00010000	n17	CAN-Sync is not locked
17 / 0x00020000	n18	Using Multiturn encoder feedback, a overrun over the maximum number of resolutions (+/-2048) was detected (starting with firmware 4.91)
18...30	n19 ..n31	Reserved
31 / 0x80000000	n32	=1 Beta version of the firmware

Also see about this

- I2TLIM [▸ 107]
- PBALMAX [▸ 82]
- PEMAX [▸ 282]
- CLRFAULT [▸ 35]
- EXTWD [▸ 95]
- SWE1 [▸ 302]
- SWE2 [▸ 303]
- HSAVE [▸ 198]
- SAVE [▸ 51]
- IN1MODE [▸ 116]

4.8.18 STATUS

ASCII - Command	STATUS		
Syntax Transmit	STATUS		
Syntax Receive	STATUS <Data>	Available in	
Type	Variable ro	MMI	Yes
ASCII Format	Integer16 Integer32 Integer16 Integer16 Integer16	CANBus Object Number	35FD (hex)
DIM	-	PROFIBUS PNU	1853 (dec) IND = 0000xxxx (bin)
Range	int (=Word); long int (=DoubleWord)	DPR	253 (dec)
Default	-		
Opmode	All	Data Type Bus/DPR	-
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.3
Function Group	Drive Status	EEPROM	No
Short Description	Detailed Amplifier Status		

Description

The STATUS command returns the detailed status information in the form of a 5 status variables in hexadecimal format.

Word no. 1 Format Hxxxx
Bit 0 =0 if hardware enable is set (ENABLE input = 24V)
Bit 1 =0 if software enable is set
Bit 2 reserve
Bit 3 =0 if amplifier is ready for operation (BTB / no fault)

Word no. 2 Format Hxxxx
Bits 0 ... 31 fault variable (see ERRCODE [▸ 75])

Word no. 3 Format Hxxxx
Word no. 4 Format Hxxxx
=0 no service function active
=1 service function □constant current/velocity□ is active
=2 jog mode MJOX

Word no. 5 Format Hxxxx

Bit Value Interpretation
0 0x0001 =1 motion block / homing movement / jog mode is active
1 0x0002 =1 reference point set
2 0x0004 =1 reference switch occupied (home position)
3 0x0008 =1 IN-POSITION signal
4 0x0010 =1 position has been latched (positive edge)
5 0x0020 =1 homing in progress
6 0x0040 =1 jog mode is running
7 0x0080 =1 position has been latched (negative edge)
8 ... 15 reserve

4.8.19 TRJSTAT

ASCII - Command	**TRJSTAT**		
Syntax Transmit	TRJSTAT		
Syntax Receive	TRJSTAT <Data>	**Available in**	
Type	Variable ro	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	3613 (hex)
DIM	-	**PROFIBUS PNU**	1875 (dec) IND = 0000xxxx (bin)
Range	0 .. 0xFFFFFFFF	**DPR**	275 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	2.03		
Configuration	No	**Revision**	1.5
Function Group	Drive Status	**EEPROM**	No
Short Description	Status2 Information		

Description

The TRJSTAT command returns the internal status information in the form of a bit-variable.The status information is primarily used for internal functions. Only the bits that are marked by an □*□ can be used for external functions (control system).

Bits 16 ... 20 are also mirrored in the DRVSTAT [▸ 171] status variable.

Bit	Significance	Meaning
0	0x00000001	=1 the output INPOS2 is updated every msec
1	0x00000002	=1 At the end of the actual motion task, the drive outputs no IN-POSITION signal (a motion task sequence was activated.
2*	0x00000004	=1 Toggle Bit "Motion task finished". Is toggled at the end of a motion task. The toggling of the Bit is done if the target position is reached, and the profile generator is switched off. This is different to the functionality of the IN-POSITION Bit. When the drive is switched on, this Bit is set to low. (Firmware 3.41)
3...15		Reserved
16*	0x00010000	=1 Motion task active (position control) Is set, if a motion task is started (motion task, Jog, Homing). Is cleared if a motion task has finished or is stopped (STOP [▸ 297]).
17*	0x00020000	=1 Reference point set Is set, if the homing move has successfully finished or if the feedback device is a multiturn encoder. Is cleared when a homing move is started.
18*	0x00040000	=1 Home position Is high, if the homing switch is active, otherwise low.
19*	0x00080000	=1 In-Position Is set, if the difference between the actual position and the target position is smaller than PEINPOS [▸ 282]. Is cleared, if the distance is greater.
20*	0x00100000	=1 Position latch activated (positive latch) Is set, if a positive edge at Latch input 2 (configured by IN2MODE [▸ 123]=26) was detected. Is cleared, if the position is read by LATCH16 [▸ 260] / LATCH32 [▸ 261].
21*	0x00200000	=1 Homing move is active Is set, if a Homing move was started. Is cleared, if the homing move is successful or stopped (STOP [▸ 297]).

Bit	Significance	Meaning
22*	0x00400000	=1 Jog move active Is set, if a Jog move is started. Is cleared, if the Jog move is stopped.
23	0x00800000	=1 Position latch activated (negative latch) Is set, if a negative edge at Latch input 2 (configured by IN2MODE [▸ 123]=26) was detected. Is cleared, if the position is read by LATCH16N [▸ 260] / LATCH32N [▸ 261].
24	0x01000000	=1 Emergency stop active Is set, if an emergency stop has occurred (DEC-phase after an error, active hardware limit switches, Input configured as Emergency stop with level low.
25	0x02000000	=1 position latch at input1 (positive transition), if a rising edge at input 1 is detected, when input1 is defined as latch input (IN1MODE [▸ 116]=26). Is reset, if the latched position is read by LATCHX16 [▸ 262] or LATCHX32 [▸ 263]. (4.61 Firmware)
26	0x04000000	=1 position latch at input1 (negative transition), if a falling edge at input 1 is detected, when input1 is defined as latch input (IN1MODE [▸ 116]=26). Is reset, if the latched position is read by LATCHX16N [▸ 262] or LATCHX32N [▸ 264]. (4.61 Firmware)
27 .. 31		Reserved

4.9 Feedback

4.9.1 CALCCOG

ASCII - Command	CALCCOG		
Syntax Transmit	CALCCOG [Data]		
Syntax Receive	CALCCOG	Available in	
Type	Command	MMI	No
ASCII Format	-	CANBus Object Number	No
DIM	rpm	PROFIBUS PNU	No
Range	0 .. 5	DPR	No
Default	2		
Opmode	0	Data Type Bus/DPR	-
Drive State	Enabled	Weightning	
Start Firmware	5.41		
Configuration	No	Revision	1.9
Function Group	Feedback	EEPROM	-
Short Description	Determing the Cogging Table		

Description

CALCCOG starts the automatic determination of the cogging table (see also COGGING [▸ 189]). To do this, the output stage must be enabled, and the motor must be able to move freely, ideally without any mechanical coupling. The gain of the velocity controller GV [▸ 332] should be tuned as high as possible.

While this command is being executed, the motor makes two full turns at the predefined speed. During the first turn, the cogging table values is coarsely identified. The second turn makes a fine identification for the cogging table values.

After this function has been completed, the 24 V power supply must be switched off and on to copy the determined table to the FLASH.

Until now, The CALCCOG function is reasonable when a Resolver, Hiperface- or an EnDAT-feedback device has been selected as the feedback device (FBTYPE [▸ 190]=0, 2 or 4).

Before starting this processing, the parameter COGGING [▸ 189] must be firstly set to one. Please reference to the ASCII command COGGING [▸ 189].

4.9.2 CALCHP

ASCII - Command	**CALCHP**	**For Manufacturer Use only**	
Syntax Transmit	CALCHP [Data] , [Data]		
Syntax Receive	CALCHP	**Available in**	
Type	Command	**MMI**	No
ASCII Format	-	**CANBus Object Number**	3512 (hex)
DIM	rpm	**PROFIBUS PNU**	1618 (dec) IND = 0000xxxx (bin)
Range	0 .. 200	**DPR**	18 (dec)
Default	5		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	Enabled	**Weightning**	
Start Firmware	1.34		
Configuration	No	**Revision**	1.9
Function Group	Feedback	**EEPROM**	-
Short Description	Determining the Hiperface Parameters		

Description

This command can be used to start the automatic determination of the Hiperface parameters. To do this, the output stage must be enabled and the drive must be able to move freely. While this command is being carried out, the motor makes a full turn at the predefined velocity. During this phase, the offset parameters (HISOFFS [▸ 196]/HICOFFS [▸ 195]) and the sine/cosine gain factor (HIFACT1 [▸ 195]) are calculated. After this function has been completed, the newly determined parameters can be stored in the encoder, using the HSAVE [▸ 198] command for FBTYPE [▸ 190] = 2 or 4 or using the command SAVE [▸ 51] forFBTYPE [▸ 190] = 7.

The CALCHP function is only available when a Hiperface- or an EnDAT-Encoder or sine encoder has been selected as the feedback device (FBTYPE [▸ 190]=2 or 4 or 7).

Starting with firmware 5.41, the command allows also two parameters. The first is to select a certain speed in rpm and the second the angle of the motor that should be moved.

E.g.: CALCHP 5 10, start CALCHP with 5 rpm and move the motor 10 degrees.

4.9.3 CALCRK

ASCII - Command	**CALCRK**		**For Manufacturer Use only**	
Syntax Transmit	CALCRK [Data]			
Syntax Receive	CALCRK		**Available in**	
Type	Command		**MMI**	No
ASCII Format	-		**CANBus Object Number**	3513 (hex)
DIM	rpm		**PROFIBUS PNU**	1619 (dec) IND = 0000xxxx (bin)
Range	0 .. 200		**DPR**	19 (dec)
Default	5			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	Enabled		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Feedback		**EEPROM**	-
Short Description	Calculate resolver parameters			

Description

This command can be used to start the automatic determination of the resolver parameter RK [▸ 202] (sine/cosine gain factor). To do this, the output stage must be enabled and the drive must be able to move freely. While this command is being carried out, the motor makes a full turn at the given velocity. If CALCRK is started without parameter, the default value is used. After this function has been completed, the newly determined RK [▸ 202] parameter can be stored in the EEPROM, using the SAVE [▸ 51] command.

This command can be used to reduce the current ripple of the motor at high velocity. It can only be used with resolver feedback.

4.9.4 CALCRP

ASCII - Command	**CALCRP**		**For Manufacturer Use only**	
Syntax Transmit	CALCRP			
Syntax Receive	CALCRP		**Available in**	
Type	Command		**MMI**	No
ASCII Format	-		**CANBus Object Number**	3514 (hex)
DIM	-		**PROFIBUS PNU**	1620 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	20 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	Disabled + Reset (Coldstart)		**Weightning**	
Start Firmware	1.20			
Configuration	Yes		**Revision**	1.3
Function Group	Feedback		**EEPROM**	-
Short Description	Calculate resolver phase			

4.9.5 COGGING

ASCII - Command	**COGGING**			
Syntax Transmit	COGGING [Data]			
Syntax Receive	COGGING <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer8		**CANBus Object Number**	36CF (hex)
DIM	-		**PROFIBUS PNU**	1663 (dec) IND = 0001xxxx (bin)
Range	0, 1		**DPR**	463 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	Disable + Restet (Coldstart)		**Weightning**	
Start Firmware	5.41			
Configuration	Yes		**Revision**	1.9
Function Group	Feedback		**EEPROM**	Yes
Short Description	Enable of Cogging Compensation			

Description

COGGING enables a cogging compensation function, which adds a cogging current to the current controller in function of the angle of the motor.

Before identifying the cogging parameter of the motor by CALCCOG [▸ 186], the parameter COGGING must be firstly set to "1".

When COGGING = 1, the cogging effect of the PM motor will be online compensated if the corresponding table was created by CALCCOG [▸ 186]. If COGGING = 0, the cogging compensation will be switched off and the table is erased. The cogging-compensation works only in FBTYPE [▸ 190] = 0, 2 and 4.

4.9.6 ENCCAPT

ASCII - Command	**ENCCAPT**			
Syntax Transmit	ENCCAPT [Data]			
Syntax Receive	ENCCAPT <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer8		**CANBus Object Number**	3531 (hex)
DIM	-		**PROFIBUS PNU**	1649 (dec) IND = 0000xxxx (bin)
Range	0, 1		**DPR**	49 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	Disabled		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.6
Function Group	Feedback		**EEPROM**	Yes
Short Description	no function			

Description

This command has been implemented for compatibility reasons.

4.9.7 FBTYPE

ASCII - Command	FBTYPE		
Syntax Transmit	FBTYPE [Data]		
Syntax Receive	FBTYPE <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	353B (hex)
DIM	-	**PROFIBUS PNU**	1659 (dec) IND = 0000xxxx (bin)
Range	0 .. 20	**DPR**	59 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	Disabled + Reset (Coldstart)	**Weightning**	
Start Firmware	1.20		
Configuration	Yes	**Revision**	2.0
Function Group	Feedback	**EEPROM**	Yes
Short Description	Selection of Encoder or Resolver		

Description

The FBTYPE command is used to select the type of feedback device. The type of encoder that is set is only initialized when the amplifier is switched on, which means that the amplifier must be switched off and then on again after every change of this variable.

The encoder parameters can be altered by using the appropriate ASCII command (in brackets) and then saved in the encoder EEPROM, using the HSAVE [▸ 198] command. The encoder commands (HISOFFS [▸ 196], HICOFFS [▸ 195], HIFACT1 [▸ 195], HSAVE [▸ 198]) are only available when communication with the connected encoder has been established.

If communication with the encoder is not possible, then the error message ENCODER FAULT F04 is displayed.

When the data are loaded from the encoder, the setting for the motor number in the encoder is compared with the internal setting (MNUMBER [▸ 233]). If the numbers are different, then an attempt is made to load a motor data set from the internal motor database that has the same motor number as that stored in the encoder. At the same time, the warning □NEW MOTOR DATA SET□ n12 is displayed.

In order to prevent a warning being generated at the next power-on, the latest MNUMBER [▸ 233] setting should be saved in the EEPROM, using the SAVE [▸ 51] command. If it was not possible to load a valid motor number from the encoder (for instance, when an encoder is used for the first time), then no motor data will be loaded. However, the n12 warning will still be generated. The HSAVE [▸ 198] command can be used to save the preset setting for the motor number (MNUMBER [▸ 233]) in the encoder, so that no warning will be produced at the next power-on.

When using an encoder without a parameter channel (FBTYPE [▸ 190]=7/16), and thus without the facility for storing parameters, the offset values HISOFFS [▸ 196] / HICOFFS [▸ 195] / HIFACT1 [▸ 195] will be saved in the serial EEPROM of the amplifier. After an alteration, these values can be permanently stored by using the SAVE [▸ 51] command.

Status	Type of Feed-back System	Description
FBTYPE = 0	Resolver	Data is loaded from the drive EEPROM.
FBTYPE = 2	Hiperface (Stegmann)	In the initialization phase, all the data is loaded that is stored in the encoder EEPROM. These are: Offset compensation Sine (HISOFFS [▸ 196]) Offset compensation Cosine (HICOFFS [▸ 195])

Status	Type of Feed-back System	Description
		Amplitude scaling (HIFACT1 [▶ 195]) Motor number (MNUMBER [▶ 233]) Motorphase (MPHASE [▶ 199])
FBTYPE = 3	Resolver, EnDAT oder Hiperface	Automatic selection of the feedback device. First the drive tries to communicate with an EnDAT device (FBTYPE [▶ 190]=4). If there is no reply, the drive tries to communicate with an Hiperface device (FBTYPE [▶ 190]=2). If there is also no communication, the drive selects resolver feedback (FBTYPE [▶ 190]=0) and starts to work. The setting of FBTYPE [▶ 190]=3 is not effected by this search. There is a possibility to read the selected type by the command "M [▶ 42] FBTYPE".
FBTYPE = 4	EnDAT (Heidenhain)	In the initialization phase, all the data is loaded that is stored in the encoder EEPROM. These are: Offset compensation Sine (HISOFFS [▶ 196]) Offset compensation Cosine (HICOFFS [▶ 195]) Amplitude scaling (HIFACT1 [▶ 195]) Motor number (MNUMBER [▶ 233]) Motorphase (MPHASE [▶ 199])
FBTYPE = 6	Sine/Cosine Encoder	Sine/Cosine encoder without parameter channel. MPHASE [▶ 199] is stored in the drive EEPROM. HISOFFS [▶ 196], HICOFFS [▶ 195], HIFACT1 [▶ 195] is also stored in the drive EEPROM.
FBTYPE = 7	Sine/Cosine Encoder	Sine/Cosine encoder without parameter channel. Automatic detection of MPHASE [▶ 199] by Wake&Shake mode. HISOFFS [▶ 196], HICOFFS [▶ 195], HIFACT1 [▶ 195] is also stored in the drive EEPROM. If the 24V are switched on or the drive has got a COLDSTART [▶ 170], a warning n14 is present. After enabling the drive, a wake & shake mode is activated, to get the commutation angle. The motor is doning a short move to do that. After that, the n14 is cleared.
FBTYPE=8	RS422 & Wake&Shake	This setting can only be used, if GEARMODE [▶ 213]=3 and ENCMODE [▶ 314]=0 (see ENCLINES [▶ 223]). If FPGA [▶ 77]=1 the position output at X5 (Drive 400 X4) gives the position information of the incremental encoder.
FBTYPE=9	RS422 Feedback Device MPHASE [▶ 199] is loaded out of the EEPROM	This setting can only be used, if GEARMODE [▶ 213]=3 and ENCMODE [▶ 314]=0 If FPGA [▶ 77]=1 the position output at X5 (Drive 400 X4) gives the position information of the incremental encoder.
FBTYPE=10	Without Feedback Device (sensorless)	
FBTYPE=11	Sine encoder feedback with hall's	
FBTYPE=12	RS422 feedback device (A quad B) with hall's	

Status	Type of Feed-back System	Description
FBTYPE=13	Hall's only	The hall sensor is used as only feedback. In this case, the drive can be operated in torque and also velocity mode (OPMODE [▸ 50] = 1,2 or 3,4). The performance at low speed is poor. To extend the low speed performance, use MVR [▸ 239] to enable the speed estimation at low speed.
FBTYPE = 16	Start-up with resolver (commutation), then switch over to Sine/Cosine encoder (FBTYPE [▸ 190]=7)	Commutation information read by resolver feedback. The switch-over to sine/cosine is done after a switch-on delay time together with the encoder simulation.

4.9.8 FBTYPEX

ASCII - Command	**FBTYPEX**			
Syntax Transmit	FBTYPEX			
Syntax Receive	FBTYPEX <Data>		**Available in**	
Type	Variable ro		**MMI**	No
ASCII Format	Integer8		**CANBus Object Number**	369B (hex)
DIM	-		**PROFIBUS PNU**	2011 (dec) IND = 0000xxxx (bin)
Range	Int8		**DPR**	411 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	-		**Weightning**	
Start Firmware	4.86			
Configuration	No		**Revision**	1.6
Function Group	Feedback		**EEPROM**	No
Short Description	Display the detected feedback device			

Description

The detected feedback device can be displayed by FBTYPEX. Especially using FBTYPE [▸ 190]=3, this object is useful to get the information, which feedback device was detected.

It is also possible to display if the drive has detected a multiturn encoder. If so, an offset of d100 is added to the feedback type.

E.g.: Endat multiturn was detected, FBTYPEX displays d104.

4.9.9 HACOFFS

ASCII - Command	HACOFFS	For Manufacturer Use only	
Syntax Transmit	HACOFFS [Data]		
Syntax Receive	HACOFFS <Data>	Available in	
Type	Variable rw	MMI	No
ASCII Format	Integer16	CANBus Object Number	354E (hex)
DIM	Millivolts	PROFIBUS PNU	1678 (dec) IND = 0000xxxx (bin)
Range	-10000 .. 10000	DPR	78 (dec)
Default	0		
Opmode	All	Data Type Bus/DPR	Integer16
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.3
Function Group	Feedback	EEPROM	Encoder
Short Description	Hiperface Cosinus Offset (absolut)		

Description

The HACOFFS command sets the offset correction (in mV) for the cosine signal of the absolute track (SinCoder)

The command is only available when a sin/cos encoder has been selected as the feedback device (FBTYPE [▶ 190]=2,4,7). Depending on the type of encoder used, the HACOFFS setting is stored in the EEPROM of the encoder (FBTYPE [▶ 190]=2,4, command HSAVE [▶ 198])

When using an encoder without a parameter channel (FBTYPE [▶ 190]=7), and thus without an internal EEPROM, this setting will be saved in the EEPROM of the amplifier (command SAVE [▶ 51]).

4.9.10 HAFACT1

ASCII - Command	HAFACT1	For Manufacturer Use only	
Syntax Transmit	HAFACT1 [Data]		
Syntax Receive	HAFACT1 <Data>	Available in	
Type	Variable rw	MMI	No
ASCII Format	Integer16	CANBus Object Number	354F (hex)
DIM	-	PROFIBUS PNU	1679 (dec) IND = 0000xxxx (bin)
Range	12000 .. 19000	DPR	79 (dec)
Default	16384		
Opmode	All	Data Type Bus/DPR	Integer16
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.3
Function Group	Feedback	EEPROM	Encoder
Short Description	Hiperface Gain Factor (absolut)		

Description

The HAFACT1 command sets the amplitude scaling for the sine signal of the absolute track (SinCoder). The amplitude scaling is for the value 16384 = 1.

The command is only available when a sin/cos encoder has been selected as the feedback device (FBTYPE [▸ 190]=2, 4, 7). Depending on the type of encoder used, the HACACT1 setting is stored in the EEPROM of the encoder (FBTYPE [▸ 190]=2, 4, command HSAVE [▸ 198]).

When using an encoder without a parameter channel (FBTYPE [▸ 190]=7), and thus without an internal EEPROM, this setting will be saved in the EEPROM of the amplifier (command SAVE [▸ 51]).

4.9.11 HASOFFS

ASCII - Command	**HASOFFS**		**For Manufacturer Use only**	
Syntax Transmit	HASOFFS [Data]			
Syntax Receive	HASOFFS <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer16		**CANBus Object Number**	3550 (hex)
DIM	Millivolts		**PROFIBUS PNU**	1680 (dec) IND = 0000xxxx (bin)
Range	-10000 .. 10000		**DPR**	80 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Feedback		**EEPROM**	Encoder
Short Description	Hiperface Sinus Offset (absolut)			

Description

The HASOFFS command sets the offset correction (in mV) for the sine signal of the absolute track (SinCoder).

The command is only available when a sin/cos encoder has been selected as the feedback device (FBTYPE [▸ 190]=2, 4, 7). Depending on the type of encoder used, the HASOFFS setting is stored in the EEPROM of the encoder (FBTYPE [▸ 190]=2, 4, command HSAVE [▸ 198]).

When using an encoder without a parameter channel (FBTYPE [▸ 190]=7), and thus without an internal EEPROM, this setting will be saved in the EEPROM of the amplifier (command SAVE [▸ 51]).

4.9.12 HDUMP

ASCII - Command	**HDUMP**			
Syntax Transmit	HDUMP			
Syntax Receive	HDUMP <Data>		**Available in**	
Type	Multi-line Return Command		**MMI**	No
ASCII Format	String		**CANBus Object Number**	3551 (hex)
DIM	-		**PROFIBUS PNU**	1681 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	81 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Feedback		**EEPROM**	-

Short Description	Output all sin/cos (Hiperface) variables

Description

Produces an output of all the sin/cos feedback variables.

4.9.13 HICOFFS

ASCII - Command	**HICOFFS**		**For Manufacturer Use only**	
Syntax Transmit	HICOFFS [Data]			
Syntax Receive	HICOFFS <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer16		**CANBus Object Number**	3552 (hex)
DIM	Millivolts		**PROFIBUS PNU**	1682 (dec) IND = 0000xxxx (bin)
Range	-10000 .. 10000		**DPR**	82 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Feedback		**EEPROM**	Yes
Short Description	Hiperface: Cosine-Offset (incremental track)			

Description

The HICOFFS command sets the offset correction (in mV) for the cosine signal of the incremental track.

The command is only available when a sin/cos encoder has been selected as the feedback device (FBTYPE [▸ 190]=2,4,7). Depending on the type of encoder used, the HICOFFS setting is stored in the EEPROM of the encoder (FBTYPE [▸ 190]=2,4, command HSAVE [▸ 198]).

When using an encoder without a parameter channel (FBTYPE [▸ 190]=7), and thus without an internal EEPROM, this setting will be saved in the EEPROM of the amplifier (command SAVE [▸ 51]).

4.9.14 HIFACT1

ASCII - Command	**HIFACT1**		**For Manufacturer Use only**	
Syntax Transmit	HIFACT1 [Data]			
Syntax Receive	HIFACT1 <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer16		**CANBus Object Number**	3553 (hex)
DIM	-		**PROFIBUS PNU**	1683 (dec) IND = 0000xxxx (bin)
Range	12000 .. 19000		**DPR**	83 (dec)
Default	16384			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Feedback		**EEPROM**	Encoder
Short Description	Hiperface: Sin/Cos Gain Factor (incremental track)			

Description

The HIFACT1 command sets the amplitude scaling for the sine signal of the absolute track (SinCoder). The amplitude scaling is for the value 16384 = 1.

The command is only available when a sin/cos encoder has been selected as the feedback device (FBTYPE [▸ 190]=2,4,7). Depending on the type of encoder used, the HIFACT1 setting is stored in the EEPROM of the encoder (FBTYPE [▸ 190]=2,4, command HSAVE [▸ 198]).

When using an encoder without a parameter channel (FBTYPE [▸ 190]=7), and thus without an internal EEPROM, this setting will be saved in the EEPROM of the amplifier (command SAVE [▸ 51]).

4.9.15 HISOFFS

ASCII - Command	**HISOFFS**		**For Manufacturer Use only**	
Syntax Transmit	HISOFFS [Data]			
Syntax Receive	HISOFFS <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer16		**CANBus Object Number**	3554 (hex)
DIM	Millivolts		**PROFIBUS PNU**	1684 (dec) IND = 0000xxxx (bin)
Range	-10000 .. 10000		**DPR**	84 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Feedback		**EEPROM**	Encoder
Short Description	Hiperface: Sin/Cos Offset (incremental track)			

Description

The HISOFFS command sets the offset correction (in mV) for the sine signal of the incremental track.

The command is only available when a sin/cos encoder has been selected as the feedback device (FBTYPE [▸ 190]=2,4,7). Depending on the type of encoder used, the HISOFFS setting is stored in the EEPROM of the encoder (FBTYPE [▸ 190]=2,4, command HSAVE [▸ 198]).

When using an encoder without a parameter channel (FBTYPE [▸ 190]=7), and thus without an internal EEPROM, this setting will be saved in the EEPROM of the amplifier (command SAVE [▸ 51]).

4.9.16 HRESET

ASCII - Command	HRESET		For Manufacturer Use only	
Syntax Transmit	HRESET			
Syntax Receive	HRESET		Available in	
Type	Command		MMI	No
ASCII Format	-		CANBus Object Number	3555 (hex)
DIM	-		PROFIBUS PNU	1685 (dec) IND = 0000xxxx (bin)
Range	-		DPR	85 (dec)
Default	-			
Opmode	All		Data Type Bus/DPR	-
Drive State	-		Weightning	
Start Firmware	1.30			
Configuration	No		Revision	1.3
Function Group	Feedback		EEPROM	-
Short Description	Hiperface: Load Default Parameters			

Description

The HRESET returns the default values. This command is only available when FBTYPE [▸ 190] = 2/4/7 has been set, and the amplifier detects a sin/cos encoder. If a SinCoder (Stegmann type SNS50/60) is detected, then the correction values from data field 5 will be transferred to the encoder variables (in other cases, the default values).

- FBTYPE [▸ 190] = 2 HIPERFACE (Stegmann)
- FBTYPE [▸ 190] = 4 EnDat (Heidenhain)
- FBTYPE [▸ 190] = 7 SINCOS □ encoder without its own serial EEPROM
- HRESET The HRESET sets the following default values.
  HACOFFS [▸ 193]
- HASOFFS [▸ 194]
- HAFACT1 [▸ 193]
- HICOFFS [▸ 195]
- HISOFFS [▸ 196]
- HIFACT1 [▸ 195]

4.9.17 HSAVE

ASCII - Command	HSAVE	For Manufacturer Use only	
Syntax Transmit	HSAVE		
Syntax Receive	HSAVE	Available in	
Type	Command	MMI	No
ASCII Format	-	CANBus Object Number	3556 (hex)
DIM	-	PROFIBUS PNU	1686 (dec) IND = 0000xxxx (bin)
Range	-	DPR	86 (dec)
Default	-		
Opmode	All	Data Type Bus/DPR	-
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.3
Function Group	Feedback	EEPROM	-
Short Description	Hiperface: Save Parameters in Encoder		

Description

If FBTYPE [▶ 190]=2 or =4 is set, the HSAVE [▶ 198] command saves the variables for the encoder (HIPERFACE / EnDat) in the serial EEPROM of the encoder. With the setting FBTYPE [▶ 190]=7, only the variables for the incremental track, as well as MNUMBER [▶ 233] and MPHASE [▶ 199] are saved in the serial EEPROM of the amplifier.

This command is only available if the amplifier has detected a sin/cos encoder (FBTYPE [▶ 190] =2;4;7).

- FBTYPE [▶ 190] = 2 HIPERFACE (Stegmann)
- FBTYPE [▶ 190] = 4 EnDat (Heidenhain)
- FBTYPE [▶ 190] = 7 SINCOS □ encoder without its own serial EEPROM

The HSAVE command saves the following variables.

- MNUMBER [▶ 233]*
- MPHASE [▶ 199]*
- HACOFFS [▶ 193]
- HASOFFS [▶ 194]
- HAFACT1 [▶ 193]
- HICOFFS [▶ 195]*
- HISOFFS [▶ 196]*
- HIFACT1 [▶ 195]*
    - * If FBTYPE = 7, these parameters are saved in the serial EEPROM of the amplifier.

      Starting with firmware 5.41, Hiperface encoder, which have contents, can be erased by "HSAVE ERASE".

4.9.18 MPHASE

ASCII - Command	**MPHASE**		
Syntax Transmit	MPHASE [Data]		
Syntax Receive	MPHASE <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	359C (hex)
DIM	Electrical Degrees	**PROFIBUS PNU**	1756 (dec) IND = 0000xxxx (bin)
Range	0 .. 360	**DPR**	156 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	Disabled	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Feedback	**EEPROM**	Yes
Short Description	Motor Phase, Feedback Offset		

Description

The MOTOR PHASE parameter is handled in different ways, depending on the type of feedback (FBTYPE [▶ 190]) that is used.

- FBTYPE=0 resolver MPHASE is saved in the serial EEPROM of the amplifier (SAVE [▶ 51] command) and is transferred after every power-on of the amplifier.
- FBTYPE=2, 4 Hiperface/Endat MPHASE is saved in the serial EEPROM of the encoder (HSAVE [▶ 198] command) and is read out from the encoder after every power-on of the amplifier. So if an encoder is exchanged, the MPHASE setting goes with the encoder. When a new encoder is fitted, the MPHASE value must be re-established and stored in the encoder (HSAVE command).
- FBTYPE=7 sin/cos encoder without an internal EEPROM
  MPHASE will be determined automatically at the first enable of the output stage (Wake & Shake)
  It is not necessary to make a separate determination of the MPHASE value, or to save it.

4.9.19 MRESBW

ASCII - Command	**MRESBW**		
Syntax Transmit	MRESBW [Data]		
Syntax Receive	MRESBW <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	35A0 (hex)
DIM	Hz	**PROFIBUS PNU**	1760 (dec) IND = 0000xxxx (bin)
Range	25 .. 1200	**DPR**	160 (dec)
Default	600		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	1.38		
Configuration	No	**Revision**	1.8
Function Group	Feedback	**EEPROM**	Yes
Short Description	Resolver Bandwidth		

Description

MRESBW is a tuning parameter that sets the bandwidth (in Hz) of the inner control loop. A high value (>800 Hz) results in a fast (low phase lag) and noisy velocity signal. A low value (<400 Hz) results in a slow (higher phase lag) and smooth velocity signal. The default value of 600 Hz is a compromise between phase lag and noise. The phase lag can be reduced by providing the acceleration feed forward signal (VLO [▶ 205] = 1).

- >= 1.57 for Resolver Feedback
- >= 3.10 for High Resolution Feedback

With a wide bandwidth, the drive responds more rapidly to control loop deviations and there is a smaller following error (reduced lag). A very wide bandwidth only makes sense with low moments of inertia, low KP, and very high acceleration values. A narrower bandwidth produces a filter effect. The rotational velocity and positional control are smoother (encoder equivalent output is quieter as well).

For the sensor less drive, the Luenberger Observer is used as the adaptive controller. Therefore, the parameter MRESBW corresponds to the bandwidth of the adaptive controller. It is normally set between 25 and 100 Hz.

4.9.20 MRESD

ASCII - Command	**MRESD**		
Syntax Transmit	MRESD [Data]		
Syntax Receive	MRESD <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Float	**CANBus Object Number**	3697 (hex)
DIM	-	**PROFIBUS PNU**	2007 (dec) IND = 0000xxxx (bin)
Range	0.5 .. 2	**DPR**	407 (dec)
Default	1		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	4.78		
Configuration	No	**Revision**	1.6
Function Group	Feedback	**EEPROM**	Yes
Short Description	Damping of the Luenberger Observer		

Description

MRESD is a variable to ensure the firmware compatibility for the bandwidth of the Luenberger velocity observer.

If the parameter settings for the firmware version from 3.00 to 3.38 and from 4.00 to 4.77 are used for the firmware version above 4.78, this parameter should be set to 0.5.

4.9.21 MRESPOLES

ASCII - Command	**MRESPOLES**			
Syntax Transmit	MRESPOLES [Data]			
Syntax Receive	MRESPOLES <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer8		**CANBus Object Number**	35A1 (hex)
DIM	-		**PROFIBUS PNU**	1761 (dec) IND = 0000xxxx (bin)
Range	2, 4, .. 32		**DPR**	161 (dec)
Default	2			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	Disabled		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.6
Function Group	Feedback		**EEPROM**	Yes
Short Description	Number of Resolver Poles (Multispeed)			

Description

The number of resolver poles (multispeed resolver) per turn.

4.9.22 READNIMP

ASCII - Command	**READNIMP**			
Syntax Transmit	READNIMP			
Syntax Receive	READNIMP		**Available in**	
Type	Command		**MMI**	No
ASCII Format	-		**CANBus Object Number**	35DC (hex)
DIM	-		**PROFIBUS PNU**	1820 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	220 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.67			
Configuration	No		**Revision**	1.8
Function Group	Feedback		**EEPROM**	-
Short Description	Read/Set the EEO (ROD) Zero-Pulse Offset			

Description

READNIMP calculates the actual position, depending on the resolution set for the Encoder Equivalent Output (EEO), and enters it as the ENCZERO [▶ 316] variable. This function ensures that the Encoder Equivalent Output (EEO) zero-pulse is always generated at the actual position (within a single turn). If this setting is to be permanently accepted, use the SAVE [▶ 51] command (save in the serial EEPROM).

4.9.23 RESPHASE

ASCII - Command	**RESPHASE**			
Syntax Transmit	RESPHASE [Data]			
Syntax Receive	RESPHASE <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer16		**CANBus Object Number**	35E5 (hex)
DIM	-		**PROFIBUS PNU**	1829 (dec) IND = 0000xxxx (bin)
Range	-300 .. 50		**DPR**	229 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.6
Function Group	Feedback		**EEPROM**	Yes
Short Description	Resolver Phase			

4.9.24 RK

ASCII - Command	**RK**			
Syntax Transmit	RK [Data]			
Syntax Receive	RK <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer16		**CANBus Object Number**	35E6 (hex)
DIM	Counts		**PROFIBUS PNU**	1830 (dec) IND = 0000xxxx (bin)
Range	12000 ..19000		**DPR**	230 (dec)
Default	16384			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Feedback		**EEPROM**	Yes
Short Description	Gain Adjust for Resolver Sine Signal			

Description

The RK parameter can be used to correct any amplitude difference that may exist between the sine and cosine signals from the resolver. The relationships are as follows:

- RK = 16384 no alteration of the amplitude of the sine signal
- RK < 16384 sine signal amplitude is reduced
- RK > 16384 sine signal amplitude is increased

An incorrect setting of this correction factor will result in velocity/velocity variations (ripple) which are strongly dependent on the position.

The CALCRK [▸ 188] command enables an automatic determination of the correction factor RK.

This value will not be changed by a parameter download, since it only depends on the equipment.

4.9.25 ROFFS0

ASCII - Command	**ROFFS0**			
Syntax Transmit	ROFFS0 [Data]			
Syntax Receive	ROFFS0 <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer32		**CANBus Object Number**	365C (hex)
DIM	-		**PROFIBUS PNU**	1948 (dec) IND = 0000xxxx (bin)
Range	long int		**DPR**	348 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	3.43			
Configuration	No		**Revision**	1.3
Function Group	Feedback		**EEPROM**	Yes
Short Description	Reference Offset for the second Encoder Feedback			

Description

The command ROFFS0 is a reference position of the second encoder. The position is set to this position when a successful homing move is done. After the homing move, external position can be read by PFB0 [▸ 27]. This function is only available in mode EXTPOS [▸ 253] 2 and 3.

4.9.26 SMNUMBER

ASCII - Command	**SMNUMBER**			
Syntax Transmit	SMNUMBER [Data]			
Syntax Receive	SMNUMBER <Data>		**Available in**	
Type	Variable r		**MMI**	No
ASCII Format	Integer16		**CANBus Object Number**	3695 (hex)
DIM	-		**PROFIBUS PNU**	2005 (dec) IND = 0000xxxx (bin)
Range	0 .. 32767		**DPR**	405 (dec)
Default	0			
Opmode	-		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	4.74			
Configuration	No		**Revision**	1.5
Function Group	Feedback		**EEPROM**	No
Short Description	Stored Motor Number in the feedback Device			

Description

SMNUMBER gives the motor number, that is stored in the feedback device (EnDAT or HIPERFACE).

This Object makes sense with FBTYPE [▸ 190] = 2 or 4, otherwise "0" is returned.

4.9.27 SSIGRAY

ASCII - Command	**SSIGRAY**		
Syntax Transmit	SSIGRAY [Data]		
Syntax Receive	SSIGRAY <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	35F6 (hex)
DIM	-	**PROFIBUS PNU**	1846 (dec) IND = 0000xxxx (bin)
Range	0, 1	**DPR**	246 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	Disabled	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	2.0
Function Group	Feedback	**EEPROM**	Yes
Short Description	Select SSI Code		

Description

The SSIGRAY command can be used to define the format to be used for the output of the SSI information on connector X5 (Drive 400 X4).

- SSIGRAY=0 binary code
- SSIGRAY=1 Gray code

4.9.28 SSIINV

ASCII - Command	**SSIINV**		
Syntax Transmit	SSIINV [Data]		
Syntax Receive	SSIINV <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	35F7 (hex)
DIM	-	**PROFIBUS PNU**	1847 (dec) IND = 0000xxxx (bin)
Range	0, 1	**DPR**	247 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	Disabled	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	2.0
Function Group	Feedback	**EEPROM**	Yes
Short Description	SSI Clock		

Description

The behavior of the SSI interface at X5 (Drive 400 X4).
The SSIINV command has different interpretations, depending on whether the SSI is configured as an output or an input.

1. SSI output (GEARMODE [▸ 213] != 7, ENCMODE [▸ 314] = 2)
   SSIINV=0 normal clock level
   SSIINV=1 inverted clock level

2. SSI read-in (GEARMODE [▶ 213] = 7, ENCMODE [▶ 314] = 2)
   SSIINV=0 MSB transmitted first
   SSIINV=1 LSB transmitted first

4.9.29 SSIOUT

ASCII - Command	**SSIOUT**			
Syntax Transmit	SSIOUT [Data]			
Syntax Receive	SSIOUT <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer8		**CANBus Object Number**	35F9 (hex)
DIM	-		**PROFIBUS PNU**	1849 (dec) IND = 0000xxxx (bin)
Range	0 .. 31		**DPR**	249 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	Disabled		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	2.0
Function Group	Feedback		**EEPROM**	Yes
Short Description	SSI Baud Rate			

Description

The SSIOUT command has different interpretations, depending on whether the SSI is configured as an output or an input at X5 (Drive 400 X4).

1. SSI output (GEARMODE [▶ 213] != 7, ENCMODE [▶ 314] = 2)
   SSIOUT = 0 baud rate 200 kbaud
   SSIOUT = 1 baud rate 1 Mbaud
2. SSI read-in (GEARMODE [▶ 213] = 7, ENCMODE [▶ 314] = 2)
   SSIOUT = number of data bits (25)

4.9.30 VLO

ASCII - Command	**VLO**			
Syntax Transmit	VLO [Data]			
Syntax Receive	VLO <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Float		**CANBus Object Number**	363D (hex)
DIM	-		**PROFIBUS PNU**	1917 (dec) IND = 0000xxxx (bin)
Range	0.0 .. 5.0		**DPR**	317 (dec)
Default	1.0			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	2.49			
Configuration	No		**Revision**	1.6
Function Group	Feedback		**EEPROM**	Yes
Short Description	Software Resolver/Digital Converter Feedforward			

Description

VLO is a parameter of the Luenberger Velocity Observer. To reduce the delay of the derivation the observer can be served with the torque component of the current. The effective inertia is estimated by the gain of the velocity loop (GV). Setting VLO to zero the acceleration will not influence the observer. With a value of 1 the acceleration is full enabled. With VLO 0.5 the Observer will use 50% of the acceleration torque. Reducing VLO can result in an instable velocity loop.

4.9.31 WSAMPL

ASCII - Command	**WSAMPL**			
Syntax Transmit	WSAMPL [Data]			
Syntax Receive	WSAMPL <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer32		**CANBus Object Number**	36D1 (hex)
DIM	-		**PROFIBUS PNU**	1665 (dec) IND = 0001xxxx (bin)
Range	0 .. 2^31-1		**DPR**	465 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	5.41			
Configuration	No		**Revision**	1.9
Function Group	Feedback		**EEPROM**	Yes
Short Description	Minimum Move of W&S Mode			

Description

Gives the minimum Move for W&S - function in FBTYPE [▶ 190] = 7 and 8. The move is given in internal counts.

In WSAMPL = 0, the calculation is automatically done with ENCLINES [▶ 223]

4.9.32 WSTIME

ASCII - Command	**WSTIME**			
Syntax Transmit	WSTIME [Data]			
Syntax Receive	WSTIME <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer16		**CANBus Object Number**	36D0 (hex)
DIM	msec		**PROFIBUS PNU**	1664 (dec) IND = 0001xxxx (bin)
Range	0 .. 100		**DPR**	464 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	5.41			
Configuration	No		**Revision**	1.9
Function Group	Feedback		**EEPROM**	Yes
Short Description	Action Time of the W&S - Funktion			

Description

WSTIME defines the action time of the W&S - function in FBTYPE [▶ 190] = 7 and 8. The different current vectors are switched on for that time and increase the move distance. See also WSAMPL [▶ 206].

If WSTIME is set to "0", the calculation depending on GV [▸ 332] is done automatically.

4.10 Fieldbus

4.10.1 DPRILIMIT

ASCII - Command	**DPRILIMIT**			
Syntax Transmit	DPRILIMIT [Data]			
Syntax Receive	DPRILIMIT <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Int16		**CANBus Object Number**	3658 (hex)
DIM	-		**PROFIBUS PNU**	1944 (dec) IND = 0000xxxx (bin)
Range	0 .. 3280		**DPR**	344 (dec)
Default	3280			
Opmode	All		**Data Type Bus/DPR**	Int16
Drive State	-		**Weightning**	
Start Firmware	4.96			
Configuration	No		**Revision**	1.7
Function Group	Fieldbus		**EEPROM**	No
Short Description	Digital Limiting of the peak Current via DPR			

Description

Digital Limiting of the peak current via DPR.

The scaling is:

- DPRILIMIT=3280 Current limited to DIPEAK
- DPRILIMIT=0 Current limited to 0 A

If the drive is switched on, DPRILIMIT is set to 3280 (no current limit). DPRILIMIT is not stored in EEPROM. So, to enable the limit, write the data to the variable via fieldbus, RS232 or I/O command buffer.

To enable this function, DILIM must be set to "1".

4.10.2 INTERPOL

ASCII - Command	**INTERPOL**			
Syntax Transmit	INTERPOL [Data]			
Syntax Receive	INTERPOL <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer8		**CANBus Object Number**	3684 (hex)
DIM	-		**PROFIBUS PNU**	1988 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2		**DPR**	388 (dec)
Default	0			
Opmode	5, 6		**Data Type Bus/DPR**	Integer8
Drive State	Disabled + Reset (Coldstart)		**Weightning**	
Start Firmware	4.78			
Configuration	Yes		**Revision**	1.5
Function Group	Fieldbus		**EEPROM**	Yes
Short Description	Type of Interpolation in OPMODE 5 and 6			

Description

INTERPOL defines the type of interpolation for external trajectory mode (OPMODE [▶ 50] 5 and 6). This functionality can only be used, selecting the synchronization that can be activated by SYNCSRC [▶ 209].

- 0: Linear Interpolation
- 1: (reserved) Sercos Spline Interpolation
- 2: Interpolation 2. Order for CAN

4.10.3 RXPDO1A

ASCII - Command	**RXPDO1A**		
Syntax Transmit	RXPDO1A [Data]		
Syntax Receive	RXPDO1A <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Single Line Multi String	**CANBus Object Number**	No
DIM	-	**PROFIBUS PNU**	No
Range	-	**DPR**	No
Default	-		
Opmode	All	**Data Type Bus/DPR**	Single Line Multi String
Drive State	-	**Weightning**	
Start Firmware	5.00		
Configuration	No	**Revision**	2.0
Function Group	Fieldbus	**EEPROM**	Yes
Short Description	RX-PDO 1 parameter selection		

Description

By the command RXPDO1A the listed CANopen-PDO RX-PDO 1 parameter can be selected (in brackets: corresponding SDOs by controlling via CAN):

1. Selection of the used PDO-Mappings for Receive-PDO 1 (2600, 0), decimal.
2. COB-Identifier for Receive-PDO 1 (1400, 1), hexadecimal.
3. Transmission type for Receive-PDO 1 (1400, 2), decimal.
4. Inhibit time for Receive-PDO 1 (1400, 3), decimal.
5. Priority group for Receive-PDO 1 (1400, 4), decimal.

Example

Read the actual status: Command: RXPDO1A
Write: Command: RXPDO1A 1 0x201 255 0 2, all parameters have to be set

4.10.4 RXPDO1B

ASCII - Command	**RXPDO1B**			
Syntax Transmit	RXPDO1A [Data]			
Syntax Receive	RXPDO1B <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Single Line Multi String		**CANBus Object Number**	No
DIM	-		**PROFIBUS PNU**	No
Range	-		**DPR**	No
Default	-			
Opmode	All		**Data Type Bus/DPR**	Single Line Multi String
Drive State	-		**Weightning**	
Start Firmware	5.00			
Configuration	No		**Revision**	2.0
Function Group	Fieldbus		**EEPROM**	Yes
Short Description	RX-PDO 1 Mapping Settings			

Description

The CANopen-PDOs TX-PDO 1mapping can be set by the command RX PDO 1 (in brackets: corresponding SDOs):

This is only possible, if free PDO mapping was selected, for example by RXPDO1A.

To read the actual mapping the command has to be set to: RXPDO1B

The mapping form is: 6040002

The format is different to the bus format. The syntax of the mapping-values xxxxyyz is:

- xxxx Hex-number for SDO index (for the example 6040)
- yy Hex-number for SDO subindex (for the example 00)
- z number for byte quantity in SDO setting (for the example 2 Byte = 16 Bit)

The input of free mappable PDO is analog to the output, for example: TXPDO1B 6041002 6061001 for CANopen status word setting and the CANopen-OPMODE via TX-PDO 1.

4.10.5 SYNCSRC

ASCII - Command	**SYNCSRC**			
Syntax Transmit	SYNCSRC [Data]			
Syntax Receive	SYNCSRC <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer16		**CANBus Object Number**	3683 (hex)
DIM	-		**PROFIBUS PNU**	1987 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2, 3		**DPR**	387 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	Disabled + Reset (Coldstart)		**Weightning**	
Start Firmware	4.78			
Configuration	Yes		**Revision**	1.5
Function Group	Fieldbus		**EEPROM**	Yes

Short Description	Source for Fieldbus Synchronization

Description

This parameter defines the source for the synchronization of the control loops to the external fieldbus. Synchronization via CAN needs a special hardware PLL which is included in a special FPGA [▶ 77] program, enabled by FPGA [▶ 77]=3.

- 0: No synchronization
- 1: (reserved) synchronization via Sercos
- 2: Synchronization via KS3000 Fire-Wire option board
- 3: Synchronization via CANopen

4.10.6 TXPDO1A

ASCII - Command	TXPDO1A		
Syntax Transmit	RXPDO1A [Data]		
Syntax Receive	TXPDO1A <Data>	Available in	
Type	Variable rw	MMI	No
ASCII Format	Single Line Multi String	CANBus Object Number	No
DIM	-	PROFIBUS PNU	No
Range	-	DPR	No
Default	-		
Opmode	All	Data Type Bus/DPR	Single Line Multi String
Drive State	-	Weightning	
Start Firmware	5.00		
Configuration	No	Revision	2.0
Function Group	Fieldbus	EEPROM	Yes
Short Description	TX-PDO1 Mapping - Setup		

Description

By the command TXPDO1A the listed CANopen-PDO TX-PDO 1 parameter can be selected (in brackets: corresponding SDOs by controlling via CAN):

1. Selection of the used PDO-Mappings for Transmit -PDO 1 (2A00, 0), decimal.
2. COB-Identifier for Transmit-PDO 1 (1400, 1), hexadecimal.
3. Transmission type for Transmit-PDO 1 (1400, 2), decimal.
4. Inhibit time for Transmit-PDO 1 (1400, 3), decimal.
5. Priority group for Transmit-PDO 1 (1400, 4), decimal.

Example

Read the actual status: Command: TXPDO1A

Write: Command: TXPDO1A 1 0x201 255 0 2 0XFFFFFFFF 0XFFFFFFFF,
all parameter have to be set

4.10.7 TXPDO1B

ASCII - Command	**TXPDO1B**		
Syntax Transmit	RXPDO1A [Data]		
Syntax Receive	TXPDO1B <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Single Line Multi String	**CANBus Object Number**	No
DIM	-	**PROFIBUS PNU**	No
Range	-	**DPR**	No
Default	-		
Opmode	All	**Data Type Bus/DPR**	Single Line Multi String
Drive State	-	**Weightning**	
Start Firmware	5.00		
Configuration	No	**Revision**	2.0
Function Group	Fieldbus	**EEPROM**	Yes
Short Description	TX-PDO1 Mapping - Setup		

Description

The CANopen-PDOs TX-PDO 1mapping can be set by the command TXPDO1B (in brackets: corresponding SDOs):

This is only possible, if free PDO mapping was selected, for example by TXPDO1A.

To read the actual mapping the command has to be set to: TXPDO1B The mapping form is: TXPDO1B 6041002

The format is different to the bus format. The syntax of the mapping-values xxxxyyz is:

- xxxx Hex-number for SDO index (im Beispiel 6040)
- yy Hex-number for SDO subindex (im Beispiel 00)
- z number for byte quantity in SDO setting (im Beispiel 2 Byte = 16 Bit)

The input of free mappable PDO is analog to the output, for example: TXPDO1B 6041002 6061001 for CANopen status word setting and the CANopen-OPMODE via TX-PDO 1.

4.11 Gearing

4.11.1 ENCIN

ASCII - Command	**ENCIN**		
Syntax Transmit	ENCIN [Data]		
Syntax Receive	ENCIN <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	3532 (hex)
DIM	Pulse/Umdr.	**PROFIBUS PNU**	1650 (dec) IND = 0000xxxx (bin)
Range	256,512,.. ,65536	**DPR**	50 (dec)
Default	4096		
Opmode	4	**Data Type Bus/DPR**	Integer32
Drive State	Disabled + Reset (Coldstart)	**Weightning**	
Start Firmware	1.20		
Configuration	Yes	**Revision**	1.3
Function Group	Gearing	**EEPROM**	Yes
Short Description	Encoder Pulse Input		

Description

In master/slave applications (OPMODE [▸ 50]=4) this parameter can be used to set the number of EEO (ROD) pulses per turn of the encoder. For pulse numbers that cannot be represented as a binary power, a □nearby□ pulse number must be entered. The difference in the resolution can then be adjusted by using the gearing factor, e.g.

Number of pulses = 500

- ENCIN [▸ 212]=512
- GEARI [▸ 212]=500
- GEARO [▸ 218]=512

4.11.2 GEARI

ASCII - Command	**GEARI**		
Syntax Transmit	GEARI [Data]		
Syntax Receive	GEARI <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	353E (hex)
DIM	-	**PROFIBUS PNU**	1662 (dec) IND = 0000xxxx (bin)
Range	1 .. 32767	**DPR**	62 (dec)
Default	8192		
Opmode	4	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Gearing	**EEPROM**	Yes
Short Description	Input Factor for Electronic Gearing		

Description

In master/slave applications (OPMODE [▸ 50]=4) this parameter can be used to set the master/slave translation ratio.

The relationship is as follows:

- For PRBASE [▸ 286]=20
  distance to move = input pulses * 1048576 / ENCIN [▸ 212] * GEARO [▸ 218] / GEARI
- For PRBASE [▸ 286]=16
  distance to move = input pulses * 65536 / ENCIN [▸ 212] * GEARO [▸ 218] / GEARI

The □distance to move□ is always referred to the resolution that has been set for the position control loop (PRBASE [▸ 286]) (65536 pulses / motor turn for PRBASE [▸ 286]=16 or 1048576 pulses / motor turn for PRBASE [▸ 286]=20).

4.11.3 GEARMODE

ASCII - Command	**GEARMODE**		
Syntax Transmit	GEARMODE [Data]		
Syntax Receive	GEARMODE <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	353F (hex)
DIM	-	**PROFIBUS PNU**	1663 (dec) IND = 0000xxxx (bin)
Range	0 .. 17	**DPR**	63 (dec)
Default	6		
Opmode	4	**Data Type Bus/DPR**	Integer8
Drive State	Disabled	**Weightning**	
Start Firmware	1.20		
Configuration	Yes	**Revision**	2.0
Function Group	Gearing	**EEPROM**	Yes
Short Description	Electronic Gearing Mode		

Description

The servo amplifier is controlled through different interfaces from various sources. The GEARMODE variable configures the source that provides the master setpoint (position). For the connector pin assignments, see the Installation Manual.

Starting with firmware 4.96, all devices (resolver (X2) (Drive 400 X5), SinCos (X1) (Drive 400 X2) and incremental signals can be used at the same time. Resolver for commutation and speed control, SinCos for position control and incremental encoder for electronic gearing.
Following settings must be made:

- FPGA [▸ 77]=1 (advanced FPGA-program with second counter)
- EXTPOS [▸ 253]=1 .. 3 (External actual position)
- GEARMODE=10 ..17

In addition to GEARMODE 0 ..7, a sine encoder at connector X1 (Drive 400 X2) can be used for position control with EXTPOS = 1.

Status	Description
GEARMODE=0	Encoder Follower Digital I/O 24V (X3) With an incremental encoder (track A/B, 24V signal level) connected to the digital inputs DIGITAL-IN 1/2, terminals X3/11, 12, an additional function assignment for the inputs is not necessary and any assignments on the screen page, Digital I/O, are ignored.

Status	Description
GEARMODE=1	Pulse And Direction Digital I/O 24V (X3) With a stepper motor control (pulse/direction, 24V signal level) connected to the digital inputs DIGITAL-IN 1/2, terminals X3/11, 12, an additional function assignment for the inputs is not necessary and any assignments on the screen page, Digital I/O, are ignored. INPUT1=direction (Low = positive, High = negative) INPUT2=pulse
GEARMODE=2	Encoder Follower Digital I/O 24V (X3) With an incremental encoder (track A/B, 24V signal level) connected to the digital inputs DIGITAL-IN 1/2, terminals X3/11, 12, an additional function assignment for the inputs is not necessary and any assignments on the screen page, Digital I/O, are ignored.
GEARMODE=3	Encoder Follower Digital I/O 5V X5 (Drive 400 X4) With an incremental encoder connected to connector X5 (Drive 400 X4), terminals 4, 5, 6, 7. ENCMODE [▶ 314] has to be set to "0".
GEARMODE=4	Pulse And Direction Digital I/O 5V X5 (Drive 400 X4) With a stepper motor control connected to connector X5 (Drive 400 X4), terminals 4, 5, 6, 7. INPUT1=direction (Low = positive, High = negative) INPUT2=pulse ENCMODE [▶ 314] has to be set to "0"
GEARMODE=5	Encoder Follower Digital I/O 5V X5 (Drive 400 X4) With an incremental encoder connected to connector X5 (Drive 400 X4), terminals 4, 5, 6, 7. ENCMODE [▶ 314] has to be set to "0".
GEARMODE=6	With a sine encoder connected to X1 (Drive 400 X2). Only the zero crossing of the sine(cosine signals are used. No analog processing.

Status	Description
GEARMODE=7	SSI input X5 (Drive 400 X4). For the Master/Slave mode with two drives you need settings as follow: Master: (is sending the SSI position) ENCMODE [▸ 314] 2 setting for encoder emulation (1=ROD, 2=SSI) SSIGRAY [▸ 204] = 0 data format (0=binary/1=gray) SSIINV [▸ 204] = 1 SSI-Clock (0=standard, 1=inverted) SSIMODE [▸ 316] = 1 0=single turn / 1= multi turn SSIOUT [▸ 205] = 0 baudrate 0=200 Kbaud / 1=1MBaud Slave: (is reading the SSI position) GEARMODE = 7 Setting for the master interface OPMODE [▸ 50] = 4 Master/Slave mode ENCMODE [▸ 314] = 2 always 2 when GEARMODE = 7 (SSI) SSIGRAY [▸ 204] = 0 data format (0=binary/1=gray) SSIINV [▸ 204] = 0 start transmission with MSB (=0) or LSB (=1) SSIMODE [▸ 316] = 0 alarm bit at begin (=1) or at end (=2) or off (=0) SSIOUT [▸ 205] = 25 data bits□1 (26) IN1MODE [▸ 116] = 16 Start input for the motion task IN1TRIG [▸ 122] = 0 0 means: motion task is a homing move NREF [▸ 267] = 8 Number for homing move. The settings, GEARMODE = 7 and OPMODE [▸ 50] =4, activate the read of the SSI-position about the encoder-input. The drive reads the SSI-position every 250 µs and calculates the difference to the previous position. This difference is multiplied by a scaling-factor and added to the last position command value. With PRBASE [▸ 286] = 20: Scaling-factor = 2^(33-SSIOUT [▸ 205])* GEARO [▸ 218] / GEARI [▸ 212] With SSIOUT [▸ 205] = 25: Scaling-factor = 256*GEARO [▸ 218] / GEARI [▸ 212] The absolute position from Master/Slave could move with ROFFS [▸ 293]. To adjust the absolute position between Master and Slave, it is necessary to do a homing with the slave drive and NREF [▸ 267] = 8 (start with digital input). At the beginning of the homing the Slave drive reads the absolute position from the Master, does the scaling and uses this position as the target position for the homing. The drive changes the operation mode to OPMODE [▸ 50] = 8 and starts the homing move to the target position with VREF [▸ 310] and ramps ACCR [▸ 247] / DECR [▸ 249]. When the drive reaches the target position, it sets the INPOSITION message. The PLC resets the start input to activate the Master/Slave-mode (OPMODE [▸ 50] = 4). For testing (with the setting MSG [▸ 99] = 2), it is possible to display the Master SSI-position at the Slave drive with the command, "M [▸ 42] NEWSSI," in the terminal program of the drive. It is possible to get the Slave position with the command, "M [▸ 42] PFB."
GEARMODE=8	EnDAT-Encoder at input X1 (Drive 400 X2). The difference to GEARMODE=6 is, that the sine/cosine signals of the encoder are read analog. This increases the resolution significantly.
GEARMODE=9	EnDAT-Encoder at input X1 (Drive 400 X2). The difference to GEARMODE=6 is, that the parameter channel of the encoder is read and the absolute position is transfered to the position register. ENCLINES [▸ 223] is calculated automatically to thie internal resolution of 20 Bit per rev of the encoder. This setting can be used in position mode under EXTPOS [▸ 253]=1. The sine/cosine signals of the encoder are read analog. This increases the resolution significantly.
GEARMODE=10	Encoder follower X3, 24V Sine encoder at X1 (Drive 400 X2) for position control (EXTPOS [▸ 253] = 1)

Status	Description
GEARMODE=11	Pulse and direction X3, 24V INPUT1=direction (Low=positive, High=negative) INPUT2=Pulse Sine encoder at X1 (Drive 400 X2) for position control (EXTPOS [▸ 253] = 1)
GEARMODE=12	Encoder Follower Digital I/O 24V (X3) With an incremental encoder (track A/B, 24V signal level) connected to the digital inputs DIGITAL-IN 1/2, terminals X3/11, 12, an additional function assignment for the inputs is not necessary and any assignments on the screen page, Digital I/O, are ignored. In addition to that, a sine encoder at X1 (Drive 400 X2) can be used for position control (EXTPOS [▸ 253] = 1)
GEARMODE=13	Encoder follower X5 (Drive 400 X4), 5V Sine encoder at X1 (Drive 400 X2) for position control (EXTPOS [▸ 253] = 1)
GEARMODE=14	Pulse and direction X5 (Drive 400 X4), 5V Sine encoder at X1 (Drive 400 X2) for position control (EXTPOS [▸ 253] = 1)
GEARMODE=15	Encoder Follower Digital I/O 5V (X5) With an incremental encoder connected to connector X5, terminals 4, 5, 6, 7. ENCMODE [▸ 314] has to be set to "0". In addition to that, a sine encoder at X1 (Drive 400 X2) for position control (EXTPOS [▸ 253] = 1)
GEARMODE=16	With a sine encoder connected to X1 (Drive 400 X2). Only the zero crossing of the sine(cosine signals are used. No analog processing.

Status	Description
GEARMODE=17	SSI input X5 (Drive 400 X4). For the Master/Slave mode with two drives you need settings as follow: Master: (is sending the SSI position) ENCMODE [▸ 314] = 2 setting for encoder emulation (1=ROD, 2=SSI) SSIGRAY [▸ 204] = 0 data format (0=binary/1=gray) SSIINV [▸ 204] = 1 SSI-Clock (0=standard, 1=inverted) SSIMODE [▸ 316] = 1 0=single turn / 1= multi turn SSIOUT [▸ 205] = 0 baudrate 0=200 Kbaud / 1=1MBaud Slave: (is reading the SSI position) GEARMODE = 7 Setting for the master interface OPMODE [▸ 50] = 4 Master/Slave mode ENCMODE [▸ 314] = 2 always 2 when GEARMODE = 7 (SSI) SSIGRAY [▸ 204] =0 data format (0=binary/1=gray) SSIINV [▸ 204] = 0 start transmission with MSB (=0) or LSB (=1) SSIMODE [▸ 316] = 0 alarm bit at begin (=1) or at end (=2) or off (=0) SSIOUT [▸ 205] = 25 data bits□1 (26) IN1MODE [▸ 116] = 16 Start input for the motion task IN1TRIG [▸ 122] = 0 0 means: motion task is a homing move NREF [▸ 267] = 8 Number for homing move The settings GEARMODE = 7 and OPMODE [▸ 50] =4 activate the function read the SSI-position about the encoder-input. The drive read then the SSI-position every 250 µs and calculate the difference to the old position before. This difference will be multiply with a scaling-factor and add to the last position command value. Scaling-factor = 2^(33-SSIOUT [▸ 205])* GEARO [▸ 218] / GEARI [▸ 212] ; with PRBASE [▸ 286]=20 With SSIOUT [▸ 205]=25: Scaling-factor = 256*GEARO [▸ 218] / GEARI [▸ 212] The absolute position from Master/Slave could move with the parameter ROFFS [▸ 293]. To adjust the absolute position between Master and Slave, it is necessary to do a homing with the slave drive and NREF [▸ 267] = 8 (start with digital input). At the beginning of the homing the Slave drive read the absolute position from the Master, do the scaling and use this position as his target position for the homing. Then the drive change the operation mode to OPMODE [▸ 50] = 8 and start the homing move, to the target position with VREF [▸ 310] and the ramps ACCR [▸ 247] / DECR [▸ 249]. When the drive reach the target position he set the INPOSITION message. Then the PLC should reset the start input, to activate the Master/Slave-mode (OPMODE [▸ 50] = 4) again. For testing (with the setting MSG [▸ 99] = 2) it is possible to display the Master SSI-position at the Slave drive with the command □M [▸ 42] NEWSSI□ in the terminal program of the drive. It is possible to get the Slave position with the command "M [▸ 42] PFB". The relation between this positions is corresponding to the scaling-factor. Sine encoder at X1 (Drive 400 X2) for position control (EXTPOS [▸ 253] = 1)

Also see about this

- MH [▸ 264]
- ROFFS2 [▸ 218]

4.11.4 GEARO

ASCII - Command	**GEARO**		
Syntax Transmit	GEARO [Data]		
Syntax Receive	GEARO <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	3540 (hex)
DIM	-	**PROFIBUS PNU**	1664 (dec) IND = 0000xxxx (bin)
Range	-32767 .. 32767	**DPR**	64 (dec)
Default	8192		
Opmode	4	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Gearing	**EEPROM**	Yes
Short Description	Output Factor for Electronic Gearing		

Description

In master/slave applications (OPMODE [▶ 50]=4) this parameter can be used to set the master/slave translation ratio.

The relationship is as follows:

- For PRBASE [▶ 286]=20
  distance to move = input pulses * 1048576 / ENCIN [▶ 212] * GEARO [▶ 218] / GEARI [▶ 212]
- For PRBASE [▶ 286]=16
  distance to move = input pulses * 65536 / ENCIN [▶ 212] * GEARO [▶ 218] / GEARI [▶ 212]

The □distance to move□ is always referred to the resolution that has been set for the position control loop (PRBASE [▶ 286]) (65536 pulses / motor turn for PRBASE [▶ 286]=16 or 1048576 pulses / motor turn for PRBASE [▶ 286]=20).

If a negative value is entered for GEARO, the slave runs in the opposite direction to the master.

With the configuration ANCNFG [▶ 60]=6 the GEARO parameter can be influenced by the analog input SW. The correction factor is given in % by VSCALE2 [▶ 70]. e.g. VSCALE2=20

- SW2= +10V GEAROeff = GEARO*1.2
- SW2= -10V GEAROeff = GEARO*0.8
- SW2= 0V GEAROeff = GEARO

4.11.5 ROFFS2

ASCII - Command	**ROFFS2**

Syntax Transmit	ROFFS2 [Data]			
Syntax Receive	ROFFS2 <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer32		**CANBus Object Number**	3656 (hex)
DIM	PUNIT		**PROFIBUS PNU**	1942 (dec) IND = 0000xxxx (bin)
Range	long int		**DPR**	342 (dec)
Default	0			
Opmode	8		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	3.20			
Configuration	No		**Revision**	1.3
Function Group	Gearing		**EEPROM**	Yes
Short Description	Position offset for "absolute Gearing"			

Description

This parameter gives an offset to the absolute SSI-position of the master, read by the slave. This parameter is only used while starting the reference move NREF [▶ 267]=8 (move to absolute position). Starting the reference move, the absolute SSI position (GEARMODE [▶ 213]=7) is read by the slave using the resolution PGEARI [▶ 283] / PGEARO [▶ 284]. The ROFFS2 offset is added then and move to this target position is started.

4.12 Modbus

4.12.1 GDTX

ASCII - Command	**GDTX**		**Not supported with Standard Drive**	
Syntax Transmit	GDTX [Data]			
Syntax Receive	GDTX <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer8		**CANBus Object Number**	368A (hex)
DIM	-		**PROFIBUS PNU**	1994 (dec) IND = 0000xxxx (bin)
Range	0 .. 18		**DPR**	394 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	-		**Weightning**	
Start Firmware	4.04			
Configuration	No		**Revision**	1.5
Function Group	Modbus		**EEPROM**	Yes
Short Description	Number of Actual Value Data Words via Modbus			

Description

This parameter defines the number of cyclic updated actual values in 16 bit data words, which are updated every cycle between the drive and the Modbus board.

4.12.2 MBPDRVSTAT

ASCII - Command	**MBPDRVSTAT**		**Not supported with Standard Drive**	
Syntax Transmit	MBPDRVSTAT [Data]			
Syntax Receive	MBPDRVSTAT <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer16		**CANBus Object Number**	368D (hex)
DIM	-		**PROFIBUS PNU**	1997 (dec) IND = 0000xxxx (bin)
Range	0 .. 15		**DPR**	397 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	4.04			
Configuration	No		**Revision**	1.5
Function Group	Modbus		**EEPROM**	Yes
Short Description	State of the Modbus+ Network			

Description

MBPDRVSTAT gives the state of the Modbus+ network of the drive. The bit 3 can be written by the drive and can be saved. If the bit is set, Modbus+ network errors are indicated at the drive.

4.12.3 MBPSET

ASCII - Command	**MBPSET**		**Not supported with Standard Drive**	
Syntax Transmit	MBPSET [Data]			
Syntax Receive	MBPSET <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer8		**CANBus Object Number**	368E (hex)
DIM	-		**PROFIBUS PNU**	1998 (dec) IND = 0000xxxx (bin)
Range	0,1		**DPR**	398 (dec)
Default	1			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	-		**Weightning**	
Start Firmware	4.04			
Configuration	No		**Revision**	1.5
Function Group	Modbus		**EEPROM**	Yes
Short Description	Address selection of Modbus+			

Description

This parameter defines the direction of the address selection of the Modbus+ board in the initialization phase.

- MBPSET=0 The address is given by the Modbus board.
- MBPSET=1 The address is given by the drive and it's address in ADDR [▶ 70].

4.12.4 PEERCOP

ASCII - Command	**PEERCOP**		**Not supported with Standard Drive**	
Syntax Transmit	PEERCOP [Data]			
Syntax Receive	PEERCOP <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer8		**CANBus Object Number**	368F (hex)
DIM	-		**PROFIBUS PNU**	1999 (dec) IND = 0000xxxx (bin)
Range	0 .. 9		**DPR**	399 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	-		**Weightning**	
Start Firmware	4.04			
Configuration	No		**Revision**	1.5
Function Group	Modbus		**EEPROM**	Yes
Short Description	Number of Data Words (Command) at Modbus+			

Description

This parameter defines the number of data words (command), which are updated cyclic. Data, which is enabled as process data, cannot be written by the SDO channel (messaging).

4.12.5 PEERCOPS

ASCII - Command	**PEERCOPS**		**Not supported with Standard Drive**	
Syntax Transmit	PEERCOPS [Data]			
Syntax Receive	PEERCOPS <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer8		**CANBus Object Number**	3690 (hex)
DIM	-		**PROFIBUS PNU**	2000 (dec) IND = 0000xxxx (bin)
Range	1 .. 64		**DPR**	400 (dec)
Default	1			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	-		**Weightning**	
Start Firmware	4.04			
Configuration	No		**Revision**	1.6
Function Group	Modbus		**EEPROM**	Yes
Short Description	Number of Data Words (Command) at Modbus+			

Description

This parameter gives the address of the Modbus-Master for this drive. While initializing, this parameter is sent from the drive to the Modbus board. The board then transmits only data to the drive, that was send from this master.

4.12.6 TIMEMBP

ASCII - Command	**TIMEMBP**	**Not supported with Standard Drive**	
Syntax Transmit	TIMEMBP [Data]		
Syntax Receive	TIMEMBP <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer16	**CANBus Object Number**	3692 (hex)
DIM	10 ms	**PROFIBUS PNU**	2002 (dec) IND = 0000xxxx (bin)
Range	1 .. 6000	**DPR**	402 (dec)
Default	100		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	4.04		
Configuration	No	**Revision**	1.5
Function Group	Modbus	**EEPROM**	Yes
Short Description	Number of Data Words (Command) at Modbus+		

Description

This parameter defines the time-out of the Modbus communication in 10ms steps. If the drive gets no interrupt from the board in that time, the drive is disabled and the communication in MBPDRVSTAT [▶ 220] is displayed as faulty.

4.13 Motor

4.13.1 CPHASE

ASCII - Command	**CPHASE**		
Syntax Transmit	CPHASE [Data]		
Syntax Receive	CPHASE <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	No
DIM	-	**PROFIBUS PNU**	No
Range	0, 1	**DPR**	No
Default	1		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	Disable + Reset (Coldstart)	**Weightning**	
Start Firmware	5.41		
Configuration	Yes	**Revision**	1.9
Function Group	Motor	**EEPROM**	Yes
Short Description	Deactivate Motor Connection Detection		

Description

The motor connection detection trips if a cable is broken or not connected. CPHASE = 0 disables this function

4.13.2 ENCLINES

ASCII - Command	**ENCLINES**		
Syntax Transmit	ENCLINES [Data]		
Syntax Receive	ENCLINES <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	3533 (hex)
DIM	-	**PROFIBUS PNU**	1651 (dec) IND = 0000xxxx (bin)
Range	0 .. 32767, 32768 (5.41), 65535 (6.00)	**DPR**	51 (dec)
Default	1000		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	Disabled + Reset (Coldstart)	**Weightning**	
Start Firmware	1.71		
Configuration	Yes	**Revision**	1.9
Function Group	Motor	**EEPROM**	Yes
Short Description	SinCos Encoder Resolution		

Description

ENCLINES sets the resolution (number of lines) of the encoder input channel using an ENCODER as feedback unit. In case of Rotary Motors, it is the number of lines per revolution, in case of linear Motors it is the number of lines per pole pitch. With an ENDAT or Hiperface Encoder ENCLINES is read automatically during the initialization process.

- Starting with firmware 5.41, the range of ENCLINES is extended to 32767.
- Starting with firmware 6.00, the range of ENCLINES is extended to 65535.

4.13.3 FLUXM

ASCII - Command	**FLUXM**		
Syntax Transmit	FLUXM [Data]		
Syntax Receive	FLUXM <Data>	**Available in**	
Type	Variable r	**MMI**	No
ASCII Format	Float	**CANBus Object Number**	3689 (hex)
DIM	-	**PROFIBUS PNU**	1993 (dec) IND = 0000xxxx (bin)
Range	0 .. 10000	**DPR**	393 (dec)
Default	4500		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	4.40		
Configuration	No	**Revision**	1.6
Function Group	Motor	**EEPROM**	-
Short Description	Rated Flux Level of Permanent Magnet Motor		

Description

FLUXM is read-only and corresponds to the rated flux level of permanent magnet motor. This variable is determined by the motor torque constant, Kt. The unit is counts and the scaling factor is 22000. This means that Fluxm(counts)/22000 = VoltSec.

This variable is applicable only for the sensorless drive of permanent magnet motor.

4.13.4 GF

ASCII - Command	GF	For Manufacturer Use only	
Syntax Transmit	GF [Data]		
Syntax Receive	GF <Data>	Available in	
Type	Variable rw	MMI	No
ASCII Format	Float	CANBus Object Number	3667 (hex)
DIM	-	PROFIBUS PNU	1959 (dec) IND = 0000xxxx (bin)
Range	0 ... 2000	DPR	359 (dec)
Default	15		
Opmode	All	Data Type Bus/DPR	Integer32
Drive State	-	Weightning	1000
Start Firmware	3.40		
Configuration	No	Revision	1.4
Function Group	Motor	EEPROM	Yes
Short Description	Proportional Gain of the Flux Controller		

Description

This command is only for the induction motor mode (MTYPE [▸ 237] = 3).

This rotor flux controller utilizes a standard PI controller.

Rotor flux control loop: propotional gain. (3.40) (see also GFTN [▸ 224])

4.13.5 GFTN

ASCII - Command	GFTN	For Manufacturer Use only	
Syntax Transmit	GFTN [Data]		
Syntax Receive	GFTN <Data>	Available in	
Type	Variable rw	MMI	No
ASCII Format	Float	CANBus Object Number	3668 (hex)
DIM	ms	PROFIBUS PNU	1960 (dec) IND = 0000xxxx (bin)
Range	0 ... 1000	DPR	360 (dec)
Default	50		
Opmode	All	Data Type Bus/DPR	Integer32
Drive State	-	Weightning	1000
Start Firmware	3.40		
Configuration	No	Revision	1.4
Function Group	Motor	EEPROM	Yes
Short Description	Integral Action Time of the Flux Controller		

Description

This command is only for the induction motor mode (MTYPE [▸ 237] = 3).

This rotor flux controller utilizes a standard PI controller.

Rotor flux control loop: integral action time constant. (3.40) (see also GF [▸ 224])

4.13.6 GKC

ASCII - Command	**GKC**		**For Manufacturer Use only**	
Syntax Transmit	GKC [Data]			
Syntax Receive	GKC <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Float		**CANBus Object Number**	3687 (hex)
DIM	ms		**PROFIBUS PNU**	1991 (dec) IND = 0000xxxx (bin)
Range	0 .. 100		**DPR**	391 (dec)
Default	10			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	4.72			
Configuration	No		**Revision**	1.5
Function Group	Motor		**EEPROM**	Yes
Short Description	Compensation Gain of the Flux Controller			

Description

This parameter is only for the sensorless drive (FBTYPE [▸ 190]=10). It corresponds to the compensation gain of the rotor flux, and normally is set to 10 ms.

4.13.7 L

ASCII - Command	**L**			
Syntax Transmit	L [Data]			
Syntax Receive	L <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Float		**CANBus Object Number**	3577 (hex)
DIM	mH		**PROFIBUS PNU**	1719 (dec) IND = 0000xxxx (bin)
Range	0 .. 100		**DPR**	119 (dec)
Default	10			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	4.72			
Configuration	No		**Revision**	1.5
Function Group	Motor		**EEPROM**	Yes
Short Description	Stator Inductance of the Motor			

Description

The parameter describes the stator inductance between phase and phase in mH.

4.13.8 LDUMP

ASCII - Command	LDUMP		
Syntax Transmit	LDUMP [data]		
Syntax Receive	LDUMP <Data>	Available in	
Type	Multi-line Return Command	MMI	No
ASCII Format	String	CANBus Object Number	No
DIM	-	PROFIBUS PNU	No
Range	-	DPR	No
Default	-		
Opmode	All	Data Type Bus/DPR	-
Drive State	-	Weightning	
Start Firmware	1.30		
Configuration	No	Revision	1.3
Function Group	Motor	EEPROM	-
Short Description	Parameter Output of Motor Data		

Description

The command LDUMP <name> can be used to output the parameters for the motor data set <name> from the internal database. The <name> that is entered must be a valid motor designation from the motor database (see MDBLIST [▶ 229]). If the <name> parameter is not entered, the motor parameters that are loaded at present will be displayed.

4.13.9 MBRAKE

ASCII - Command	MBRAKE		
Syntax Transmit	MBRAKE [Data]		
Syntax Receive	MBRAKE <Data>	Available in	
Type	Variable rw	MMI	Yes
ASCII Format	Integer8	CANBus Object Number	3587 (hex)
DIM	-	PROFIBUS PNU	1735 (dec) IND = 0000xxxx (bin)
Range	0, 1	DPR	135 (dec)
Default	0		
Opmode	All	Data Type Bus/DPR	Integer8
Drive State	Disabled + Reset (Coldstart)	Weightning	
Start Firmware	1.20		
Configuration	Yes	Revision	2.0
Function Group	Motor	EEPROM	Yes
Short Description	Select Motor Holding Brake		

Description

MBRAKE enables the brake function for a 24V holding brake in the motor directly from the servo amplifier.

- MBRAKE = 0 Brake function is disabled
- MBRAKE = 1 Brake function is enabled. The output at the BRAKE terminal is 24V if the ENABLE signal is present (brake off) and 0V if the ENABLE signal is missing (brake activated).
- MBRAKE = 2 If the wake&shake mode is activated (FBTYPE [▶ 190] = 7 or 8) the holding brake is deactivated after the wake&shake mode (starting with firmware version 5.05).

4.13.10 MCFW

ASCII - Command	**MCFW**		
Syntax Transmit	MCFW [Data]		
Syntax Receive	MCFW <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Float	**CANBus Object Number**	3669 (hex)
DIM	-	**PROFIBUS PNU**	1961 (dec) IND = 0000xxxx (bin)
Range	1 ... 5	**DPR**	361 (dec)
Default	1.1		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	3.40		
Configuration	No	**Revision**	1.6
Function Group	Motor	**EEPROM**	Yes
Short Description	The Correction Factor of the Field Weakening		

Description

This command is only for the induction motor mode (MTYPE [▸ 237] = 3).

The correction factor of the field weakening.

This correction factor is introduced to compensate the nonlinearity of the magnetizing inductance since the magnetizing current is decreased according to the rotor mechanical velocity during the field weakening.

4.13.11 MCTR

ASCII - Command	**MCTR**		
Syntax Transmit	MCTR [Data]		
Syntax Receive	MCTR <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Float	**CANBus Object Number**	366A (hex)
DIM	-	**PROFIBUS PNU**	1962 (dec) IND = 0000xxxx (bin)
Range	1 ... 5	**DPR**	362 (dec)
Default	1.1		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	3.40		
Configuration	No	**Revision**	1.6
Function Group	Motor	**EEPROM**	Yes
Short Description	Correction Factor of the rotor time constant		

Description

This command is only for the induction motor mode (MTYPE [▸ 237] = 3).

The correction factor of the rotor time constant for the field weakening, which is introduced to improve the torque performance at the steady state in the field weakening.

4.13.12 MDBCNT

ASCII - Command	**MDBCNT**		
Syntax Transmit	MDBCNT		
Syntax Receive	MDBCNT <Data>	**Available in**	
Type	Variable ro	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	3588 (hex)
DIM	-	**PROFIBUS PNU**	1736 (dec) IND = 0000xxxx (bin)
Range	1 .. 127	**DPR**	136 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Motor	**EEPROM**	-
Short Description	Number of Motor Data Sets		

Description

MDBCNT returns the number of motor data sets that can be loaded for the present combination of output stage + feedback. A change of the feedback setting FBTYPE [▸ 190] is used, for instance, to ensure that only the data sets for resolver motors or EnDat motors are used.

4.13.13 MDBGET

ASCII - Command	**MDBGET**		
Syntax Transmit	MDBGET		
Syntax Receive	MDBGET <Data>	**Available in**	
Type	Command	**MMI**	Yes
ASCII Format	String	**CANBus Object Number**	3589 (hex)
DIM	-	**PROFIBUS PNU**	1737 (dec) IND = 0000xxxx (bin)
Range	1 .. MDBCNT	**DPR**	137 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Motor	**EEPROM**	-
Short Description	Get Actual Motor Data Set		

Description

The MDBGET command generates an information line (directory entry) for the latest motor data set that was selected with the MDBSET [▸ 230] command. The information line consists of the following elements: data set number, motor name, motor number, motor family

The individual elements are interpreted as follows:

- Data set number: the number for the data set within the motor database. This number can be used to address a motor data set with the MDBSET [▸ 230] command. This number is increased automatically at every MDBGET call.
- Motor name: a symbolic motor designation (max. length 12 characters).

- Motor number: a number that can be used to uniquely identify a particular motor. This number is used to load a data set from the motor database with the MNUMBER [▸ 233] command.
- Motor family: an additional designation (for internal use only).

The group of commands MDBCNT [▸ 228],MDBSET [▸ 230],MDBGET can be used by an external control system, to read out the contents of the motor database. The procedure is as follows:

1. Read out the number of available data sets, using the MDBCNT [▸ 228] command.
2. Set the data set pointer to the first data set, using the MDBSET [▸ 230] 1 command.
3. Read out the first directory entry, using the MDBGET command.
4. Repeat step 3 until the number of available data sets (MDBCNT [▸ 228]) has been read.

The MDBLIST [▸ 229] command offers an alternative. This command can be used to display the complete list.

4.13.14 MDBLIST

ASCII - Command	**MDBLIST**		
Syntax Transmit	MDBLIST [*]		
Syntax Receive	MDBLIST <Data>	**Available in**	
Type	Multi-line Return Command	**MMI**	No
ASCII Format	String	**CANBus Object Number**	No
DIM	-	**PROFIBUS PNU**	No
Range	-	**DPR**	No
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Motor	**EEPROM**	No
Short Description	List of Motor Data Sets		

Description

The MDBLIST command returns the list of contents for the motor database (for the present combination of output stage + feedback). One motor database entry is displayed per line on the screen, in the following format: motor name, motor number, motor family, amplifier designation.

The individual elements are interpreted as follows:

- Motor name: a symbolic motor designation (max. length 12 characters).
- Motor number: a number that can be used to uniquely identify a particular motor. This number is used to load a data set from the motor database with the MNUMBER [▸ 233] command.
- Motor family: an additional designation (for internal use only).

If PROMPT [▸ 102] 2 is set, a formatted output appears, which is especially suitable for terminal display.

The MDBLIST * command can be used to display the complete list of contents for the motor database. The difference to the output generated by MDBLIST is that the contents also include motor data sets that are not suitable for the present combination of output stage and feedback. These data sets will be displayed, but they cannot be loaded. Compared with the MDBLIST output, the MDBLIST * output has been enlarged by the columns "Amplifier designation" and "Feedback". These designations can be used to find out for which output stage or FBTYPE [▸ 190] setting this data set was created.

- Amplifier designation 6xx, where xx = current rating
- Feedback: 0=Resolver, 2=Hiperface, 4=Endat

4.13.15 MDBSET

ASCII - Command	**MDBSET**		
Syntax Transmit	MDBSET [Data]		
Syntax Receive	MDBSET <Data>	**Available in**	
Type	Command	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	358A (hex)
DIM	-	**PROFIBUS PNU**	1738 (dec) IND = 0000xxxx (bin)
Range	1 .. MDBCNT	**DPR**	138 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Motor	**EEPROM**	-
Short Description	Set Actual Motor Data Set		

Description

The MDBSET command can be used to address a specific data set from the motor database. The subsequent MDBGET [▶ 228] command provides the directory entry for the selected motor data set

4.13.16 MDUMP

ASCII - Command	**MDUMP**		
Syntax Transmit	MDUMP		
Syntax Receive	MDUMP <Data>	**Available in**	
Type	Multi-line Return Command	**MMI**	No
ASCII Format	String	**CANBus Object Number**	358B (hex)
DIM	-	**PROFIBUS PNU**	1739 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	139 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Motor	**EEPROM**	-
Short Description	Display Present Motor Parameters		

Description

Displays the currently valid motor parameters.

4.13.17 MICONT

ASCII - Command	**MICONT**			
Syntax Transmit	MICONT [Data]			
Syntax Receive	MICONT <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Float		**CANBus Object Number**	358E (hex)
DIM	Amperes		**PROFIBUS PNU**	1742 (dec) IND = 0000xxxx (bin)
Range	10% of DICONT, .. 2* DICONT		**DPR**	142 (dec)
Default	DICONT			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Motor		**EEPROM**	Yes
Short Description	Motor Continuous Current Rating			

Description

This parameter limits the ICONT [▶ 108] setting of the amplifier, depending on the maximum continuous current rating of the motor.

4.13.18 MIMR

ASCII - Command	**MIMR**			
Syntax Transmit	MIMR [Data]			
Syntax Receive	MIMR <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Float		**CANBus Object Number**	366B (hex)
DIM	A		**PROFIBUS PNU**	1963 (dec) IND = 0000xxxx (bin)
Range	(0.0 ... 0.8) * ICONT		**DPR**	363 (dec)
Default	0.0			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	3.40			
Configuration	No		**Revision**	1.9
Function Group	Motor		**EEPROM**	Yes
Short Description	Magnetizing Current (Induction Motor)			

Description

This command is only for the induction motor mode (MTYPE [▶ 237] = 3).

The MIMR defines the magnetizing current of induction motors, which is normally set in the range of 40%~50% of the rated current of the induction motor.

The magnetizing current maintains constant under the rated velocity. If the motor runs over the rated velocity, the magnetizing current will be decreased according to the rotor mechanical velocity.

This value will be also limited between 10% and 80% of the rated current.

For sensorless drive of PM motor (MTYPE [▶ 237]=2, FBTYPE [▶ 190]=10), this command determines the starting current. In the low velocity operation range, an injecting current control the PM motor starting and operation.

4.13.19 MIPEAK

ASCII - Command	MIPEAK		
Syntax Transmit	MIPEAK [Data]		
Syntax Receive	MIPEAK <Data>	Available in	
Type	Variable rw	MMI	Yes
ASCII Format	Float	CANBus Object Number	358F (hex)
DIM	Amperes	PROFIBUS PNU	1743 (dec) IND = 0000xxxx (bin)
Range	10% of DIPEAK, .. 2*DIPEAK	DPR	143 (dec)
Default	DIPEAK		
Opmode	All	Data Type Bus/DPR	Integer32
Drive State	-	Weightning	1000
Start Firmware	1.20		
Configuration	No	Revision	1.8
Function Group	Motor	EEPROM	Yes
Short Description	Motor Peak Current Rating		

Description

MIPEAK limits the IPEAK [▶ 110] setting of the amplifier, depending on the maximum peak current rating of the motor. The peak current should not exceed 4 times the rated current (MICONT [▶ 231]) of the motor. The actual value is also determined by the peak current (DIPEAK [▶ 38]) of the servo amplifier used (defines the maximum value for the entry of IPEAK [▶ 110] in the current controller).

4.13.20 MKT

ASCII - Command	MKT	For Manufacturer Use only	
Syntax Transmit	MKT [Data]		
Syntax Receive	MKT <Data>	Available in	
Type	Variable rw	MMI	Yes
ASCII Format	Float	CANBus Object Number	3593 (hex)
DIM	-	PROFIBUS PNU	1747 (dec) IND = 0000xxxx (bin)
Range	0.0 .. 10.0	DPR	147 (dec)
Default	1.0		
Opmode	All	Data Type Bus/DPR	Integer32
Drive State	-	Weightning	1000
Start Firmware	1.20		
Configuration	No	Revision	1.6
Function Group	Motor	EEPROM	Yes
Short Description	Motor KT		

Description

The torque constant of the motor in Nm/A.

This parameter is used for sensorless control. The value can be online checked according to the following equation:

Kt = 60 * SQRT (3) * Ui / (2 * PI * n)

- Ui induced voltage of the motor
- n actual rotor velocity

4.13.21 MNAME

ASCII - Command	MNAME			
Syntax Transmit	MNAME [Data]			
Syntax Receive	MNAME <Data>		Available in	
Type	Variable rw		MMI	Yes
ASCII Format	String		CANBus Object Number	No
DIM	-		PROFIBUS PNU	No
Range	max 12 ASCII Characters		DPR	No
Default	Blanks			
Opmode	All		Data Type Bus/DPR	-
Drive State	-		Weightning	
Start Firmware	1.20			
Configuration	No		Revision	1.3
Function Group	Motor		EEPROM	Yes
Short Description	Motor Name			

Description

The MNAME parameter is directly related to the motor number MNUMBER [▶ 233]. When a motor data set is loaded from the motor database (MNUMBER [▶ 233] command), the motor designation MNAME is also transferred. If a customer-specific motor designation is to be defined, then this can be done with the MNAME command.

When the motor name is altered, the motor number (MNUMBER [▶ 233]) is set to 0, to indicate a customer-specific motor data set.

4.13.22 MNUMBER

ASCII - Command	MNUMBER			
Syntax Transmit	MNUMBER [Data]			
Syntax Receive	MNUMBER <Data>		Available in	
Type	Variable rw		MMI	Yes
ASCII Format	Integer16		CANBus Object Number	3599 (hex)
DIM	-		PROFIBUS PNU	1753 (dec) IND = 0000xxxx (bin)
Range	int		DPR	153 (dec)
Default	0			
Opmode	All		Data Type Bus/DPR	Integer16
Drive State	Disabled		Weightning	
Start Firmware	1.20			
Configuration	No		Revision	1.3
Function Group	Motor		EEPROM	Yes
Short Description	Motor Number			

Description

The command □MNUMBER nr□ is used to load a motor data set with the number □nr□ from the motor database. If MNUMBER 0 is entered, then no data set will be loaded, but the variable MNUMBER will simply be set to 0. This setting indicates a customer-specific motor data set.

4.13.23 MPOLES

ASCII - Command	**MPOLES**		
Syntax Transmit	MPOLES [Data]		
Syntax Receive	MPOLES <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	359D (hex)
DIM	Poles	**PROFIBUS PNU**	1757 (dec) IND = 0000xxxx (bin)
Range	0, 2, 4, 6, .. , 256	**DPR**	157 (dec)
Default	6		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	Disabled	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Motor	**EEPROM**	Yes
Short Description	Number of Motor Poles		

Description

The number of motor poles per turn of the motor.
MPOLES = 0 is not saved into the drive if a Firmware > 5.07 is used. The setting MPOLES =0 is also not monitored in the MMI.

4.13.24 MRS

ASCII - Command	**MRS**		
Syntax Transmit	MRS [Data]		
Syntax Receive	MRS <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Float	**CANBus Object Number**	3686 (hex)
DIM	Ohm	**PROFIBUS PNU**	1990 (dec) IND = 0000xxxx (bin)
Range	0 .. 100	**DPR**	390 (dec)
Default	1		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	4.72		
Configuration	No	**Revision**	1.5
Function Group	Motor	**EEPROM**	Yes
Short Description	Winding Resistance of the Stator Phase-Phase		

Description

The parameter describes the stator winding resistance phase-phase in Ohm

4.13.25 MSERIALNO

ASCII - Command	**MSERIALNO**		
Syntax Transmit	MSERIALNO [Data]		
Syntax Receive	MSERIALNO <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer32	**CANBus Object Number**	36A3 (hex)
DIM	-	**PROFIBUS PNU**	2019 (dec) IND = 0000xxxx (bin)
Range	Long Int	**DPR**	419 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	4.93		
Configuration	No	**Revision**	1.6
Function Group	Motor	**EEPROM**	No
Short Description	Serial no of the motor for encoder feedback		

Description

MSERIALNO give the possibility to add a serial number of the motor. It is stored in the encoder with parameter channel (EnDAT or Hiperface) of the motor by typing in HSAVE [▶ 198]. MSERIALNO gives the serial number of the connected motor with encoder feedback.

This command can only be used, if a motor with encoder is connected.

4.13.26 MSPEED

ASCII - Command	**MSPEED**		
Syntax Transmit	MSPEED [Data]		
Syntax Receive	MSPEED <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Float	**CANBus Object Number**	35A3 (hex)
DIM	rpm	**PROFIBUS PNU**	1763 (dec) IND = 0000xxxx (bin)
Range	0.0 .. 12000.0	**DPR**	163 (dec)
Default	3000		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Motor	**EEPROM**	Yes
Short Description	Maximum Rated Motor Velocity		

Description

The MSPEED setting fixes the upper limit for the following amplifier parameters: VLIM [▶ 338], VLIMN [▶ 339], 5/6 * VOSPD [▶ 341].

4.13.27 MTANGLP

ASCII - Command	**MTANGLP**			
Syntax Transmit	MTANGLP [Data]			
Syntax Receive	MTANGLP <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer16		**CANBus Object Number**	35A5 (hex)
DIM	Electrical Degrees		**PROFIBUS PNU**	1765 (dec) IND = 0000xxxx (bin)
Range	0 .. 45		**DPR**	165 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.8
Function Group	Motor		**EEPROM**	Yes
Short Description	Current Lead			

Description

The current-dependent phase lead that is applied to make use of the reluctance torque at motor peak current (MIPEAK [▸ 232])

4.13.28 MTR

ASCII - Command	**MTR**			
Syntax Transmit	MTR [Data]			
Syntax Receive	MTR <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Float		**CANBus Object Number**	366C (hex)
DIM	ms		**PROFIBUS PNU**	1964 (dec) IND = 0000xxxx (bin)
Range	30 .. 1000		**DPR**	364 (dec)
Default	200			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	3.40			
Configuration	No		**Revision**	1.4
Function Group	Motor		**EEPROM**	Yes
Short Description	Rotor Time Constant			

Description

This command is only for the induction motor mode (MTYPE [▸ 237] = 3).

The MTR defines the rotor time constant at the rated operating point (Tr = Lh/Rr), where Lh and Rr are the magnetizing inductance and rotor resistance, respectively.

4.13.29 MTYPE

ASCII - Command	**MTYPE**		
Syntax Transmit	MTYPE [Data]		
Syntax Receive	MTYPE <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	35A6 (hex)
DIM	-	**PROFIBUS PNU**	1766 (dec) IND = 0000xxxx (bin)
Range	1, 2, 3	**DPR**	166 (dec)
Default	1		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	4.00		
Configuration	No	**Revision**	1.5
Function Group	Motor	**EEPROM**	Yes
Short Description	Motor Type		

Description

MTYPE sets the drive control algorithms to different motor types as follows:

- MTYPE = 1: permanent magnet motor
- MTYPE = 2: permanent magnet motor with Id current control. The one case is for the linear permanent magnet motor, the other case is for the sensor less drive of permanent magnet motor.
- MTYPE = 3: asynchronous motor (Induction motor)

4.13.30 MVANGLB

ASCII - Command	**MVANGLB**		
Syntax Transmit	MVANGLB [Data]		
Syntax Receive	MVANGLB <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	35A7 (hex)
DIM	rpm	**PROFIBUS PNU**	1767 (dec) IND = 0000xxxx (bin)
Range	0 .. 15000	**DPR**	167 (dec)
Default	3000		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.6
Function Group	Motor	**EEPROM**	Yes
Short Description	Velocity-dependent Lead (Start Phi)		

Description

This is a compensation for the inductive phase shift between the motor voltage and the motor current at high velocitys. With defined voltage relationships, it permits a higher torque at the final limit velocity. Alternatively, the achievable final limit velocity can be increased by up to 30%. Depending on the motor velocity, the phase shift (commutation angle) is increased linearly from the Start Phi point up to the Limit Phi value (MVANGLF [▸ 238]) at the final limit velocity. The most favorable setting depends on the type of motor and the final limit velocity.

4.13.31 MVANGLF

ASCII - Command	**MVANGLF**			
Syntax Transmit	MVANGLF [Data]			
Syntax Receive	MVANGLF <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer16		**CANBus Object Number**	35A8 (hex)
DIM	Electrical Degrees		**PROFIBUS PNU**	1768 (dec) IND = 0000xxxx (bin)
Range	0 .. 45		**DPR**	168 (dec)
Default	20			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Motor		**EEPROM**	Yes
Short Description	Velocity-dependent Lead (Limit Phi)			

Description

This is a compensation for the inductive phase shift between the motor voltage and the motor current at high velocitys. With defined voltage relationships, this permits a higher torque at the final limit velocity. Alternatively, the achievable final limit velocity can be increased by up to 30%. Depending on the motor velocity, the phase shift is increased linearly from the Start Phi point (MVANGLB [▸ 237]) up to the End Phi value at the final limit velocity. The most favorable setting depends on the type of motor and the final limit velocity.

4.13.32 MVANGLP

ASCII - Command	**MVANGLP**			
Syntax Transmit	MVANGLP [Data]			
Syntax Receive	MVANGLP <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer16		**CANBus Object Number**	3592 (hex)
DIM	Electrical Degrees		**PROFIBUS PNU**	1746 (dec) IND = 0000xxxx (bin)
Range	0 .. 60		**DPR**	146 (dec)
Default	20			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	2.42			
Configuration	No		**Revision**	1.8
Function Group	Motor		**EEPROM**	Yes
Short Description	Velocity-dependent Lead (Commutation Angle)			

Description

The inductive phase shift between the motor current and the motor voltage is compensated at high velocities. With the given voltage conditions, a higher torque is achieved at the velocity limit.

Alternatively, the achievable velocity limit is increased by 30%. The phase shift is increased linearly from a value of 0 degrees at MVANGLB [▸ 237] up to a final value of MVANGLF [▸ 238] degrees at VLIM [▸ 338]. The optimum setting depends on the type of motor and velocity limit.

4.13.33 MVR

ASCII - Command	**MVR**			
Syntax Transmit	MVR [Data]			
Syntax Receive	MVR <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Float		**CANBus Object Number**	366D (hex)
DIM	rpm		**PROFIBUS PNU**	1965 (dec) IND = 0000xxxx (bin)
Range	0 .. 10000		**DPR**	365 (dec)
Default	6000			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	4.72			
Configuration	No		**Revision**	1.9
Function Group	Motor		**EEPROM**	Yes
Short Description	Beginning Velocity of the field weakening			

Description

For the induction motor mode (MTYPE [▸ 237] = 3), the parameter MVR is set to the rated rotor mechanical velocity. This vale determines the beginning of the field weakening.

The value for 50 Hz induction motors with two poles is 3000 rpm. For the 50 Hz induction motor with four poles, MVR is 1500 rpm.

If the induction motor is for 60 Hz power line, the correspondent rated velocity should be given.

In the case of sensorless drives of PM motor (MTYPE [▸ 237]=2, FBTYPE [▸ 190]=10), the parameter MVR determines the switching value from scalar control to vector control. It is normally set to 10% ~ 20% of the rated velocity of the motor.

When using Hall' only, the parameter MVR determines the threshold speed where the actual speed is switched between speed estimation by using sensorless control method and speed calculation by the Hall's. So parameters of the PM motor MKT [▸ 232], MKS and L [▸ 225] must be set correctly. MTYPE [▸ 237] must be set to 2.

If MVR = 0, the compensation for the low speed will be switched off. An optimal setting of this parameter depends on the relation of the motor poles and the rated speed. Normally it is set to 20 - 30% of the rated speed VLIM [▸ 338].

4.14 Oscilloscope

4.14.1 GET

ASCII - Command	**GET**		
Syntax Transmit	GET		
Syntax Receive	GET <Data>	**Available in**	
Type	Command	**MMI**	Yes
ASCII Format	String	**CANBus Object Number**	3541 (hex)
DIM	-	**PROFIBUS PNU**	1665 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	65 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Oscilloscope	**EEPROM**	-
Short Description	Scope: output data		

Description

The GET command returns a list with all the most recently recorded SCOPE data. The list consists of n+3 lines (n = no. items of data recorded)

Line 1: commentary e.g. □Drive Recording□

Line 2: n, timebase in msec e.g. 10, 0.25 (10 data lines, timebase 250 microseconds)

Line 3: var1, var2, var3 names of the recorded variables, e.g. VCMD [▸ 32], V [▸ 31], ICMD [▸ 107]

Line 4: data1, data2, data3 recorded data, e.g. 0, 20.3, -0.5

Line 5: data1, dat2, data3

..

Line 1: data1, data2, data3

See also RECORD [▸ 243], RECTRIG [▸ 244]

4.14.2 J

ASCII - Command	**J**		
Syntax Transmit	J [Data]		
Syntax Receive	J <Data>	**Available in**	
Type	Command	**MMI**	Yes
ASCII Format	Float	**CANBus Object Number**	No
DIM	rpm (velocity) / Milliseconds (Time)	**PROFIBUS PNU**	No
Range	-15000.0 .. 15000.0 (=velocity),long int (Time)	**DPR**	No
Default	-		
Opmode	0	**Data Type Bus/DPR**	-
Drive State	Enabled	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Oscilloscope	**EEPROM**	-
Short Description	Service Function: Constant Velocity		

Description

The command □J <n> <t>□ can be used to define a constant velocity <n> (in rpm) for a defined time <t> (in msec). If the <t> entry is missing, the drive runs continuously.

4.14.3 RECDONE

ASCII - Command	**RECDONE**		
Syntax Transmit	RECDONE		
Syntax Receive	RECDONE <Data>	**Available in**	
Type	Variable ro	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	35DE (hex)
DIM	-	**PROFIBUS PNU**	1822 (dec) IND = 0000xxxx (bin)
Range	0, 1	**DPR**	222 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Oscilloscope	**EEPROM**	No
Short Description	Scope: Recording Done		

Description

The RECDONE command can be used to request the status of the SCOPE recording. The command returns a 1 if the recording is finished and the data can now be requested with the GET [▶ 240] command.

4.14.4 RECING

ASCII - Command	**RECING**		
Syntax Transmit	RECING		
Syntax Receive	RECING <Data>	**Available in**	
Type	Variable ro	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	35DF (hex)
DIM	-	**PROFIBUS PNU**	1823 (dec) IND = 0000xxxx (bin)
Range	0, 1	**DPR**	223 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Oscilloscope	**EEPROM**	No
Short Description	Scope: Recording in Progress		

Description

Returns a 1 if the recording is active. At the end of a recording, or if the recording has not started, a 0 is returned

4.14.5 RECOFF

ASCII - Command	**RECOFF**		
Syntax Transmit	RECOFF		
Syntax Receive	RECOFF	**Available in**	
Type	Command	**MMI**	Yes
ASCII Format	-	**CANBus Object Number**	35E0 (hex)
DIM	-	**PROFIBUS PNU**	1824 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	224 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Oscilloscope	**EEPROM**	-
Short Description	Scope: Cancel Scope Recording		

Description

RECOFF stops the SCOPE recording (if started). State after RECOFF: RECRDY [▸ 244]=1, RECING [▸ 242]=0, RECDONE [▸ 241]=0.

4.14.6 RECORD

ASCII - Command	**RECORD**			
Syntax Transmit	RECORD [Data]			
Syntax Receive	RECORD <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	String		**CANBus Object Number**	No
DIM	-		**PROFIBUS PNU**	No
Range	1 .. 10000 (=Time);1 .. 1024(=Points); ASCII String (=Var)		**DPR**	No
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Oscilloscope		**EEPROM**	No
Short Description	Scope: Capture Data for Recording			

Description

The RECORD command can be used to define the data for the next SCOPE recording. The command is used in the following form.

RECORD time number var1 [var2] [var3]

- time: the sampling interval in 250 microsecond steps
- number: the number of sample points to be recorded.

The maximum possible number depends on the number and size of the variables to be recorded.

If the number entered is too large, it will automatically be limited (when recording Long/Float variables, a maximum of 512 sample points can be recorded). var1,var2,var3 - names of the variables to be recorded.

Apart from the names for macro variables, the following names can be used.

I [▸ 21] - actual value of current

- ICMDVAL
  - setpoint for current
  - PE [▸ 26] - following error
  - V [▸ 31] - actual value of velocity
  - VCMD [▸ 32] - setpoint for velocity
  - VBUS [▸ 32] - DC-bus (DC-link) voltage
  - PFB [▸ 27] - actual position

4.14.7 RECRDY

ASCII - Command	**RECRDY**		
Syntax Transmit	RECRDY		
Syntax Receive	RECRDY <Data>	**Available in**	
Type	Variable ro	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	35E1 (hex)
DIM	-	**PROFIBUS PNU**	1825 (dec) IND = 0000xxxx (bin)
Range	0, 1	**DPR**	225 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Oszilloscope	**EEPROM**	No
Short Description	Scope: Status of RECORD Function		

Description

After the recording has been made trigger-ready by RECORD [▶ 243] / RECTRIG [▶ 244], the RECRDY command generates a 0. As soon as the trigger condition defined by RECRDY is fulfilled, and the recording starts, RECRDY generates a 1.

(RECRDY=0 means □waiting for trigger event□)

4.14.8 RECTRIG

ASCII - Command	**RECTRIG**		
Syntax Transmit	RECTRIG [Data]		
Syntax Receive	RECTRIG <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	String	**CANBus Object Number**	No
DIM	-	**PROFIBUS PNU**	No
Range	ASCII String (=Mode);Depends upon Mode (=Level);0 .. 1023 (=Loc.); 0, 1 (=Dir.)	**DPR**	No
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Oscilloscope	**EEPROM**	No
Short Description	Scope: Activate Recording Function		

Description

The RECTRIG command prepares the SCOPE function for a data recording. The command is used in the following form.

RECTRIG mode level location direction

- mode: designates the name of a variable that is to be used to trigger the recording. If the designation IMM is used, the recording starts immediately. In this case, the parameters □level□, □location□ and □direction□ do not have to be specified.
- level: specifies the value of the variable that must be reached to trigger the recording.
- location: give the number of points that are to be recorded before the moment of the trigger event.
- direction: specifies in which direction the value must pass the threshold □level□ of the □mode□ variable to trigger the recording.
    - direction=0 falling (variable value falls below threshold level)
    - direction=1 rising (variable value goes above threshold level)

4.14.9 S

ASCII - Command	**S**		
Syntax Transmit	S		
Syntax Receive	S	**Available in**	
Type	Command	**MMI**	Yes
ASCII Format	-	**CANBus Object Number**	35EA (hex)
DIM	-	**PROFIBUS PNU**	1834 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	234 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Oscilloscope	**EEPROM**	-
Short Description	Stop Motor and Disable Drive		

Description

The S command stops the drive (using the braking ramp DECSTOP [▸ 330]). As soon as the velocity/velocity falls below the standstill threshold (VEL0 [▸ 338]) the output stage is disabled.
The S command corresponds to the command K [▸ 40] (or DIS [▸ 38]) if the STOPMODE [▸ 85] option is set to 1.

4.14.10 STEP

ASCII - Command	**STEP**		
Syntax Transmit	STEP [Data]		
Syntax Receive	STEP <Data>	**Available in**	
Type	Command	**MMI**	Yes
ASCII Format	Integer16 Float Integer16 Float	**CANBus Object Number**	No
DIM	Milliseconds (DurationN) / rpm (velocityN)	**PROFIBUS PNU**	No
Range	Duration:0 to 32767; velocity:-VLIM to +VLIM	**DPR**	No
Default	Duration:1000; velocity1/2: 100/-100		
Opmode	0	**Data Type Bus/DPR**	-
Drive State	Enabled	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Oscilloscope	**EEPROM**	-
Short Description	Service Operation (STEP Command)		

Description

The STEP command is used to implement a service function through the operating mode □digital velocity control□ (OPMODE [▸ 50]=0). The command can be used in the following forms.

1. STEP The command provides the present settings for the service function.
2. STEP T1 V1 A digital setpoint V1 (RPM) is provided for time T1 (in msec). After T1 has elapsed, the digital setpoint is set to 0.
3. STEP T1 V1 T2 V2 A digital setpoint V1 (RPM) is provided for time T1 (in msec). After T1 has elapsed, a digital setpoint V2 (RPM) is provided for time T2 (in msec). After T2 has elapsed, the T1/V1 cycle starts again.

This command can be used to create an endless reversing operation.
e.g. STEP 1000 500 1000 -500

The service operation can always be cancelled by using the STOP [▸ 297] command. The □digital velocity control□ operating mode is a precondition for implementing the STEP command.

4.14.11 T

ASCII - Command	**T**		
Syntax Transmit	T [Data]		
Syntax Receive	T <Data>	**Available in**	
Type	Command	**MMI**	Yes
ASCII Format	Float	**CANBus Object Number**	360E (hex)
DIM	Amperes	**PROFIBUS PNU**	1870 (dec) IND = 0000xxxx (bin)
Range	-DIPEAK .. DIPEAK	**DPR**	270 (dec)
Default	-		
Opmode	2	**Data Type Bus/DPR**	Integer32
Drive State	Enabled	**Weightning**	1000
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Oscilloscope	**EEPROM**	-
Short Description	Digital Current Setpoint		

Description

The □T□ command can be used to define a constant current setpoint <i> (in A).
This current setpoint remains effective until a new T / STOP [▶ 297] / OPMODE [▶ 50] command is executed.

4.15 Position Controler

4.15.1 ACCR

ASCII - Command	**ACCR**		
Syntax Transmit	ACCR [Data]		
Syntax Receive	ACCR <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	3502 (hex)
DIM	Milliseconds	**PROFIBUS PNU**	1602 (dec) IND = 0000xxxx (bin)
Range	1 .. 32767	**DPR**	2 (dec)
Default	10		
Opmode	8	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Acceleration Ramp for homing/jog modes		

Description

This variable defines the acceleration ramp used for jogging and homing with the internal position control loop. The entry is made in msec and is in reference to the final limit velocity for the selected mode (VJOG [▶ 309] for jogging or VREF [▶ 310] for homing). When starting the homing or jog mode, the ACCR acceleration ramp can (in some circumstances) be limited by the minimum acceleration time PTMIN [▶ 289].

4.15.2 AUTOHOME

ASCII - Command	AUTOHOME		
Syntax Transmit	AUTOHOME [Data]		
Syntax Receive	AUTOHOME <Data>	Available in	
Type	rw	MMI	No
ASCII Format	Integer8	CANBus Object Number	36D7 (hex)
DIM		PROFIBUS PNU	1671 (dec) IND = 0001xxxx (bin)
Range	0,1	DPR	471 (dec)
Default	0		
Opmode	8	Data Type Bus/DPR	Integer8
Drive State	Disable	Weightning	
Start Firmware	5.53		
Configuration	No	Revision	2.0
Function Group	Position Controller	EEPROM	Yes
Short Description			

Description

The Command AUTOHOME selects, if the homing procedure will be started automatically.

- AUTOHOME=1 After the drive is enabled, the homing procedure MH [▶ 264] gets started automatically
- AUTOHOME=0 no automatic homing procedure started

4.15.3 CLRORDER

ASCII - Command	CLRORDER		
Syntax Transmit	CLRORDER [Data]		
Syntax Receive	-	Available in	
Type	Command	MMI	Yes
ASCII Format	Integer16	CANBus Object Number	351A (hex)
DIM	-	PROFIBUS PNU	1626 (dec) IND = 0000xxxx (bin)
Range	0;1 ..180; 192 .. 255	DPR	26 (dec)
Default	-		
Opmode	All	Data Type Bus/DPR	Integer16
Drive State	Enabled (only RAM) / Disabled	Weightning	
Start Firmware	2.00		
Configuration	No	Revision	1.3
Function Group	Position Controller	EEPROM	-
Short Description	Deleting a Motion Task		

Description

The command CLRORDER is used to delete a motion task given by the variable (e.g. CLRORDER 10, means: motion task 10 is deleted).

4.15.4 CONTINUE

ASCII - Command	**CONTINUE**			
Syntax Transmit	CONTINUE			
Syntax Receive	CONTINUE		**Available in**	
Type	Command		**MMI**	No
ASCII Format	-		**CANBus Object Number**	351D (hex)
DIM	-		**PROFIBUS PNU**	1629 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	29 (dec)
Default	-			
Opmode	8		**Data Type Bus/DPR**	-
Drive State	Enabled		**Weightning**	
Start Firmware	1.30			
Configuration	No		**Revision**	1.3
Function Group	Position Controller		**EEPROM**	-
Short Description	Continue last position order			

Description

The CONTINUE command can be used to continue (and complete) a motion block that was previously interrupted by the STOP [▶ 297] command. This is especially important for a motion block with relative paths.

4.15.5 DECR

ASCII - Command	**DECR**			
Syntax Transmit	DECR [Data]			
Syntax Receive	DECR <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer16		**CANBus Object Number**	3524 (hex)
DIM	Milliseconds		**PROFIBUS PNU**	1636 (dec) IND = 0000xxxx (bin)
Range	1 .. 32767		**DPR**	36 (dec)
Default	10			
Opmode	8		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Position Controller		**EEPROM**	Yes
Short Description	Deceleration Ramp for homing/jog modes			

Description

The DECR command defines the braking ramp for jog mode or homing with the internal position control loop. The entry is made in msec and is referred to the final limit velocity for the corresponding operating mode: VJOG [▶ 309] for jog operation, or VREF [▶ 310] for homing.

When starting the homing/jog mode, the DECR deceleration ramp can, in some circumstances, be limited by the minimum acceleration time PTMIN [▶ 289] (see description of the PTMIN [▶ 289] parameter).

4.15.6 DOVRIDE

ASCII - Command	**DOVRIDE**			
Syntax Transmit	DOVRIDE [Data]			
Syntax Receive	DOVRIDE <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Int16		**CANBus Object Number**	36B6 (hex)
DIM	-		**PROFIBUS PNU**	2038 (dec) IND = 0000xxxx (bin)
Range	0 .. 8192		**DPR**	438 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Int16
Drive State	-		**Weightning**	
Start Firmware	5.00			
Configuration	No		**Revision**	1.7
Function Group	Position Controller		**EEPROM**	Yes
Short Description	Digital Override Factor			

Description

If the digital Override function is selected (see OVRIDE [▶ 280]=3), DOVRIDE gives the possibility to change the digital scaling.

The scaling is:

- DOVRIDE=0 Motion task speed is 0 %
- DOVRIDE=8192 Motion task speed is 100 %

4.15.7 DREF

ASCII - Command	**DREF**			
Syntax Transmit	DREF [Data]			
Syntax Receive	DREF <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer8		**CANBus Object Number**	352C (hex)
DIM	-		**PROFIBUS PNU**	1644 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2		**DPR**	44 (dec)
Default	0			
Opmode	8		**Data Type Bus/DPR**	Integer8
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Position Controller		**EEPROM**	Yes
Short Description	Direction for Homing			

Description

The DREF parameter can be used to define the preferred direction of motion for a homing operation and for positioning with a modulo-axes (POSCNFG [▶ 285]=2).

Status	Short description	Description
DREF = 0	Negative movement	A homing move is always started in the negative direction (-VREF [▸ 310]). Using a modulo axes type a position is always searched in negative direction (is DIR [▸ 331]=1, the target position is searched in the positive direction).
DREF = 1	Positive movement	A homing move is always started in the positive direction (+VREF [▸ 310]). Using a modulo axes type a position is always searched in positive direction (is DIR [▸ 331]=1, the target position is searched in the negative direction).
DREF = 2	Optimized movement	The shortest distance between the starting position and the zero pulse of the resolver is executed in Homing Mode No. 5 (NREF [▸ 267]=5). When a modulo axes type is used, the drive always searches for the shortest distance to reach the position.

4.15.8 ERND

ASCII - Command	**ERND**		
Syntax Transmit	ERND [Data]		
Syntax Receive	ERND <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer32	**CANBus Object Number**	3638 (hex)
DIM	-	**PROFIBUS PNU**	1912 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	312 (dec)
Default	2^31-1		
Opmode	-	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	2.45		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	Yes
Short Description	End position of modulo axes		

Description

The ERND parameter is used to define the end of the range of movement for a modulo axes (POSCNFG [▸ 285]=2). The start of the range can be set by the SRND [▸ 296] command. All positioning operations are made in the positioning range <SRND [▸ 296]...ERND-1>.

The entry for ERND is made in SI units (taking account of PGEARI [▸ 283], PGEARO [▸ 284]).

4.15.9 EXTLATCH

ASCII - Command	EXTLATCH			
Syntax Transmit	EXTLATCH [Data]			
Syntax Receive	EXTLATCH <Data>		Available in	
Type	Variable rw		MMI	No
ASCII Format	Integer8		CANBus Object Number	3681 (hex)
DIM	-		PROFIBUS PNU	1985 (dec) IND = 0000xxxx (bin)
Range	0 .. 2		DPR	385 (dec)
Default	0			
Opmode	All		Data Type Bus/DPR	Integer8
Drive State	Disabled + Reset (Coldstart)		Weightning	
Start Firmware	4.61			
Configuration	Yes		Revision	1.5
Function Group	Position Controller		EEPROM	Yes
Short Description	Selection of the Source of the Latch Inputs			

Description

The configuration variable EXTLATCH defines the source for the position information using the Latch functions of the digital inputs (**IN1MODE** =26 and/or **IN2MODE** =26). If more than one inputs (1 or 2) are configured as Latch input, EXTLATCH defines thedifferent sources. If only one input is configured as Latch input, both different sources are stored at the same time.

Status	Latch with Input 1	Latch with Input 2
EXTLATCH=0	Resolver/EnDAT/Hiperface depends on FBTYPE [▸ 190]	Resolver/EnDAT/Hiperface depends on FBTYPE [▸ 190]
EXTLATCH=1	external encoder	Resolver/EnDAT/Hiperface depends on FBTYPE [▸ 190]
EXTLATCH=2	external encoder	external encoder

Also see about this

- IN1MODE [▸ 116]
- IN2MODE [▸ 123]

4.15.10 EXTMUL

ASCII - Command	EXTMUL			
Syntax Transmit	EXTMUL [Data]			
Syntax Receive	EXTMUL <Data>		Available in	
Type	Variable rw		MMI	Yes
ASCII Format	Integer16		CANBus Object Number	3538 (hex)
DIM	-		PROFIBUS PNU	1656 (dec) IND = 0000xxxx (bin)
Range	-32768 .. 32767		DPR	56 (dec)
Default	256			
Opmode	All		Data Type Bus/DPR	Integer16
Drive State	-		Weightning	
Start Firmware	1.62			
Configuration	No		Revision	1.3
Function Group	Position Controller		EEPROM	Yes

Short Description	ext. Encoder multiplier

Description

The EXTMUL parameter can be used to adjust the resolution of the external encoder to match the resolution of the internal position control loop. EXTMUL can be calculated according to the following formula:

EXTMUL = 1048576 / (NN x 4) NN is the resolution of the external encoder, in pulses/turn

4.15.11 EXTPOS

ASCII - Command	**EXTPOS**			
Syntax Transmit	EXTPOS [Data]			
Syntax Receive	EXTPOS <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer8		**CANBus Object Number**	3539 (hex)
DIM	-		**PROFIBUS PNU**	1657 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2, 3, 4		**DPR**	57 (dec)
Default	4			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	Disabled + Reset (Coldstart)		**Weightning**	
Start Firmware	1.62			
Configuration	Yes		**Revision**	2.0
Function Group	Position Controller		**EEPROM**	Yes
Short Description	Position Feedback + Control Type			

Description

The EXTPOS command defines the feedback source for the internal position control loop. Furthermore, the type of position control (P/PI) can be defined.

Parameters for the PI position control

- The PI position control can be set up by using the following parameters:
  - GP [▸ 255]: proportional gain (position)
  - GPTN [▸ 257]: rest time (position), integral-action time
  - GPFFV [▸ 256]: feed-forward (position)
  - GPV [▸ 257]: proportional gain (speed)

Parameters for the P position / PI speed control

- The P position control can be set up by using the following parameters:
  - GP [▸ 255]: proportional gain (position)
  - GPFFV [▸ 256]: feed-forward (position)

The usual velocity-control loop parameters can be used for the following speed control loop.

Status	Position Control uses/ Type of Position Controller	Reading of an external Encoder (PFB0 [▸ 27])
EXTPOS=0	Feedback system select by FBTYPE [▸ 190], PI-type Position Controller, P-type Speed Controller	There is no possibility to read an external Encoder on X1 (Drive 400 X2) or X5 (Drive 400 X4).

Status	Position Control uses/ Type of Position Controller	Reading of an external Encoder (PFB0 [▸ 27])
EXTPOS=1	Feedback via external source selectable by GEARMODE [▸ 213], P-type Position Controller, PI-type Speed Controller	External Encoder selectable GEARMODE [▸ 213], Read by PFB0 [▸ 27]
EXTPOS=2	Feedback system select by FBTYPE [▸ 190], PI-type Position Controller, P-type Speed Controller	External Encoder selectable by GEARMODE [▸ 213], Read by PFB0 [▸ 27]
EXTPOS=3	Feedback system select by FBTYPE [▸ 190], P-type Position Controller, PI-type Speed Controller	External Encoder selectable by GEARMODE [▸ 213], Read by PFB0 [▸ 27]
EXTPOS=4	Feedback system select by FBTYPE [▸ 190], P-type Position Controller, PI-type Speed Controller	There is no possibility to read an external Encoder on X1 (Drive 400 X2) or X5 (Drive 400 X4).

4.15.12 FB2RES

ASCII - Command	**FB2RES**		
Syntax Transmit	FB2RES [Data]		
Syntax Receive	FB2RES <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer32	**CANBus Object Number**	3688 (hex)
DIM	-	**PROFIBUS PNU**	1992 (dec) IND = 0000xxxx (bin)
Range	Long Int	**DPR**	392 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	3.58		
Configuration	No	**Revision**	1.6
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Number of Counts of an ext. Encoder per Motorturn		

Description

This parameter defines the number of counts of an external encoder per motor turn. The drive calculates automatically EXTMUL [▸ 252] and other parameters if necessary.

The actual position of the external encoder is now scaled to the actual sercos scaling for positions (see also Sercos Manual IDN117).

4.15.13 GP

ASCII - Command	GP			
Syntax Transmit	GP [Data]			
Syntax Receive	GP <Data>		Available in	
Type	Variable rw		MMI	Yes
ASCII Format	Float		CANBus Object Number	3542 (hex)
DIM	-		PROFIBUS PNU	1666 (dec) IND = 0000xxxx (bin)
Range	0.001 .. 25.0		DPR	66 (dec)
Default	0.15			
Opmode	4, 5, 8		Data Type Bus/DPR	Integer32
Drive State	-		Weightning	1000
Start Firmware	1.20			
Configuration	No		Revision	1.8
Function Group	Position Controller		EEPROM	Yes
Short Description	Position Control Loop: Proportional Gain			

Description

This variable is used both in the P position control loop (EXTPOS=1, 3, 4), and in the PI position control loop (EXTPOS=0, 2). If GP is set too low, the lag or settling time is too long and the drive is too soft. If GP is set too high, the drive oscillates.

Also see about this

EXTPOS [253]

4.15.14 GPFBT

ASCII - Command	GPFBT			
Syntax Transmit	GPFBT [Data]			
Syntax Receive	GPFBT <Data>		Available in	
Type	Variable rw		MMI	Yes
ASCII Format	Float		CANBus Object Number	3543 (hex)
DIM	-		PROFIBUS PNU	1667 (dec) IND = 0000xxxx (bin)
Range	0.0 .. 2.0		DPR	67 (dec)
Default	1.0			
Opmode	4, 5, 8		Data Type Bus/DPR	Integer32
Drive State	-		Weightning	1000
Start Firmware	1.20			
Configuration	No		Revision	1.3
Function Group	Position Controller		EEPROM	Yes
Short Description	Position Control Loop: Feed Forward for Actual Current			

Description

Position control loop: feed forward for the actual value of current. This parameter is only used for the PI position control loop (EXTPOS [253]=0, 2).

4.15.15 GPFFT

ASCII - Command	**GPFFT**		
Syntax Transmit	GPFFT [Data]		
Syntax Receive	GPFFT <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Float	**CANBus Object Number**	3544 (hex)
DIM	-	**PROFIBUS PNU**	1668 (dec) IND = 0000xxxx (bin)
Range	0.0 .. 5.0	**DPR**	68 (dec)
Default	1		
Opmode	4, 5, 8	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Position Control Loop: Feed Forward for Current Setpoint		

Description

Position control loop: feed forward for the current setpoint. Has to be set, that the contouring error is minimized.

This parameter is only used for the PI position control loop (EXTPOS [▸ 253]=0, 2).

In addition to that, the parameter has effect with starting firmware 4.78 using table based motion task enabled with bit in O_C [▸ 272] and using SPSET [▸ 295] = 3.

If GV [▸ 332] is changed after optimizing GPFFT, GPFFT must be changed also inversely proportional.

4.15.16 GPFFV

ASCII - Command	**GPFFV**		
Syntax Transmit	GPFFV [Data]		
Syntax Receive	GPFFV <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Float	**CANBus Object Number**	3545 (hex)
DIM	-	**PROFIBUS PNU**	1669 (dec) IND = 0000xxxx (bin)
Range	0.0 .. 50.0	**DPR**	69 (dec)
Default	1.0		
Opmode	4, 5, 8	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Position Control Loop: Feed Forward for Velocity		

Description

This variable is used both in the P position control loop (EXTPOS=1, 3, 4), and in the PI position control loop (EXTPOS=0, 2). Feed forward is used to ease the position controller task. A better setting for GPFFV means better utilization of the dynamic range of the position controller. The most favorable setting (usually about 1.0) depends on factors external to the drive such as friction, dynamic resistance, and stiffness. If GPFFV is set too low, the drive lags. If GPFFV is set too high, the drive oversteers.

Also see about this

- EXTPOS [▸ 253]

4.15.17 GPTN

ASCII - Command	**GPTN**		
Syntax Transmit	GPTN [Data]		
Syntax Receive	GPTN <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Float	**CANBus Object Number**	3546 (hex)
DIM	Milliseconds	**PROFIBUS PNU**	1670 (dec) IND = 0000xxxx (bin)
Range	1.0 .. 200.0	**DPR**	70 (dec)
Default	50		
Opmode	4, 5, 8	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Position Control Loop: Integral-Action Time		

Description

This parameter is only used for the PI position control loop (EXTPOS [▸ 253] = 0,2). There is no possibility to switch off the integral part. If a P position control loop and P velocity control loop should be used, set EXTPOS [▸ 253] = 1,3,4 and set GVTN [▸ 336] = 0

4.15.18 GPV

ASCII - Command	**GPV**		
Syntax Transmit	GPV [Data]		
Syntax Receive	GPV <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Float	**CANBus Object Number**	3547 (hex)
DIM	-	**PROFIBUS PNU**	1671 (dec) IND = 0000xxxx (bin)
Range	0.1 .. 60.0	**DPR**	71 (dec)
Default	3		
Opmode	4, 5,8	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Proportional Gain of the Velocity Controller		

Description

This variable is used only for the PI position control loop (EXTPOS [▸ 253]=0, 2). Adjust GPV by increasing the value to the level where the motor starts to oscillate. Then, back it off until the oscillations have clearly stopped. Typical values are the same as for the GV [▸ 332] gain of the velocity controller. If the GPV value is too low, the drive is too soft and has poor damping. If the GPV value is too high, the drive whistles or runs roughly.

4.15.19 IN2PM

ASCII - Command	**IN2PM**		
Syntax Transmit	IN2PM [Data]		
Syntax Receive	IN2PM <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	362E (hex)
DIM	-	**PROFIBUS PNU**	1902 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2	**DPR**	302 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	2.44		
Configuration	No	**Revision**	1.8
Function Group	Position Controller	**EEPROM**	Yes
Short Description	In-Position 2 Mode		

Description

The IN2PM command is used to configure the function of the interim message during a motion task (motion block) sequence.

The function □Interim message during a motion task sequence□ (NextInPos) is available if an I/O expansion card is used (terminal X11B4) or a digital output of the drive is configured with the function OxMODE [▸ 150]=16. At the start of the first motion block (motion task), the □NextInPos□ output is always set to 0. The response of the output during the execution of the motion block sequence depends on the configuration variable IN2PM.

- IN2PM=0 □ the output is inverted at the start of the next block.
- IN2PM=2 □ the output is inverted at the end of a block.
- IN2PM=1 □ the output is set to 0 at the start of a motion block and set to HIGH at the end of a motion block.

With a sequence of motion blocks where the blocks are started immediately, only the IN2PM=0 or IN2PM=2 settings make sense. If the setting is IN2PM=1, the HIGH state is so short that it may not be registered at all by the external control system.

If a following task is started with the aid of an I/O (INxMODE [▸ 116]=15), then the IN2PM=2 or IN2PM=1 setting should be used. With this setting, the end of a motion block is signaled by the HIGH state (IN2PM=1) or the change of state (IN2PM=2) at the □NextInPos□ output. The external control system can then initialize the continuation of the motion task sequence via the □Start next task□ input.

4.15.20 INPOS

ASCII - Command	**INPOS**			
Syntax Transmit	INPOS			
Syntax Receive	INPOS <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Integer8		**CANBus Object Number**	356D (hex)
DIM	-		**PROFIBUS PNU**	1709 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	109 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Position Controller		**EEPROM**	-
Short Description	Status of In-Position Signal			

Description

The INPOS command returns the status of the IN-Position bit of the status register (DRVSTAT [▸ 171]).

As long as the difference between the last target position (motion task) and the actual position (PFB [▸ 27]) is within the width of the preset In-Position window (PEINPOS [▸ 282]), a 1 is signaled, otherwise a 0. see also INPT [▸ 259]

4.15.21 INPT

ASCII - Command	**INPT**			
Syntax Transmit	INPT [Data]			
Syntax Receive	INPT <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer16		**CANBus Object Number**	3630 (hex)
DIM	Milliseconds		**PROFIBUS PNU**	1904 (dec) IND = 0000xxxx (bin)
Range	1 .. 32000		**DPR**	304 (dec)
Default	10			
Opmode	8		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	2.08			
Configuration	No		**Revision**	1.3
Function Group	Position Controller		**EEPROM**	Yes
Short Description	In-Position Delay			

Description

The INPT command defines a delay time for the In-Position signal. At the start of a motion block, the In-Position signal is removed, and the monitoring of the In-Position window is only activated again after the end of this preset time. This function is especially important for positioning tasks within the In-Position window. In such a case, it ensures that the In-Position signal is always removed for a definite time.See also INPOS [▸ 259]

4.15.22 LATCH16

ASCII - Command	**LATCH16**		
Syntax Transmit	LATCH16		
Syntax Receive	LATCH16 <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	3578 (hex)
DIM	-	**PROFIBUS PNU**	1720 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	120 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	1.66		
Configuration	No	**Revision**	1.7
Function Group	Position Controller	**EEPROM**	-
Short Description	Latched 16-bit Position (positive edge)		

Description

The LATCH16 command returns the position where latching was performed by the last positive (rising) edge on digital input 2 (IN2MODE [▸ 123]=26). The position value is absolute within one turn, and is given out in the internal units (counts 0 ... 65535). In order to get the absolute 32-bit position in SI units (taking account of the position control loop resolution PGEARI [▸ 283]/PGEARO [▸ 284]), the command LATCH32 [▸ 261] should be used.

The commands LATCH16 and LATCH32 [▸ 261] have the effect of erasing the status bit 20 □positive latch made□ in the status register DRVSTAT [▸ 171].

4.15.23 LATCH16N

ASCII - Command	**LATCH16N**		
Syntax Transmit	LATCH16N		
Syntax Receive	LATCH16N <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer16	**CANBus Object Number**	3579 (hex)
DIM	-	**PROFIBUS PNU**	1721 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	121 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	2.03		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	No
Short Description	Latched 16-bit Position (negative edge)		

Description

The LATCH16N command returns the position where latching was performed by the last negative (falling) edge on digital input 2 (IN2MODE [▸ 123]=26). The position value is absolute within one turn, and is given out in the internal units (counts 0 ... 65535). In order to get the absolute 32-bit position in SI units (taking account of the position control loop resolution PGEARI [▸ 283]/PGEARO [▸ 284]), the command LATCH32N should be used.

The commands LATCH16N and LATCH32N [▶ 261] have the effect of erasing the status bit 23 □negative latch made□ in the status register TRJSTAT [▶ 185].

4.15.24 LATCH32

ASCII - Command	**LATCH32**		
Syntax Transmit	LATCH32		
Syntax Receive	LATCH32 <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	357A (hex)
DIM	-	**PROFIBUS PNU**	1722 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	122 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	1.66		
Configuration	No	**Revision**	1.7
Function Group	Position Controller	**EEPROM**	-
Short Description	Latched 32-bit Position (positive edge)		

Description

The LATCH32 command returns the position where latching was performed by the last positive (rising) edge on digital input 2 (IN2MODE [▶ 123]=26). The position value is absolute within 4096 turns, and is given out in microns (taking account of the position control loop resolution PGEARI [▶ 283]/PGEARO [▶ 284]). To obtain an absolute position within one turn, the LATCH16 [▶ 260] command should be used.

The commands LATCH16 [▶ 260] and LATCH32 have the effect of erasing the status bit 20 □positive latch made□ in the status register DRVSTAT [▶ 171].

4.15.25 LATCH32N

ASCII - Command	**LATCH32N**		
Syntax Transmit	LATCH32N		
Syntax Receive	LATCH32N <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer32	**CANBus Object Number**	357B (hex)
DIM	-	**PROFIBUS PNU**	1723 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	123 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	2.03		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	No
Short Description	Latched 32-bit Position (negative edge)		

Description

The LATCH32N command returns the position where latching was performed by the last negative (falling) edge on digital input 2 (IN2MODE [▸ 123]=26). The position value is absolute within 4096 turns, and is given out in microns (taking account of the position control loop resolution PGEARI [▸ 283]/PGEARO [▸ 284]). To obtain an absolute position within one turn, the LATCH16N [▸ 260] command should be used.

The commands LATCH16N [▸ 260] and LATCH32N have the effect of erasing the status bit 23 □negative latch made□ in the status register TRJSTAT [▸ 185]

4.15.26 LATCHX16

ASCII - Command	LATCHX16		
Syntax Transmit	LATCH16		
Syntax Receive	LATCHX16 <Data>	Available in	
Type	Variable rw	MMI	Yes
ASCII Format	Integer16	CANBus Object Number	367F (hex)
DIM	-	PROFIBUS PNU	1983 (dec) IND = 0000xxxx (bin)
Range	-	DPR	383 (dec)
Default	-		
Opmode	All	Data Type Bus/DPR	Integer16
Drive State	-	Weightning	
Start Firmware	4.61		
Configuration	No	Revision	1.7
Function Group	Position Controller	EEPROM	-
Short Description	Latched 16-bit Position (positive edge)		

Description

The LATCHX16 command returns the position where latching was performed by the last positive (rising) edge on digital input 2 (IN1MODE [▸ 116]=26). The position value is absolute within one turn, and is given out in the internal units (counts 0 ... 65535). In order to get the absolute 32-bit position in SI units (taking account of the position control loop resolution PGEARI [▸ 283]/PGEARO [▸ 284]), the command LATCHX32 [▸ 263] should be used.

The commands LATCHX16 and LATCHX32 [▸ 263] have the effect of erasing the status bit 25 □positive latch made□ in the status register TRJSTAT [▸ 185].

4.15.27 LATCHX16N

ASCII - Command	LATCHX16N		
Syntax Transmit	LATCH16N		
Syntax Receive	LATCHX16N <Data>	Available in	
Type	Variable rw	MMI	No
ASCII Format	Integer16	CANBus Object Number	3680 (hex)
DIM	-	PROFIBUS PNU	1984 (dec) IND = 0000xxxx (bin)
Range	-	DPR	384 (dec)
Default	-		
Opmode	All	Data Type Bus/DPR	Integer16
Drive State	-	Weightning	
Start Firmware	4.61		
Configuration	No	Revision	1.7
Function Group	Position Controller	EEPROM	No

Short Description	Latched 16-bit Position (negative edge)

Description

The LATCH16XN command returns the position where latching was performed by the last negative (falling) edge on digital input1 (IN1MODE [▸ 116]=26). The position value is absolute within one turn, and is given out in the internal units (counts 0 ... 65535). In order to get the absolute 32-bit position in SI units (taking account of the position control loop resolution PGEARI [▸ 283]/PGEARO [▸ 284]), the command LATCHX32N [▸ 264] should be used.

The commands LATCHX16N and LATCHX32N [▸ 264] have the effect of erasing the status bit 26 □negative latch made□ in the status register TRJSTAT [▸ 185].

4.15.28 LATCHX32

ASCII - Command	**LATCHX32**		
Syntax Transmit	LATCHX32		
Syntax Receive	LATCHX32 <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	357C (hex)
DIM	-	**PROFIBUS PNU**	1724 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	124 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	2.07		
Configuration	No	**Revision**	1.7
Function Group	Position Controller	**EEPROM**	No
Short Description	Latched External 32-bit Position (positive edge)		

Description

If read-in from an external encoder position is activated (EXTPOS [▸ 253]=1,2), then this position will be stored automatically when a latch event (IN1MODE [▸ 116]=26) occurs.

The LATCHX32 command returns the position where latching was performed by the last positive (rising) edge on digital input 1 (IN1MODE [▸ 116]=26). The position value is absolute within 4096 turns, and is given out in microns (taking account of the position control loop resolution PGEARI [▸ 283]/PGEARO [▸ 284]). To obtain an absolute position within one turn, the LATCHX16 command should be used.

The commands LATCHX16 [▸ 262] and LATCHX32 have the effect of erasing the status bit 25 □positive latch made□ in the status register TRJSTAT [▸ 185].

4.15.29 LATCHX32N

ASCII - Command	**LATCHX32N**		
Syntax Transmit	LATCHX32N		
Syntax Receive	LATCHX32N <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	357D (hex)
DIM	-	**PROFIBUS PNU**	1725 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	125 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	2.07		
Configuration	No	**Revision**	1.7
Function Group	Position Controller	**EEPROM**	No
Short Description	Latched External 32-bit Position (negative edge)		

Description

If read-in from an external encoder position is activated (EXTPOS [▶ 253]=1,2), then this position will be stored automatically when a latch event (IN1MODE [▶ 116]=26) occurs.

The LATCHX32N command returns the position where latching was performed by the last negative (falling) edge on digital input 1 (IN1MODE [▶ 116]=26). The position value is absolute within 4096 turns, and is given out in microns (taking account of the position control loop resolution PGEARI [▶ 283]/PGEARO [▶ 284]). To obtain an absolute position within one turn, the LATCHX16N [▶ 262] command should be used.

The commands LATCHX16N [▶ 262] and LATCHX32N have the effect of erasing the status bit 26 □negative latch made□ in the status register TRJSTAT [▶ 185].

4.15.30 MH

ASCII - Command	**MH**		
Syntax Transmit	MH		
Syntax Receive	MH	**Available in**	
Type	Command	**MMI**	Yes
ASCII Format	-	**CANBus Object Number**	358D (hex)
DIM	-	**PROFIBUS PNU**	1741 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	141 (dec)
Default	-		
Opmode	8	**Data Type Bus/DPR**	-
Drive State	Enabled	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	-
Short Description	Start Homing		

Description

The MH (move home) command is used to start a homing movement (reference traverse) via the serial interface. Homing type, direction and speed are taken from the NREF [▶ 267], DREF [▶ 250] and VREF [▶ 310] parameters.

4.15.31 MJOG

ASCII - Command	**MJOG**		
Syntax Transmit	MJOG		
Syntax Receive	MJOG	**Available in**	
Type	Command	**MMI**	Yes
ASCII Format	-	**CANBus Object Number**	3591 (hex)
DIM	-	**PROFIBUS PNU**	1745 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	145 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	Enabled	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Position Controller	**EEPROM**	-
Short Description	Start Jog Mode		

Description

MJOG starts the jog mode via the serial interface. The velocity in the jog mode is taken from VJOG [▸ 309] (with ± sign). Jog mode is defined as a continuous motion at a constant velocity. This type of operation is started without a reference point being set (without homing). The hardware limit switches are monitored. Software limit switches are only monitored if a reference point is set (the drive has been homed).

Acceleration and deceleration ramps are taken from the settings for homing (see ACCR [▸ 247], DECR [▸ 249], and VJOG [▸ 309]).

4.15.32 MOVE

ASCII - Command	**MOVE**		
Syntax Transmit	MOVE [Data]		
Syntax Receive	MOVE <Data>	**Available in**	
Type	Command	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	3642 (hex)
DIM	-	**PROFIBUS PNU**	1922 (dec) IND = 0000xxxx (bin)
Range	0,1,..,180,192 .. 255	**DPR**	322 (dec)
Default	-		
Opmode	8	**Data Type Bus/DPR**	Integer16
Drive State	Enabled	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	-
Short Description	Start Motion Task		

Description

The command □MOVE nr□ starts the motion task □nr□ from the motion task memory.

If the command is used without a parameter, then the number of the most recently started task will be displayed.

4.15.33 MRD

ASCII - Command	**MRD**		
Syntax Transmit	MRD		
Syntax Receive	MRD	**Available in**	
Type	Command	**MMI**	Yes
ASCII Format	-	**CANBus Object Number**	359E (hex)
DIM	-	**PROFIBUS PNU**	1758 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	158 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	Enabled	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	-
Short Description	Homing to Resolver Zero, Mode 5		

Description

The command MRD initiates a homing movement type 5 (NREF [▸ 267]=5, to the next zero crossing point of the resolver). The velocity and the direction of movement are taken from the VREF [▸ 310] and DREF [▸ 250] variables.

4.15.34 MTMUX

ASCII - Command	**MTMUX**		
Syntax Transmit	MTMUX [Data]		
Syntax Receive	MTMUX <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer16	**CANBus Object Number**	365B (hex)
DIM	-	**PROFIBUS PNU**	1947 (dec) IND = 0000xxxx (bin)
Range	0, 192 ... 255	**DPR**	347 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	3.43		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	No
Short Description	Presetting for motion task that is processed later		

Description

The command MTMUX presets the number of a motion task that is then prepared to work with commands O_P [▸ 276],O_V [▸ 277],O_C [▸ 272],O_ACC1 [▸ 270],O_ACC2 [▸ 271],O_DEC1 [▸ 274],O_DEC2 [▸ 275],O_FT [▸ 276],O_FN [▸ 275].
All this commands then have access to the selected motion task.

MTMUX is only allowed to work with RAM motion tasks.

MTMUX is not stored in EEPROM. While start-up of the drive, MTMUX is automatically set to "0".

4.15.35 MUNIT

ASCII - Command	**MUNIT**		
Syntax Transmit	MUNIT [Data]		
Syntax Receive	MUNIT <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	3674 (hex)
DIM		**PROFIBUS PNU**	1972 (dec) IND = 0000xxxx (bin)
Range	0, 1	**DPR**	372 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	4.02		
Configuration	No	**Revision**	1.5
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Unit of the Velocity dependant motor parameters		

Description

MUNIT changes the units of the velocity dependant motor parameters, e.g. MVANGLP [▶ 238] and MSPEED [▶ 235].

- MUNIT = 0 rpm
- MUNIT = 1 the setting of VUNIT [▶ 342] is used

4.15.36 NREF

ASCII - Command	**NREF**		
Syntax Transmit	NREF [Data]		
Syntax Receive	NREF <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	35AD (hex)
DIM	-	**PROFIBUS PNU**	1773 (dec) IND = 0000xxxx (bin)
Range	0 .. 20	**DPR**	173 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Homing Mode		

Description

For linear movements, before a positioning movement can be started, a homing operation must be carried out. The reference point set in this operation is valid until the next hardware reset of the amplifier. An attempt to start positioning without a reference point being set causes a warning (LCD display n09). Any previously set reference point is cancelled before the homing operation begins.

A preset zero-point offset is considered for the position output and display. Exception: Homing=5. In this case, the true current position is displayed. You can shift the zero-crossing point of the motor shaft within one turn by using ENCZERO [▶ 316].

Zero-point recognition: The reference point is set to the first zero-crossing point of the feedback unit (zero mark) after recognition of the reference switch transition. Two-pole resolvers and all encoders have just one zero-crossing per turn, so the positioning at the zero mark is unambiguous within a motor turn. For four-pole resolvers, there are two zero-crossings per turn. For six-pole resolvers, there are three zero-crossings per turn. If the transition of the reference switch lies very close to the zero-crossing point of the feedback unit, the positioning to the zero mark can vary by one motor turn. The repetition accuracy of homing operations made without zero-point recognition depends on the traversing velocity and the mechanical design of the reference or limit switch.

For homing modes 1 and 3, a digital input must be configured as a zero-mark input (home position) (INxMODE [▸ 116]=12 or I/O expansion card).

For homing modes 2 and 4, a digital input must be configured as a hardware limit switch (INxMODE [▸ 116]=2 or INxMODE [▸ 116]=3).

For homing modes 1, 2, 3, 4, 5, and 7, the setting of the zero-pulse offset for the Encoder Equivalent Output (EEO) output is taken into account (the zero point is set so both the output of the zero pulse and the display of the zero position appear at zero-pulse offset).
The setting of the reference offset (ROFFS [▸ 293]) is taken into account for all homing modes. The zero point is assigned to a freely chosen absolute position value.

If a multiturn encoder is used, every homing move can be started. If the homing move is ready, RSOFFS is calculated automatically and a

SAVE [▸ 51] command is executed. Wenn the drive is switched off and on, the drive has the same position.

Status	Short Description	Additional Description
NREF=0	Set Reference at actual position	The actual position becomes the refernce point (the target and the actual position are set to ROFFS [▸ 293]). The distance between the actual and the target position is lost.
NREF=1	Traverse to the reference switch with zero-mark recognition.	The drive starts a move using DREF [▸ 250], until a positive edge at the reference switch is detected. Then the distance to the next zero point of the resolver is calculated and a move to this position is started. If the reference switch is present (input signal = high), a move in the opposite direction of DREF [▸ 250] is startet until a negative edge is detected. Then the homing move is started. If a hardware limit switch is detected (start of the homing move behind the reference switch), the direction is changed and a move is started, until a positive and a negative level edge is detected at the reference switch. Then the homing move is started. At the end of the homing move, the target and the actual position are set to ROFFS [▸ 293].
NREF=2	Move to hardware limit-switch, with zero-mark recognition.	The drive starts a move using DREF [▸ 250] until the hardware limit switch is reached. Then the direction is changed and the drive moves to the next zero point of the resolver. At the end of the homing move, the target and the actual position are set to ROFFS [▸ 293].
NREF=3	Move to reference switch, without zero-mark recognition.	The drive starts a move using DREF [▸ 250], until a positive edge at the reference switch is detected. The position at the edge of the reference move is eqivalent to ROFFS [▸ 293]. Then the drive stops. If the reference switch is present (input signal = high), a move in the opposite direction of DREF [▸ 250] is startet until a negative edge is detected. Then the homing move is started. If a hardware limit switch is detected (start of the homing move behind the reference switch), the direction is changed and a move is started, until a positive and a negative level edge is detected at the reference switch. Then the homing move is started. The real stop position is not the edge of the reference switch and depends on the selected speed and the deceleration ramp.

Status	Short Description	Additional Description
NREF=4	Move to hardware limit-switch, without zero-mark recognition.	The drive starts a move using DREF [▸ 250], until the hardware limit switch is detected. Then the direction is changed, and a move is started until the hardware limit switch is high again. The position at the edge of the hardware limit switch is eqivalent to ROFFS [▸ 293]. Then the drive stops. The real stop position is not the edge of the hardware limit switch and depends on the selected speed and the deceleration ramp.
NREF=5	Move to the next zero-mark of the feedback unit.	Homing to the next zero point of the resolver. The moving direction is given by variable DREF [▸ 250]. DREF [▸ 250]=0 negative DREF [▸ 250]=1 positive DREF [▸ 250]=2 the direction is given by the shortest distance.
NREF=6	Set Reference at actual position, without loosing target position	The actual position becomes the reference point (the position setpoint and the actual position are set to ROFFS [▸ 293]). The difference to NREF=0 is, that the distance between target and actual position is not lost (position error).
NREF=7	Move to mechanical stop with zero-mark recognition	When the homing mode 7 is started, the peak current limit threshold IPEAK [▸ 110] is set to REFIP [▸ 116] (peak current for the homing mode in A) in the drirection given by DREF [▸ 250] (DREF [▸ 250]=0 positive, DREF [▸ 250]=1 negative). When the drive moves the motor, the contouring error is monitored and if the error becomes higher than PEMAX [▸ 282] / 2 (half of the contouring error window), the direction is changed and a move to the next zero point of the resolver is startet. The motor stops in that position and sets the actual and the target position to ROFFS [▸ 293]. The peak current of the drive is set back to the original value of IPEAK [▸ 110].
NREF=8	Move to absolute SSI-position	When a homing mode 8 is started, the actual position of an external Multiturn SSI encoder (GEARMODE [▸ 213]=7) is read, calculated with GEARI [▸ 212] and GEARO [▸ 218] to internal counts and an offset value ROFFS2 [▸ 218] is added. The result is a target position for a motion task that is started. When the target position is reached, the IN-POSITION bit is set. This function is done for a gantry application with multiturn encoder feedback and coupling of the two drive using SSI multiturn,
NREF=9	Move to mechanical stop without zero-mark recognition	When the homing mode 7 is started, the peak current limit threshold IPEAK [▸ 110] is set to REFIP [▸ 116] (peak current for the homing mode in A) in the drirection given by DREF [▸ 250] (DREF [▸ 250]=0 positive, DREF [▸ 250]=1 negative). When the drive moves the motor, the contouring error is monitored and if the error becomes higher than PEMAX [▸ 282] / 2 (half of the contouring error window), this position is used to set the actual and the target position to ROFFS [▸ 293]. The peak current of the drive is set back to the original value of IPEAK [▸ 110].

4.15.37 NREFMT

ASCII - Command	**NREFMT**		
Syntax Transmit	NREFMT [Data]		
Syntax Receive	NREFMT <Data>	**Available in**	
Type	rw	**MMI**	No
ASCII Format	Integer16	**CANBus Object Number**	36D2 (hex)
DIM		**PROFIBUS PNU**	1666 (dec) IND = 0001xxxx (bin)
Range	0 ... 511	**DPR**	466 (dec)
Default	0		
Opmode	8	**Data Type Bus/DPR**	Integer16
Drive State		**Weightning**	
Start Firmware	5.41		
Configuration	No	**Revision**	2.0
Function Group	Position Controller	**EEPROM**	
Short Description	Homing wih following motion task		

Description

The command NREFMT will start automatically a motion task at the end of the homing.

The parameter NREFMT is a bit-variable (16 bit) FEDCBA9876543210xxxxxxxcnnnnnnnn

- Bits 0..7 (nnnnnnnn) number of the automatic startet motion task
- number = 0 no motion task will be started.
- Bit 8 =0 motion task nn will be startet after the motor stopped. The bits □homing active=0□ and □homing finnished=1□ are set before motion task nn is started.
    - =1 motion task nn starts immediately. The bits □homing active=0□ and □homing finnished=1□ are set after motion task nn is finnished.

(This option is available > firmware 5.70)

4.15.38 O_ACC1

ASCII - Command	**O_ACC1**		
Syntax Transmit	O_ACC1 [Data]		
Syntax Receive	O_ACC1 <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	35B7 (hex)
DIM	Milliseconds, mm/sec ^2	**PROFIBUS PNU**	1783 (dec) IND = 0000xxxx (bin)
Range	1 .. 32000	**DPR**	183 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	No
Short Description	Acceleration Time 1 for Motion Task 0		

Description

The command O_ACC1 can be used to define the acceleration ramp for motion task 0 (direct motion block). The scaling of the acceleration time depends on the PGEARI [▸ 283], PGEARO [▸ 284] and O_C [▸ 272] parameters.

1. Bit 12 of the motion task control variable O_C [▸ 272] is = 0.The acceleration time is given in milliseconds for acceleration from 0 to the target speed O_V [▸ 277].
2. Bit 12 of the motion task control variable O_C [▸ 272] is = 1.The acceleration is given in mm/sec^2. The resulting run-up time is calculated at the start of the motion task.

If the resolution is set to 1 (PGEARI [▸ 283]=PGEARO [▸ 284]) then internal units (counts) will be used for the speed, position and acceleration. In this case, O_ACC1 is interpreted as a run-up time in msec.

4.15.39 O_ACC2

ASCII - Command	**O_ACC2**		
Syntax Transmit	O_ACC2 [Data]		
Syntax Receive	O_ACC2 <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	35B8 (hex)
DIM	Milliseconds	**PROFIBUS PNU**	1784 (dec) IND = 0000xxxx (bin)
Range	1 .. 32000	**DPR**	184 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Position Controller	**EEPROM**	No
Short Description	Acceleration Time 2 for Motion Task 0		

Description

The command O_ACC2 defines the time taken to build up the initial acceleration for motion task 0 (direct motion task).

The following settings are possible:

- O_ACC2 = 0 the acceleration is applied instantly (V-ramp = trapeze)
- O_ACC2 = 0.5 * O_ACC1 [▸ 270] the acceleration is built up linearly (V-ramp = sine^2 form / S-curve)
- O_ACC2 < 0.5 * O_ACC1 [▸ 270] Set internally to 0.5 * O_ACC1 [▸ 270]).

Starting with firmware 4.86, also a table motion task is available. To enable this function, the bit 9 of O_C [▸ 272] must be set to "1". In this case, O_ACC2 is not used as acceleration time, but as number of the selected table. Before, that table(s) must be downloaded by UPDATE [▸ 53] Lookup.

4.15.40 O_C

ASCII - Command	O_C		
Syntax Transmit	O_C [Data]		
Syntax Receive	O_C <Data>	Available in	
Type	Variable rw	MMI	Yes
ASCII Format	Integer16	CANBus Object Number	35B9 (hex)
DIM	-	PROFIBUS PNU	1785 (dec) IND = 0000xxxx (bin)
Range	int	DPR	185 (dec)
Default	-		
Opmode	All	Data Type Bus/DPR	Integer16
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.3
Function Group	Position Controler	EEPROM	No
Short Description	Control Variable for Motion Task 0		

Description

The O_C command defines the type of motion task for the local motion task 0 (direct motion task).

A bit-variable (16 bits) is transferred as the parameter. The individual bits of this variable are interpreted as follows:

Bit	Significance	Meaning
0	0x0001	Bit for the type of motion task (relative or absolute) (see table 2)
1	0x0002	Bit for the type of the relative motion task (see table 2)
2	0x0004	Bit for the type of the relative motion task (see table 2)
3	0x0008	=0 no next motion task, at the end of the motion task, the drive stops. =1 Next motion task selected, at the end of the motion task, automatically the next motion task is started. The number of the next motion task is given by O_FN [▶ 275]
4	0x0010	Bit for the type of next motion task (see table 3)
5	0x0020	Bit for the type of next motion task (see table 3)
6	0x0040	Bit for the type of next motion task (see table 3)
7	0x0080	Bit for the type of next motion task (see table 3)
8	0x0100	Bit for the type of next motion task (see table 3)
9	0x0200	=0 The motion task is executed via the internal trajectory generator. =1 A stored lookup table profile is started. The table has to be stored in the flash of the drive. O_ACC2 [▶ 271] gives the number of the selected table. The sum of O_ACC1 [▶ 270] and O_DEC1 [▶ 274] gives the moving time of the profile. O_V [▶ 277] and O_DEC2 [▶ 275] are ignored.
10	0x0400	=0 The profile is executed in the given direction. =1 The profile is executed in the inverse direction.
11	0x0800	reserved
12	0x1000	=0 the acc and dec of the motion tasks is given in msek from "0" to the target speed. =1 the acc and dec of the motion task is given in mm/sec² (see also commands O_ACC1 [▶ 270], O_ACC2 [▶ 271], O_DEC1 [▶ 274], O_DEC2 [▶ 275]).
13	0x2000	=0 The target position and target speed of the motion task is interpreted as counts (there is no calculation needed). =1 The target position and target speed is given in SI units. There must be a calulation with PGEARI [▶ 283] and PGEARO [▶ 284] to get the internal counts (see also O_S, O_V [▶ 277], PGEARI [▶ 283], PGEARO [▶ 284]).

Bit	Significance	Meaning
14	0x4000	=0 The speed that is given in the motion task is the target speed. =1 The target speed is given by the analog setpoint 1 SW1. When a motion task is started, the anlog input is read and becomes the target speed of the motion task (Scaling: 10V=VSCALE1 [▶ 69]). The absolute of SW1 is used.
15	0x8000	Bit 3 of the type of the relative motion task (see separate table)

Type of relative/absolute Motion Task

Bit 15/2/1/0	Meaning
xxx0	Absolute Motion Task, the position value in the motion task is the new target position
x001	Relative Motion Task, the position value in the motion task is added to the old target position. The target position depends on the IN-POSITION message: IN-POSITION=1 target position = last target position + relative position of the motion task IN-POSITION = 0 target position = actual position + relative position of the motion task
x011	Relative Motion Task, the position value in the motion task is added to the old target position. target position = last target position + relative position of the motion task
x101	Relative Motion Task, the position value in the motion task is added to the old target position. target position = actual position + relative position of the motion task
0111	Relative Motion Task, the position value in the motion task is added to the old target position. target position = latched position at the positive edge of the input + relative position of the motion task (see object LATCH32 [▶ 261])
1111	Relative Motion Task, the position value in the motion task is added to the old target position. target position = latched position at the negative edge of the input + relative move of the motion task (see object LATCH32N [▶ 261]

Type of Next motion task

Bit 8/7/6/5/4	Meaning
00000	Switch over to next motion task with stop. The drive stops at the target position of the actual motion task. Then it starts the next motion task in the sequence.
00001	Switch over to next motion task without stop. The drive moves to the target position with target speed of the actual motion task. Then it starts the next motion task in the sequence.
10001	Switch over to next motion task without stop. The drive calculates the brake point, that the speed of the motor at target position becomes the speedof the next motion task in the sequence.
00010	Switch over to next motion task with stop. The drive stops at the target position of the actual motion task. The next motion task in the sequence is started, if the digital input selected by INxMODE [▶ 116]=15 is switched to low.
00110	Switch over to next motion task with stop. The drive stops at the target position of the actual motion task. The next motion task in the sequence is started, if the digital input selected by INxMODE [▶ 116]=15 is switched to high.
01000	Switch over to next motion task with stop. The drive stops at the target position of the actual motion task. The next motion task in the sequence is started after the selected delay time defined by O_FT [▶ 276]).

Bit 8/7/6/5/4	Meaning
01010	Switch over to next motion task with stop. The drive stops at the target position of the actual motion task. The next motion task in the sequence is started after the selected delay time defined by O_FT [▸ 276]) or if the digital input selected by INxMODE [▸ 116]=15 is set to low.
01110	Switch over to next motion task with stop. The drive stops at the target position of the actual motion task. The next motion task in the sequence is started after the selected delay time defined by O_FT [▸ 276]) or if the digital input selected by INxMODE [▸ 116]=15 is set to high.

4.15.41 O_DEC1

ASCII - Command	O_DEC1			
Syntax Transmit	O_DEC1 [Data]			
Syntax Receive	O_DEC1 <Data>		Available in	
Type	Variable rw		MMI	Yes
ASCII Format	Integer16		CANBus Object Number	35BA (hex)
DIM	Milliseconds, mm/sec ^2		PROFIBUS PNU	1786 (dec) IND = 0000xxxx (bin)
Range	1 .. 32000		DPR	186 (dec)
Default	-			
Opmode	All		Data Type Bus/DPR	Integer16
Drive State	-		Weightning	
Start Firmware	1.20			
Configuration	No		Revision	1.3
Function Group	Position Controller		EEPROM	No
Short Description	Braking Time 1 for Motion Task 0			

Description

The command O_DEC1 can be used to define the deceleration (braking) ramp for motion task 0 (direct motion task). The scaling of the deceleration/braking time depends on the PGEARI [▸ 283], PGEARO [▸ 284] and O_C [▸ 272] parameters.

1. Bit 12 of the motion task control variable O_C [▸ 272] is = 0. The braking time is given in milliseconds for deceleration from target speed O_V [▸ 277] down to 0.
2. Bit 12 of the motion task control variable O_C [▸ 272] is = 1. The deceleration is given in mm/sec². The resulting run-down time is calculated at the start of the motion task.

If the resolution is set to 1 (PGEARI [▸ 283]=PGEARO [▸ 284]) then internal units (counts) will be used for the speed, position and acceleration. In this case, O_DEC1 is interpreted as a run-down time in msec.

4.15.42 O_DEC2

ASCII - Command	**O_DEC2**			
Syntax Transmit	O_DEC2 [Data]			
Syntax Receive	O_DEC2 <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer16		**CANBus Object Number**	35BB (hex)
DIM	Milliseconds		**PROFIBUS PNU**	1787 (dec) IND = 0000xxxx (bin)
Range	1 .. 32000		**DPR**	187 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.8
Function Group	Position Controller		**EEPROM**	No
Short Description	Deceleration Time 2 for Motion Task 0			

Description

The command O_DEC2 defins the time taken to build up the initial deceleration for motion task 0 (direct motion task).

The following settings are possible:

- O_ADEC2 = 0 the deceleration is applied instantly (V-ramp = trapeze)
- O_DEC2 = 0.5 * O_DEC1 [▸ 274] the deceleration is built up linearly (V-ramp = sine² form / S-curve)
- O_DEC2 < 0.5 * O_DEC1 [▸ 274] Set internally to 0.5 * O_DEC1 [▸ 274])

4.15.43 O_FN

ASCII - Command	**O_FN**			
Syntax Transmit	O_FN [Data]			
Syntax Receive	O_FN <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer16		**CANBus Object Number**	35BC (hex)
DIM	-		**PROFIBUS PNU**	1788 (dec) IND = 0000xxxx (bin)
Range	0, 1, .. ,180,192 .. 255		**DPR**	188 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Position Controller		**EEPROM**	No
Short Description	Next Task Number for Motion Task 0			

Description

The command O_FN can be used to define the number of the following motion block. This number is only used if bit 3 (next block activated) of the motion block control word is set to 1.

The motion block number can have the following values:

- 0 - direct motion block
- 1 ... 180 motion block from the Flash EEPROM
- 192 ... 255 motion block from the RAM

4.15.44 O_FT

ASCII - Command	**O_FT**		
Syntax Transmit	O_FT [data]		
Syntax Receive	O_FT	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	35BD (hex)
DIM	Milliseconds	**PROFIBUS PNU**	1789 (dec) IND = 0000xxxx (bin)
Range	1 .. 32767	**DPR**	189 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	No
Short Description	Delay before Next Motion Task		

Description

This parameter can be used to delay the start of the next motion task (if one is defined). This parameter is only evaluated if bit 3 (next task activated) and bit 7 (delay time activated) of the motion task control word are set.

4.15.45 O_P

ASCII - Command	**O_P**		
Syntax Transmit	O_P [data]		
Syntax Receive	O_P	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	35BE (hex)
DIM	-	**PROFIBUS PNU**	1790 (dec) IND = 0000xxxx (bin)
Range	long int	**DPR**	190 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	No
Short Description	Target Position/Path for Motion Task 0		

Description

The command O_P can be used to define the target position for motion block 0 (direct motion block). Depending on the type of motion task (absolute or relative) this parameter will be interpreted as an absolute target position or a relative path movement. The scaling of the position depends on the PGEARI [▸ 283], PGEARO [▸ 284], PRBASE [▸ 286] and O_C [▸ 272] parameters.

1. Bit 13 of the motion block control word = 0 (given in internal units)
   The position /path is given in counts.
   Scaling: PRBASE [▸ 286]=20 -> 1048576 increments per turn
   PRBASE [▸ 286]=16 -> 65536 increments per turn
2. Bit 13 of the motion block control variable is = 1 (taking the resolution into account)
   The position is converted according to the following formula:
   Position [increments] = O_P [▸ 276] * PGEARO [▸ 284] / PGEARI [▸ 283]

If the resolution is set to 1 (PGEARI [▸ 283]=PGEARO [▸ 284]) then internal units (counts) will be used for the speed, position and acceleration.

4.15.46 O_V

ASCII - Command	O_V		
Syntax Transmit	O_V [data]		
Syntax Receive	O_V	Available in	
Type	Variable rw	MMI	Yes
ASCII Format	Integer32	CANBus Object Number	35BF (hex)
DIM	-	PROFIBUS PNU	1791 (dec) IND = 0000xxxx (bin)
Range	long int	DPR	191 (dec)
Default	-		
Opmode	All	Data Type Bus/DPR	Integer32
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.3
Function Group	Position Controller	EEPROM	No
Short Description	Target Speed for Motion Task 0		

Description

The command O_V can be used to define the target speed for motion block 0 (direct motion block). The scaling of the speed depends on the PGEARI [▸ 283], PGEARO [▸ 284], PRBASE [▸ 286] and O_C [▸ 272] parameters.

1. Bit 13 of the motion block control word = 0 (given in internal units)
   The speed is given in counts.
   Scaling: PRBASE [▸ 286]=20 -> 140/32 increments per rpm
   PRBASE [▸ 286]=16 -> 140/512 increments per rpm
2. Bit 13 of the motion block control variable is = 1 (taking the resolution into account)
   The speed is converted according to the following formula:
   Speed [increments] = O_P [▸ 276] * PGEARO [▸ 284] / PGEARI [▸ 283] / 4000

If the resolution is set to 1 (PGEARI=PGEARO) then internal units (counts) will be used for the speed.

Also see about this

O_V [▸ 277]

4.15.47 OCOPY

ASCII - Command	**OCOPY**		
Syntax Transmit	OCOPY [- Data]		
Syntax Receive	OCOPY <Data>	**Available in**	
Type	Command	**MMI**	No
ASCII Format	Integer8 Integer8	**CANBus Object Number**	No
DIM	-	**PROFIBUS PNU**	No
Range	0,1,..,180,192..255	**DPR**	No
Default	-		
Opmode	8	**Data Type Bus/DPR**	-
Drive State	Enabled (only RAM) / Disabled	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	No
Short Description	Save/copy Motion Tasks		

Description

The OCOPY command can be used to copy motion tasks from one storage location to another. The motion block number can have the following values:

- 0 - direct/local motion block
- 1 ... 180 motion blocks from the ROM. The ROM motion blocks are stored in a segment of the internal Flash EEPROM. They remain in the amplifier memory even after the 24V supply has been switched off. Write access to these motion blocks is only permitted if the output stage has been disabled.
- 192 ... 180 motion blocks from the RAM. The RAM motion blocks can also be written while the output stage is enabled. But the contents of these motion blocks will be lost if the 24V supply is switched off. When the controller is switched on, the RAM motion blocks will be initialized with the contents of ROM motion blocks 1 ... 64.

e.g.

OCOPY 0 1 save the local motion block (direct motion block /RAM) as ROM motion task 1
(The output stage must be inhibited while this command is carried out)
OCOPY 1 192 copy the first ROM motion block to the RAM (number 192)
OCOPY 1 - 16 192 copy ROM motion blocks 1 ... 16 to the RAM (192 ... 207)

4.15.48 OLIST

ASCII - Command	**OLIST**		
Syntax Transmit	OLIST [Data] [Data]		
Syntax Receive	OLIST <Data>	**Available in**	
Type	Multi-line Return Command	**MMI**	No
ASCII Format	String	**CANBus Object Number**	No
DIM	-	**PROFIBUS PNU**	No
Range	-	**DPR**	No
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	-

Short Description	List of Motion Task Data

Description

The command □OLIST x number□ is used to output the contents of □number□ motion blocks (= motion orders) one after another, starting with block □x□. The interpretation and sequence of the parameters that are shown corresponds to the parameters of the ORDER [▶ 279] command.

If the □number□ parameter is missing, then just the contents of motion block □x□ will be shown.

If both the □x□ and the □number□ parameter are missing, then the contents of all the valid motion blocks will be shown (i.e. motion blocks with valid data and correct checksums).

4.15.49 ORDER

ASCII - Command	**ORDER**		
Syntax Transmit	ORDER [Data1...Data10]		
Syntax Receive	ORDER	**Available in**	
Type	Command	**MMI**	No
ASCII Format	Integer32 ... Integer32	**CANBus Object Number**	No
DIM	-	**PROFIBUS PNU**	No
Range	0 .. 180, 192 .. 255	**DPR**	No
Default	-		
Opmode	8	**Data Type Bus/DPR**	-
Drive State	Enabled (only RAM) / Disabled	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Set Motion Task Parameters		

Description

The ORDER command can be used to define any RAM/ROM motion task (= order). The ORDER command can be used in one of three forms:

ORDER the contents of the direct motion task (nr = 0) are shown ORDER nr the contents of motion task number □nr□ are shown ORDER nr o_p o_v o_c o_acc1 o_dec1 o_acc2 o_dec2 o_fn o_ft = definition of motion task □nr□

- The □nr□ parameter specifies the number of the motion task that is to be defined. The motion task number can have the following values:
- 0 - direct/local motion task
- 1 ... 180 motion tasks from the ROM. The ROM motion tasks are stored in a segment of the internal Flash EEPROM. They remain in the amplifier memory even after the 24V supply has been switched off. Write access to these motion tasks is only permitted if the output stage has been disabled.
- 192 ... 255 motion tasks from the RAM. The RAM motion tasks can also be written while the output stage is enabled. But the contents of these motion tasks will be lost if the 24V supply is switched off. When the controller is switched on, the RAM motion tasks will be initialized with the contents of ROM motion tasks 1 ... 64.

The individual elements o_p ... o_ft have the same interpretation as the corresponding ASCII commands.

- O_P [▶ 276] target position/path for the motion task
- O_V [▶ 277] target speed/velocity
- O_C [▶ 272] type of motion task (control word)

- O_ACC1 [▸ 270] acceleration ramp /starting acceleration
- O_DEC1 [▸ 274] braking ramp / deceleration
- O_ACC2 [▸ 271] build-up time for the starting acceleration (>0 for sine² / S-curve)
- O_DEC2 [▸ 275] build-up time for the deceleration (>0 for sine² / S-curve)
- O_FN [▸ 275] number of following motion tasks
- O_FT [▸ 276] delay before starting next motion task

4.15.50 OVRIDE

ASCII - Command	**OVRIDE**		
Syntax Transmit	OVRIDE [Data]		
Syntax Receive	OVRIDE <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	35B6 (hex)
DIM	-	**PROFIBUS PNU**	1782 (dec) IND = 0000xxxx (bin)
Range	0 .. 3	**DPR**	182 (dec)
Default	0		
Opmode	8	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	2.08		
Configuration	No	**Revision**	1.9
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Override Function for Motion Tasks		

Description

The override function can be used to influence the speed/velocity for a motion block through the analog/digital interface. When this function is activated, the analog setpoint is read in every millisecond, and used for scaling the velocity for the motion block.

SW=10V motion block velocity = the target velocity that is programmed in the motion block

SW=5V motion block velocity = 50% of the programmed target velocity

The override function does not work with sin^2 curves.

The following settings are possible:

- OVRIDE=0 override function is switched off
- OVRIDE=1 SW1 input is activated for the override function
- OVRIDE=2 SW2 input is activated for the override function
- OVRIDE=3 Digital interface is activated for the override function.

The digital Interface can be:Sercos, CAN, PROFIBUS, DPR and all other field busses.

4.15.51 P1P16

ASCII - Command	**P1...P16**		
Syntax Transmit	P1 [Data]		
Syntax Receive	P1...P16 <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer32	**CANBus Object Number**	3644 (hex)
DIM	-	**PROFIBUS PNU**	1924 (dec) IND = 0000xxxx (bin)
Range	long int	**DPR**	324 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	3.20		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Fast Position Register 1 ... 16		

Description

The variables P1 ... P16 contain the position values for the position thresholds 1 ... 16.
The scaling of the position depends on the PGEARI [▸ 283], PGEARO [▸ 284], PRBASE [▸ 286] parameters, and is calculated according to the following formula:

P[increments] = P[entered] * PGEARO [▸ 284] / PGEARI [▸ 283]

- 1048576 increments/turn at PRBASE [▸ 286]=20
- 65536 increments/turn at PRBASE [▸ 286]=16

see also description of WPOS [▸ 311], WPOSE [▸ 313], WPOSP [▸ 313], WPOSX [▸ 314], POSRSTAT [▸ 286]

The object number is given for P1. The other object numbers up to P16 are the next ones.

4.15.52 PDUMP

ASCII - Command	**PDUMP**		
Syntax Transmit	PDUMP		
Syntax Receive	PDUMP <Data>	**Available in**	
Type	Multi-line Return Command	**MMI**	No
ASCII Format	String	**CANBus Object Number**	35C4 (hex)
DIM	-	**PROFIBUS PNU**	1796 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	196 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	-
Short Description	List All Position Control Variables		

Description

Produces a list of all the position control parameters.

4.15.53 PEINPOS

ASCII - Command	**PEINPOS**		
Syntax Transmit	PEINPOS [Data]		
Syntax Receive	PEINPOS <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	35C6 (hex)
DIM	PUNIT	**PROFIBUS PNU**	1798 (dec) IND = 0000xxxx (bin)
Range	long int	**DPR**	198 (dec)
Default	4000		
Opmode	>=4	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	Yes
Short Description	In-Position Window		

Description

If the distance between the actual position and the target position during the execution of an internal motion block is less than the window width that has been set, then the In-Position signal is generated (status message, digital output).

The In-Position window is entered in the same units as the position control loop (PGEARI [▶ 283] / PGEARO [▶ 284]). See description of PFB [▶ 27]

4.15.54 PEMAX

ASCII - Command	**PEMAX**		
Syntax Transmit	PEMAX [Data]		
Syntax Receive	PEMAX <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	35C7 (hex)
DIM	µm	**PROFIBUS PNU**	1799 (dec) IND = 0000xxxx (bin)
Range	long int	**DPR**	199 (dec)
Default	262144		
Opmode	>=4	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Max. Following Error		

Description

If the momentary following error (PE [▶ 26]) goes beyond the maximum value set, the motion block is stopped and the Following Error warning is generated. The motion block can only be continued (CONTINUE [▶ 249]) or restarted after the warning has been acknowledged (CLRFAULT [▶ 35], digital input INxMODE [▶ 116]=14). PEMAX=0 switches off the following error monitoring.

If a following error occurs, negative values (starting version 4.78) do not result in a stop of the axis. The status bit and warning are still there. A following motion task or new motion task cannot be started until the following error is cleared.

4.15.55 PGEARI

ASCII - Command	**PGEARI**		
Syntax Transmit	PGEARI [Data]		
Syntax Receive	PGEARI <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	35CA (hex)
DIM	µm	**PROFIBUS PNU**	1802 (dec) IND = 0000xxxx (bin)
Range	long int	**DPR**	202 (dec)
Default	10000		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	Disabled + Reset (Coldstart)	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.7
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Position Resolution (Numerator)		

Description

The parameter PGEARI is used in conjunction with the PGEARO [▶ 284] parameter to convert the control loop position and speed from SI units into increments.

The PGEARO [▶ 284] parameter contains the number of increments that are moved if the path to be moved has a length of PGEARI.

The conversion is made according to the following formula:

- Position[increments] = Position[SI] * PGEARO [▶ 284] / PGEARI
- Velocity[increments] = Velocity[SI] * PGEARO [▶ 284] / PGEARI / 4000

If PGEARI=PGEARO [▶ 284], then there will be no conversion from SI units into increments. In this case, the position and velocity must be given in increments.

- Position: 1046576 increments/turn for PRBASE [▶ 286] = 20, or 65536 increments/turn for PRBASE [▶ 286] = 16
- Velocity: 140/32 * speed in RPM

1. Example
   PGEARI = 10000
   PGEARO [▶ 284] = 1048576
   PRBASE [▶ 286] = 20

   The motion task position should be given in µm with a resolution of 10 mm/rev. The internal resolution of 20 Bit/rev (PRBASE=20) is used. Following settings have to be made:

   Position: 1046576 counts/rev at PRBASE [▶ 286] = 20 or 65536 counts/rev at PRBASE [▶ 286] = 16.

   All settings regarding position (PFB [▶ 27], O_P [▶ 276], PE [▶ 26], PEMAX [▶ 282], PEINPOS [▶ 282]) are made in µm, the settings regarding speed/velocity in µm/sec, all settings regarding acceleration in 1000µm/sec²
2. 2. Example
   PGEARI = 3600
   PGEARO [▶ 284] = 65536
   PRBASE [▶ 286] = 16

   The position is given in 0.1 degree steps with a resolution of 360.0 degrees/rev. The internal resolution of 16 Bit/rev (PRBASE [▶ 286] = 16) is used. Following settings have to be made:

All settings regarding position (PFB [▶ 27], O_P [▶ 276], PE [▶ 26], PEMAX [▶ 282], PEINPOS [▶ 282]) are made in 0.1*degree, the settings regarding speed/velocity in 0.1*degree/sec, all settings regarding acceleration in 1000*0.1*degree/sec^2

4.15.56 PGEARO

ASCII - Command	**PGEARO**		
Syntax Transmit	PGEARO [Data]		
Syntax Receive	PGEARO <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	35CB (hex)
DIM	µm	**PROFIBUS PNU**	1803 (dec) IND = 0000xxxx (bin)
Range	long int	**DPR**	203 (dec)
Default	1048576		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	Disabled + Reset (Coldstart)	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.7
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Position Resolution (Denominator)		

Description

The parameter PGEARI [▶ 283] is used in conjunction with the PGEARO parameter to convert the control loop position and speed from SI units into increments.

The PGEARO parameter contains the number of increments that are moved if the path to be moved has a length of PGEARI [▶ 283].

The conversion is made according to the following formula:

- Position[increments] = Position[SI] * PGEARO / PGEARI [▶ 283]
- Velocity[increments] = Velocity[SI] * PGEARO / PGEARI [▶ 283] / 4000

If PGEARI [▶ 283] = PGEARO, then there will be no conversion from SI units into increments. In this case, the position and velocity must be given in increments.

- Position: 1046576 increments/turn for PRBASE [▶ 286] = 20, or 65536 increments/turn for PRBASE [▶ 286] = 16
- Velocity: 140/32 * speed in RPM For an example: see PGEARI [▶ 283]

4.15.57 POSCNFG

ASCII - Command	**POSCNFG**			
Syntax Transmit	POSCNFG [Data]			
Syntax Receive	POSCNFG <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer8		**CANBus Object Number**	35CF (hex)
DIM	-		**PROFIBUS PNU**	1807 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2		**DPR**	207 (dec)
Default	0			
Opmode	8		**Data Type Bus/DPR**	Integer8
Drive State	Disabled + Reset (Coldstart)		**Weightning**	
Start Firmware	1.20			
Configuration	Yes		**Revision**	1.8
Function Group	Position Controller		**EEPROM**	Yes
Short Description	Axes Type			

Description

Use the axis type to select whether the axis is treated as a linear or rotary axis. This variable does not imply that the motor is a linear or rotary motor, but instead, affects the way the software limit switches are used by the Firmware. The software limit switches are treated in different ways, depending on the selection. The possible settings are:

POSCNFG=0	Linear Axes	Axes with a limited range of movement. The zero position for position tracking is fixed by a homing operation. With this setting, a set reference point is a precondition to be able to implement motion blocks. After the homing movement (setting the reference point) has been completed, the position is continuously tracked for the control loop, and remains valid until the amplifier is switched off.
POSCNFG=1	Rotary Axes	A rotary axes is an axes with unlimited travel. The software limit-switches have no significance in this case. A rotary axes always makes a relative movement, even if the tasks are entered as absolute ones. The actual position is set to zero with every start. A reference point is not required.
POSCNFG=2	Modulo Axis	Axes with a limited range of movement. The minimum position is SRND [▸ 296] and the maximum position is ERND [▸ 251]-1. If the maimum position ERND [▸ 251]-1 is reached, it automatically switches over to SRND [▸ 296]. The absolute target positions have to be in the defined range. If a motion task is startet, which has a absolute position outside the range, a warning "n08" is displayed (wrong motion task). Relative moves are calculated in a way, that the target position always is in the defined range. A positioning in a axes like this, gives two possibilities of direction to the target position. DREF [▸ 250] gives the possibility to restrict the direction. This axes type also needs a homing move.

Also see about this

- OPMODE [▸ 50]
- PFB [▸ 27]

4.15.58 POSRSTAT

ASCII - Command	**POSRSTAT**		
Syntax Transmit	POSRSTAT [Data]		
Syntax Receive	POSRSTAT <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer32	**CANBus Object Number**	3643 (hex)
DIM	-	**PROFIBUS PNU**	1923 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	323 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	3.20		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	No
Short Description	Status of Fast Position Registers 1 ... 16		

Description

The variable POSRSTAT returns the present status of the fast position registers.
This variable can be considered as a 32-bit variable, whereby the lower 16 bits (bits 0 ... 15) are used for the status information of position registers P1 ... P16.

- Bit=0 position signaling inactive
- Bit=1 position signaling active (position overrun for WPOSP [▶ 313]=0 or underrun for WPOSP [▶ 313]=1).

See also WPOS [▶ 311]

4.15.59 PRBASE

ASCII - Command	**PRBASE**		
Syntax Transmit	PRBASE [Data]		
Syntax Receive	PRBASE <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	35D1 (hex)
DIM	-	**PROFIBUS PNU**	1809 (dec) IND = 0000xxxx (bin)
Range	16, 20	**DPR**	209 (dec)
Default	20		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	Disabled + Reset (Coldstart)	**Weightning**	
Start Firmware	1.20		
Configuration	Yes	**Revision**	1.8
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Position Resolution		

Description

PRBASE changes the internal position resolution between 16 and 20 bits/turn. The resolution is only activated when the amplifier is switched off and then on again. The actual position is 32-bits wide. This setting dictates how many shaft revolutions are registered.

- 20 bits incremental/turn 0 to 1048575
    - Maximum path length (absolute) ±2047 turns
- 16 bits incremental/turn 0 to 65535
    - Maximum path length (absolute) ±32767 turns

When PRBASE is changed, PGEARO is automatically adjusted.
Change from 16 to 20 bit: PGEARO [▸ 284] = PGEARO [▸ 284] * 16
Change from 20 to 16 bit: PGEARO [▸ 284] = PGEARO [▸ 284] / 16
See PGEARO [▸ 284] and PGEARI [▸ 283] for additional details.

4.15.60 PTARGET

ASCII - Command	**PTARGET**			
Syntax Transmit	PTARGET			
Syntax Receive	PTARGET <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer32		**CANBus Object Number**	3654 (hex)
DIM	-		**PROFIBUS PNU**	1940 (dec) IND = 0000xxxx (bin)
Range	long int		**DPR**	340 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	3.20			
Configuration	No		**Revision**	1.3
Function Group	Position Controller		**EEPROM**	Yes
Short Description	Last Target Position			

Description

The command PTARGET can be used to request the target position for the last motion task that was started (and possibly already interrupted). This position is accepted as a new target position, as soon as the CONTINUE [▸ 249] command is executed (to continue the last motion task).

4.15.61 PTBASE

ASCII - Command	**PTBASE**			
Syntax Transmit	PTBASE [Data]			
Syntax Receive	PTBASE <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer8		**CANBus Object Number**	35D5 (hex)
DIM	-		**PROFIBUS PNU**	1813 (dec) IND = 0000xxxx (bin)
Range	1 .. 127		**DPR**	213 (dec)
Default	4 (1 msec)			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State			**Weightning**	
Start Firmware	1.30			
Configuration	No		**Revision**	1.6
Function Group	Position Controller		**EEPROM**	Yes
Short Description	Time base for the external trajectory			

Description

The PTBASE parameter is used to define the interpolation time for the external trajectory (OPMODE [▶ 50]=5). The time is set in 250 microsecond steps, and defined the time period in which the drive should reach the next position setpoint. Since the internal position control loop works in 250 microsecond steps, an interpolation of the given position setpoint (external trajectory) is also given in 250 microsecond steps.

4.15.62 PTEACH

ASCII - Command	**PTEACH**		
Syntax Transmit	PTEACH [Data]		
Syntax Receive	PTEACH <Data>	**Available in**	
Type	Command	**MMI**	No
ASCII Format	Integer8 Integer8	**CANBus Object Number**	No
DIM	-	**PROFIBUS PNU**	No
Range	0,1,..,180,192 .. 255	**DPR**	No
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	Enabled (only RAM) / Disabled	**Weightning**	
Start Firmware	1.67		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	-
Short Description	Teach-In Function		

Description

The command PTEACH can be used to accept the present position (from the position control loop) as the target position for a motion block.

Syntax: PTEACH QNR [ZNR]

- QNR Number of the source motion block
- ZNR Number of the target motion block

With the PTEACH command, the motion task ZNR is loaded in to a buffer store, the actual position is entered as the target position, and the complete motion task is written to the memory location for ZNR. If the number ZNR is not entered, then the motion task is written back to the memory location for QNR.

When the actual position is accepted, and depending on the state of the F_ART_CALCDAT bit for the type of motion task, the position is calculated either in increments (F_ART_CALCDAT=0) or as SI units (FART_CALCDAT=1).

If the number of the target motion task is within the Flash EEPROM range (1 ... 180) then the PTEACH command is only permitted while the output stage is disabled.

4.15.63 PTMIN

ASCII - Command	**PTMIN**		
Syntax Transmit	PTMIN [Data]		
Syntax Receive	PTMIN <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	35D6 (hex)
DIM	Milliseconds	**PROFIBUS PNU**	1814 (dec) IND = 0000xxxx (bin)
Range	1 .. 32767	**DPR**	214 (dec)
Default	10		
Opmode	8	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.6
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Min. Acceleration Ramp for Motion Tasks		

Description

The minimum acceleration ramp PTMIN defines the minimum time that is permitted for a velocity change from 0 to PVMAX [▸ 290]. Regardless of how the acceleration value is entered (milliseconds, SI units), the acceleration that is used is limited to PVMAX [▸ 290] / PTMIN at the start of a motion task.

With the help of the PVMAX [▸ 290] and PTMIN parameters it is possible to control the behavior of the system, especially during the commissioning phase, without having to alter the individual motion tasks.

4.15.64 PUNIT

ASCII - Command	**PUNIT**		
Syntax Transmit	PUNIT [Data]		
Syntax Receive	PUNIT <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	3660 (hex)
DIM	-	**PROFIBUS PNU**	1952 (dec) IND = 0000xxxx (bin)
Range	0 .. 10	**DPR**	352 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	4.00		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Set Resolution of the Position		

Description

PUNIT enables a systemwide setting for the unit of position for the position controller. The following settings are possible:

- PUNIT=0 internal Unit (user specific)
- PUNIT=1 1 dm (0.1 m)
- PUNIT=2 1 cm (0.01 m)
- PUNIT=6 1 µm
- PUNIT=7 0.1 µm
- PUNIT=8 0.01 µm

- PUNIT=3 1 mm
- PUNIT=4 0.1 mm
- PUNIT=5 0.01 mm
- PUNIT=9 1 nm
- PUNIT=10 0.1 nm

The parameter PUNIT is only used for the MMI. It calculates different units for the MMI. All internal calculations (position controller resolution PGEARI [▸ 283] and motion tasks are not effected.

IF PUNIT=0 there is no difference to older firmware versions. The unit is defined only by PGEARI [▸ 283].

e.g: PGEARI [▸ 283]=360 (Unit = Degree)
PGEARI [▸ 283]=3600 (Unit= 0.1 Degree)

4.15.65 PVMAX

ASCII - Command	**PVMAX**		
Syntax Transmit	PVMAX [Data]		
Syntax Receive	PVMAX <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	35D8 (hex)
DIM	VUNIT	**PROFIBUS PNU**	1816 (dec) IND = 0000xxxx (bin)
Range	0 .. long int	**DPR**	216 (dec)
Default	10000		
Opmode	8	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Max. Velocity for Position Control		

Description

The parameter PVMAX defines the maximum velocity that is permitted for a motion task. When a motion task is started, the target velocity for the motion task is limited to the value of PVMAX.

With the help of the PVMAX and PTMIN [▸ 289] parameters it is possible to control the behavior of the system, especially during the commissioning phase, without having to alter the individual motion tasks.

When used together with the PVMAXN [▸ 291] parameter, it is possible to implement a directionally-dependent velocity limit. The PVMAX determines the maximum velocity for positive and negative directions together. By making a subsequent entry for PVMAXN [▸ 291], the limit for the negative direction can be set separately.

4.15.66 PVMAXN

ASCII - Command	**PVMAXN**		
Syntax Transmit	PVMAXN [Data]		
Syntax Receive	PVMAXN <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	35D9 (hex)
DIM	VUNIT	**PROFIBUS PNU**	1817 (dec) IND = 0000xxxx (bin)
Range	0 .. long int	**DPR**	217 (dec)
Default	10000		
Opmode	8	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Max. (Negative) Velocity for Position Control		

Description

The parameter PVMAXN defines the maximum velocity (in the negative direction) that is permitted for a motion task. When a motion task is started, the target velocity for the motion task is limited to the value of PVMAXN.

When the maximum velocity for the positive direction (PVMAX [▸ 290]) is defined, the PVMAXN parameter is set to the PVMAX [▸ 290] value at the same time. So, if a separate setting is required for the negative direction of movement, the value for PVMAXN must be entered separately, afterwards.

With the help of the PVMAX [▸ 290], PTMIN [▸ 289] and PVMAXN parameters it is possible to control the behavior of the system, especially during the commissioning phase, without having to alter the individual motion tasks.

4.15.67 REFLS

ASCII - Command	**REFLS**		
Syntax Transmit	REFLS [Data]		
Syntax Receive	REFLS <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer32	**CANBus Object Number**	365D (hex)
DIM	-	**PROFIBUS PNU**	1949 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2, 3	**DPR**	349 (dec)
Default	0		
Opmode	8	**Data Type Bus/DPR**	Integer32
Drive State		**Weightning**	
Start Firmware	3.43		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Behavior of the Hardware Limit switches at Homing Move		

Description

REFLS defines the behavior of the hardware limit switch processing at homing move.

- REFLS=0 Change direction at NSTOP and PSTOP

- REFLS=1 Change direction at PSTOP, create error message F26 (limit switch) at NSTOP
- REFLS=2 Change direction at NSTOP, create error message F26 (limit switch) at PSTOP
- REFLS=3 Create error message F26 (limit switch) at NSTOP and PSTOP

Can be used for homing mode 1 and 3

4.15.68 REFMODE

ASCII - Command	**REFMODE**		
Syntax Transmit	REFMODE [Data]		
Syntax Receive	REFMODE <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	363C (hex)
DIM	-	**PROFIBUS PNU**	1916 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2, 3, 4, 5, 6, 7	**DPR**	316 (dec)
Default	0		
Opmode	8	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	2.49		
Configuration	No	**Revision**	2.0
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Source of the Zero Pulse in Homing Mode		

Description

The command REFMODE selects the source of the zero pulse in homing mode.

- REFMODE=0 Resolver- or singleturn encoder-zero, Zero crossing per rev of a multiturn encoder (Firmware 5.41) / at EXTPOS [▸ 253]=1 Data-Pin X1-connector (Drive 400 X2)
- REFMODE=1 digital INPUT1
- REFMODE=2 digital INPUT2
- REFMODE=3 digital INPUT3
- REFMODE=4 digital INPUT4
- REFMODE=5 Data-Pin of X1-Connector (Drive 400 X2)
- REFMODE=6 Zero pulse of the connector X5 (Drive 400 X4) is used (Firmware 3.43) only with FPGA [▸ 77]=1 and ENCMODE [▸ 314]=0
- REFMODE=7 Zero crossing of the absolute multiturn encoder (per rev) (Firmware 4.34)

4.15.69 REFPOS

ASCII - Command	**REFPOS**		
Syntax Transmit	REFPOS		
Syntax Receive	REFPOS <Data>	**Available in**	
Type	Variable ro	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	35E3 (hex)
DIM	Counts	**PROFIBUS PNU**	1827 (dec) IND = 0000xxxx (bin)
Range	0 .. 1048575	**DPR**	227 (dec)
Default	-		
Opmode	8	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	1.78		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	No
Short Description	Reference Switch Position		

Description

The REFPOS command returns the position (20-bit, within one turn) to be used for detecting the “Reference criterion” during the homing movement. The “Reference criterion” depends on NREF [▸ 267], the type of homing movement.

- NREF [▸ 267]=0,5,6 REFPOS = position for starting the homing movement
- NREF [▸ 267]=1,3 REFPOS = position for detecting the rising edge of the reference switch
- NREF [▸ 267]=2,4 REFPOS = position for detecting the falling edge of the reference switch
- NREF=7 REFPOS = position for detecting a stop (PE [▸ 26]>PEMAX [▸ 282] / 2)
- NREF=8 REFPOS is not altered

4.15.70 ROFFS

ASCII - Command	**ROFFS**		
Syntax Transmit	ROFFS [Data]		
Syntax Receive	ROFFS	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	35E7 (hex)
DIM	µm	**PROFIBUS PNU**	1831 (dec) IND = 0000xxxx (bin)
Range	long int	**DPR**	231 (dec)
Default	0		
Opmode	8	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Reference Offset		

Description

The ROFFS parameter can have various functions, depending on the type of position sensor that is used for the position control loop.

1. Resolver or single-turn encoder (homing to reference point is possible)
   The ROFFS parameter can be used to assign a freely chosen absolute position as the reference position (zero position) that will be reached at the end of a homing movement.
   The scaling of the position depends on the settings for PGEARI [▶ 283], PGEARO [▶ 284], PRBASE [▶ 286].
   If the resolution is set to 1 (PGEARI [▶ 283]=PGEARO [▶ 284]), then internal units (counts) will be used.
2. Absolute encoder (multi-turn, homing to reference point is also possible)
   If the position value of the absolute encoder is to be altered, this can be done with the help of the ROFFS variable. When the amplifier is switched on, the value of the ROFFS variable is added once to the position value of the absolute encoder. Since this correction is only made when the amplifier is switched on, the parameter value must be saved in the EEPROM (using the SAVE [▶ 51] command) after every alteration of the ROFFS variable, and the amplifier must then be switched off and on again (COLDSTART [▶ 170] command).

Example:

If a position 10000 is shown when the amplifier is switched on, with ROFFS=0, then entering ROFFS □10000 will shift the position to the value 0.

4.15.71 SETREF

ASCII - Command	**SETREF**		
Syntax Transmit	SETREF		
Syntax Receive	SETREF	**Available in**	
Type	Command	**MMI**	Yes
ASCII Format	-	**CANBus Object Number**	35F0 (hex)
DIM	-	**PROFIBUS PNU**	1840 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	240 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	-
Short Description	Set Reference Point		

Description

The SETREF command is used to declare the present position as the reference point (i.e. the actual position is set to the value of ROFFS [▶ 293]) and to set the bit that permits the execution of motion blocks.

The SETREF command corresponds to the execution of a homing to a reference with NREF [▶ 267]=0.

4.15.72 SETROFFS

ASCII - Command	**SETROFFS**			
Syntax Transmit	SETROFFS			
Syntax Receive	-		**Available in**	
Type	Command		**MMI**	No
ASCII Format	-		**CANBus Object Number**	35F1 (hex)
DIM	-		**PROFIBUS PNU**	1841 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	241 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	2.00			
Configuration	No		**Revision**	1.3
Function Group	Position Controller		**EEPROM**	-
Short Description	Automatic setting of ROFFS			

Description

The command SETROFFS automatically changes the reference offset variable ROFFS [▸ 293] according to the actual position (PFB [▸ 27]). This enables a automatic setting of ROFFS according to the mechanical requirements.

Example 1:

ROFFS [▸ 293]=0 PFB [▸ 27]=100
to SETROFFS ROFFS [▸ 293]=-100

Example 2:

ROFFS [▸ 293]=100 PFB [▸ 27]=70
to SETROFFS ROFFS [▸ 293]=30

4.15.73 SPSET

ASCII - Command	**SPSET**			
Syntax Transmit	SPSET [Data]			
Syntax Receive	SPSET <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer8		**CANBus Object Number**	35F5 (hex)
DIM	-		**PROFIBUS PNU**	1845 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2, 3		**DPR**	245 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	-		**Weightning**	
Start Firmware	1.81			
Configuration	No		**Revision**	1.9
Function Group	Position Controller		**EEPROM**	Yes
Short Description	Enable for S-curve			

Description

The SPSET variable generates an enable for the S-curve (= sin² ramp) for executing internal motion tasks (OPMODE=8).

- SPSET=0 Motion blocks are carried out exclusively with trapeze-form ramps.
- SPSET=1 Motion blocks are carried out with ramps which are defined within the motion tasks.
  With sin^2 curves, following motion tasks with flying change of the velocity are allways executed with intermediate stop.
  A change of this variable from 1 to 0 means that all the S-curve motion tasks will be executed as trapeze-form motion tasks, without having to modify any elements of the motion tasks.
- SPSET=2 The same as SPSET=1, except the motion blocks are always carried out with ramps which are defined within the motion tasks and the acc/dec is not changed if e.g. PVMAX [▶ 290] is changed. Especially with very small moves, the time to get to the next position was very long. (3.42)
  With sin^2 curves, following motion tasks with flying change of the speed are allways executed with intermediate stop.
  A change of this variable from 1 to 0 means that all the S-curve motion tasks will be executed as trapeze-form motion tasks, without having to modify any elements of the motion tasks.
- SPSET=3 (starting with firmware 4.91)
  Sin^2 motion tasks can be run more dynamically (tarjectory is updated every 250µs instead of 1ms) and with speed and current feed forward. The result is, that the position error while moving is much smaller and the transient oscillation at the end of the move is much better. Speed feed forward can be set by GPFFV [▶ 256] and current feed forward by GPFFT [▶ 256].
  Sin^2 motion tasks, that were programmed with SPSET=1 have the same function under SPSET=3. To select this curve generator, the Sin^2 table has to be in the Flash. Other shapes cannot be used.
- SPSET=4 (starting with firmware 5.41)
  In Addition to SPSET=3, SPSET=4 enables a table that is like the sin^2 table, but also the deviation of the acceleration is continual. This helps in case of very low frequency ringing mechanics.

4.15.74 SRND

ASCII - Command	**SRND**		
Syntax Transmit	SRND [Data]		
Syntax Receive	SRND <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer32	**CANBus Object Number**	3637 (hex)
DIM	-	**PROFIBUS PNU**	1911 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	311 (dec)
Default	- 2^31		
Opmode	-	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	2.45		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Start Position of Modulo Axes		

Description

The SRND parameter is used to define the start of the range of movement for a modulo axes (POSCNFG [▶ 285]=2). The end of the range can be set by the ERND [▶ 251] command. All positioning operations are made in the positioning range <SRND...ERND [▶ 251]-1>.

The entry for SRND is made in SI units (taking account of PGEARI [▶ 283], PGEARO [▶ 284]).

4.15.75 STOP

ASCII - Command	**STOP**			
Syntax Transmit	STOP			
Syntax Receive	STOP		**Available in**	
Type	Command		**MMI**	Yes
ASCII Format	-		**CANBus Object Number**	35FE (hex)
DIM	-		**PROFIBUS PNU**	1854 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	254 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	Enabled		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.9
Function Group	Position Controller		**EEPROM**	-
Short Description	Stop Motion Task			

Description

The STOP command breaks off the drive movement. The response of the drive varies according to the operating mode that is valid now.

1. OPMODE [▶ 50]=0 (digital velocity control)
   The STOP command has the effect of setting the velocity setpoint to 0.
   The drive brakes along the preset braking ramp for the velocity control loop (DEC [▶ 329]).
2. OPMODE [▶ 50]=2 (digital current control)
   The STOP command has the effect of setting the current setpoint to 0.
   The drive coasts down.
3. OPMODE [▶ 50]=8 (internal motion tasks)
   The STOP command has the effect of breaking off the present motion task (jog mode / homing movement).
   The drive brakes along the decel ramp that is defined in the motion task. The motion task can be restarted by CONTINUE [▶ 249] or digital input defined with INxMODE [▶ 116]=22.

The STOP command has no function in the OPMODE [▶ 50]=1,3,4,5,6,7 operating modes.

4.15.76 SWCNFG

ASCII - Command	**SWCNFG**			
Syntax Transmit	SWCNFG [Data]			
Syntax Receive	SWCNFG <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Unsigned16		**CANBus Object Number**	3600 (hex)
DIM	-		**PROFIBUS PNU**	1856 (dec) IND = 0000xxxx (bin)
Range	0 .. 65536		**DPR**	256 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Unsigned16
Drive State	Disabled + Reset (Coldstart)		**Weightning**	
Start Firmware	1.30			
Configuration	Yes		**Revision**	1.3
Function Group	Position Controller		**EEPROM**	Yes

Short Description	Configuration of Position Registers 1 ... 4

Description

6 position registers (SWE0 [▸ 301] ... SWE5 [▸ 307]) are available for monitoring functions, that can be configured as software limit-switches or cam contacts (position switches).

Registers SWE1 [▸ 302] ... SWE4 [▸ 305] can be configured with the help of the SWCNFG configuration parameter.

Registers SWE0 [▸ 301] and SWE5 [▸ 307] can be configured with the help of the SWCNFG2 [▸ 300] parameter.
A corresponding cam register SWExN is assigned to each position register SWEx.

The cam registers are only used if the cam function is activated.

The SWCNFG variable can be considered as a bit-variable. The individual bits are interpreted as follows:

- Bit 0 =0 Position/cam register SWE1 [▸ 302] is not active
    - =1 Position/cam register SWE1 [▸ 302] is active
- Bit 1 =0 Signal on going above the position (PFB [▸ 27] > SWE1 [▸ 302]) Signal if SWE1 [▸ 302] < PFB [▸ 27] < SWE1N [▸ 302] and cam function is activated
    - =1 Signal on going below the position (PFB [▸ 27] < SWE1 [▸ 302]) Signal if SWE1 [▸ 302] > PFB [▸ 27] > SWE1N [▸ 302] and cam function is activated
- Bit 2 =0 SWE1 [▸ 302] functions as signal threshold
    - =1 SWE1 [▸ 302] functions as software limit-switch 1 (left)
- Bit 3 =1 Cam function for SWE1 [▸ 302] / SWE1N [▸ 302]
- Bit 4 =0 Position/cam register SWE2 [▸ 303] is not active
    - =1 Position/cam register SWE2 [▸ 303] is active
- Bit 5 =0 Signal on going above the position (PFB [▸ 27] > SWE2 [▸ 303]) Signal if SWE2 [▸ 303] < PFB [▸ 27] < SWE2N [▸ 304] and cam function is activated
    - =1 Signal on going below the position (PFB [▸ 27] < SWE2 [▸ 303]) Signal if SWE2 [▸ 303] > PFB [▸ 27] > SWE2N [▸ 304] and cam function is activated
- Bit 6 =0 SWE2 [▸ 303] functions as signal threshold
    - =2 SWE2 [▸ 303] functions as software limit-switch 2 (right)
- Bit 7 =1 Cam function for SWE2 [▸ 303] / SWE2N [▸ 304]
- Bit 8 =0 Position/cam register SWE3 [▸ 304] is not active
    - =1 Position/cam register SWE3 [▸ 304] is active
- Bit 9 =0 Signal on going above the position (PFB [▸ 27] > SWE3 [▸ 304]) Signal if SWE3 [▸ 304] < PFB [▸ 27] < SWE3N [▸ 305] and cam function is activated
    - =1 Signal on going below the position (PFB [▸ 27] < SWE3 [▸ 304]) Signal if SWE3 [▸ 304] > PFB [▸ 27] > SWE3N [▸ 305] and cam function is activated
- Bit 10 Reserve
- Bit 11 =1 Cam function for SWE3 [▸ 304] / SWE3N [▸ 305]
- Bit 12 =0 Position/cam register SWE4 [▸ 305] is not active
    - =1 Position/cam register SWE4 [▸ 305] is active
- Bit 13 =0 Signal on going above the position (PFB [▸ 27] > SWE4 [▸ 305]) Signal if SWE4 [▸ 305] < PFB [▸ 27] < SWE4N [▸ 306] and cam function is activated
    - =1 Signal on going below the position (PFB [▸ 27] < SWE4 [▸ 305]) Signal if SWE4 [▸ 305] > PFB [▸ 27] > SWE4N [▸ 306] and cam function is activated

- Bit 14 Reserve
- Bit 15 =1 Cam function for SWE4 [▸ 305] / SWE4N [▸ 306]

The cam function is activated with the help of the cam bits (bits 3/7/11/15 of SWCNFG and bits 3/7 of SWCNFG2 [▸ 300]).

If a cam bit is set, a cam signal is generated if the actual position lies between the positions SWEx and SWExN

(x = 0 ... 5). The polarity of the cam signal can be defined by the direction bit (bits 1/5/9/13 of SWCNFG and bits 1/5 of SWCNFG2 [▸ 300]).

Output of the position signal through a digital output.

- If an I/O-expansion card (slot card) is available, the individual position signals are generated at the following outputs.
  - SWE0: Next-InPos X11B.4
  - SWE1: PosReg1 X11B.6
  - SWE2: PosReg2 X11B.7
  - SWE3: PosReg3 X11B.8
  - SWE4: PosReg4 X11B.9
  - SWE5: Reserve X11B.10

The Next-InPos and SWE0-signal functions use the same output X11B.4, so they must not be used simultaneously. If position register SWE0 has been configured, then the Next-InPos function is inhibited through the I/O card. If necessary, this function can be diverted to a digital output on the motherboard (O1MODE=16 or O2MODE=16).

- - if no I/O card is available, then the individual position signals can be given out through the outputs on the motherboard.
  - SWE0: OxMODE=28 x=1,2
  - SWE1: OxMODE=12 x=1,2
  - SWE2: OxMODE=13 x=1,2
  - SWE3: OxMODE=14 x=1,2
  - SWE4: OxMODE=15 x=1,2
  - SWE5: OxMODE=29 x=1,2

All position signals are recorded in a status register, regardless of the outputs via the digital outputs, and can be read out through the serial interface as well as through the CAN/PROFIBUS interface.

- SWE0: Bit 21 (0x00200000) of DRVSTAT
- SWE1: Bit 22 (0x00400000) of DRVSTAT
- SWE2: Bit 23 (0x00800000) of DRVSTAT
- SWE3: Bit 24 (0x01000000) of DRVSTAT
- SWE4: Bit 25 (0x02000000) of DRVSTAT
- SWE5: Bit 27 (0x08000000) of DRVSTAT

4.15.77 SWCNFG2

ASCII - Command	**SWCNFG2**		
Syntax Transmit	SWCNFG2 [Data]		
Syntax Receive	SWCNFG2 <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Unsigned16	**CANBus Object Number**	3601 (hex)
DIM	-	**PROFIBUS PNU**	1857 (dec) IND = 0000xxxx (bin)
Range	0 .. 65535	**DPR**	257 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Unsigned16
Drive State	Disabled + Reset (Coldstart)	**Weightning**	
Start Firmware	1.71		
Configuration	Yes	**Revision**	1.6
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Configuration of Position Registers 0 and 5		

Description

The configuration variable SWCNFG2 can be used to define the function of position registers 0 and 5. The SWCNFG2 variable can be considered as a bit-variable. The individual bits are interpreted as follows:

- Bit 0 =0 Position/cam register SWE0 [▸ 301] is not active
  - =1 Position/cam register SWE0 [▸ 301] is active
- Bit 1 =0 Signal on going above the position (PFB [▸ 27] > SWE0 [▸ 301]) Signal if SWE0 [▸ 301] < PFB [▸ 27] < SWE0N [▸ 301] and cam function is activated
  - =1 Signal on going below the position (PFB [▸ 27] < SWE0 [▸ 301])
    Signal if SWE0 [▸ 301] > PFB [▸ 27] > SWE0N [▸ 301] and cam function is activated
- Bit 2 Reserve
- Bit 3 =1 Cam function for SWE0 [▸ 301] / SWE0N [▸ 301]
- Bit 4 =0 Position/cam register SWE5 [▸ 307] is not active
  - =1 Position/cam register SWE5 [▸ 307] is active
- Bit 5 =0 Signal on going above the position (PFB [▸ 27] > SWE5 [▸ 307]) Signal if SWE5 [▸ 307] < PFB [▸ 27] < SWE5N [▸ 307] and cam function is activated
  - =1 Signal on going below the position (PFB [▸ 27] < SWE5 [▸ 307]) Signal if SWE5 [▸ 307] > PFB [▸ 27] > SWE5N [▸ 307] and cam function is activated
- Bit 6 Reserve
- Bit 7 =1 Cam function for SWE5 [▸ 307] / SWE5N [▸ 307]

see also description of SWCNFG [▸ 297]

4.15.78 SWE0

ASCII - Command	**SWE0**			
Syntax Transmit	SWE0 [Data]			
Syntax Receive	SWE0 <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer32		**CANBus Object Number**	3602 (hex)
DIM	-		**PROFIBUS PNU**	1858 (dec) IND = 0000xxxx (bin)
Range	long int		**DPR**	258 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	1.71			
Configuration	No		**Revision**	1.3
Function Group	Position Controller		**EEPROM**	Yes
Short Description	Position register 0			

Description

The variable SWE0 contains the position value for position register 0. The scaling of the position depends on the PGEARI [▶ 283] / PGEARO [▶ 284] / PRBASE [▶ 286] parameters, and is calculated according to the following formula:

SWE0[increments] = SWE0[input] * PGEARO [▶ 284] / PGEARI [▶ 283]

- 1048576 increments/turn for PRBASE [▶ 286]=20
- 65536 increments/turn for PRBASE [▶ 286]=16

see also description of SWCNFG [▶ 297]

4.15.79 SWE0N

ASCII - Command	**SWE0N**			
Syntax Transmit	SWE0N [Data]			
Syntax Receive	SWE0N <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer32		**CANBus Object Number**	3603 (hex)
DIM	-		**PROFIBUS PNU**	1859 (dec) IND = 0000xxxx (bin)
Range	long int		**DPR**	259 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	1.71			
Configuration	No		**Revision**	1.3
Function Group	Position Controller		**EEPROM**	Yes
Short Description	Position register 0 (Cam)			

Description

The variable SWE0N contains the cam position value for position register 0. The scaling of the position depends on the PGEARI [▶ 283] / PGEARO [▶ 284] / PRBASE [▶ 286] parameters, and is calculated according to the following formula:

SWE0N[increments] = SWE0N[input] * PGEARO [▶ 284] / PGEARI [▶ 283]

- 1048576 increments/turn for PRBASE [▶ 286]=20
- 65536 increments/turn for PRBASE [▶ 286]=16

see also description of SWCNFG [▶ 297]

4.15.80 SWE1

ASCII - Command	SWE1			
Syntax Transmit	SWE1 [Data]			
Syntax Receive	SWE1 <Data>		Available in	
Type	Variable rw		MMI	Yes
ASCII Format	Integer32		CANBus Object Number	3604 (hex)
DIM	-		PROFIBUS PNU	1860 (dec) IND = 0000xxxx (bin)
Range	long int		DPR	260 (dec)
Default	0			
Opmode	All		Data Type Bus/DPR	Integer32
Drive State	-		Weightning	
Start Firmware	1.30			
Configuration	No		Revision	1.3
Function Group	Position Controller		EEPROM	Yes
Short Description	Position register 1			

Description

The variable SWE1 contains the position value for position register 1. The scaling of the position depends on the PGEARI [▶ 283] / PGEARO [▶ 284] / PRBASE [▶ 286] parameters, and is calculated according to the following formula:

SWE1[increments] = SWE1[input] * PGEARO [▶ 284] / PGEARI [▶ 283]

- 1048576 increments/turn for PRBASE [▶ 286]=20
- 65536 increments/turn for PRBASE [▶ 286]=16

see also description of SWCNFG [▶ 297]

4.15.81 SWE1N

ASCII - Command	SWE1N			
Syntax Transmit	SWE1N [Data]			
Syntax Receive	SWE1N <Data>		Available in	
Type	Variable rw		MMI	Yes
ASCII Format	Integer32		CANBus Object Number	3605 (hex)
DIM	-		PROFIBUS PNU	1861 (dec) IND = 0000xxxx (bin)
Range	long int		DPR	261 (dec)
Default	0			
Opmode	All		Data Type Bus/DPR	Integer32
Drive State	-		Weightning	
Start Firmware	1.71			
Configuration	No		Revision	1.3
Function Group	Position Controller		EEPROM	Yes

Short Description	Position register 1 (Cam)

Description

The variable SWE1N contains the cam position value for position register 1. The scaling of the position depends on the PGEARI [▸ 283] / PGEARO [▸ 284] / PRBASE [▸ 286] parameters, and is calculated according to the following formula:

SWE1N[increments] = SWE1N[input] * PGEARO [▸ 284] / PGEARI [▸ 283]

- 1048576 increments/turn for PRBASE [▸ 286]=20
- 65536 increments/turn for PRBASE [▸ 286]=16

see also description of SWCNFG [▸ 297]

4.15.82 SWE2

ASCII - Command	**SWE2**			
Syntax Transmit	SWE2 [Data]			
Syntax Receive	SWE2 <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer32		**CANBus Object Number**	3606 (hex)
DIM	-		**PROFIBUS PNU**	1862 (dec) IND = 0000xxxx (bin)
Range	long int		**DPR**	262 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	1.30			
Configuration	No		**Revision**	1.3
Function Group	Position Controller		**EEPROM**	Yes
Short Description	Position register 2			

Description

The variable SWE2 contains the position value for position register 2. The scaling of the position depends on the PGEARI [▸ 283] / PGEARO [▸ 284] / PRBASE [▸ 286] parameters, and is calculated according to the following formula:

SWE2[increments] = SWE2[input] * PGEARO [▸ 284] / PGEARI [▸ 283]

- 1048576 increments/turn for PRBASE [▸ 286]=20
- 65536 increments/turn for PRBASE [▸ 286]=16

see also description of SWCNFG [▸ 297]

4.15.83 SWE2N

ASCII - Command	**SWE2N**		
Syntax Transmit	SWE2N [Data]		
Syntax Receive	SWE2N <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	3607 (hex)
DIM	-	**PROFIBUS PNU**	1863 (dec) IND = 0000xxxx (bin)
Range	long int	**DPR**	263 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	1.71		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Position register 2 (Cam)		

Description

The variable SWE2N contains the cam position value for position register 2. The scaling of the position depends on the PGEARI [▶ 283] / PGEARO [▶ 284] / PRBASE [▶ 286] parameters, and is calculated according to the following formula:

SWE2N[increments] = SWE2N[input] * PGEARO [▶ 284] / PGEARI [▶ 283]

- 1048576 increments/turn for PRBASE [▶ 286]=20
- 65536 increments/turn for PRBASE [▶ 286]=16

see also description of SWCNFG [▶ 297]

4.15.84 SWE3

ASCII - Command	**SWE3**		
Syntax Transmit	SWE3 [Data]		
Syntax Receive	SWE3 <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	3608 (hex)
DIM	-	**PROFIBUS PNU**	1864 (dec) IND = 0000xxxx (bin)
Range	long int	**DPR**	264 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	1.30		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Position register 3		

Description

The variable SWE3 contains the position value for position register 3.The scaling of the position depends on the PGEARI [▶ 283] / PGEARO [▶ 284] / PRBASE [▶ 286] parameters, and is calculated according to the following formula:

SWE3[increments] = SWE3[input] * PGEARO [▶ 284] / PGEARI [▶ 283]

- 1048576 increments/turn for PRBASE [▶ 286]=20
- 65536 increments/turn for PRBASE [▶ 286]=16

see also description of SWCNFG [▶ 297]

4.15.85 SWE3N

ASCII - Command	**SWE3N**		
Syntax Transmit	SWE3N [Data]		
Syntax Receive	SWE3N <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	3609 (hex)
DIM	-	**PROFIBUS PNU**	1865 (dec) IND = 0000xxxx (bin)
Range	long int	**DPR**	265 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	1.71		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Position register 3 (Cam)		

Description

The variable SWE3N contains the cam position value for position register 3. The scaling of the position depends on the PGEARI [▶ 283] / PGEARO [▶ 284] / PRBASE [▶ 286] parameters, and is calculated according to the following formula:

SWE3N[increments] = SWE3N[input] * PGEARO [▶ 284] / PGEARI [▶ 283]

- 1048576 increments/turn for PRBASE [▶ 286]=20
- 65536 increments/turn for PRBASE [▶ 286]=16

see also description of SWCNFG [▶ 297]

4.15.86 SWE4

ASCII - Command	**SWE4**		
Syntax Transmit	SWE4 [Data]		
Syntax Receive	SWE4 <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	360A (hex)
DIM	-	**PROFIBUS PNU**	1866 (dec) IND = 0000xxxx (bin)
Range	long int	**DPR**	266 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	1.30		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	Yes

Short Description	Position register 4

Description

The variable SWE4 contains the position value for position register 4. The scaling of the position depends on the PGEARI [▶ 283] / PGEARO [▶ 284] / PRBASE [▶ 286] parameters, and is calculated according to the following formula:

SWE4[increments] = SWE4[input] * PGEARO [▶ 284] / PGEARI [▶ 283]

- 1048576 increments/turn for PRBASE [▶ 286]=20
- 65536 increments/turn for PRBASE [▶ 286]=16

see also description of SWCNFG [▶ 297]

4.15.87 SWE4N

ASCII - Command	**SWE4N**		
Syntax Transmit	SWE4N [Data]		
Syntax Receive	SWE4N <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer32	**CANBus Object Number**	360B (hex)
DIM	-	**PROFIBUS PNU**	1867 (dec) IND = 0000xxxx (bin)
Range	long int	**DPR**	267 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	1.71		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Position register 4 (Cam)		

Description

The variable SWE4N contains the cam position value for position register 4. The scaling of the position depends on the PGEARI [▶ 283] / PGEARO [▶ 284] / PRBASE [▶ 286] parameters, and is calculated according to the following formula:

SWE4N[increments] = SWE4N[input] * PGEARO [▶ 284] / PGEARI [▶ 283]

- 1048576 increments/turn for PRBASE [▶ 286]=20
- 65536 increments/turn for PRBASE [▶ 286]=16

see also description of SWCNFG [▶ 297]

4.15.88 SWE5

ASCII - Command	SWE5			
Syntax Transmit	SWE5 [Data]			
Syntax Receive	SWE5 <Data>		Available in	
Type	Variable rw		MMI	Yes
ASCII Format	Integer32		CANBus Object Number	360C (hex)
DIM	-		PROFIBUS PNU	1868 (dec) IND = 0000xxxx (bin)
Range	long int		DPR	268 (dec)
Default	0			
Opmode	All		Data Type Bus/DPR	Integer32
Drive State	-		Weightning	
Start Firmware	1.71			
Configuration	No		Revision	1.3
Function Group	Position Controller		EEPROM	Yes
Short Description	Position register 5			

Description

The variable SWE5 contains the position value for position register 5. The scaling of the position depends on the PGEARI [▸ 283] / PGEARO [▸ 284] / PRBASE [▸ 286] parameters, and is calculated according to the following formula:

SWE5[increments] = SWE5[input] * PGEARO [▸ 284] / PGEARI [▸ 283]

- 1048576 increments/turn for PRBASE [▸ 286]=20
- 65536 increments/turn for PRBASE [▸ 286]=16

see also description of SWCNFG [▸ 297]

4.15.89 SWE5N

ASCII - Command	SWE5N			
Syntax Transmit	SWE5N [Data]			
Syntax Receive	SWE5N <Data>		Available in	
Type	Variable rw		MMI	Yes
ASCII Format	Integer32		CANBus Object Number	360D (hex)
DIM	-		PROFIBUS PNU	1869 (dec) IND = 0000xxxx (bin)
Range	long int		DPR	269 (dec)
Default	0			
Opmode	All		Data Type Bus/DPR	Integer32
Drive State	-		Weightning	
Start Firmware	1.71			
Configuration	No		Revision	1.3
Function Group	Position Controller		EEPROM	Yes
Short Description	Position register 5 (Cam)			

Description

The variable SWE5N contains the cam position value for position register 5.
The scaling of the position depends on the PGEARI [▸ 283] / PGEARO [▸ 284] / PRBASE [▸ 286] parameters, and is calculated according to the following formula:

SWE5N[increments] = SWE5N[input] * PGEARO [▸ 284] / PGEARI [▸ 283]

- 1048576 increments/turn for PRBASE [▸ 286]=20
- 65536 increments/turn for PRBASE [▸ 286]=16

see also description of SWCNFG [▸ 297]

4.15.90 UCOMP

ASCII - Command	**UCOMP**		
Syntax Transmit	UCOMP [Data]		
Syntax Receive	UCOMP <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer32	**CANBus Object Number**	3631 (hex)
DIM	PUNIT	**PROFIBUS PNU**	1905 (dec) IND = 0000xxxx (bin)
Range	long int	**DPR**	305 (dec)
Default	0		
Opmode	8	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	
Start Firmware	2.20		
Configuration	No	**Revision**	1.3
Function Group	Position Controller	**EEPROM**	Yes
Short Description	Backlash Compensation		

Description

For many applications it is necessary to approach motion block positions from one direction only (to avoid backlash errors arising from the interplay of the rack and pinion). To do this, at the start of a motion block the target position for the motion block is shifted by a correction value, and the motion block is only started for the real target value when this corrected position has been reached. The behavior of this function is controlled by the UCOMP parameter. The value of this parameter is the size of the correction, the sign shows the direction in which the correction is to be made. If the sign is positive, the correction is only made for positive velocities (i.e. the target position is always approached from the right), if it is negative, the correction is only made for negative velocities. This function is switched off if UCOMP is set to 0 (default setting). e.g.

1. Actual position = 0, target position = 1000, UCOMP = 100 -> the drive moves to position 1100, reverses, and stops at position 1000.
2. Actual position = 1000, target position = 0, UCOMP = 100 -> the drive moves directly to position 0
3. Actual position = 1000, target position = 0, UCOMP =-100 -> the drive moves to position -100, reverses, and stops at position 0.

4.15.91 VEXTRES

ASCII - Command	VEXTRES		
Syntax Transmit	VEXTRES [Data]		
Syntax Receive	VEXTRES <Data>	Available in	
Type	Variable rw	MMI	No
ASCII Format	Float	CANBus Object Number	3694 (hex)
DIM	-	PROFIBUS PNU	2004 (dec) IND = 0000xxxx (bin)
Range	0 .. 127	DPR	404 (dec)
Default	1		
Opmode	All	Data Type Bus/DPR	Integer32
Drive State	-	Weightning	1000
Start Firmware	4.74		
Configuration	No	Revision	1.6
Function Group	Position Controller	EEPROM	Yes
Short Description	Adjustment of the speed of the external Encoder		

Description

VEXTRES changes the scaling of the speed of an external encoder. When EXTPOS [▸ 253] = 1 is used (position information of an external encoder for the position controller), and VMIX [▸ 340] is < 1 (the speed of the external encoder is also used for the speed controller), this parameter gives the gear factor to the drive.

For example:

1. Gearing factor 12 : 1, means 1 motor turn for 12 encoder turns, then VEXTRES = 12
2. Gearing factor 1 : 12, means 12 motor turn for 1 encoder turns, then VEXTRES = 0.083 VEXTRES has up to 3 fractional digits.

4.15.92 VJOG

ASCII - Command	VJOG		
Syntax Transmit	VJOG [Data]		
Syntax Receive	VJOG <Data>	Available in	
Type	Variable rw	MMI	Yes
ASCII Format	Integer32	CANBus Object Number	3621 (hex)
DIM	µm/s	PROFIBUS PNU	1889 (dec) IND = 0000xxxx (bin)
Range	long int	DPR	289 (dec)
Default	10000		
Opmode	8	Data Type Bus/DPR	Integer32
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.6
Function Group	Position Controller	EEPROM	Yes
Short Description	Speed for Jog Mode		

Description

Jog mode is effectively an endless motion task and is implemented by the internal position control loop.

The sign for the speed indicates the direction for jog operation. The scaling of the velocity is given in position control loop units, and depends on the PGEARI [▸ 283] and PGEARO [▸ 284] parameters.

4.15.93 VREF

ASCII - Command	**VREF**			
Syntax Transmit	VREF [Data]			
Syntax Receive	VREF <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer32		**CANBus Object Number**	3628 (hex)
DIM	µm/s		**PROFIBUS PNU**	1896 (dec) IND = 0000xxxx (bin)
Range	0 .. long int		**DPR**	296 (dec)
Default	10000			
Opmode	8		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.6
Function Group	Position Controller		**EEPROM**	Yes
Short Description	Speed for Homing			

Description

The VREF is used to define the velocity value (VREF>0) for homing to a reference. The direction of the reference is taken from the DREF [▶ 250] variable.

The scaling of the velocity is given in position control loop units, and depends on the PGEARI [▶ 283] and PGEARO [▶ 284] parameters.

4.15.94 VREF0

ASCII - Command	**VREF0**			
Syntax Transmit	VREF0 [Data]			
Syntax Receive	VREF0 <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Float		**CANBus Object Number**	3698 (hex)
DIM	-		**PROFIBUS PNU**	2008 (dec) IND = 0000xxxx (bin)
Range	0.01 .. 2.0		**DPR**	408 (dec)
Default	0.125			
Opmode	8		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	4.78			
Configuration	No		**Revision**	1.5
Function Group	Position Controller		**EEPROM**	Yes
Short Description	Homing Mode Reduction factor			

Description

The parameter VREF0 reduces the homing speed, after the load reaches e.g. the reference switch, while searching for a zero pulse of an external encoder. The selection of the source of the zero pulse can be selected by REFMODE [▶ 292]. The second homing speed can be reduced by VREF0 in % of VREF [▶ 310].

Example 1:

REFMODE [▶ 292]=1 Zeropulse via digital input 1
NREF [▶ 267]=1 Homing move with reference switch with zero pulse
VREF [▶ 310]=10000 Homing speed 10000 µm/sec
VREF0=0.2 Reduction of the speed to 2000 µm/sec

Starting a homing move, the drive starts to find the reference switch with the speed of 10000 µm/sec. If the reference switch was found, the speed is reduced to 2000 µm/sec and then the search for the zero pulse is started. If the zero pulse was recognized at digital input 1(high level), the homing move is stopped.

Example 2:

REFMODE [▶ 292]=2 Zero pulse via digiatl input 2
NREF [▶ 267]=5 Zero pulse in one turn of the motor
VREF [▶ 310]=10000
VREF0=0.2

The criteria for the search of the zero pulse is fulfilled, so the reference move is directly started with 2000 µm/sec.

4.15.95 WPOS

ASCII - Command	**WPOS**			
Syntax Transmit	WPOS			
Syntax Receive	WPOS <Data>		**Available in**	
Type	Variable ro		**MMI**	No
ASCII Format	Integer8		**CANBus Object Number**	3636 (hex)
DIM	-		**PROFIBUS PNU**	1910 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2		**DPR**	310 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	Disabled + Reset (Coldstart)		**Weightning**	
Start Firmware	3.20			
Configuration	Yes		**Revision**	1.6
Function Group	Position Controller		**EEPROM**	No
Short Description	Enable Position Registers			

Description

In addition to the existing software limit-switches/position thresholds (SWCNFG [▶ 297] / SWCNFG2 [▶ 300]) there is a further option for monitoring positions. Unlike the existing solution, this new monitoring function operates in a deterministic manner. Going above/below a position is detected and signaled within 1 millisecond. The functional range of this monitoring is also expanded (continually/once).

The fast position registers are enabled through the WPOS configuration variable.

- WPOS=0 Position register disabled
- WPOS=1 Position register enabled, no spontaneous CAN message on change of status.
- WPOS=2 Position register enabled, spontaneous CAN message on change of status (this setting is only via CAN-Bus possible).

Changes of the WPOS variable between 0 and >0 can only be made offline (SAVE [▶ 51] and COLDSTART [▶ 170]), a change between 1 and 2 can also be made online.

There is a total of 16 position registers P1 ... P16, that can be configured with the help of 3 control variables. The position signals are indicated through a status variable. All control/status variables can be considered as 32-bit variables, whereby the lower 16 bits (bits 0 ... 15) are used for the configuration of the position registers P1 ... P16.

Control variables

- WPOSE [▶ 313] Enable/disable a position register.
  - Bit=0 the corresponding position register is not monitored
  - Bit=1 the position register is monitored
- WPOSP [▶ 313] Polarity for the position signaling.
  - Bit=0 Position signal is generated on going above/beyond (overrun) the position
  - Bit=1 Position signal is generated on going below/behind (underrun) the position
- WPOSX [▶ 314] Type of position monitoring
  - Bit=0 position is monitored continuously
  - Bit=1 position is monitored once. When the position signal is generated, the corresponding enable bit (WPOSE) is set to 0, so that the monitoring is disabled for this position register,

Status variable

- POSRSTAT [▶ 286] (z_data.Posrstat) Position signaling
  - Bit=0 position signaling inactive
  - Bit=1 position signaling active (position overrun for WPOSP [▶ 313]=0 or underrun for WPOSP [▶ 313]=1).

Position register

The position registers 1 to 16 can be accessed by the ASCII command P1 ... P16. Position values are displayed in the same units as the position control loop (PGEARI [▶ 283] / PGEARO [▶ 284] conversion).

The variables that are required for the fast position registers (WPOSE [▶ 313], WPOSP [▶ 313], WPOSX [▶ 314], P1 ... P16) can be saved in the serial EEPROM by using the SAVE command. Those position registers P1 ... P16 which are not used should be set to 0 (since the default value for a position register is 0, no space will be occupied in the serial EEPROM).

The individual position signals from the status register POSRSTAT [▶ 286] can be output from the digital outputs of the motherboard.

- OxMODE [▶ 150]=40
  This function is used to produce the result of a logical OR operation (on the bit-variable POSRSTAT [▶ 286] and a bit-mask from the auxiliary variable OxTRIG [▶ 155]) at the digital output x.
- OxMODE [▶ 150]=41
  This function is used to produce the result of a logical AND operation (on the bit-variable POSRSTAT [▶ 286] and a bit-mask from the auxiliary variable OxTRIG [▶ 155]) at the digital output x.

4.15.96 WPOSE

ASCII - Command	WPOSE			
Syntax Transmit	WPOSE			
Syntax Receive	WPOSE <Data>		Available in	
Type	Variable ro		MMI	No
ASCII Format	Integer32		CANBus Object Number	363F (hex)
DIM	-		PROFIBUS PNU	1919 (dec) IND = 0000xxxx (bin)
Range	0 ..65535		DPR	319 (dec)
Default	0			
Opmode	All		Data Type Bus/DPR	Integer32
Drive State	-		Weightning	
Start Firmware	3.20			
Configuration	No		Revision	1.6
Function Group	Position Controller		EEPROM	No
Short Description	Enable Fast Position Registers 1 ... 16			

Description

The bit-variable WPOSE can be used to enable or disable the fast position registers P1 ... P16. The WPOSE variable can be considered as a 32-bit variable, whereby the lower 16 bits (bits 0 ... 15) are used for the configuration of the position registers P1 ... P16.

- Bit=0 the corresponding position register is not monitored
  Bit=1 the position register is monitored

See also WPOS [▸ 311]

4.15.97 WPOSP

ASCII - Command	WPOSP			
Syntax Transmit	WPOSP [Data]			
Syntax Receive	WPOSP <Data>		Available in	
Type	Variable rw		MMI	No
ASCII Format	Integer32		CANBus Object Number	3640 (hex)
DIM	-		PROFIBUS PNU	1920 (dec) IND = 0000xxxx (bin)
Range	0 .. 65535		DPR	320 (dec)
Default	0			
Opmode	All		Data Type Bus/DPR	Integer32
Drive State	-		Weightning	
Start Firmware	3.20			
Configuration	No		Revision	1.6
Function Group	Position Controller		EEPROM	No
Short Description	Polarity of Fast Position Registers 1 ... 16			

Description

The bit-variable WPOSP can be used to configure the fast position registers P1 ... P16 individually. The WPOSP variable can be considered as a 32-bit variable, whereby the lower 16 bits (bits 0 ... 15) are used for the configuration of the position registers P1 ... P16.

- Bit=0 Position signal is generated on going above/beyond (overrun) the position
- Bit=1 Position signal is generated on going below/behind (underrun) the position

See also WPOS [▸ 311]

4.15.98 WPOSX

ASCII - Command	**WPOSX**			
Syntax Transmit	WPOSX [Data]			
Syntax Receive	WPOSX <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer32		**CANBus Object Number**	3641 (hex)
DIM	-		**PROFIBUS PNU**	1921 (dec) IND = 0000xxxx (bin)
Range	0 .. 65535		**DPR**	321 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	
Start Firmware	3.20			
Configuration	No		**Revision**	1.6
Function Group	Position Controller		**EEPROM**	No
Short Description	Mode of Fast Position Registers 1 ... 16			

Description

The bit-variable WPOSX can be used to configure the fast position registers P1 ... P16 individually. The WPOSX variable can be considered as a 32-bit variable, whereby the lower 16 bits (bits 0 ... 15) are used for the configuration of the position registers P1 ... P16.

- Bit=0 position is monitored continuously
- Bit=1 position is monitored once. When the position signal is generated, the corresponding enable bit (WPOSE [▸ 313]) is set to 0, so that the monitoring is disabled for this position register,

See also WPOS [▸ 311]

4.16 Position Output

4.16.1 ENCMODE

ASCII - Command	**ENCMODE**			
Syntax Transmit	ENCMODE [Data]			
Syntax Receive	ENCMODE <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer8		**CANBus Object Number**	3534 (hex)
DIM	-		**PROFIBUS PNU**	1652 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2, 3		**DPR**	52 (dec)
Default	1			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	2.0
Function Group	Position Output		**EEPROM**	Yes
Short Description	Selection of Encoder Emulation			

Description

Selection of the encoder emulation

- ENCMODE=0 Encoder emulation switched off
- ENCMODE=1 EEO (ROD) output
- ENCMODE=2 SSI output
- ENCMODE=3 EEO (ROD) interpolation mode
  This mode is available with high resolution feedback device (FBTYPE [▸ 190]>0).
  The encoder output brings ENCOUT [▸ 315] * ENCLINES [▸ 223] lines per motor rev. Following settings are possible: 4,8,16,32,64,128

4.16.2 ENCOUT

ASCII - Command	**ENCOUT**		
Syntax Transmit	ENCOUT [Data]		
Syntax Receive	ENCOUT <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	3535 (hex)
DIM	CPR	**PROFIBUS PNU**	1653 (dec) IND = 0000xxxx (bin)
Range	see Description	**DPR**	53 (dec)
Default	1024		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.9
Function Group	Position Output	**EEPROM**	Yes
Short Description	Resolution Encoder Emulation EEO (ROD)		

Description

The resolution of the encoder emulation EEO (ROD)

ENCOUT defines the number of lines that are given out by the EEO (ROD) interface for one turn of the motor.

Resolver feedback (FBTYPE [▸ 190]=0) allows lines per rev from 256 to 4096 with all integer numbers between. One zero pulse per rev.

Encoder feedback (FBTYPE [▸ 190]=2,4,7) allows all numbers from 256 to 524288, but only degrees of 2 (256, 512, 1024, .. , 262144, 524288).

Additional values have been added in version 4.32.

Starting with firmware 4.94 all integer numbers between 256 and 4096 are enabled also for FBTYPE [▸ 190]= 2 and 4.

4.16.3 ENCZERO

ASCII - Command	**ENCZERO**		
Syntax Transmit	ENCZERO [Data]		
Syntax Receive	ENCZERO <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	3537 (hex)
DIM	-	**PROFIBUS PNU**	1655 (dec) IND = 0000xxxx (bin)
Range	0 .. ENCOUT-1	**DPR**	55 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Position Output	**EEPROM**	Yes
Short Description	Zero Pulse Offset EEO (ROD)		

Description

The ENCZERO command can be used to shift the output of the EEO (ROD) zero pulse over the range of one turn. The shift is made in the clockwise direction, e.g.

- ENCOUT [▶ 315] 1024
- ENCZERO 256

The zero pulse is given out at the 90° position.This is also effective for SSI outputs.

4.16.4 SSIMODE

ASCII - Command	**SSIMODE**		
Syntax Transmit	SSIMODE [Data]		
Syntax Receive	SSIMODE <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	35F8 (hex)
DIM	-	**PROFIBUS PNU**	1848 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2	**DPR**	248 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	2.12		
Configuration	No	**Revision**	2.0
Function Group	Position Output	**EEPROM**	Yes
Short Description	SSI Mode		

Description

The SSIMODE parameter defines the type of SSI output or SSI read-in procedure at connector X5 (Drive 400 X4).

1. SSI output (GEARMODE [▶ 213]<>7, ENCMODE [▶ 314]=2)
   With SSI output it is possible to switch between single-turn and multi-turn output (from firmware version 2.12).

SSIMODE 0 single-turn
SSIMODE 1 multi-turn
To switch over from 12 Bit Format per rev to 15 Bit Format per rev, set FPGA [▶ 77]=4.

2. SSI read-in
When reading in an SSI value (GEARMODE [▶ 213]=7,ENCMODE [▶ 314]=2)
it is possible to use the SSIMODE parameter to define the position within the SSI bit-stream which is used for transmitting the alarm bit.
    - SSIMODE=0 no alarm bit
    - SSIMODE=1 alarm bit first
    - SSIMODE=2 alarm bit last

4.17 Rack Drive Panel

4.17.1 RDP

ASCII - Command	**RDP**		
Syntax Transmit	RDP [Data]		
Syntax Receive	RDP <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Char	**CANBus Object Number**	36B7 (hex)
DIM	-	**PROFIBUS PNU**	2039 (dec) IND = 0000xxxx (bin)
Range	0 .. 2	**DPR**	439 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Char
Drive State	Disabled + Reset (Coldstart)	**Weightning**	
Start Firmware	4.96		
Configuration	Yes	**Revision**	1.9
Function Group	Rack Drive Panel	**EEPROM**	Yes
Short Description	Activate Racjk Drive Panel Mode		

Description

RDP is used to enable RDP (Rack Drive Panel anti-backlash control) and to select the RDP operating mode. See the Application Note □Rack Drive Panel (RDP):

Controlling Backlash□ for more information.

- 0 RDP disabled; normal operation of the amplifier.
- 1 RDP mode enabled; Enable mode = □Fault Disables Other Drive.□
- 2 RDP mode enabled; Enable mode = □Fault Forces Other Drive From RDP.□

See also: RDPBIAS [▶ 318], RDPCLAMP [▶ 318], RDPKI [▶ 319], RDPKP [▶ 320], RDPON [▶ 320], RDPINT [▶ 319]

4.17.2 RDPBIAS

ASCII - Command	**RDPBIAS**			
Syntax Transmit	RDPBIAS [Data]			
Syntax Receive	RDPBIAS <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Float		**CANBus Object Number**	36B8 (hex)
DIM	Amp		**PROFIBUS PNU**	2040 (dec) IND = 0000xxxx (bin)
Range	0 .. ICONT		**DPR**	440 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	4.96			
Configuration	No		**Revision**	1.9
Function Group	Rack Drive Panel		**EEPROM**	Yes
Short Description	Rack Drive Panel Bias Current			

Description

The RDP bias current in Amps. Normally set at 25% - 50% of ICONT [▸ 108]. Set positive in one drive and negative in the other. See also: RDP [▸ 317]

4.17.3 RDPCLAMP

ASCII - Command	**RDPCLAMP**			
Syntax Transmit	RDPCLAMP [Data]			
Syntax Receive	RDPCLAMP <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Float		**CANBus Object Number**	36B9 (hex)
DIM	Rpm		**PROFIBUS PNU**	2041 (dec) IND = 0000xxxx (bin)
Range	0 .. VLIM		**DPR**	441 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	4.96			
Configuration	No		**Revision**	1.9
Function Group	Rack Drive Panel		**EEPROM**	Yes
Short Description	Max. Velocity Offset of the Rack Drive Panel Circuit			

Description

Maximum output of the RDP equalization circuit in rpm. Normally set at 50 rpm in the master drive. Set to 0 in the slave drive. See also: RDP [▸ 317]

4.17.4 RDPINT

ASCII - Command	RDPINT		
Syntax Transmit	RDPINT		
Syntax Receive	RDPINT <Data>	Available in	
Type	Variable ro	MMI	No
ASCII Format	Integer32	CANBus Object Number	36BA (hex)
DIM	-	PROFIBUS PNU	2042 (dec) IND = 0000xxxx (bin)
Range	-	DPR	442 (dec)
Default	-		
Opmode	All	Data Type Bus/DPR	Integer32
Drive State	-	Weightning	
Start Firmware	4.96		
Configuration	No	Revision	1.9
Function Group	Rack Drive Panel	EEPROM	No
Short Description	Rack Drive Panel Test Variable		

Description

Rack Drive Panel test variable. Equals the integral value of the Rack Drive Panel equalization circuit. Scaled for 9000000 = 1 rpm (this is 9 million = 1 rpm). Limited by RDCLAMP (scaled in rpm). See also RDP [▶ 317]

4.17.5 RDPKI

ASCII - Command	RDPKI		
Syntax Transmit	RDPKI [Data]		
Syntax Receive	RDPKI <Data>	Available in	
Type	Variable rw	MMI	No
ASCII Format	Integer16	CANBus Object Number	36BB (hex)
DIM	-	PROFIBUS PNU	2043 (dec) IND = 0000xxxx (bin)
Range	-	DPR	443 (dec)
Default	0		
Opmode	All	Data Type Bus/DPR	Integer16
Drive State	-	Weightning	
Start Firmware	4.96		
Configuration	No	Revision	1.9
Function Group	Rack Drive Panel	EEPROM	Yes
Short Description	Integral gain of Rach Drive Panel		

Description

Integral Gain of the RDP equalization circuit. Normally set at 13 in the master drive. Set to 0 in the slave drive. See also RDP [▶ 317]

4.17.6 RDPKP

ASCII - Command	**RDPKP**		
Syntax Transmit	RDPKP [Data]		
Syntax Receive	RDPKP <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer16	**CANBus Object Number**	36BC (hex)
DIM	-	**PROFIBUS PNU**	2044 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	444 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	4.96		
Configuration	No	**Revision**	1.9
Function Group	Rack Drive Panel	**EEPROM**	Yes
Short Description	Proportional Gain of Rach Drive Panel		

Description

Proportional Gain of equalization circuit. Normall set at 300 in the master drive. Set to 0 in the slave drive. See also RDP [▶ 317]

4.17.7 RDPON

ASCII - Command	**RDPON**		
Syntax Transmit	RDPON		
Syntax Receive	RDPON <Data>	**Available in**	
Type	Variable ro	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	36BD (hex)
DIM	-	**PROFIBUS PNU**	2045 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	445 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	4.96		
Configuration	No	**Revision**	1.9
Function Group	Rack Drive Panel	**EEPROM**	No
Short Description	Test Variable Rack Drive Panel		

Description

Rack Drive Panel test variable. Will read 1 only when Rack Drive Panel is active, otherwise 0. See also RDP [▶ 317]

4.18 Sercos

4.18.1 SBAUD

ASCII - Command	SBAUD			
Syntax Transmit	SBAUD [Data]			
Syntax Receive	SBAUD <Data>		Available in	
Type	Variable rw		MMI	Yes
ASCII Format	Integer8		CANBus Object Number	35EC (hex)
DIM	Mbaud		PROFIBUS PNU	1836 (dec) IND = 0000xxxx (bin)
Range	2, 4		DPR	236 (dec)
Default	4			
Opmode	All		Data Type Bus/DPR	Integer8
Drive State	-		Weightning	
Start Firmware	1.67			
Configuration	No		Revision	1.3
Function Group	Sercos		EEPROM	Yes
Short Description	Sercos: Baud Rate			

Description

This parameter sets the transmission rate for SERCOS in MBAUD. The possible settings are 2 and 4 MBAUD

4.18.2 SERCERR

ASCII - Command	SERCERR			
Syntax Transmit	SERCERR			
Syntax Receive	SERCERR <Data>		Available in	
Type	Variable ro		MMI	No
ASCII Format	Integer32		CANBus Object Number	No
DIM	-		PROFIBUS PNU	No
Range	0 .. 8		DPR	No
Default	0			
Opmode	All		Data Type Bus/DPR	-
Drive State	-		Weightning	
Start Firmware				
Configuration	No		Revision	1.3
Function Group	Sercos		EEPROM	No
Short Description	Display Error State of Object SERCOS			

Description

The command SERERR displays an error generated by an wrong access with the command SERCOS [▶ 322] to an IDN. See also object SERCOS [▶ 322].

4.18.3 SERCLIST

ASCII - Command	**SERCLIST**			
Syntax Transmit	SERCLIST [Data]			
Syntax Receive	SERCLIST <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer32		**CANBus Object Number**	No
DIM			**PROFIBUS PNU**	No
Range	0 .. 8		**DPR**	No
Default	0			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware				
Configuration	No		**Revision**	1.3
Function Group	Sercos		**EEPROM**	No
Short Description	Set Sercos IDN Pointer			

Description

The command SERCLIST enables the access to an element of the IDN list. After that, the IDN can be read by command SERCOS [▶ 322].

4.18.4 SERCOS

ASCII - Command	**SERCOS**			
Syntax Transmit	SERCOS [Data]			
Syntax Receive	SERCOS <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer32		**CANBus Object Number**	No
DIM	-		**PROFIBUS PNU**	No
Range	0 .. 8		**DPR**	No
Default	0			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware				
Configuration	No		**Revision**	1.3
Function Group	Sercos		**EEPROM**	No
Short Description	Read the Data of an Sercos IDN			

Description

Gives the contents of an Sercos IDN. Write access changes the selected number of the IDN, read access gives the contents of the selected IDN. If the selected IDN is a list, only the list value where SERCLIST [▶ 322] points to is displayed. If SERCOS generates an error (e.g. wrong IDN number), SERCERR [▶ 321] is set to "1" and a value of "0" is displayed.

4.18.5 SERCSET

ASCII - Command	**SERCSET**		
Syntax Transmit	SERCSET [Data]		
Syntax Receive	SERCSET <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer32	**CANBus Object Number**	3691 (hex)
DIM	-	**PROFIBUS PNU**	2001 (dec) IND = 0000xxxx (bin)
Range	Long Int	**DPR**	401 (dec)
Default	0		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware			
Configuration	No	**Revision**	1.6
Function Group	Sercos	**EEPROM**	No
Short Description	Set Sercos Settings		

Description

The object SERCSET gives the possiblity to change some of the Sercos settings. Changes must be saved in the EEPROM and effect at the next start-up of the drive. The not described bits effect other Sercos settings. In so far, this command should only be used in combination with the MMI. See also Sercos IDN Manual.

- Bit 0: Hardware Limit Switch Effect (P-IDN 3015)
- Bit 1: CLRFAULT [▸ 35] Command Effect (P-IDN 3016)
- Bit 4: Polarity Target Position (S-IDN 55)
- Bit 6: Polaity Actual Position 1 (S-IDN 55)
- Bit 7: Polaity Actual Position 2 (S-IDN 55)
- Bit 12: Polarity Target Speed (S-IDN 43)
- Bit 14: Polarity Actual Speed (S-IDN 43)

4.18.6 SLEN

ASCII - Command	**SLEN**		
Syntax Transmit	SLEN [Data]		
Syntax Receive	SLEN <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	35F2 (hex)
DIM	m	**PROFIBUS PNU**	1842 (dec) IND = 0000xxxx (bin)
Range	0 .. 45	**DPR**	242 (dec)
Default	5		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	1.67		
Configuration	No	**Revision**	1.3
Function Group	Sercos	**EEPROM**	Yes
Short Description	Sercos Optical Range		

Description

This parameter can be used to set the optical range (in meters) for a standardized 1mm² plastic optical fiber cable.

4.18.7 SPHAS

ASCII - Command	**SPHAS**		
Syntax Transmit	SPHAS		
Syntax Receive	SPHAS <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Integer8	**CANBus Object Number**	35F4 (hex)
DIM	-	**PROFIBUS PNU**	1844 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	244 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	-	**Weightning**	
Start Firmware	1.67		
Configuration	No	**Revision**	1.3
Function Group	Sercos	**EEPROM**	No
Short Description	Sercos Phase		

Description

Shows the present Sercos phase.

- Phase 0 Close ring and reset
- Phase 1 Drive identification
- Phase 2 Communication initialization
- Phase 3 Parameter initialization
- Phase 4 Ready for operation

4.18.8 SSTAT

ASCII - Command	**SSTAT**		
Syntax Transmit	SSTAT		
Syntax Receive	SSTAT <Data>	**Available in**	
Type	Variable ro	**MMI**	Yes
ASCII Format	String	**CANBus Object Number**	35FA (hex)
DIM	-	**PROFIBUS PNU**	1850 (dec) IND = 0000xxxx (bin)
Range	-	**DPR**	250 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	-
Drive State	-	**Weightning**	
Start Firmware	1.67		
Configuration	No	**Revision**	1.3
Function Group	Sercos	**EEPROM**	No
Short Description	Sercos Status		

Description

Presents the actual status of the Sercos interface, as a text string.

4.19 Velocity Controller

4.19.1 ACC

ASCII - Command	**ACC**			
Syntax Transmit	ACC [Data]			
Syntax Receive	ACC <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer16		**CANBus Object Number**	3501 (hex)
DIM	Milliseconds		**PROFIBUS PNU**	1601 (dec) IND = 0000xxxx (bin)
Range	1 .. 32767, VLIM * 4480 (5.41)		**DPR**	1 (dec)
Default	10			
Opmode	0, 1, 8		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.9
Function Group	Velocity Controller		**EEPROM**	Yes
Short Description	Acceleration Ramp			

Description

This variable defines the acceleration ramp for the velocity control loop (in msec), in reference to the maximum velocity (the larger value of VLIM [▸ 338] and VLIMN [▸ 339]). The acceleration ramp is only used for setpoint changes resulting in a velocity increase (acceleration). DEC [▸ 329] is used for braking (deceleration). For a setpoint step from 0 to VLIM [▸ 338] or VLIMN [▸ 339], the ramp generator generates a stepped ramp (with steps of 250 microseconds) that is completed within the set ACC time.

4.19.2 BQDC

ASCII - Command	**BQDC**			
Syntax Transmit	BQDC [Data]			
Syntax Receive	BQDC <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Float		**CANBus Object Number**	3662 (hex)
DIM	-		**PROFIBUS PNU**	1954 (dec) IND = 0000xxxx (bin)
Range	0.2 .. 1		**DPR**	354 (dec)
Default	0.3			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	4.00			
Configuration	No		**Revision**	1.7
Function Group	Velocity Controller		**EEPROM**	Yes
Short Description	Defines the Center Damping of the Bi-quad Filter			

Description

The BQDC defines the center damping of the Bi-quad filter, which can be normally set to the default value. The activation is done by BQMODE [▸ 327].

4.19.3 BQDR

ASCII - Command	**BQDR**		
Syntax Transmit	BQDR [Data]		
Syntax Receive	BQDR <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Float	**CANBus Object Number**	3663 (hex)
DIM	-	**PROFIBUS PNU**	1955 (dec) IND = 0000xxxx (bin)
Range	0.25 .. 5	**DPR**	355 (dec)
Default	2.0		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	4.00		
Configuration	No	**Revision**	1.7
Function Group	Velocity Controller	**EEPROM**	Yes
Short Description	Defines the Damping Ratio of the Bi-quad Filter		

Description

The BQDR defines the damping ratio of the Bi-quad filter, which can be normally set to the default value. The activation is done by BQMODE [▸ 327]

4.19.4 BQFC

ASCII - Command	**BQFC**		
Syntax Transmit	BQFC [Data]		
Syntax Receive	BQFC <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Float	**CANBus Object Number**	3664 (hex)
DIM	Hz	**PROFIBUS PNU**	1956 (dec) IND = 0000xxxx (bin)
Range	20 .. 1000	**DPR**	356 (dec)
Default	200		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	4.00		
Configuration	No	**Revision**	1.4
Function Group	Velocity Controller	**EEPROM**	Yes
Short Description	Center Frequency of the Bi-Quad Filter		

Description

The BQFC defines the center frequency of the Bi-quad filter, which can be calculated according to the following equation:

BQFC = SQRT(OmegaAR * OmegaR) [Hz]

Here, the anti-resonance frequency OmegaAR and the resonance frequency OmegaR can be respectively read from the bode plot of the velocity control loop.

How to make the bode plot and to set Bi-quad filter please reference the application note “Suppression of Torsional Oscillations”.

4.19.5 BQFR

ASCII - Command	**BQFR**			
Syntax Transmit	BQFR [Data]			
Syntax Receive	BQFR <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Float		**CANBus Object Number**	3665 (hex)
DIM	-		**PROFIBUS PNU**	1957 (dec) IND = 0000xxxx (bin)
Range	0.1 .. 10		**DPR**	357 (dec)
Default	2.5			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	4.00			
Configuration	No		**Revision**	1.4
Function Group	Velocity Controller		**EEPROM**	Yes
Short Description	Frequency Ratio of the Bi-quad Filter			

Description

The BQFR defines the frequency ratio of the Bi-quad filter, which is the ratio between the resonance frequency OmegaR and the anti-resonance frequency OmegaAR. That is: .

- Here, the anti-resonance frequency OmegaAR and the resonance frequency OmegaR can be respectively read from the bode plot of the velocity control loop.

How to make the bode plot and to set Bi-quad filter please reference the application note □Suppression of Torsional Oscillations□.

4.19.6 BQMODE

ASCII - Command	**BQMODE**			
Syntax Transmit	BQMODE [Data]			
Syntax Receive	BQMODE <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Integer8		**CANBus Object Number**	3666 (hex)
DIM	-		**PROFIBUS PNU**	1958 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2, 3		**DPR**	358 (dec)
Default	1			
Opmode	All		**Data Type Bus/DPR**	Integer8
Drive State	-		**Weightning**	
Start Firmware	4.00			
Configuration	No		**Revision**	1.4
Function Group	Velocity Controller		**EEPROM**	Yes
Short Description	Select Compensation Filter Mode for the Velocity Control			

Description

The BQMODE is used to set the compensation filter mode for the velocity control. There are following three possible settings:

- BQMODE = 0 : without any filter after the PI velocity controller
- BQMODE = 1 : using PID-T2 compensation filter
- BQMODE = 2 : using Bi-quad filter

- BQMODE = 3 : reserved

By default, the BQMODE is set to PID-T2 filter mode to reduce the high-frequency noise included in the torque current, which are determined by parameters GVFILT [▶ 334] and GVT2 [▶ 335].

If the two-mass servo drive system has mechanical resonance (torsional oscillation) in the frequency range between 100 Hz and 500 Hz, Bi-quad filter can be used to suppress this kind of resonance and to enhance the bandwidth of the velocity control loop (See BQFC [▶ 326], BQFR [▶ 327], BQDC [▶ 325] and BQDR [▶ 326]).

4.19.7 DAOFFSET1

ASCII - Command	**DAOFFSET1**		
Syntax Transmit	DAOFFSET1 [Data]		
Syntax Receive	DAOFFSET1 <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	3520 (hex)
DIM	Counts	**PROFIBUS PNU**	1632 (dec) IND = 0000xxxx (bin)
Range	0 .. 2500	**DPR**	32 (dec)
Default	1290		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Velocity Controller	**EEPROM**	Yes
Short Description	Analog Offset Output 1		

Description

This is an offset that is applied to the D/A converter for analog output 1. The offset value is given in internal units (counts). Scaling is as follows:

- DAOFFSET1 = 2058 -10V
- DAOFFSET1 = 1250 0V
- DAOFFSET1 = 442 10V

4.19.8 DAOFFSET2

ASCII - Command	**DAOFFSET2**		
Syntax Transmit	DAOFFSET2 [Data]		
Syntax Receive	DAOFFSET2 <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	3521 (hex)
DIM	Counts	**PROFIBUS PNU**	1633 (dec) IND = 0000xxxx (bin)
Range	0 .. 2500	**DPR**	33 (dec)
Default	1290		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Velocity Controller	**EEPROM**	Yes
Short Description	Analog Offset Output 2		

Description

This is an offset that is applied to the D/A converter for analog output 2. The offset value is given in internal units (counts). Scaling is as follows:

- DAOFFSET2 = 2058 -10V
- DAOFFSET2 = 1250 0V
- DAOFFSET2 = 442 10V

4.19.9 DEC

ASCII - Command	**DEC**			
Syntax Transmit	DEC [Data]			
Syntax Receive	DEC <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Integer16		**CANBus Object Number**	3522 (hex)
DIM	Milliseconds		**PROFIBUS PNU**	1634 (dec) IND = 0000xxxx (bin)
Range	1 .. 32767, VLIM * 4480 (5.41)		**DPR**	34 (dec)
Default	10			
Opmode	0, 1, 8 (bei EXTPOS=1,4)		**Data Type Bus/DPR**	Integer16
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.9
Function Group	Velocity Controller		**EEPROM**	Yes
Short Description	Deceleration Rate			

Description

The DEC command defines the deceleration ramp for the velocity control loop (in msec) referred to the maximum velocity (the larger value of VLIM [▶ 338] and VLIMN [▶ 339]). The DEC deceleration/braking ramp is only used for setpoint step changes that result in a velocity decrease (braking). The ACC [▶ 325] parameter is used for acceleration.

For a setpoint step from VLIM [▶ 338]/VLIMN [▶ 339] to 0, the ramp generator generates a stepped ramp (with steps of 250 microseconds) that is completed within the set DEC time.

The DEC braking ramp applies to all setpoint changes, whether they are provided in analog or digital form. Separate braking ramps (DECSTOP [▶ 330]/DECDIS [▶ 330]) are used for setpoint changes that are generated internally in emergency stop situations (e.g. amplifier fault, or removal of the amplifier enable).

4.19.10 DECDIS

ASCII - Command	**DECDIS**		
Syntax Transmit	DECDIS [Data]		
Syntax Receive	DECDIS <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	3523 (hex)
DIM	Milliseconds	**PROFIBUS PNU**	1635 (dec) IND = 0000xxxx (bin)
Range	1 .. 32767, VLIM * 4480 (5.41)	**DPR**	35 (dec)
Default	10		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Velocity Controller	**EEPROM**	Yes
Short Description	Deceleration used on Disable Output Stage		

Description

When the output stage is disabled (removal of the hardware or software enable), the internal velocity setpoint is set to 0, using the preset DECDIS ramp. The output stage is only disabled when the actual velocity has fallen below the standstill threshold (VEL0 [▸ 338]).

The DECDIS ramp only has an effect for motors with a configured brake (MBRAKE [▸ 226]=1) or with the selection STOPMODE [▸ 85]=1.

With STOPMODE [▸ 85]=0 the output stage is immediately disabled, and the drive coasts down.

4.19.11 DECSTOP

ASCII - Command	**DECSTOP**		
Syntax Transmit	DECSTOP [Data]		
Syntax Receive	DECSTOP <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer16	**CANBus Object Number**	3525 (hex)
DIM	Milliseconds	**PROFIBUS PNU**	1637 (dec) IND = 0000xxxx (bin)
Range	1 .. 32767, VLIM * 4480 (5.41)	**DPR**	37 (dec)
Default	10		
Opmode	All	**Data Type Bus/DPR**	Integer16
Drive State	-	**Weightning**	
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Velocity Controller	**EEPROM**	Yes
Short Description	Quick Stop □ braking ramp for emergency situations		

Description

In emergency stop situations, the internal setpoint goes to 0 using the preset DECSTOP ramp. The output stage is only disabled when the actual velocity has fallen below the standstill threshold (VEL0 [▸ 338]).

An emergency stop situation exists in the following cases:

- amplifier fault (with ACTFAULT [▸ 34]=1)
- contouring/following error
- threshold monitoring (fieldbus devices)
- hardware/software limit switch activated
- emergency stop function through the digital input (INxMODE [▸ 116]=27)
- emergency stop function through the fieldbus (control word)

4.19.12 DIR

ASCII - Command	**DIR**		
Syntax Transmit	DIR [Data]		
Syntax Receive	DIR <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Integer8	**CANBus Object Number**	352A (hex)
DIM	-	**PROFIBUS PNU**	1642 (dec) IND = 0000xxxx (bin)
Range	0, 1	**DPR**	42 (dec)
Default	1		
Opmode	All	**Data Type Bus/DPR**	Integer8
Drive State	Disabled + Reset (Coldstart)	**Weightning**	
Start Firmware	1.20		
Configuration	Yes	**Revision**	1.3
Function Group	Velocity Controller	**EEPROM**	Yes
Short Description	Count Direction		

Description

The DIR variable defines the count direction for evaluation and entries of position information.
DIR = 0 negative count direction □ positive velocity and current entries cause the motor shaft to rotate in an anti-clockwise (CCW) direction.

DIR = 1 positive count direction □ positive velocity and current entries cause the motor shaft to rotate in a clockwise (CW) direction.

The definition of the count direction affects all controller modes (OPMODE [▸ 50]).

4.19.13 ESPEED

ASCII - Command	**ESPEED**		
Syntax Transmit	-		
Syntax Receive	ESPEED <Data>	**Available in**	
Type	Variable r	**MMI**	No
ASCII Format	Float	**CANBus Object Number**	3675 (hex)
DIM	rpm	**PROFIBUS PNU**	1973 (dec) IND = 0000xxxx (bin)
Range	0 .. 16000	**DPR**	373 (dec)
Default	-		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	4.02		
Configuration	No	**Revision**	1.4
Function Group	Velocity Controller	**EEPROM**	-

Short Description	Maximum velocity corresponding to the Feedback Type

Description

The command ESPEED gives the maximum velocity of the motor corresponding to the selected feedback type (FBTYPE [▶ 190]).

4.19.14 FILTMODE

ASCII - Command	**FILTMODE**		
Syntax Transmit	FILTMODE [Data]		
Syntax Receive	FILTMODE <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Unsigned8	**CANBus Object Number**	353C (hex)
DIM	-	**PROFIBUS PNU**	1660 (dec) IND = 0000xxxx (bin)
Range	0, 1, 2, 3	**DPR**	60 (dec)
Default	2		
Opmode	All	**Data Type Bus/DPR**	Unsigned8
Drive State	Disabled + Reset (Coldstart)	**Weightning**	
Start Firmware	1.71		
Configuration	Yes	**Revision**	1.7
Function Group	Velocity Controller	**EEPROM**	Yes
Short Description	Feedback Filter Mode		

Description

- FILTMODE=0 16 KHz Update without Luenberger Observer
- FILTMODE=1 4 KHz Update without Luenberger Observer
- FILTMODE=2 16 KHz Update with Luenberger Observer
- FILTMODE=3 4 KHz Update with Luenberger Observer

4.19.15 GV

ASCII - Command	**GV**		
Syntax Transmit	GV [Data]		
Syntax Receive	GV <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Float	**CANBus Object Number**	3548 (hex)
DIM	-	**PROFIBUS PNU**	1672 (dec) IND = 0000xxxx (bin)
Range	0.0 .. 62.5*GVTN	**DPR**	72 (dec)
Default	1		
Opmode	0, 1	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Velocity Controller	**EEPROM**	Yes
Short Description	Velocity Control Loop: Proportional Gain		

Description

That means, that a difference between N_cmd and N_actual of 3000 rpm with GV = 1 results in the peak current of the drive.

This variable determines the proportional gain (also known as AC-gain). Adjust this variable by increasing the value to the level where the motor starts to oscillate. Then, back it off until the oscillations have clearly stopped. Typical values are between 10 and 20. If the GV value is too low, the drive is too soft and has poor damping. If the GV value is too high, the drive whistles or runs roughly.

The gain is defined, that a velocity deviation of 3000rpm with GV = 1 results in the peak-current of the drive.

4.19.16 GVD

ASCII - Command	**GVD**			
Syntax Transmit	GVD [Data]			
Syntax Receive	GVD <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Float		**CANBus Object Number**	368B (hex)
DIM	-		**PROFIBUS PNU**	1995 (dec) IND = 0000xxxx (bin)
Range	0 .. 2		**DPR**	395 (dec)
Default	0			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	4.30			
Configuration	No		**Revision**	1.5
Function Group	Velocity Controller		**EEPROM**	Yes
Short Description	Derivate Part in the Velocity Controller			

Description

To compensate the margin of the system stability in the high frequency, the conventional PI velocity controller is extended by PID controller.

GVD is a tuning variable of the PID velocity controller, which sets the gain of the derivative feedback of the actual velocity. The other related parameter is the filter time constant GVDT [▸ 333].

4.19.17 GVDT

ASCII - Command	**GVDT**			
Syntax Transmit	GVDT [Data]			
Syntax Receive	GVDT <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Float		**CANBus Object Number**	368C (hex)
DIM	ms		**PROFIBUS PNU**	1996 (dec) IND = 0000xxxx (bin)
Range	0 .. 1		**DPR**	396 (dec)
Default	0.3			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	4.30			
Configuration	No		**Revision**	1.5
Function Group	Velocity Controller		**EEPROM**	Yes
Short Description	Filter Time Constant of the D-Part of the Velocity Controller			

Description

In the PID velocity controller, the derivative feedback of the velocity is directly obtained by derivation of the actual velocity. To reduce the derivative noise, the derivative signal will be filtered by a first low pass filter.

GVDT sets the time constant of the filter.

4.19.18 GVFBT

ASCII - Command	GVFBT		
Syntax Transmit	GVFBT [Data]		
Syntax Receive	GVFBT <Data>	Available in	
Type	Variable rw	MMI	Yes
ASCII Format	Float	CANBus Object Number	3549 (hex)
DIM	Milliseconds	PROFIBUS PNU	1673 (dec) IND = 0000xxxx (bin)
Range	0.0 .. 30.0	DPR	73 (dec)
Default	0.4		
Opmode	0, 1	Data Type Bus/DPR	Integer32
Drive State	-	Weightning	1000
Start Firmware	1.20		
Configuration	No	Revision	1.8
Function Group	Velocity Controller	EEPROM	Yes
Short Description	Velocity Control Loop: Time Constant First Order Tacho Filter		

Description

If necessary, the time constant for the PT-1 filter in the actual velocity feedback is altered (default=0.6 ms). This may improve the step response and smoothness of running, particularly for very small, highly dynamic motors. If the GVFBT value is set too low, the motor runs roughly. If the GVFBT value is set too high, the velocity control becomes soft and unstable.

4.19.19 GVFILT

ASCII - Command	GVFILT		
Syntax Transmit	GVFILT [Data]		
Syntax Receive	GVFILT <Data>	Available in	
Type	Variable rw	MMI	Yes
ASCII Format	Integer8	CANBus Object Number	354A (hex)
DIM	%	PROFIBUS PNU	1674 (dec) IND = 0000xxxx (bin)
Range	0 .. 100	DPR	74 (dec)
Default	85		
Opmode	0, 1, 4, 5, 8	Data Type Bus/DPR	Integer8
Drive State	-	Weightning	
Start Firmware	1.20		
Configuration	No	Revision	1.3
Function Group	Velocity Controller	EEPROM	Yes
Short Description	Velocity Control Loop: Part of the Output that is filtered [%] by GVT2		

Description

velocity control loop: Part of the Output that is filtered [%] by GVT2 [▶ 335] (GVFILT = 85 means, 85% are filtered and 15% are not filtered)

4.19.20 GVFR

ASCII - Command	**GVFR**		
Syntax Transmit	GVFR [Data]		
Syntax Receive	GVFR <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Float	**CANBus Object Number**	354B (hex)
DIM	-	**PROFIBUS PNU**	1675 (dec) IND = 0000xxxx (bin)
Range	0.0 .. 1.0	**DPR**	75 (dec)
Default	1.0		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	1.77		
Configuration	No	**Revision**	1.3
Function Group	Velocity Controller	**EEPROM**	Yes
Short Description	PI-PLUS Actual Velocity Feedforward		

Description

GVFR is a tuning variable of the velocity control loop which sets the feed-forward to feedback gain ratio for the Pseudo Derivative Feedback with Feed-Forward. (PDFF or PI+) . With GVFR 1 the behavior of the velocity control loop is like a standard PI controller. GVFR 0.65 is a value which suppresses step response overshoot.

4.19.21 GVT2

ASCII - Command	**GVT2**		
Syntax Transmit	GVT2 [Data]		
Syntax Receive	GVT2 <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Float	**CANBus Object Number**	354C (hex)
DIM	Milliseconds	**PROFIBUS PNU**	1676 (dec) IND = 0000xxxx (bin)
Range	0.0 .. 30.0	**DPR**	76 (dec)
Default	1.0		
Opmode	0, 1	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Velocity Controller	**EEPROM**	Yes
Short Description	Velocity Control Loop: Second Time Constant		

Description

This variable affects the proportional gain (P-gain) at medium frequencies. It is often possible to improve the damping of the velocity of the velocity control loop by increasing this value to about GVTN [▸ 336] / 3. If required, set this value after the basic setting of GV [▸ 332] and GVTN [▸ 336]. If the GVT2 value is too low, the drive is very stiff. If the GVT2 value is too high, the drive is not stiff enough.

The part, which is filtered can be set by GVFILT [▸ 334].

4.19.22 GVTN

ASCII - Command	**GVTN**			
Syntax Transmit	GVTN [Data]			
Syntax Receive	GVTN <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Float		**CANBus Object Number**	354D (hex)
DIM	Milliseconds		**PROFIBUS PNU**	1677 (dec) IND = 0000xxxx (bin)
Range	0.0 , GV/62.5 .. 1000.0		**DPR**	77 (dec)
Default	10			
Opmode	0, 1		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	1.20			
Configuration	No		**Revision**	1.8
Function Group	Velocity Controller		**EEPROM**	Yes
Short Description	Velocity Control Loop: I-Integration Time			

Description

This variable determines the integral-action time/integral time constant. Smaller motors permit shorter integration times. Larger motors or high moments of inertia in the load usually require integration times of 20ms or more. With GVTN=0ms, the I-component is switched off. If the GVTN value is too low, the drive runs roughly or strongly overshoots with high inertia loads. If the GVTN value is too high, the drive is too soft.

4.19.23 ISTFR

ASCII - Command	**ISTFR**			
Syntax Transmit	ISTFR [Data]			
Syntax Receive	ISTFR <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Float		**CANBus Object Number**	36A4 (hex)
DIM	A		**PROFIBUS PNU**	2020 (dec) IND = 0000xxxx (bin)
Range	0 .. IPEAK		**DPR**	420 (dec)
Default	0			
Opmode	0,1,4,5,6,7,8		**Data Type Bus/DPR**	Integer32
Drive State	Disabled		**Weightning**	1000
Start Firmware	4.96			
Configuration	No		**Revision**	1.7
Function Group	Velocity Controller		**EEPROM**	Yes
Short Description	Velocity dependant Friction Compensation			

Description

The two objects ISTFR and VSTFR [▸ 341] define the friction compensation curve. ISTFR enables the function if it is not "0". The friction compensation changes the additional current from -ISTFR to ISTFR if the velocity changes from -VSTFR [▸ 341] to VSTFR [▸ 341].

It is a configuration parameter if it is changed from "0" to another value, other changes can be done online.

V [▶ 31]=0 -> IFRICT = 0
V [▶ 31]= 50% of VSTFR [▶ 341] -> IFRICT = 50% of ISTFR
V [▶ 31]>=VSTFR [▶ 341] -> IFRICT = ISTFR
V [▶ 31]= -50% of VSTFR [▶ 341] -> IFRICT = -50% of ISTFR
V [▶ 31]<=-VSTFR [▶ 341] -> IFRICT = -ISTFR

4.19.24 SDUMP

ASCII - Command	**SDUMP**			
Syntax Transmit	SDUMP			
Syntax Receive	SDUMP <Data>		**Available in**	
Type	Multi-line Return Command		**MMI**	Yes
ASCII Format	String		**CANBus Object Number**	35EE (hex)
DIM	-		**PROFIBUS PNU**	1838 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	238 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Velocity Controller		**EEPROM**	-
Short Description	List Speed/Velocity Limits			

Description

Outputs a list of the speed/velocity limits.

4.19.25 VDUMP

ASCII - Command	**VDUMP**			
Syntax Transmit	VDUMP			
Syntax Receive	VDUMP <Data>		**Available in**	
Type	Multi-line Return Command		**MMI**	Yes
ASCII Format	String		**CANBus Object Number**	361F (hex)
DIM	-		**PROFIBUS PNU**	1887 (dec) IND = 0000xxxx (bin)
Range	-		**DPR**	287 (dec)
Default	-			
Opmode	All		**Data Type Bus/DPR**	-
Drive State	-		**Weightning**	
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Velocity Controller		**EEPROM**	-
Short Description	List all Velocity Controller Variables			

Description

A listing of all the parameters for the velocity control loop.

4.19.26 VEL0

ASCII - Command	**VEL0**		
Syntax Transmit	VEL0 [Data]		
Syntax Receive	VEL0 <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Float	**CANBus Object Number**	3620 (hex)
DIM	-	**PROFIBUS PNU**	1888 (dec) IND = 0000xxxx (bin)
Range	0.0 .. long int	**DPR**	288 (dec)
Default	5		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State		**Weightning**	1000
Start Firmware	1.20		
Configuration	No	**Revision**	1.8
Function Group	Velocity Controller	**EEPROM**	Yes
Short Description	Standstill Threshold		

Description

The VEL0 (Velocity "0") parameter defines the velocity threshold (in RPM) for the standstill signal.

The standstill signal is required for the following functions:

1. Standstill signal in the status register DRVSTAT [▸ 171].
2. If the brake is configured (MBRAKE [▸ 226]=1), then, if the output stage is disabled, first of all the velocity is reduced to 0, and the brake is only applied after the velocity has fallen below the standstill threshold.
3. If the ACTFAULT [▸ 34] option is activated (active braking in the event of a fault), or the STOPMODE [▸ 85] option (active braking if the output stage is disabled), then the standstill threshold defines the velocity below which the output stage will actually be disabled.

4.19.27 VLIM

ASCII - Command	**VLIM**		
Syntax Transmit	VLIM [Data]		
Syntax Receive	VLIM <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Float	**CANBus Object Number**	3622 (hex)
DIM	rpm	**PROFIBUS PNU**	1890 (dec) IND = 0000xxxx (bin)
Range	0.0 .. MSPEED	**DPR**	290 (dec)
Default	3000		
Opmode	0, 1	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Velocity Controller	**EEPROM**	Yes
Short Description	Max. Velocity		

Description

The VLIM parameter defines the maximum velocity for the velocity control loop in RPM.

VLIM is also used for limiting the following parameters:

1. MVANGLB [▸ 237] <= 0.9*VLIM
2. MSPEED [▸ 235] >= VLIM
3. PVMAX [▸ 290] <= (VLIM * PGEARI [▸ 283] * 2^PRBASE [▸ 286]) / (60 * PGEARO [▸ 284])

When used together with the VLIMN [▸ 339] parameter, it is possible to implement a directionally dependent rotational velocity limit. The VLIM command determines the maximum velocity for both positive and negative directions. By making a subsequent entry for VLIMN [▸ 339], the limit for the negative direction can be set separately.

4.19.28 VLIMN

ASCII - Command	**VLIMN**			
Syntax Transmit	VLIMN [Data]			
Syntax Receive	VLIMN <Data>		**Available in**	
Type	Variable rw		**MMI**	No
ASCII Format	Float		**CANBus Object Number**	3623 (hex)
DIM	rpm		**PROFIBUS PNU**	1891 (dec) IND = 0000xxxx (bin)
Range	0.0 .. MSPEED		**DPR**	291 (dec)
Default	3000			
Opmode	0, 1		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	1.20			
Configuration	No		**Revision**	1.3
Function Group	Velocity Controller		**EEPROM**	Yes
Short Description	Max. Negative Velocity			

Description

The VLIMN parameter defines the maximum velocity for the negative direction (velocity control loop) in RPM.VLIMN is also used for limiting PVMAXN [▸ 291]:

PVMAXN <= (VLIMN * PGEARI [▸ 283] * 2^PRBASE [▸ 286]) / (60 * PGEARO [▸ 284])

When used together with the VLIM [▸ 338] parameter, it is possible to implement a directionally dependent rotational velocity limit. The VLIM [▸ 338] command determines the maximum velocity for both positive and negative directions. By making a subsequent entry for VLIMN, the limit for the negative direction can be set separately.

4.19.29 VMAX

ASCII - Command	**VMAX**			
Syntax Transmit	VMAX			
Syntax Receive	VMAX <Data>		**Available in**	
Type	Variable ro		**MMI**	Yes
ASCII Format	Float		**CANBus Object Number**	3624 (hex)
DIM	RPM		**PROFIBUS PNU**	1892 (dec) IND = 0000xxxx (bin)
Range	00 .. 12000.0		**DPR**	292 (dec)
Default	-			
Opmode	0, 1		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	1.20			
Configuration	No		**Revision**	1.8
Function Group	Velocity Controller		**EEPROM**	No
Short Description	Maximum System Speed			

Description

VMAX returns the maximum speed that can be reached by the amplifier/motor combination as set by (MSPEED [▸ 235]).

4.19.30 VMIX

ASCII - Command	**VMIX**			
Syntax Transmit	VMIX [Data]			
Syntax Receive	VMIX <Data>		**Available in**	
Type	Variable rw		**MMI**	Yes
ASCII Format	Float		**CANBus Object Number**	3625 (hex)
DIM	-		**PROFIBUS PNU**	1893 (dec) IND = 0000xxxx (bin)
Range	0.0 .. 1.0		**DPR**	293 (dec)
Default	1.0			
Opmode	All		**Data Type Bus/DPR**	Integer32
Drive State	-		**Weightning**	1000
Start Firmware	1.78			
Configuration	No		**Revision**	1.9
Function Group	Velocity Controller		**EEPROM**	Yes
Short Description	Velocity Mix: Feedback / external Encoder			

Description

If an external encoder is used for the position control EXTPOS [▸ 253]=1) then the position control is made with the position information from the external encoder, but the commutation and speed control is made with the position information from the feedback device of the motor (feedback device selected by FBTYPE [▸ 190]). The parameter VMIX defines in what ratio the speed information from the feedback device on the motor to the speed information of the external encoder is used for the actual speed calculation.e.g.

- VMIX=1.0 Velocity exclusively from the feedback device selected by FBTYPE [▸ 190] (100 %)
- VMIX=0.5 50 % feedback device selected by FBTYPE [▸ 190] / 50 % external encoder

VMIX is available if FILTMODE [▸ 332] = 0 or 1.

VMIX is available only in position mode OPMODE [▸ 50] = 8

4.19.31 VOSPD

ASCII - Command	**VOSPD**		
Syntax Transmit	VOSPD [Data]		
Syntax Receive	VOSPD <Data>	**Available in**	
Type	Variable rw	**MMI**	Yes
ASCII Format	Float	**CANBus Object Number**	3627 (hex)
DIM	rpm	**PROFIBUS PNU**	1895 (dec) IND = 0000xxxx (bin)
Range	0.0 .. 1.2*MSPEED	**DPR**	295 (dec)
Default	3600		
Opmode	All	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	1.20		
Configuration	No	**Revision**	1.3
Function Group	Velocity Controller	**EEPROM**	Yes
Short Description	Overspeed		

Description

The VOSPD parameter can be used to set the switch-off threshold for the fault message F08 (overspeed). As soon as the actual velocity exceeds the preset threshold, the fault message F08 is generated, and the output stage is disabled.

4.19.32 VSTFR

ASCII - Command	**VSTFR**		
Syntax Transmit	VSTFR [Data]		
Syntax Receive	VSTFR <Data>	**Available in**	
Type	Variable rw	**MMI**	No
ASCII Format	Float	**CANBus Object Number**	36A5 (hex)
DIM	VUNIT	**PROFIBUS PNU**	2021 (dec) IND = 0000xxxx (bin)
Range	0 .. 230 UPM	**DPR**	421 (dec)
Default	0		
Opmode	0,1,4,5,6,7,8	**Data Type Bus/DPR**	Integer32
Drive State	-	**Weightning**	1000
Start Firmware	4.96		
Configuration	No	**Revision**	1.7
Function Group	Velocity Controller	**EEPROM**	Yes
Short Description	Velocity for max. Friction Compensation		

Description

VSTFR gives the velocity, where the ISTFR [▸ 336] is added to the velocity controller. The friction compensation is enabled by ISTFR [▸ 336].

V=0 -> IFRICT = 0
V= 50% of VSTFR -> IFRICT = 50% of ISTFR [▸ 336]
V>=VSTFR -> IFRICT = ISTFR [▸ 336]
V= -50% of VSTFR -> IFRICT = -50% of ISTFR [▸ 336]
V<=-VSTFR -> IFRICT = -ISTFR [▸ 336]

4.19.33 VUNIT

ASCII - Command	VUNIT		
Syntax Transmit	VUNIT [Data]		
Syntax Receive	VUNIT <Data>	Available in	
Type	Variable rw	MMI	No
ASCII Format	Integer32	CANBus Object Number	365F (hex)
DIM	-	PROFIBUS PNU	1951 (dec) IND = 0000xxxx (bin)
Range	0 .. 8	DPR	351 (dec)
Default	0		
Opmode	All	Data Type Bus/DPR	Integer32
Drive State		Weightning	
Start Firmware	4.00		
Configuration	No	Revision	1.8
Function Group	Velocity Controller	EEPROM	Yes
Short Description	Systemwide Definition of Velocity / Speed		

Description

VUNIT gives the systemwide definition of velocity / speed resolution. This parameter effects all parameters that are related to velocity of the velocity controller and speed of the position controller.

VUNIT = 0 gives velocity in RPM and speed in µm/sec. This setting is equal to the firmware < 4.00.

- VUNIT = 1 Unit = RPM
- VUNIT = 2 Unit = Rad/Sec
- VUNIT = 3 Unit = Degree/Sec
- VUNIT = 4 Unit = Counts/250 µsec
- VUNIT = 5 Unit = PUNIT [▶ 289] / Sec
- VUNIT = 6 Unit = PUNIT [▶ 289] / Min
- VUNIT = 7 Unit = 1000 * PUNIT [▶ 289] / Sec
- VUNIT = 8 Unit = 1000 * PUNIT [▶ 289] / Min

Remark:

1. All parameters that are related to velovity have a fixed format of 32 Bit with 3 fractional digits. This causes a problem with some of the VUNIT settings (especially VUNIT=6), related to the resolution of the position controller (PGEARI [▶ 283]), that not the full range of speed can be used. Under this condition, a different setting of VUNIT is necessary.
2. All parameters that are related to speed have a fixed format of 32 Bit with no fractional digits. This causes a problem with some of the VUNIT settings (especially VUNIT=3) to give fractional digits. Under this condition, a different setting of VUNIT is necessary.

Definition of the Calculation factors

- VUNIT=1 1 UPM = 1048576*32/(4000*60) =~ 139.8 Counts
- VUNIT=2 1 Rad/sec = 1048576*32/(4000*2*PI) =~ 1335 Counts
- VUNIT=3 1 Grad/sec = 1048576*32/(4000*360) =~ 23.3 Counts
- VUNIT=4 1 Counts/250µs = 32 Counts
- VUNIT=5 1 PUNIT [▶ 289] / sec = PGEARO [▶ 284] / (125 * PGEARI [▶ 283])
- VUNIT=6 1 PUNIT [▶ 289] / min = (PGEARO [▶ 284] * 60) / (125 * PGEARI [▶ 283])
- VUNIT=7 1000 PUNIT [▶ 289] / sec = PGEARO [▶ 284] / (125 * PGEARI [▶ 283] * 1000)
- VUNIT=8 1000 PUNIT [▶ 289] / min = (PGEARO [▶ 284] * 60) / (125 * PGEARI [▶ 283] * 1000)

4.20 START

Description

ASCII Object Description, Edition REV 1.9

History

Edition	Description	Created
REV 1.2	First English Edition	01.08.00
REV 1.3	Expansion up to Firmware 3.50	27.11.00
REV 1.4	Expansion up to Firmware 4.40	10.05.01
REV 1.5	Expansion up to Firmware 4.80	23.11.01
REV 1.6	Expansion up to Firmware 4.95	15.02.02
REV 1.7	Expansion up to Firmware 4.99, Change to HTML	24.05.02
REV 1.8	New Design and some Changes	14.08.02
REV 1.9	Expansion to Firmware 5.41	04.11.02

More Information:
www.beckhoff.com/en-en/products/motion/

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com