BECKHOFF New Automation Technology

Manual | EN

TS3900

TwinCAT 2 | Solar Position Algorithm

2021-11-08 | Version: 1.1

Table of contents

1 Foreword

1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

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

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without prior announcement.
No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

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

EtherCAT®

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

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.
The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.
Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.

1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations 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 engineering who are familiar with the applicable national standards.

Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

⚠ DANGER

Serious risk of injury!

Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.

⚠ WARNING

Risk of injury!

Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.

⚠ CAUTION

Personal injuries!

Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.

NOTE

Damage to the environment or devices

Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.

Tip or pointer

This symbol indicates information that contributes to better understanding.

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.

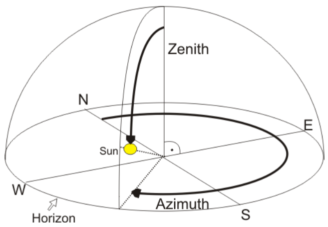
2 Overview

The TwinCAT PLC Solar Position Algorithm library (SPA) offers an option for calculating the sun position exactly at almost any time.
The times for sunrise, solar apex and sunset can also be determined.
In addition to the sun angles an angle of incidence can be issued, if the point of reference has a certain inclination. The sun angles themselves refer to the horizontal at the point of reference.

The algorithm is based on a technical report by the U.S. National Renewable Energy Laboratory (NREL). The theoretical inaccuracy of the sun angles between the year -2000 and 6000 is specified as +/-0.0003°. Based on this the function block of the TwinCAT Solar Position Algorithm library assumes an inaccuracy of +/-0.001° for the sun angles.

Sun angles

The position of the sun at a fixed observation point is normally determined by specifying two angles.
In order to calculated the sun angles using the TwinCAT Solar Position Algorithm library, the date, time, longitude, latitude and further parameters have to be specified, depending on the required accuracy.
The graphic illustrates the meaning of the main terms in this context:



The sun position represented by two angles.

Zenith	The zenith angle of the sun is defined as the angle between the vertical above the observer and the connecting line between the observer and the sun. In some cases the altitude is as used to indicate the sun elevation angle. The following applies: 90° - zenith angle = altitude
Azimuth	The azimuth coincides with the horizon. North is 0°, with the value increasing in clockwise direction (east = 90°, south=180°, west=270°).

Longitude and latitude

The latitude is specified as the distance of a place on the surface of the earth from the equator to the north or to the south in degrees. The latitude can assume a value from 0° (at the equator) to ±90° (at the poles). A positive sign thereby indicates a northern direction and a negative sign a southern direction. The longitude is an angle that can assume values up to ±180° starting from the prime meridian 0° (an artificially determined North-South line). A positive sign indicates a longitude in an eastern direction and a negative sign in a western direction. Examples:

Place	Longitude	Latitude	
Sydney, Australia	151.2°	-33.9°	
New York, USA	-74.0°	40.7°	
London, England	-0.1°	51.5°	
Moscow, Russia	37.6°	55.7°	
Peking, China	116.3°	39.9°	
Dubai, United Arab Emirates	55.3°	25.4°	
Rio de Janeiro, Brazil	-43.2°	-22.9°	
Hawaii, USA	-155.8°	20.2°	
Verl, Germany	8.5°	51.9°	

Time scale

Specification of the correct time is particularly important. Various time scales are in use. The Solar Position Algorithm is based on Universal Time (UT1).

Universal Time (UT1)

Between 1928 and 1968 was the UT was the accepted world time. It is also referred to as universal solar time. It is determined through astronomic observation of the angle of rotation of the earth and corresponds to the mean local time of the observatory at Greenwich (prime meridian). This parameter is derived from the earth's rotation and takes into account fluctuations and long-term slowdown and is therefore not strictly a uniform measure of time. On the other hand, it is always synchronised with the actual change-over between day and night.

International Atomic Time (TAI)

The International Atomic Time is specified by more than 50 time institutes worldwide, based on their atomic clocks. An atomic time is based on an atomic standard time that can be assumed to be exactly uniform.

Coordinated Universal Time (UTC)

The coordinated world time UTC has been used as the standard world time since 1968. This is the time referred to by GMT in everyday usage. Greenwich Mean Time (GMT) was the original world time before 1928.
UTC continues to use the observatory at Greenwich (prime meridian) as point of reference. The earth's time zones are derived from the coordinated world time (UTC+1 = Central European Time). In contrast to UT1, its second cycle matches the exactly uniform second cycle of the International Atomic Time (TAI). Leap seconds are used to compensate the difference between UTC and UT1. The difference between the UT1 reference time is always less than one second.
The coordinated world time UTC is therefore a compromise between UT1 and TAI.
The following formula is used to convert a time from UTC to UT1: UT1 = UTC + DUT1

Terrestrial Time (TT)

Also referred to as Terrestrial Dynamical Time (TDT). This time is used as the basis for calculating astronomic events and is based on the exactly uniform seconds of the International Atomic Time (TAI). The following applies: TT = TAI + 32.184

Leap Seconds

To synchronise the coordinated world time UTC with UT1, a leap second is added when required. This additional second is specified by the International Earth Rotation and Reference Systems Service (IERS) at irregular, non-predictable intervals. It ensures that the difference between the two time scales is always less than one second. (In the past such additional leap seconds have always been added on 31 December or 30 June after 23:59:59 UTC.)
DUT1 denotes the remaining difference. The following applies: DUT1 = UT1 - UTC
This value is derived from observations that are continuously reported.

Delta T

Delta T is the difference between Terrestrial Time and Universal Time. The following applies: Delta_t = TT - UT1
This parameter can be specified as *fDelta_t* at the input for function block FB_SPA [▸ 13]. It is derived from observations that are continuously reported. A standard value is 66 seconds.

Similar products

- Time switching functions with lower accuracy such as FB_CalcSunPosition and FB_CalcSunriseSunset from TS8010 | TwinCAT 2 PLC Building Automation Basic

Documentation last updated: 08.11.2011

3 System requirements

- Programming environment:
    - XP, XPe;
    - TwinCAT installation level: TwinCAT PLC or higher;
    - TwinCAT system version 2.10.0 build 1320 or higher
    - **TcSPA.Lib** This PLC library must be integrated in the PLC project. All other libraries are added automatically. (Standard.Lib; TcMath.Lib; TcBaseMath.Lib; TcSystem.Lib; TcBase.Lib are integrated automatically)
- Target platform:
    - PC or CX (x86): XP, XPe, CE;
    - CX (ARM): CE (image v2.18 or higher);
    - TwinCAT PLC runtime system version 2.10.0 build 1320 or higher;

In systems without floating point unit the performance is limited due to the internal complex calculations. In the event of anomalies the cycle time should be checked.

4 Copyright

The algorithm is based on the technical report "Solar Position Algorithm for Solar Radiation Application" by I. Reda & A. Andreas, National Renewable Energy Laboratory (NREL), USA (revision 14-JAN-2009).

NOTICE
Copyright (C) 2007 Alliance for Sustainable Energy, LLC, All Rights Reserved

This computer software was developed by the Alliance for Sustainable Energy, LLC, hereinafter the Contractor, under Contract DE-AC36-08GO28308 (Contract) with the Department of Energy (DOE). The United States Government has been granted for itself and others acting on its behalf a paid-up, non-exclusive, irrevocable, worldwide license in the Software to reproduce, prepare derivative works, and perform publicly and display publicly. Beginning five (5) years after the date permission to assert copyright is obtained from the DOE, and subject to any subsequent five (5) year renewals, the United States Government is granted for itself and others acting on its behalf a paid-up, non-exclusive, irrevocable, worldwide license in the Software to reproduce, prepare derivative works, distribute copies to the public, perform publicly and display publicly, and to permit others to do so. If the Contractor ceases to make this computer software available, it may be obtained from DOE's Office of Scientific and Technical Information's Energy Science and Technology Software Center (ESTSC) at P.O.Box 62, 1 Science Gov Way, Oak Ridge, TN 37831-1020. THIS SOFTWARE IS PROVIDED BY THE CONTRACTOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE CONTRACTOR OR THE U.S. GOVERNMENT BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER, INCLUDING BUT NOT LIMITED TO CLAIMS ASSOCIATED WITH THE LOSS OF DATA OR PROFITS, WHICH MAY RESULT FROM AN ACTION IN CONTRACT, NEGLIGENCE OR OTHER TORTIOUS CLAIM THAT ARISES OUT OF OR IN CONNECTION WITH THE ACCESS, USE OR PERFORMANCE OF THIS SOFTWARE.

5 Function blocks

5.1 FB_SPA

```
                         FB_SPA

stTime : ST_SPA_TIMESTRUCT            fZenith : LREAL
fTimezone : LREAL                    fAzimuth : LREAL
fDelta_t : LREAL                  fAzimuth180 : LREAL
fLongitude : LREAL                 fIncidence : LREAL
fLatitude : LREAL                 fSuntransit : LREAL
fElevation : LREAL                   fSunrise : LREAL
fPressure : LREAL                     fSunset : LREAL
fTemperature : LREAL                   bError : BOOL
fSlope : LREAL                     iErrorCode : UINT
fAzm_rotation : LREAL
fAtmos_refract : LREAL
eFunction : E_SPA_FunctionCode
```

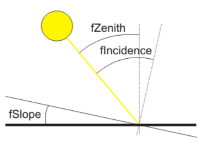
At the input all available values for the location definition and type of calculation are specified.

The calculation is performed during a function block cycle. The results are immediately available at the output.
Due to the complex internal calculation steps processing takes system performance.

VAR_INPUT

```
VAR_INPUT
    stTime      :ST_SPA_TIMESTRUCT;     (* local date and time (year, month, day, hour, minute, seco
nd) *)
    fTimezone           :LREAL;         (* Observer time zone (negative west of Greenwich)
*)
                        (* valid range: -18 TO 18 hours, error code: 8      *)
    fDelta_t        :LREAL:=66;         (* Difference between earth rotation time and terrestrial ti
me  *)
                        (* It is derived from observation only and is reported in this  *)
                        (* bulletin: http://maia.usno.navy.mil/ser7/ser7.dat,           *)
                        (* where delta_t = 32.184 + (TAI-UTC) + DUT1           *)
                        (* valid range: -8000 to 8000 seconds, error code: 7          *)
    fLongitude      :LREAL;         (* Observer longitude (negative west of Greenwich)          *)
                        (* valid range: -180 to 180 degrees, error code: 9            *)
    fLatitude           :LREAL;         (* Observer latitude (negative south of equator)        *)
                        (* valid range: -90 to 90 degrees, error code: 10             *)
    fElevation      :LREAL;         (* Observer elevation [meters]               *)
                        (* valid range: -6500000 or higher meters, error code: 11      *)
    fPressure           :LREAL:=1000;       (* Annual average local pressure [millibars]
*)
                        (* valid range: 0 to 5000 millibars, error code: 12           *)
    fTemperature    :LREAL;         (* Annual average local temperature [degrees Celsius]       *)
                        (* valid range: -273 to 6000 degrees Celsius, error code; 13    *)
    fSlope      :LREAL;         (* Surface slope (measured from the horizontal plane)       *)
                        (* valid range: -360 to 360 degrees, error code: 14           *)
    fAzm_rotation   :LREAL;         (* Surface azimuth rotation (measured from south to projection o
f       *)
                        (* surface normal on horizontal plane, negative west)        *)
                        (* valid range: -360 to 360 degrees, error code: 15           *)
    fAtmos_refract  :LREAL:=0.5667;     (* Atmospheric refraction at sunrise and sunset (0.5667 deg 
is typ.)*)
                        (* valid range: -5 to 5 degrees, error code: 16     *)
    eFunction           :E_SPA_FunctionCode:=eSPA_ZA;
(* Switch to choose functions for desired output       *)
END_VAR
```

stTime	The date and the local time are specified via stTime. This structure is of type ST_SPA_TIMESTRUCT [▶ 18].
fTimezone	The required date with the corresponding time can be specified in local time via the above variable. The respective time zone is added via fTimezone. The time zone is always based on Greenwich (London). (The prime meridian, i.e. 0° geographic longitude, also passes through Greenwich). The following applies in relation to the coordinated world time: UTC+1 = Central European Time; UTC+2 = Central European Summer Time.
fDelta_t	The input variable *fDelta_t* is used for balancing the time scales used. A standard value is 66. A more detailed description of the different time scales can be found on the overview page [▶ 8].
fLongitude	fLongitude indicates the longitude in degrees [°]. It is positive to the east of Greenwich.
fLatitude	fLatitude indicates the latitude in degrees [°]. It is positive to the north of the equator and negative to the south.
fElevation	The altitude of the location also has a small effect on the calculation of the sun angles. *fElevation* indicates the height in metres above mean sea level.
fPressure	The atmospheric pressure at the location is specified in millibar [mbar] via the input variable *fPressure*. The annual average is specified.
fTemperature	The temperature at the location is specified via the input variable *fTemperature* in °C. The annual average is specified.
fSlope	Via *fSlope* a surface inclination can be specified in degrees [°]. It is used for calculating the special angle of incidence *fIncidence*. If *fSlope* is zero, the angle of incidence is the same as the zenith angle.
fAzm_rotation	fAzm_rotation can be used to adjust the alignment (in degrees [°]) of the observer or the surface inclined by *fSlope*. For north alignment the value is 0°. From there the alignment angle increases clockwise (positive values, as does the azimuth of the sun angle). It is also used for calculating the special angle of incidence *fIncidence*. Is *fSlope* is zero, the angle of incidence is the same as the zenith angle, irrespective of *fAzm_rotation*. If *fAzm_rotation* is the same as the sun angle *fAzimuth*, the following applies: *fIncidence* = *fZenith* + *fSlope*. This is illustrated the following 2D diagram.



fAtmos_refract	Refraction in the atmosphere can have a significant effect on the zenith angle of the sun, particularly for shallow sun angles. The input variable *fAtmos_refract* is used as a correction factor for the atmospheric distraction at sunrise and sunset. A standard value is 0.5667.

eFunction

Via this enumeration value (E_SPA_FunctionCode [▶ 18]) the type of calculations can be selected. For example, the calculation can be limited to the sun angles, if information on sunrise etc. is not required.

VAR_OUTPUT

```
VAR_OUTPUT
    fZenith     :LREAL;     (* topocentric zenith angle [degrees]                *)
    fAzimuth        :LREAL;     (* topocentric azimuth angle (eastward from north) [ 0 to 360 degree
s]  *)
    fAzimuth180     :LREAL;     (* topocentric azimuth angle (westward from south) [-180 to 180 degr
ees]    *)
    fIncidence      :LREAL;     (* surface incidence angle [degrees]                    *)

    fSuntransit     :LREAL;     (* local sun transit time (or solar noon) [fractional hour]
*)
    fSunrise        :LREAL;     (* local sunrise time (+/- 30 seconds) [fractional hour]        *)
    fSunset     :LREAL;     (* local sunset TIME (+/- 30 seconds) [fractional hour]     *)

    bError      :BOOL;      (* error flag   *)
    iErrorCode      :UINT;      (* error code   *)
END_VAR
```

fZenith

The zenith angle of the sun is defined as the angle between the vertical above the observer (zenith) and the connecting line between the observer and the sun. If the sun is absolutely vertical above the observer, the zenith angle is 0°.
Sometimes the sun elevation angle (SunElevation or Altitude) is also used. The following applies: 90° - zenith angle = altitude

fAzimuth

The azimuth coincides with the horizon. North is 0°, with the value increasing in clockwise direction. (east = 90°, south=180°, west=270°). A graphical representation of the sun angles can be found on the overview page [▶ 8].

fAzimuth180

This value corresponds in meaning to the azimuth. However, the azimuth180 is assigned the value 0° in the south. From there the value increases positively in clockwise direction and negatively in counter-clockwise direction. (azimuth-180° = azimuth180)

fIncidence

fIncidence indicates the angle of solar incidence in relation to the surface specified at the input. If the surface is horizontal *fIncidence* matches the value of *fZenith*.

fSuntransit

fSuntransit indicates the time of the solar apex. It is specified in hours and used the time zone created at the input.

fSunrise

fSunrise indicates the sunrise time. It is specified in hours and used the time zone created at the input.

fSunset

fSunset indicates the sunset time. It is specified in hours and used the time zone created at the input.

bError

bError is TRUE if an error has occurred. In this case *iErrorCode* indicates the respective error code.

iErrorCode

iErrorCode indicates the error value for the calculation. If an error has occurred this value is not equal zero. All possible error values are summarized in the enumeration E_SPA_ErrorCode [▶ 19].

The following type conversion can be used for converting the variable *fSunrise* (same procedure for *fSuntransit* and *fSunset*) to TIME format:

```
tSunrise     := LREAL_TO_TIME(fbSPA.fSunrise*60*60*1000);
```

Requirements

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.10 build >= 1320	PC or CX (x86, ARM)	TcSPA.Lib

6 Functions

6.1 F_GetVersionTcSPA

This function can be used to read PLC library version information.

FUNCTION F_GetVersionTcSPA: UINT

```
VAR_INPUT
    nVersionElement : INT;
END_VAR
```

nVersionElement : Version element to be read. Possible parameters:

- 1 : major number;
- 2 : minor number;
- 3 : revision number;

Requirements

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.10 build >= 1320	PC or CX (x86, ARM)	TcSPA.Lib

7 Data types

7.1 Structures

7.1.1 ST_SPA_TIMESTRUCT

```
TYPE ST_SPA_TIMESTRUCT :
STRUCT
    iYear       :INT(-2000..6000);      (* 4-
digit year, valid range: -2000 TO 6000, error code: 1      *)
    iMonth      :INT(1..12);    (* 2-digit month, valid range: 1 to 12 (Jan.= 1), error code: 2 *)
    iDay        :INT(1..31);    (* 2-digit day, valid range: 1 to 31, error code: 3         *)
    iHour       :INT(0..24);    (* Observer local hour, valid range: 0 to 24, error code: 4     *)
    iMinute     :INT(0..59);    (* Observer local minute, valid range: 0 to 59, error code: 5   *)
    iSecond     :INT(0..59);    (* Observer local second, valid range: 0 TO 59, error code: 6   *)
END_STRUCT
END_TYPE
```

The structure *ST_SPA_TIMESTRUCT* contains information on date and time. It is used at the input for function block FB_SPA [▸ 13] in order to specify the local time at the location. This local time has seconds as the smallest unit.

Various time scales are in use. The Universal Time (UT1) is used for sun position calculations based on the time specified in ST_SPA_TIMESTRUCT. Provided that an inaccuracy of +/-0.005 ° of the sun angles is acceptable, the Coordinated Universal Time (UTC) can also be used for the time indication. Explanations can be found on the overview page [▸ 8].

When specifying the time, the summertime/wintertime changeover must be omitted. The introduction of daylight saving time in the 20th century only serves the purpose of increasing the number of hours with usable daylight. For the calculation of the sun angles with this library the standard time must be used. Standard time corresponds to winter time in Germany.

7.2 Enumerations

7.2.1 E_SPA_FunctionCode

```
TYPE E_SPA_FunctionCode :(
(* enumeration for function codes to select desired final outputs from SPA *)
    eSPA_ZA,            (*calculate zenith AND azimuth      [default setting]       *)
    eSPA_ZA_INC,        (*calculate zenith, azimuth, AND incidence              *)
    eSPA_ZA_RTS,        (*calculate zenith, azimuth, AND sun rise/transit/set values    *)
    eSPA_ALL            (*calculate all SPA output values                   *)
);
END_TYPE
```

The enumeration *E_SPA_FunctionCode* defines constant values for the different functions, which can be executed with the function block FB_SPA [▸ 13]. In addition to sun angles, sunrise and sunset can be calculated, depending on the selection. A detailed explanation of the terminology can be found on the overview page [▸ 8].

eSPA_ZA : If the function code *eSPA_ZA* is selected only the sun angles (zenith, azimuth, azimuth180) are calculated [DEFAULT].

eSPA_ZA_INC : In addition to the sun angles the angle of incidence in relation to the specified surface is issued.

eSPA_ZA_RTS : In addition to the sun angles, sunrise, solar apex and sunset is calculated.

eSPA_ALL : All offered data are calculated and displayed at the output.

The time required for a calculation strongly depends on the choice of function code.

7.2.2 E_SPA_ErrorCode

```
TYPE E_SPA_ErrorCode :(
(* enumeration for error codes returned as iErrorCode output of FB_SPA
////////////////////////////////////////////////////////////////////////
//   Note: A non-zero return error code indicates that one of the     //
//     input values did not pass simple bounds tests.         //
//////////////////////////////////////////////////////////////////////// *)
    eSPA_ERR_NoError            := 0,
    eSPA_ERR_InvalidYear,
    eSPA_ERR_InvalidMonth,
    eSPA_ERR_InvalidDay,
    eSPA_ERR_InvalidHour,
    eSPA_ERR_InvalidMinute,
    eSPA_ERR_InvalidSecond,
    eSPA_ERR_InvalidDeltaT,
    eSPA_ERR_InvalidTimezone,
    eSPA_ERR_InvalidLongitude,
    eSPA_ERR_InvalidLatitude,
    eSPA_ERR_InvalidElevation,
    eSPA_ERR_InvalidPressure,
    eSPA_ERR_InvalidTemperature,
    eSPA_ERR_InvalidSlope,
    eSPA_ERR_InvalidAZMRotation,
    eSPA_ERR_InvalidAtmosRefract,
    eSPA_ERR_InvalidFunctionCode
);
END_TYPE
```

The enumeration *E_SPA_ErrorCode* defines constant values for the different errors that can be generated internally in the library.
These values can be found in the output variable *iErrorCode* again, which indicates the associated integer value at the output of the PLC SPA function block FB_SPA [▶ 13] in the event of an error.

8 Visualisations

8.1 V_SPA_OVERVIEW

The TwinCAT Solar Position Algorithm library contains a visualisation facility that provides a quick overview of current inputs and outputs of function block FB_SPA [▶ 13]. It is therefore ideal for test purposes.

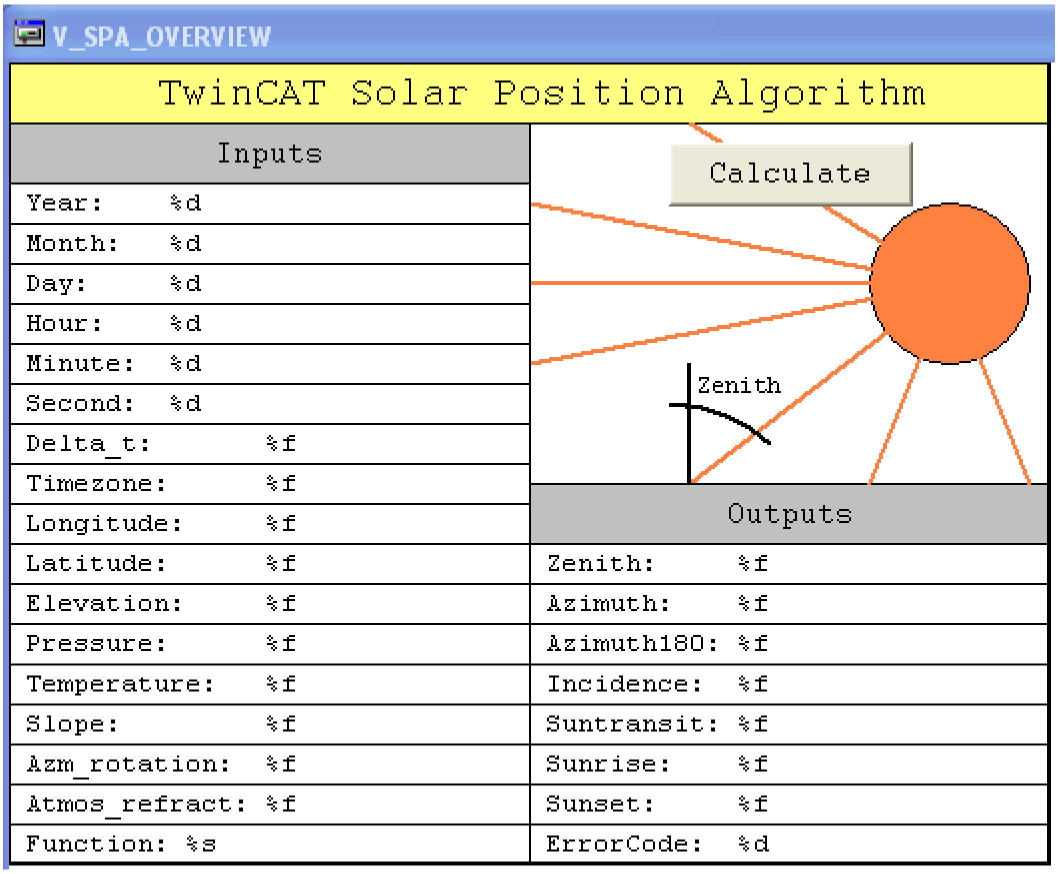
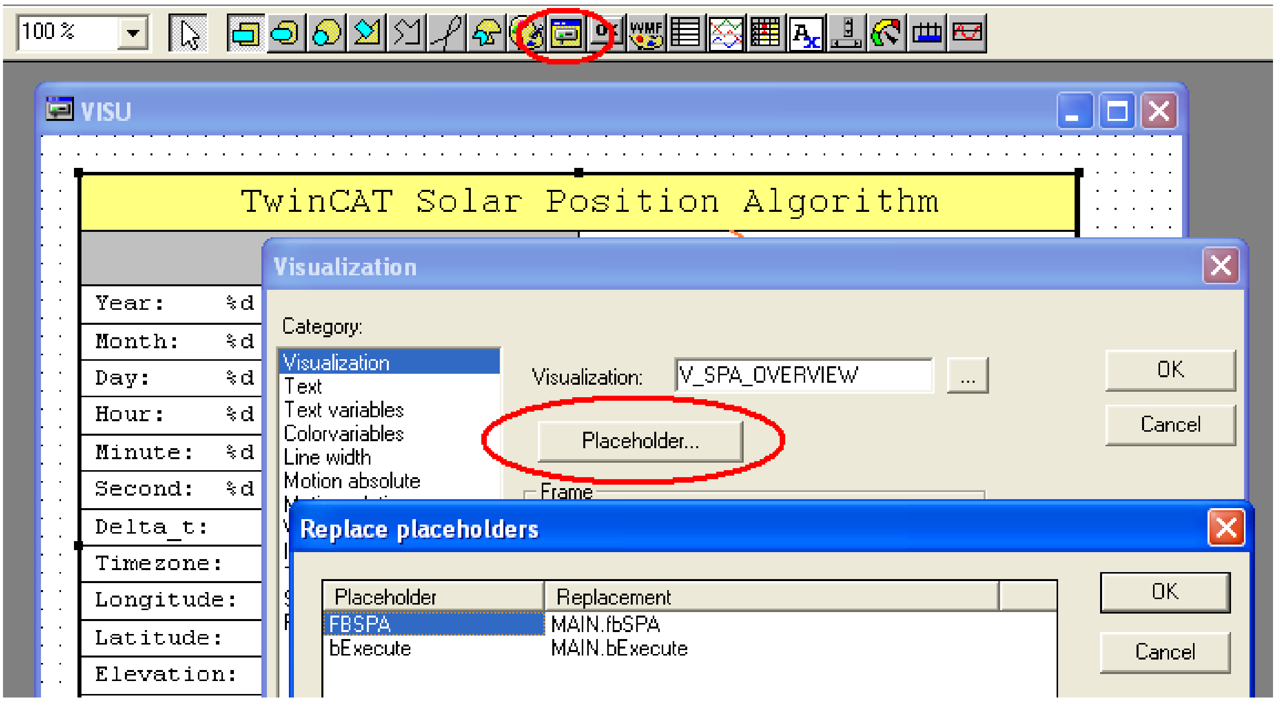


Fig. 1: SPA_Visu

To use the visualisation function a new visualisation must be created in the project. V_SPA_OVERVIEW is added via the Visualisation button.



Double-click on the added freely scalable visualisation to access the settings and the list of placeholders. Enter the instance of function block FB_SPA from the library. If required, the user can link a Boolean variable with the Calculate button in order to execute a calculation when the button is pressed.

See also the project example [▶ 22].

9 Example

This example offers an introduction into the handling of function block FB_SPA [▶ 13], which is available with the TwinCAT Solar Position Algorithm library.

The objective in this example is to determine the sun position on 4 March 2010 at 14:27:00 at the Cheops pyramid in Egypt.
Time zone: UTC + 2 hours
Latitude: 29.979, [°]
Longitude: 31.134 [°]
Height: 70 [m]
Annual average temperature: 21.7 [°C]
Other locations and times are determined similarly.

Overview

The following steps are now performed:

1. Installation of the PLC library
2. Program structure
3. Test

1. Installation of the PLC library

Start TwinCAT PLC Control.

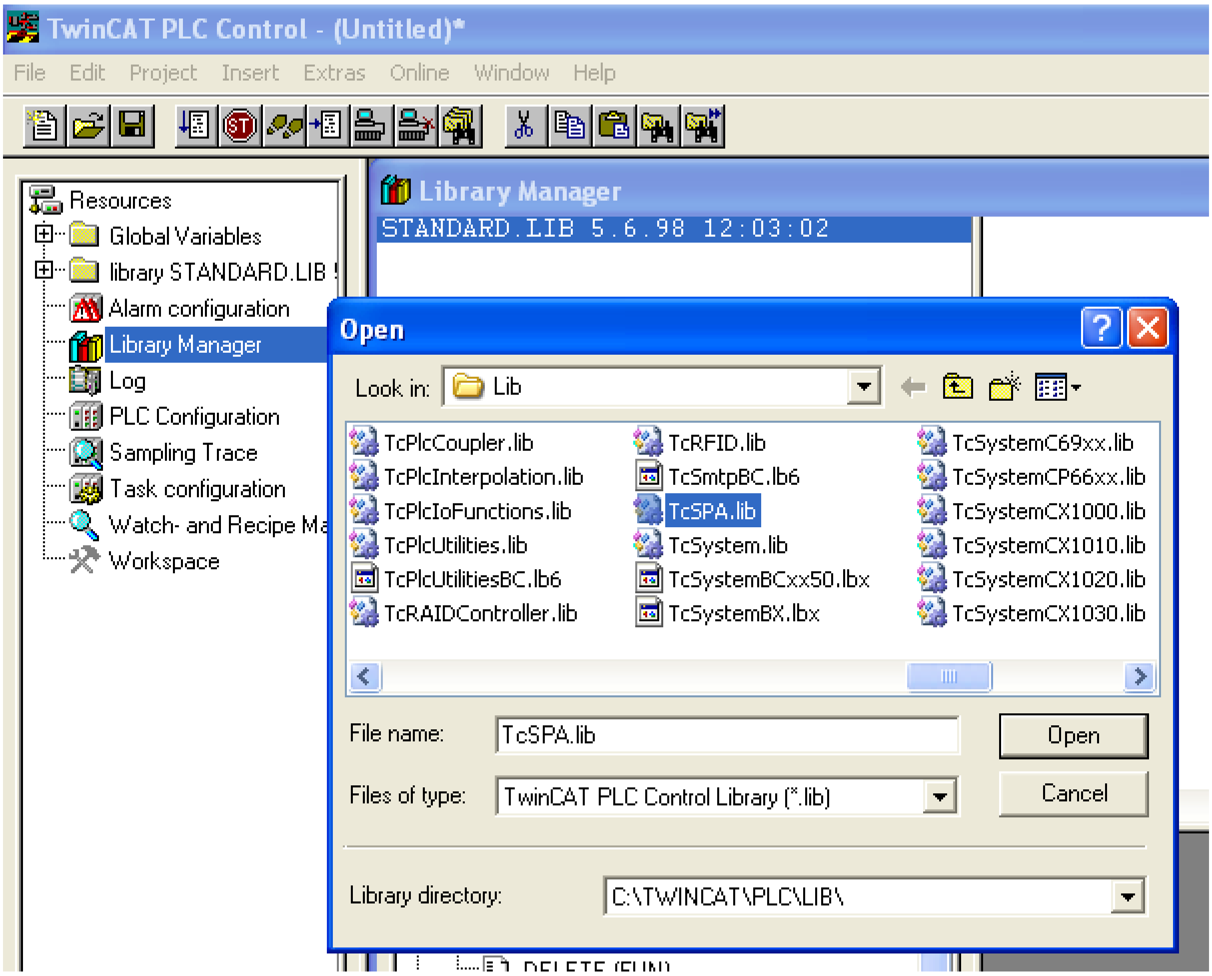
Create a new PLC project with 'File > New'.

Select your target platform PC and CX (x86) or CX (ARM).

Your first POU is a program called MAIN and in the programming language ST (Structured Text).

Open the Resources tab and the library manager.

Insert the library TcSPA.lib as shown in the picture below via 'Insert > Further library'.



All PLC blocks of the TwinCAT PLC SPA library are now available to you. All further implicitly required libraries have been automatically integrated with the TcSPA.lib.

2. Program structure

For sun position calculations you should declare an instance of function block FB_SPA [▶ 13]and local variables for allocating the required result values.

The input parameter for the calculation can be directly assigned to the inputs of the function block.
In addition to the sun angles the sunrise and sunset is required as output, which means the advanced functionality is required, which is specified via the enumeration value *eSPA_ZA_RTS* of type E_SPA_FunctionCode [▶ 18].
The output values of the function block are assigned to your local variables.

The program section should now look as follows:

```
PROGRAM MAIN
VAR
    fbSPA        : FB_SPA;
    fSunZenith       : LREAL;
    fSunAzimuth      : LREAL;
    tSunrise         : TIME;
    tSunset      : TIME;
    eErrorCode       : E_SPA_ErrorCode;
    bExecute         : BOOL;
END_VAR
```

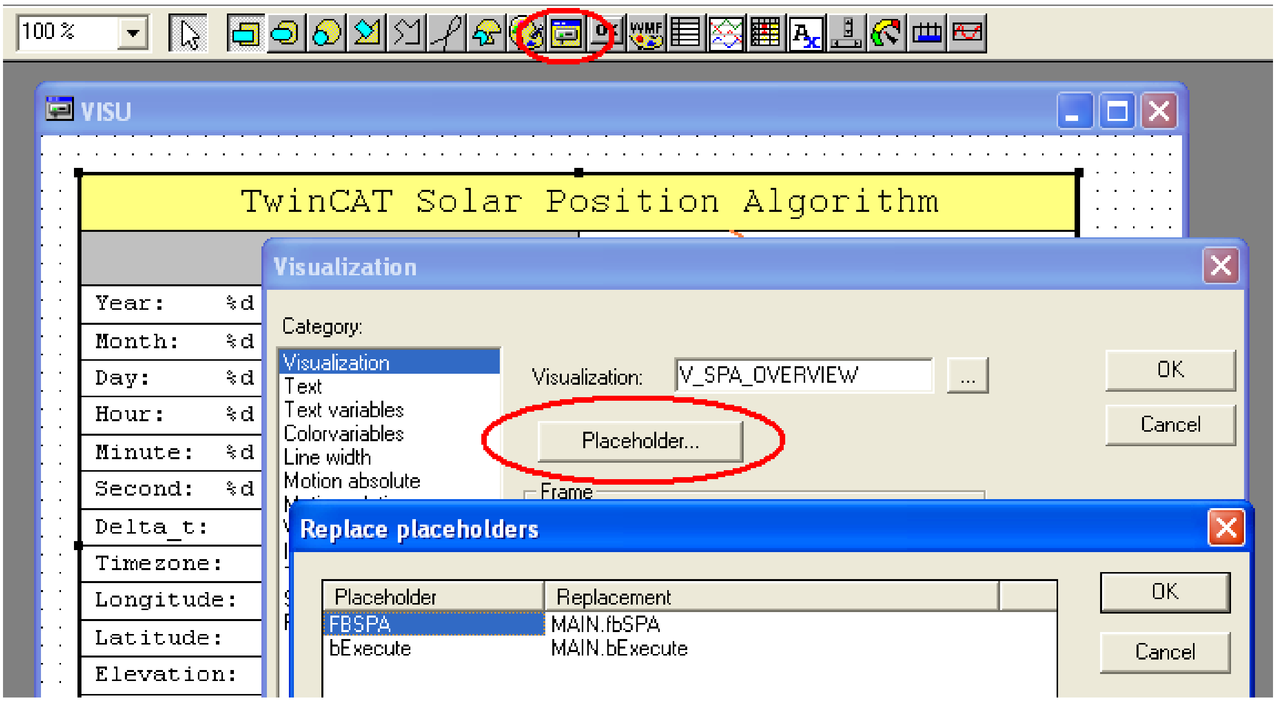
```
fbSPA.stTime.iYear          := 2010;
fbSPA.stTime.iMonth     := 3;
fbSPA.stTime.iDay           := 4;
fbSPA.stTime.iHour          := 14;
```

```
fbSPA.stTime.iMinute     := 27;
fbSPA.fTimezone      := 2;
fbSPA.fLongitude         := 31.134;
fbSPA.fLatitude      := 29.979;
fbSPA.fElevation         := 70;
fbSPA.fTemperature           := 21.7;
fbSPA.eFunction      := eSPA_ZA_RTS;

IF bExecute THEN
    fbSPA();
    eErrorCode       := fbSPA.iErrorCode;

    fSunZenith       := fbSPA.fZenith;
    fSunAzimuth      := fbSPA.fAzimuth;
    tSunrise         := LREAL_TO_TIME(fbSPA.fSunrise*60*60*1000);
    tSunset      := LREAL_TO_TIME(fbSPA.fSunset*60*60*1000);
END_IF
```

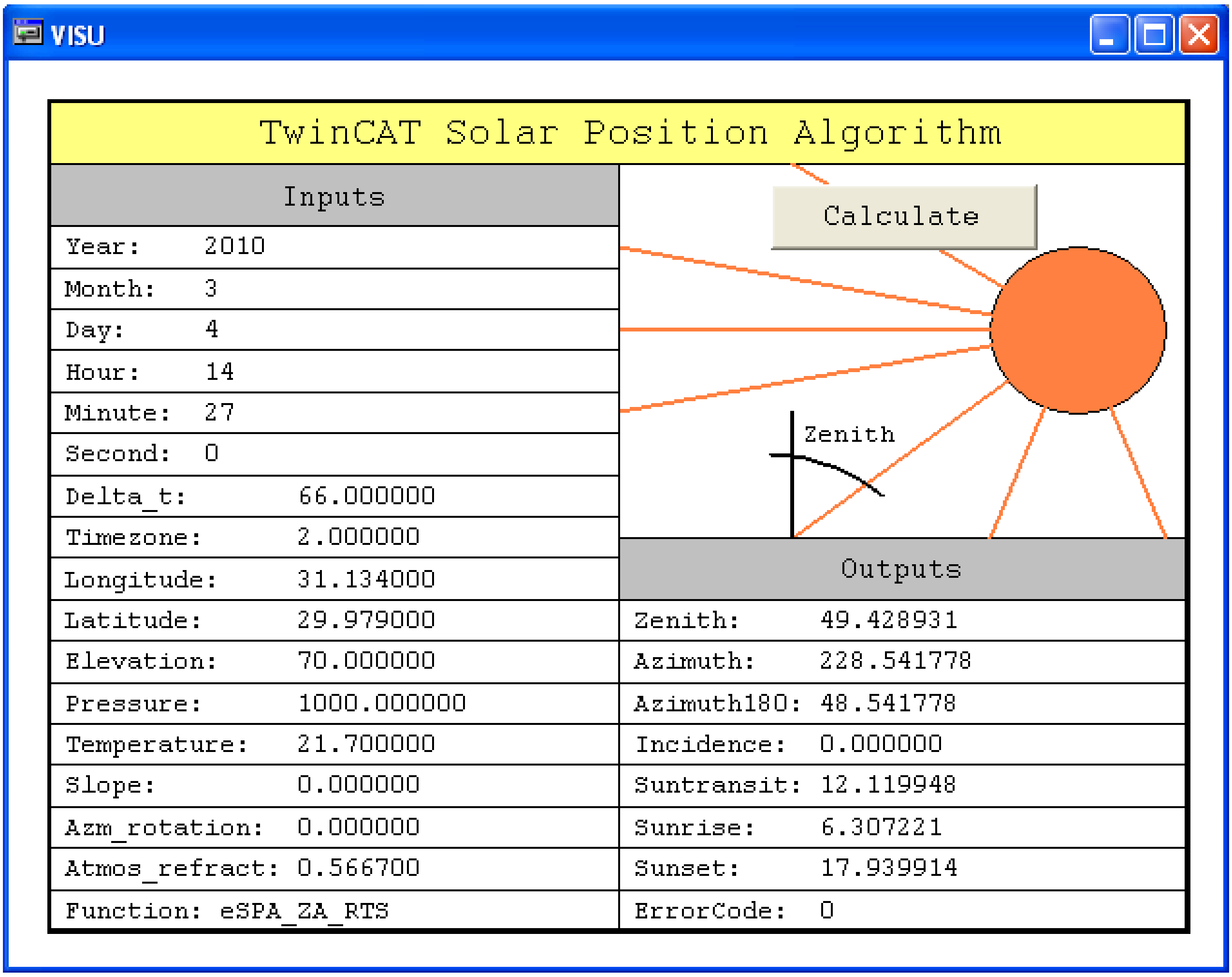
You can add the visualisation included in the library to your project as described in section Visualisation [▸ 20]. The previously declared variables are assigned to the placeholder as shown in the screenshot.



3. Test

Compile the created PLC program.
Make sure that TwinCAT is in the Run mode on the desired system.
Login to the desired run-time system from TwinCAT PLC Control. Start the PLC program.

The calculation is executed by setting the local variable *bExecute* to TRUE. This can be done via 'online write' or the corresponding button in the visualisation, for example.
The visualisation should now present the following results:



The sun angles at other locations and at other times within the given value ranges can be calculated accordingly. If an input parameter is invalid, an *eErrorCode* with the corresponding enumeration value for the error is displayed.

Click here to save this example program: https://infosys.beckhoff.com/content/1033/tcPlcLibSPA/Resources/11172780171/.zip.

Requirements

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.10 build >= 1320	PC or CX (x86, ARM)	TcSPA.Lib

More Information:
www.beckhoff.com/ts3900

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com