Library TBSWAstro

Function block “ AstroSunRiset ”

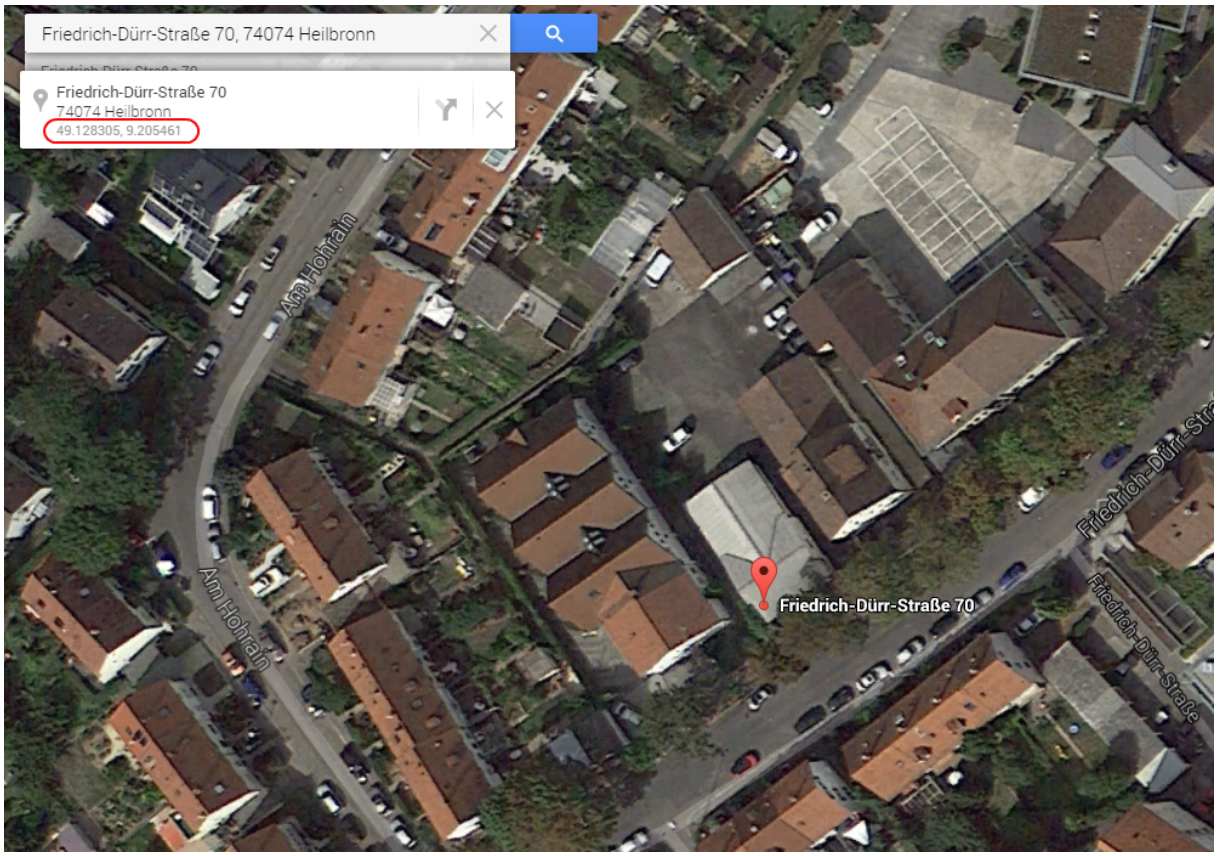
The function block calculates the times of sunrise and sunset based on a given geolocation.

The following can be calculated:

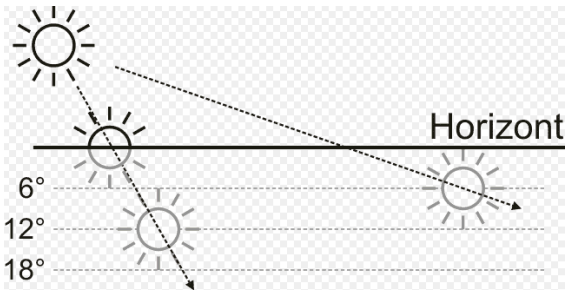
- Sunrise and sunset times for the current date
  - With or without taking into account the values set on the PLC for time zone and summer/winter time
- Sunrise and sunset on any given date

The geolocation is specified using latitude and longitude (as REAL values).
The coordinates can be determined, for example, in Google Maps by zooming in on the desired destination and clicking on it .

⇨ **Fub -Inputs “ longitude ”, “ latitude ”**



The direct time (sun crosses the horizon) can be calculated, as well as the so-called civil sunrise /sunset (sun 6° below the horizon) and the nautical sunrise/sunset (sun 12° below the horizon).



⇨ **Fub input “type” with the following constants:**

astroTYPE_NAUTICAL	USINT	☑	2
astroTYPE_DIRECT	USINT	☑	0
astroTYPE_CIVIL	USINT	☑	1

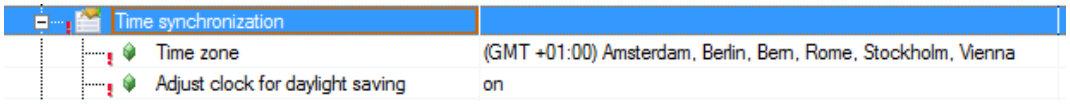
The calculation is performed for the current date (read internally by the FUB) if no specific date is set.

⇨ **Fub input “ pCalcDT ” = 0**

If a calculation is to be performed for a specific date, a pointer to a structure of type " DTStructure " must be passed to the input " pCalcDT " where the elements " year ", " month " and " day " are set to the desired values.

⇨ **Fub input “ pCalcDT ” = ADR(myDate)**

PLC settings regarding time zone and summer/winter time are taken into account in the result if the function block input “ gmtResult ” is not set.



⇨ **Fub input “ gmtResult ” = FALSE**

If the result is to be calculated without taking the settings into account, the input must be set to TRUE; in this case, the GMT time will be returned as the result.

⇨ **Fub input “ gmtResult ” = TRUE**

The result of the calculation is returned in a variable of type " AstroSunRisetCalcResult_type ". This variable must be declared in the task and connected as a pointer to the function block .

⇨ **Fub input “ pResult ” = ADR(myResult)**

AstroSunRisetCalcResult_type		
sunrise	AstroSunRiseSetTime_type	☐
sunset	AstroSunRiseSetTime_type	☐
sunriset_result	INT	☐
AstroSunRiseSetTime_type		
hour	USINT	☐
minute	USINT	☐

The result then contains sunrise and sunset times in hour and minute (e.g., " myResult.sunrise.hour " = hour of sunrise).

The element " myResult.sunriset_result " contains information about whether the calculation was correct (myResult.sunriset_result = 0). If this value is not 0, the result is invalid (e.g., because the sun does not set at the specified geolocation at the time of calculation).

The function block requires several cycles to complete the calculation; therefore, it must be called repeatedly until the FUB output “ status ” is not equal to BUSY (65535).

⇨ **Fub output “ status ” <> 65535**

FB AstroSunRiset					calculates sunrise / sunset
enable	BOOL		VAR_INPUT		enable function block
longitude	REAL		VAR_INPUT		longitude of position to calc for, set as REAL value (e.g. see google maps)
latitude	REAL		VAR_INPUT		latitude of position to calc for, set as REAL value (e.g. see google maps)
type	USINT		VAR_INPUT		type of calculation: direct or civil or nautical sunrise / sunset
pCalcDT	UDINT		VAR_INPUT		if NULL(set to 0), actual date time is used (if set, set date you want to calc for as pointer to DTStructure) type
gmtResult	BOOL		VAR_INPUT		if FALSE, timezone and daylight saving time settings are included in result
pResult	UDINT		VAR_INPUT		pointer to result variable of type "AstroSunRisetCalcResult_type"
status	UINT		VAR_OUTP...		65535 = busy

Example call

```
PROGRAM _INIT

    longitude := 9.205461;
    latitude := 49.128305;
    type := astroTYPE_DIRECT;
    enable := FALSE;
    tCalcDT;
    useGMT := FALSE;
    result;

END_PROGRAM


PROGRAM _CYCLIC

    IF blTestCalcDt = TRUE THEN
        pCalcDT := ADR(tCalcDT);
    ELSE
        pCalcDT := 0;
    END_IF

    CalcSunRiset_0(enable := enable, type := type, longitude := longitude, latitude := latitude, pCalcDT := pCalcDT, gmtResult := useGMT, pResult := ADR(result));
    IF CalcSunRiset_0.status <> 65535 THEN
        enable := FALSE;
    END_IF

END_PROGRAM
```

longitude	REAL
latitude	REAL
type	USINT
enable	BOOL
result	AstroSunRisetCalcResult_type
CalcSunRiset_0	AstroSunRiset
useGMT	BOOL
tCalcDT	DTStructure
pCalcDT	UDINT
blTestCalcDt	BOOL

Functionality used

The code is based on a public domain implementation; the functionality described above only provides a "wrapper" for the calculation for Automation Studio.
Original code from:

```
/*

SUNRISET.C - computers Sun rise/set times, start/end of Twilight, and
             the length of the day at any date and latitude

Written as DAYLEN.C, 1989-08-16

Modified to SUNRISET.C, 1992-12-01

(c) Paul Schlyter , 1989, Released in 1992

 to the public domain by Paul Schlyter , December 1992

*/
```