WAGO I/O SYSTEM 750

Library for Building Automation

Function block descriptions for time- switching functions

Last Update: 02.03.2020

Copyright © 2019 by WAGO Kontakttechnik GmbH & Co. KG
All rights reserved.

WAGO Kontakttechnik GmbH & Co. KG
Hansastraße 27
D-32423 Minden

Phone: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69

E-Mail: info@wago.com

Web: http://www.wago.com

Technical Support
Phone: +49 (0) 571/8 87 – 4 45 55
Fax: +49 (0) 571/8 87 – 84 45 55

E-Mail: tcba@wago.com

Every conceivable measure has been taken to ensure the correctness and completeness of this documentation. However, as errors can never be fully excluded we would appreciate any information or ideas at any time.

We wish to point out that the software and hardware terms as well as the trademarks of companies used and/or mentioned in the present manual are generally trademark or patent protected.

WAGO-I/O-PRO CAA Library for time-switching functions

List of contents

Important Comments 4

Copyright 4
Personnel Qualification 4
Intended Use 4
Scope of validity 4

Scheduler functions 5

Weekly time-switching program (FbScheduleWeekly) 5
Period time-switching program (FbScheduleSpecialPeriod) 8
Public holiday, period and weekly time-switching program (FbScheduler) 11
Timetable Program (FbTimetable) 21

Additional functions 27

Additional functions 27

Identification of Public Holidays (FbPublicHoliday) 27
Holiday Determination (FbHoliday) 32
General Value Specifications (FbSetScheduleValue) 34
Simulating the clock function (FbClock) 38
Storing data in the file system (FbSaveVariable) 40
Daylight saving (FbSummer_Wintertime) 41

Conversions 42

Split-up data type DT (FuDT_To_DetailTime) 42
Put together data type DT (FuDetailTime_To_DT) 43
Conversion of the data type DT to string (FuDT_TO_String) 44

RTC Module 45

Date and Time via the RTC Module (FbTime_RTC_Modul) 45
Time Conversion (FuDT_LocalTime) 46

Important Comments

To ensure fast installation and start-up of the units described in this manual, we strongly recommend that the following information and explanation is carefully read and adhered to.

Copyright

This manual is copyrighted, together with all figures and illustrations contained therein. Any use of this manual which infringes the copyright provisions stipulated herein, is not permitted. Reproduction, translation and electronic and photo-technical archiving and amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG. Non-observance will entail the right of claims for damages.

Personnel Qualification

The use of the product detailed in this manual is exclusively geared to specialists having qualifications in PLC programming, electrical specialists or persons instructed by electrical specialists who are also familiar with the valid standards. WAGO Kontakttechnik GmbH & Co. KG declines all liability resulting from improper action and damage to WAGO products and third party products due to non-observance of the information contained in this manual.

Intended Use

For each individual application, the components supplied are to work with a dedicated hardware and software configuration. Modifications are only admitted within the framework of the possibilities documented in the manuals. All other changes to the hardware and/or software and the non-conforming use of the components entail the exclusion of liability on part of WAGO Kontakttechnik GmbH & Co. KG.

Please direct any requirements pertaining to a modified and/or new hardware or software configuration directly to WAGO Kontakttechnik GmbH & Co. KG.

Scope of validity

This application note is based on the respective manufacturer's stated hardware and software and the associated documentation. This application note therefore only applies to the installation described.
New hardware and software versions may need to be treated differently.

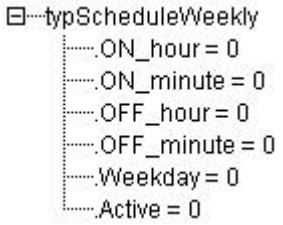
Please pay attention to the detailed description in the respective manuals.

Scheduler functions

Weekly time-switching program (FbScheduleWeekly)

WAGO-I/O-*PRO* CAA Library Elements			
Category:	Building Automation		
Name:	FbScheduleWeekly		
Type:	Function ☐	Function block ☒	Program ☐
Library name:	Scheduler_03.lib		
Applicable to:	Programmable fieldbus controllers		
Input parameter:	**Data type:**	**Comment:**	
xEnable	BOOL	Activation of the function block Default Value: TRUE	
dtActualTime	DT	Input for current date and time	
Input/output parameter:	**Data type:**	**Comment:**	
typScheduleWeekly	typSchedule Weekly	Data structure for the weekly time-switching program	
.ON_hour	WORD	Turn on time (hours)	
.ON_minute	WORD	Turn on time (minutes)	
.OFF_hour	WORD	Turn off time (hours)	
.OFF_minute	WORD	Turn off time (minutes)	
.Weekday	WORD	Weekday (bit coded) Bit 0 = Monday Bit 1 = Tuesday Bit 2 = Wednesday Bit 3 = Thursday Bit 4 = Friday Bit 5 = Saturday Bit 6 = Sunday	
.Active	WORD	Time-switching condition fulfilled	
Feedback value:	**Data type:**	**Comment:**	
xSwitchChannel	BOOL	Switching output of the weekly time-switching program	
Graphical display:			

Structure of the „typScheduleWeekly“ variable:

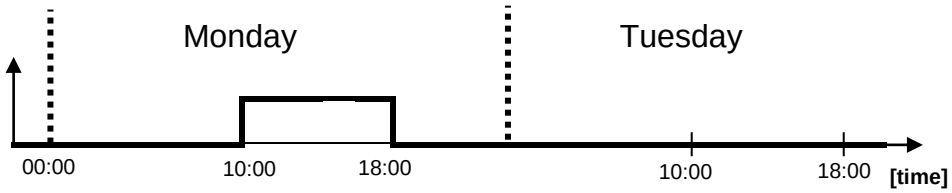


Data structure of the weekly time-switching program

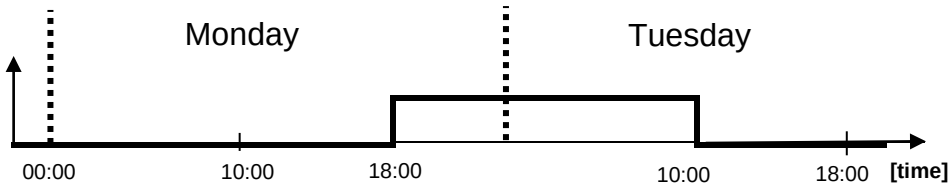
Configuration interface:

Time Referenced Behavior:

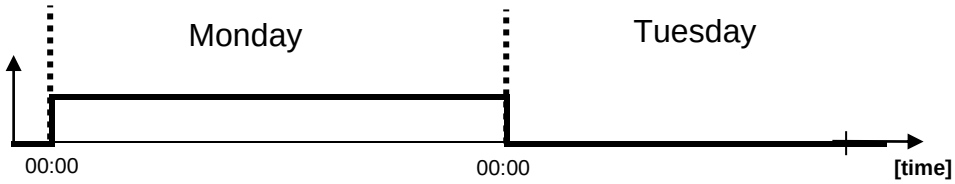
Example 1: Monday active, On: 10 am, Off: 6 pm



Example 2: Monday active, On: 6 pm, off: 10 am



Example 3: Monday active, On: 12 midnight, Off: 12 midnight



Function description:
The **FbScheduleWeekly** function block contains the functionality of a weekly time-switching program. The ***“xEnable"*** input is used for activating the function block. The ***"dtActualTime"*** input is linked to the current time. This time is used as the basis for determining the time switching program. The ***"typScheduleWeekly"*** structure variable contains all values for the time-switching programs in the **FbScheduleSingleWeekly** function block. The ***"xSwitchChannel"*** output is set when a time-switching condition is fulfilled. **Notes:** 1) The current system time of the Ethernet controller 750-841 can be determined by means of the **SysRtcGetTime** function. This function can be found in the **SysLibRtc** library. 2) The configuration interface ***"visuScheduleWeekly"*** is provided in the library for convenient configuration of the time-switching program. 3) The "*typScheduleWeekly*" structure should be declared as RETAIN so that the time-switching conditions are retained following a controller reset. 4) In the case of the Ethernet controller 750-841, the "*typScheduleWeekly*" structure should also be declared as PERSISTENT, so that the time-switching conditions are also retained after a download. 5) As an option, the "*typScheduleWeekly*" structure can be written to the file system of the Ethernet controller 750-841 in order to back-up the time-switching conditions. The data are saved by means of the **FbSaveVariable** function block.

Period time-switching program (FbScheduleSpecialPeriod)

WAGO-I/O-*PRO* CAA Library Elements			
Category:	Building Automation		
Name:	FbScheduleSpecialPeriod		
Type:	Function ☐	Function block ☒	Program ☐
Library name:	Scheduler_03.lib		
Applicable to:	Programmable fieldbus controllers		
Input parameter:	**Data type:**	**Comment:**	
xEnable	BOOL	Activation of the function block Default value: TRUE	
dtActualTime	DT	Input for current date and time	
Input/output parameter:	**Data type:**	**Comment:**	
typScheduleSpecial Period	typSchedule Special Period	Data structure of the period time-switching program	
.FirstDay_day	WORD	First day (day)	
.FirstDay_month	WORD	First day (month)	
.FirstDay_year	WORD	First day (year)	
.LastDay_day	WORD	Last day (day)	
.LastDay_month	WORD	Last day (month)	
.LastDay_year	WORD	Last day (year)	
.ON_hour	WORD	Turn on time (hours)	
.ON_minute	WORD	Turn on time (minutes)	
.OFF_hour	WORD	Turn off time (hours)	
.OFF_minute	WORD	Turn off time (minutes)	
.Weekday	WORD	Weekday (Bit coded) Bit 0 = Monday Bit 1 = Tuesday Bit 2 = Wednesday Bit 3 = Thursday Bit 4 = Friday Bit 5 = Saturday Bit 6 = Sunday	
.Active	WORD	Time-switching condition fulfilled	
Feedback value:	**Data type:**	**Comment:**	
xSwitchChannel	BOOL	Switching output of the period time-switching program	
Graphical display:			

Structure of the „typScheduleSpecialPeriod“ variable:

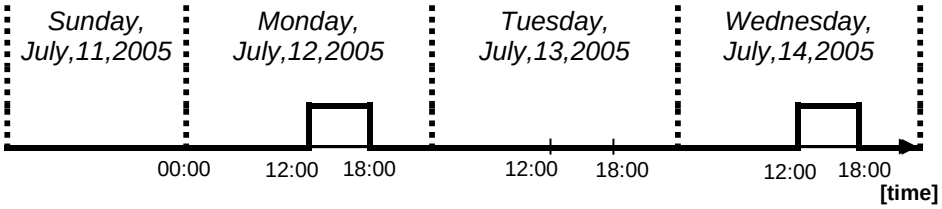
⊟ typScheduleSpecialPeriod .FirstDay_day = 1 .FirstDay_month = 1 .FirstDay_year = 2007 .LastDay_day = 1 .LastDay_month = 1 .LastDay_year = 2007 .ON_hour = 0 .ON_minute = 0 .OFF_hour = 0 .OFF_minute = 0 .Weekday = 0 .Active = 0	**Data structure of the period time-switching program**

Configuration interface:

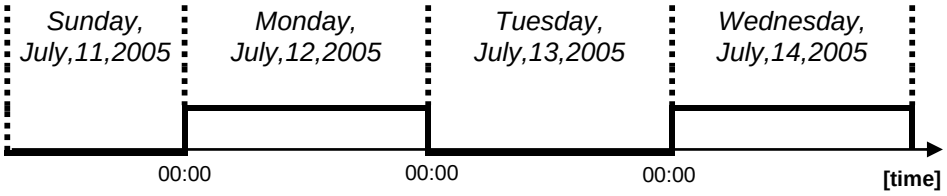
First day	Last day	ON	OFF	Weekday							Status
01-01-2007	01-01-2007	00:00	00:00	MO	TU	WE	TH	FR	SA	SU	

Time Referenced Behavior:

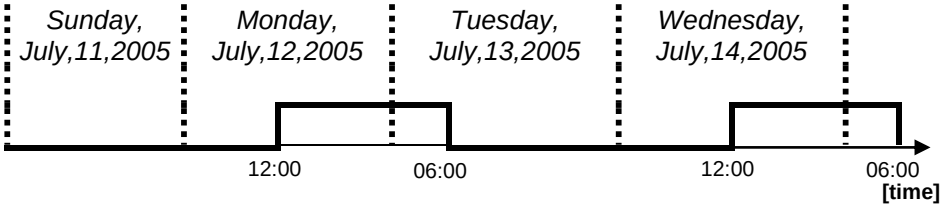
Example 1: First day: July,12,2005; Last day: July,14,2005
Monday and Wednesday are activated.
On: 12 noon, Off: 6 pm



Example 2: First day: July,12,2005; Last day: July,14,2005
Monday and Wednesday are activated.
On: 12 midnight, Off: 12 midnight

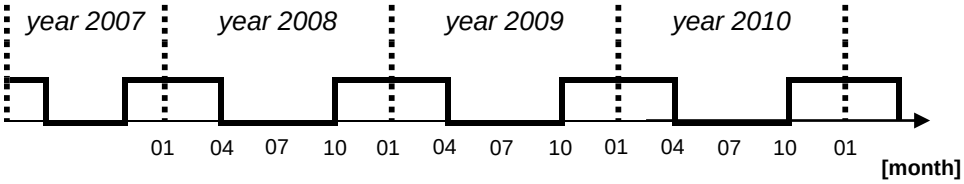


Example 3: First day: July,12,2005, Last day: July,14,2005
Monday and Wednesday are activated.
On: 12 noon, Off: 6 am



Time Referenced Behavior:

Example 4: annually recurring event (as indication of the year write a zero)
First Day: 01.10.0000, Last Day: 01.04.0000
All weekdays are active
On: 12 midnight, Off: 12 midnight



Function description:

The function module **FbScheduleSpecialPeriod** contains the functionality of a period time switching program.

The ***"xEnable"*** input is used for activating the function block.

The ***"dtActualTime"*** input is linked to the current time. This time is used as the basis for determining the time switching program.

The ***"typScheduleSpecialPeriod"*** structure variable contains all values for the time-switching programs in the **FbScheduleSpecialPeriod** function block.

The ***"xSwitchChannel"*** output is set when a time-switching condition is fulfilled.

Notes:

1) The current system time of the Ethernet controller 750-841 can be determined by means of the **SysRtcGetTime** function. This function can be found in the **SysLibRtc** library.

2) The configuration interface ***"visuScheduleSpecialPeriod"*** is provided in the library for convenient configuration of the time-switching program.

3) The "*typScheduleSpecialPeriod*" structure should be declared as RETAIN so that the time-switching conditions are retained following a controller reset.

4) In the case of the Ethernet controller 750-841, the "*typScheduleSpecialPeriod*" structure should also be declared as PERSISTENT, so that the time-switching conditions are also retained after a download.

5) As an option, the "*typScheduleSpecialPeriod*" structure can be written to the file system of the Ethernet controller 750-841 in order to back-up the time-switching conditions. The data are saved by means of the **FbSaveVariable** function block.

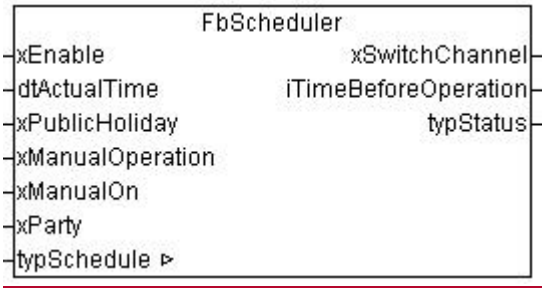
Public holiday, period and weekly time-switching program (FbScheduler)

WAGO-I/O-*PRO* CAA Library Elements		
Category:	Building Automation	
Name:	FbScheduler	
Type:	Function ☐	Function block ☒ / Program ☐
Library name:	Scheduler_03.lib	
Applicable to:	Programmable fieldbus controllers	
Input parameter:	**Data type:**	**Comment:**
xEnable	BOOL	Activation of the function block. Default Value: TRUE
dtActualTime	DT	Input for current date and time
xPublicHoliday	BOOL	Activation of the public holiday switching program
xManualOperation	BOOL	Manual mode
xManualOn	BOOL	Switching in manual mode
xParty	BOOL	Start or restart of party function
Input/output parameter:	**Data type:**	**Comment:**
typSchedule	typSchedule	Data structure for the time-switching programs
.ScheduleWeekly	ARRAY [1..10] of typSchedule Weekly	Ten weekly time-switching programs
.ON_hour	WORD	Turn on time (hours)
.ON_minute	WORD	Turn on time (minutes)
.OFF_hour	WORD	Turn off time (hours)
.OFF_minute	WORD	Turn off time (minutes)
.Weekday	WORD	Weekday (Bit coded) Bit 0 = Monday Bit 1 = Tuesday Bit 2 = Wednesday Bit 3 = Thursday Bit 4 = Friday Bit 5 = Saturday Bit 6 = Sunday
.Active	WORD	Time-switching condition fulfilled
.ScheduleSpecialPeriods	ARRAY [1..5] of typSchedule Special Period	Five period time-switching programs
.FirstDay_day	WORD	First day (day)
.FirstDay_month	WORD	First day (month)
.FirstDay_year	WORD	First day (year)
.LastDay_day	WORD	Last day (day)

.LastDay_month	WORD	Last day (month)
.LastDay_year	WORD	Last day (year)
.ON_hour	WORD	Turn on time (hours)
.ON_minute	WORD	Turn on time (minutes)
.OFF_hour	WORD	Turn off time (hours)
.OFF_minute	WORD	Turn off time (minutes)
.Weekday	WORD	Weekday (Bit coded) Bit 0 = Monday Bit 1 = Tuesday Bit 2 = Wednesday Bit 3 = Thursday Bit 4 = Friday Bit 5 = Saturday Bit 6 = Sunday
.Active	WORD	Time-switching condition fulfilled
.PublicHolidayON_hour	WORD	Turn on time on public holidays (hours)
.PublicHolidayON_minute	WORD	Turn on time on public holidays (minutes)
.PublicHolidayOFF_hour	WORD	Turn off time on public holidays (hours)
.PublicHolidayOFF_ minute	WORD	Turn off time on public holidays (minutes)
.PublicHoliday_active	WORD	Time-switching condition fulfilled
.Party_length	WORD	Duration of party function [min]
.Party_active	WORD	Party function aktive
.SwitchChannel	WORD	Copy of the switch channel output
.StatusScheduler	WORD	Status of the scheduler Bit 0 = Function block active Bit 1 = Manual operation active Bit 2 = Status manual operation Bit 3 =Public holiday Bit 4 = Party function active
.TimeBeforeOperation	INT	The remaining time up to the beginning (+) or the end of the service period (-)

Feedback value:	Data type:	Comment:
xSwitchChannel	BOOL	Switching output of time-switching program
iTimeBeforeOperation	INT	The remaining time up to the beginning (+) or the end of the service period (-)
typScheduleStatus	typScheduleStatus	Data structure for the status
.xEnable	BOOL	Module enable
.xManualOperation	BOOL	Switch manually
.xManualOn	BOOL	Manual ON
.xPublicHoliday	BOOL	Holiday
.xParty	BOOL	Party function
.xSwitchChannel	BOOL	Switching Output
.axWeekly	Array[1..10] of BOOL	Status for the ten weekly time-switching programs
.axSpecialPeriod	Array[1..5] of BOOL	Status for the five period time-switching programs

Graphical display:



Structure of the „typSchedule“ variable:	
typSchedule .ScheduleWeekly .ScheduleWeekly[1] .ON_hour = 0 .ON_minute = 0 .OFF_hour = 0 .OFF_minute = 0 .Weekday = 0 .Active = 0 .ScheduleWeekly[2] .ScheduleWeekly[3] .ScheduleWeekly[4] .ScheduleWeekly[5] .ScheduleWeekly[6] .ScheduleWeekly[7] .ScheduleWeekly[8] .ScheduleWeekly[9] .ScheduleWeekly[10]	**Weekly time-switching programs**
.ScheduleSpecialPeriods .ScheduleSpecialPeriods[1] .FirstDay_day = 1 .FirstDay_month = 1 .FirstDay_year = 2007 .LastDay_day = 1 .LastDay_month = 1 .LastDay_year = 2007 .ON_hour = 0 .ON_minute = 0 .OFF_hour = 0 .OFF_minute = 0 .Weekday = 0 .Active = 0 .ScheduleSpecialPeriods[2] .ScheduleSpecialPeriods[3] .ScheduleSpecialPeriods[4] .ScheduleSpecialPeriods[5]	**Period time-switching programs**
.PublicHolidayON_hour = 0 .PublicHolidayON_minute = 0 .PublicHolidayOFF_hour = 0 .PublicHolidayOFF_minute = 0 .PublicHoliday_active = 0	**Public holiday time-switching program**
.Party_length = 60 .Party_active = 0	**Party function**
.SwitchChannel = 0 .StatusScheduler = 0 .TimeBeforeOperation = 0	**Status information’s**

Structure of the „typScheduleStatus“ variable:

```
typScheduleStatus
    .xEnable = TRUE
    .xManualOperation = FALSE
    .xManualOn = FALSE
    .xPublicHoliday = FALSE
    .xParty = FALSE
    .xSwitchChannel = FALSE
    .axWeekly
        .axWeekly[0] = FALSE
        .axWeekly[1] = FALSE
        .axWeekly[2] = FALSE
        .axWeekly[3] = FALSE
        .axWeekly[4] = FALSE
        .axWeekly[5] = FALSE
        .axWeekly[6] = FALSE
        .axWeekly[7] = FALSE
        .axWeekly[8] = FALSE
        .axWeekly[9] = FALSE
        .axWeekly[10] = FALSE
    .axSpecialPeriod
        .axSpecialPeriod[0] = FALSE
        .axSpecialPeriod[1] = FALSE
        .axSpecialPeriod[2] = FALSE
        .axSpecialPeriod[3] = FALSE
        .axSpecialPeriod[4] = FALSE
        .axSpecialPeriod[5] = FALSE
```

Configuration interface:

FbScheduler status display:

Enable	Channel ON	Manual operation	Manual on	Time before operation
☐	☐	☐	☐	0 min

Public holiday time-switching program and party function:

Public holiday	ON	OFF	Status	Party function	Switch-on-time	Status
☐	00 : 00	00 : 00	☐	☐	60	☐

Period time-switching programs

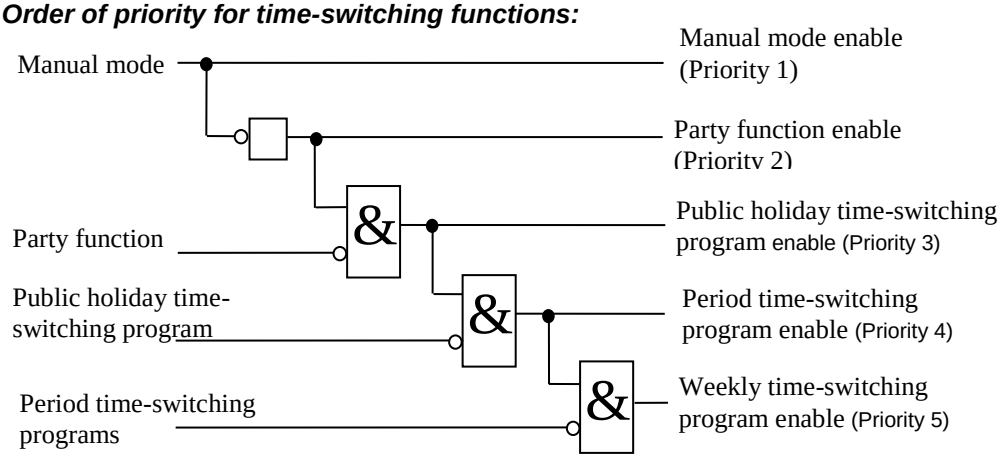
First day	Last day	ON	OFF	Weekday	Status
01-01-2007	01-01-2007	00 : 00	00 : 00	MO TU WE TH FR SA SU	☐
01-01-2007	01-01-2007	00 : 00	00 : 00	MO TU WE TH FR SA SU	☐
01-01-2007	01-01-2007	00 : 00	00 : 00	MO TU WE TH FR SA SU	☐
01-01-2007	01-01-2007	00 : 00	00 : 00	MO TU WE TH FR SA SU	☐
01-01-2007	01-01-2007	00 : 00	00 : 00	MO TU WE TH FR SA SU	☐

weekly time-switching program

ON	OFF	Weekday	Status
00 : 00	00 : 00	MO TU WE TH FR SA SU	☐
00 : 00	00 : 00	MO TU WE TH FR SA SU	☐
00 : 00	00 : 00	MO TU WE TH FR SA SU	☐
00 : 00	00 : 00	MO TU WE TH FR SA SU	☐
00 : 00	00 : 00	MO TU WE TH FR SA SU	☐
00 : 00	00 : 00	MO TU WE TH FR SA SU	☐
00 : 00	00 : 00	MO TU WE TH FR SA SU	☐
00 : 00	00 : 00	MO TU WE TH FR SA SU	☐
00 : 00	00 : 00	MO TU WE TH FR SA SU	☐
00 : 00	00 : 00	MO TU WE TH FR SA SU	☐

Order of priority:

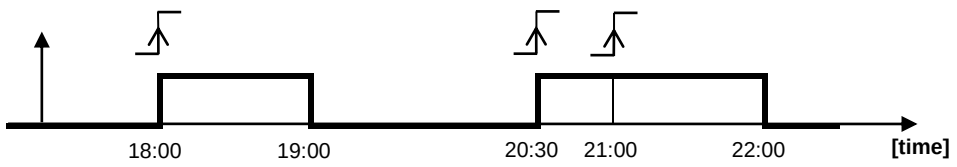
Order of priority for time-switching functions:



Time Referenced Behavior:

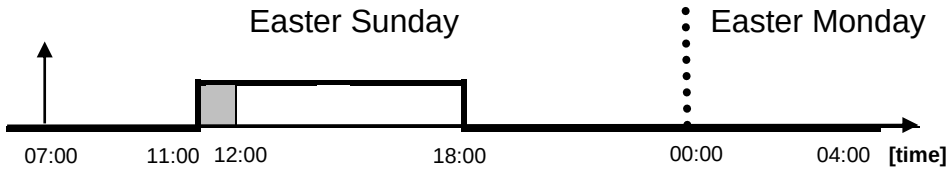
Party function:

Example: Party function 60 minutes.

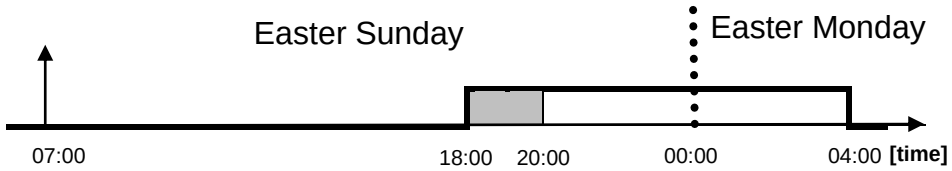


Public holiday time-switching program:

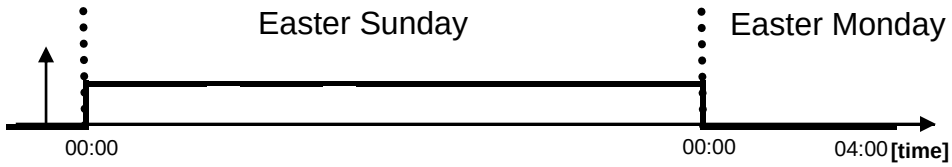
Example 1: Easter Sunday (active), On: 12 noon, Off: 6 pm,
Time before start of use 60 minutes.



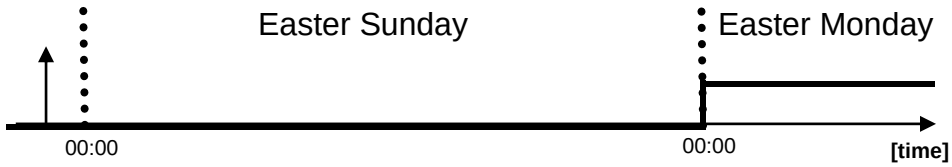
Example 2: Easter Sunday (active), On: 8 pm, Off: 4 am,
Time before start of use 120 minutes.



Example 3: Easter Sunday (active), On: 12 midnight, Off: 12 midnight

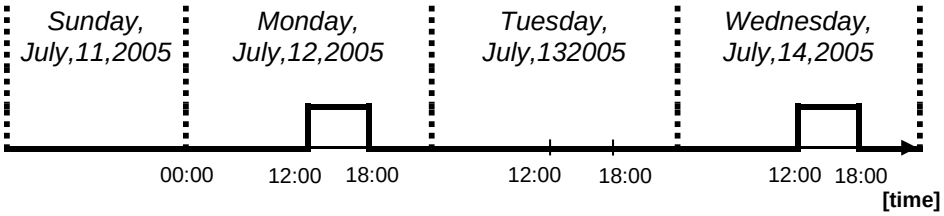


Example 4: Easter Sunday (active), On: 12 midnight, Off: 12 midnight
Easter Monday (active), On: 12 midnight, Off: 12 midnight

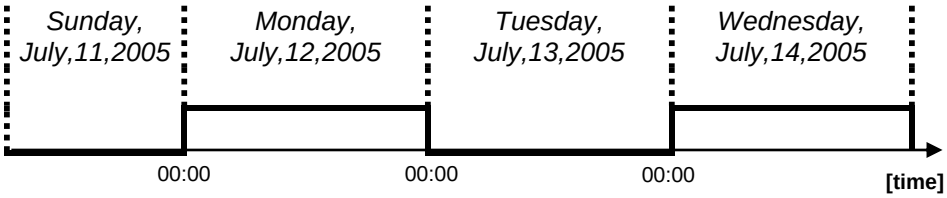


Period time-switching programs

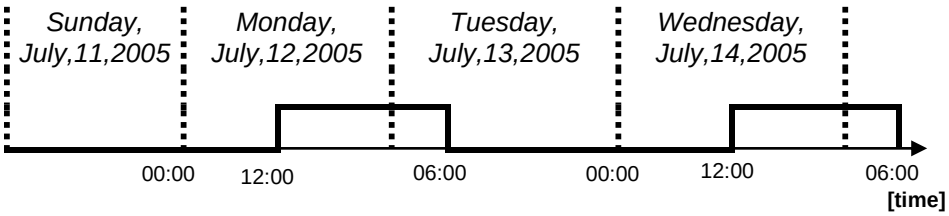
Example 1: First day: July,12,2005; Last day: July,14,2005
Monday and Wednesday are activated.
On: 12 noon, Off: 18 pm



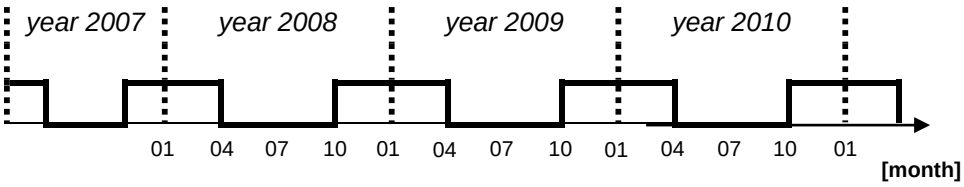
Example 2: First day: July,12,2005; Last day: July,14,2005
Monday and Wednesday are activated.
On: 12 midnight, Off: 12 midnight



Example 3: First day: July,12,2005; Last day: July,14,2005
Monday and Wednesday are activated.
On: 12 noon, Off: 6 am

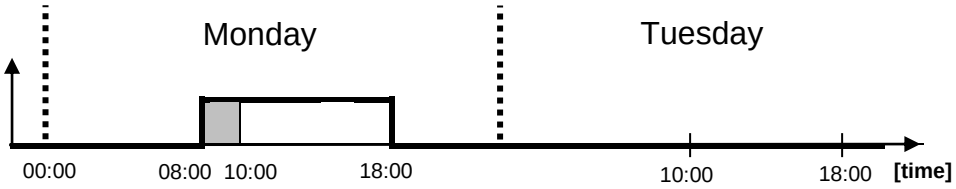


Example 4: annually recurring event (as indication of the year write a zero)
First Day: 01.10.0000, Last Day: 01.04.0000
All weekdays are active
On: 12 midnight, Off: 12 midnight

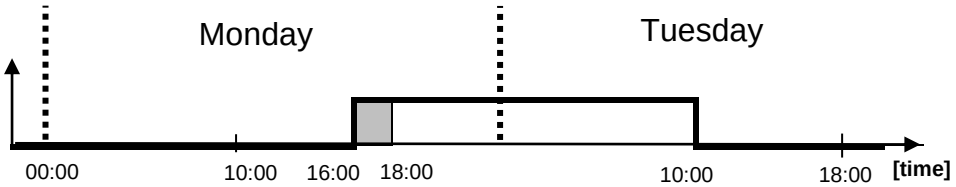


Weekly time-switching programs

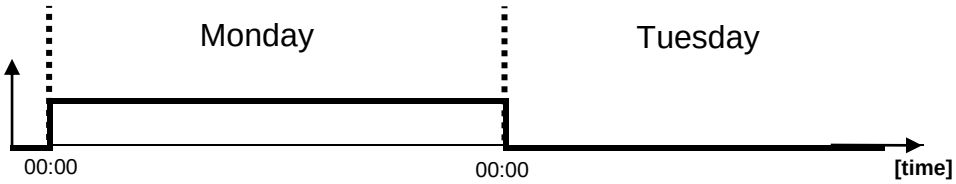
Example 1: Monday active, On: 10 am, Off: 6 pm,
Time before start of use 120 minutes.



Example 2: Monday active, On: 6 pm, Off: 10 am,
Time before start of use 120 minutes.



Example 3: Monday active, On: 12 midnight, Off: 12 midnight



Function description:

The **FbScheduler** function block includes different time-switching functions, such as the period time-switching program, public holiday time-switching program and weekly time-switching program.

Priorities are assigned to the different time-switching programs so that overlaps between the programs do not occur.

<u>Priority for time-switching functions:</u>

- Manual mode (Priority 1)
- Party function (Priority 2)
- Public holiday time-switching program (Priority 3)
- Period time-switching program (Priority 4)
- Weekly time-switching program (Priority 5)

The ***"xEnable"*** input is used for activating the **FbScheduler** function block.

The ***"dtActualTime"*** input is linked to the current time. This time is used as the basis for determining the time switching program.

The public holiday switching program is executed when the ***"xPublicHoliday"*** input is TRUE. Public holidays can be detected by means of the **FbPublicHoliday** function block.

The ***"xManualOperation"*** input activates manual mode. In manual mode, the time-switching program is controlled by the ***"xManualOn"*** input.

A positive edge on the ***"xParty"*** input starts the party function. With the party function, the output of the time-switching program is set for an adjustable time. If the function block detects another positive edge during this period, the set time is restarted.

The ***"typSchedule"*** structure variable contains all values for the time-switching programs in the **FbScheduler** function block.

The ***"xSwitchChannel"*** output is set when a time-switching condition is fulfilled.

The output ***"iTimeBeforeOperation"*** shows the remaining time up to the beginning (+) or the end of the service period (-). The switching time cannot be brought forward to before a time of 00:00 hours.

Notes:

1.) The current system time of the Ethernet controller 750-841 can be determined by means of the **SysRtcGetTime** function. This function can be found in the **SysLibRtc** library.

2.) The configuration interfaces ***"visuScheduler_Status", "visuScheduler_SpecialPeriods", "visuScheduler_Weekly"*** and ***"visuScheduler_HolidayParty"*** are provided in the library for convenient configuration of the time-switching programs.

3.) The "*typSchedule*" structure should be declared as RETAIN so that the time-switching conditions are retained following a controller reset.

4.) In the case of the Ethernet controller 750-841, the "*typSchedule*" structure should also be declared as PERSISTENT, so that the time-switching conditions are also retained after a download.

5.) As an option, the "*typSchedule*" structure can be written to the file system of the Ethernet controller 750-841 in order to back-up the time-switching conditions. The data are saved by means of the **FbSaveVariable** function block.

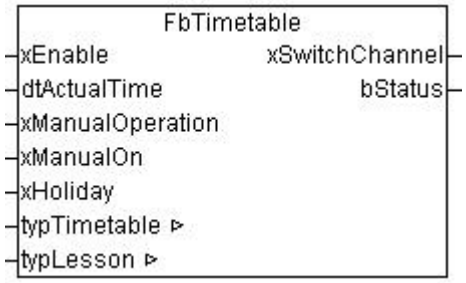
Timetable Program (FbTimetable)

WAGO-I/O-PRO CAA Library Elements			
Category:	Building Automation		
Name:	FbTimetable		
Type:	Function ☐	Function block ☒	Program ☐
Name of library:	Scheduler_03.lib		
Applicable to:	All programmable fieldbus controllers		

Input parameter:	**Data type:**	**Comment:**
xEnable	BOOL	Activation of the function block Default setting: TRUE
dtActualTime	DT	Input for the actual time
xManualOperation	BOOL	Manual mode
xManualOn	BOOL	Switching to manual mode
xHoliday	BOOL	Holiday Yes/No

Input/output parameter:	**Data type:**	**Comment:**
typTimetable	typTimetable	Data structure for the timetable program
.wDayLesson	ARRAY[1..7] of WORD	Bit-by-bit activation of the lesson units Bit 0 = Evening unit Bit 1 = 1st lesson unit Bit 2 = 2nd lesson unit etc.
.wDayLessonOnce	ARRAY[1..7] of WORD	Bit-by-bit **one-time** activation of the lesson units Bit 0 = Evening unit Bit 1 = 1st lesson unit Bit 2 = 2nd lesson unit etc.
.wEveningOn_hour	ARRAY[1..7] of WORD	Pull-in time evenings (hours)
.wEveningOff_hour	ARRAY[1..7] of WORD	Drop-out time evenings (hours)
.wEveningOn_minute	ARRAY[1..7] of WORD	Pull-in time evenings (minutes)
.wEveningOff_minute	ARRAY[1..7] of WORD	Drop-out time evenings (minutes)
.wSpecialFirstDay	ARRAY[1..3] of WORD	Start of the additional holiday period
.wSpecialLastDay	ARRAY[1..3] of WORD	End of the additional holiday period
.stRoomNr	STRING (13)	Room designation (max. 13 characters)
.bStatus	BYTE	Bit-by-bit display of the current status Bit 0 = Lead or follow-on time disabled/enabled Bit 1 = Timetable disabled/enabled Bit 2 = Holiday disabled/enabled Bit 3 = One-time switching program disabled/enabled

WAGO-I/O-PRO CAA Library Elements		
.bConfig	BYTE	Bit-by-bit display of the current configuration Bit 0 = Lesson times during the day Bit 1 = Lesson times during the night Bit 2 = One-time execution disabled/enabled
.wWeekdayOn_hour	ARRAY[1..15] of WORD	Pull-in time weekdays (hours)
.wWeekdayOff_hour	ARRAY[1..15] of WORD	Drop-out time weekdays (hours)
.wWeekdayOn_minute	ARRAY[1..15] of WORD	Pull-in time weekdays (minutes)
.wWeekdayOff_minute	ARRAY[1..15] of WORD	Drop-out time weekdays (minutes)
.wWeekendOn_hour	ARRAY[1..15] of WORD	Pull-in time weekends (hours)
.wWeekendOff_hour	ARRAY[1..15] of WORD	Drop-out time weekends (hours)
.wWeekendOn_minute	ARRAY[1..15] of WORD	Pull-in time weekends (minutes)
.wWeekendOff_minute	ARRAY[1..15] of WORD	Drop-out time weekends (minutes)
.wLeadTime	WORD	Lead time (minutes)
.wFollowupTime	WORD	Follow-up time (minutes)
Return value:	**Data type:**	**Comment:**
xSwitchChannel	BOOL	Switching output of the timetable program
bStatus	BYTE	Bit-by-bit display of the current status Bit 0 = Lead or follow-on time disabled/enabled Bit 1 = Timetable disabled/enabled Bit 2 = Holiday disabled/enabled Bit 3 = One-time switching program disabled/enabled
Graphical illustration:		

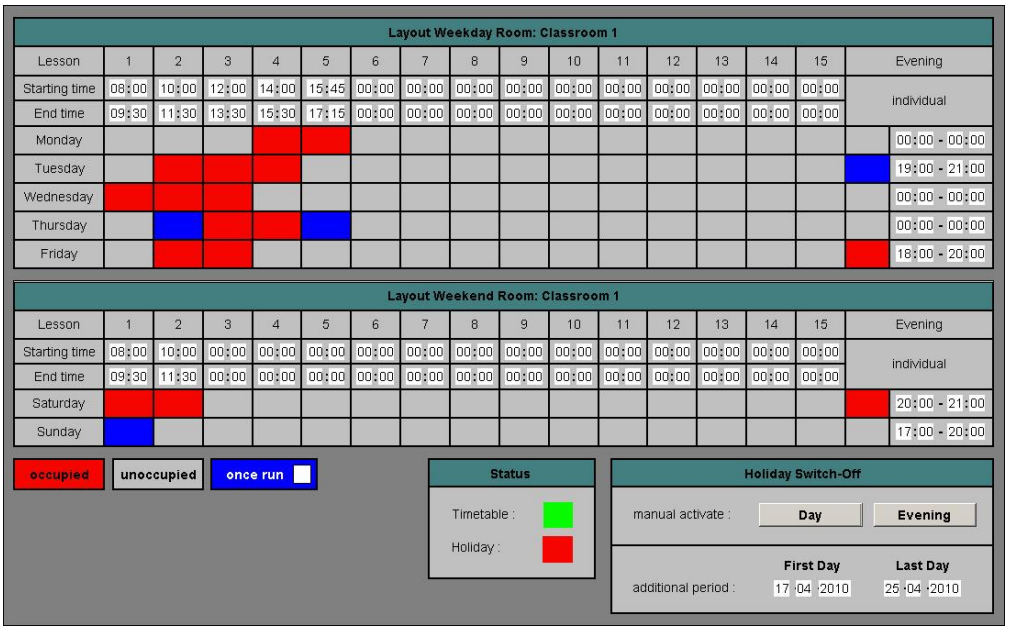


Structure of the "typTimetable" variable:	
typTimetable .wDayLesson .wDayLesson[1] = 48 .wDayLesson[2] = 28 .wDayLesson[3] = 14 .wDayLesson[4] = 24 .wDayLesson[5] = 13 .wDayLesson[6] = 7 .wDayLesson[7] = 0 .wDayLessonOnce .wEveningOn_hour .wEveningOn_hour[1] = 0 .wEveningOn_hour[2] = 19 .wEveningOn_hour[3] = 0 .wEveningOn_hour[4] = 0 .wEveningOn_hour[5] = 18 .wEveningOn_hour[6] = 20 .wEveningOn_hour[7] = 17 .wEveningOff_hour .wEveningOn_minute .wEveningOff_minute .wSpecialFirstDay .wSpecialLastDay .stRoomNr = 'Classroom 1' .bStatus = 2 .bConfig = 0	**Data structure of the timetable program**

Structure of the "typLesson" variable:	
typLesson .wWeekdayOn_hour .wWeekdayOn_hour[1] = 8 .wWeekdayOn_hour[2] = 10 .wWeekdayOn_hour[3] = 12 .wWeekdayOn_hour[4] = 14 .wWeekdayOn_hour[5] = 15 .wWeekdayOn_hour[6] = 0 .wWeekdayOn_hour[7] = 0 .wWeekdayOn_hour[8] = 0 .wWeekdayOn_hour[9] = 0 .wWeekdayOn_hour[10] = 0 .wWeekdayOn_hour[11] = 0 .wWeekdayOn_hour[12] = 0 .wWeekdayOn_hour[13] = 0 .wWeekdayOn_hour[14] = 0 .wWeekdayOn_hour[15] = 0 .wWeekdayOff_hour .wWeekdayOn_minute .wWeekdayOff_minute .wWeekendOn_hour .wWeekendOff_hour .wWeekendOn_minute .wWeekendOff_minute .wLeadTime = 10 .wFollowupTime = 5	**Data structure of the timetable program**

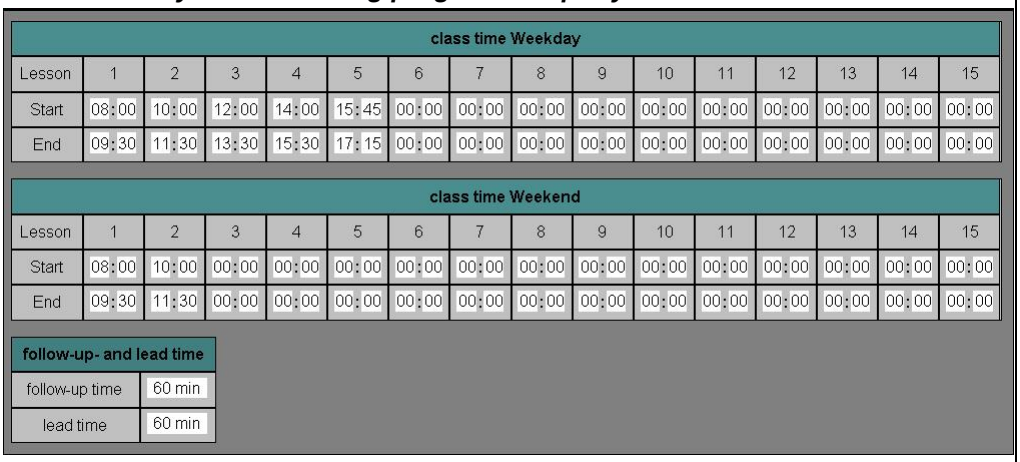
Configuration interfaces:

Status display of the FbTimetable:



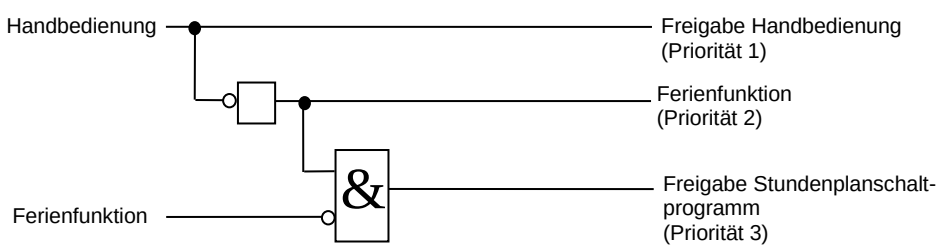
Color status: green = active; red = inactive; orange = lead-/ follow up time active; blue = once run active

Public holiday time switching program and party function:



Priority order:

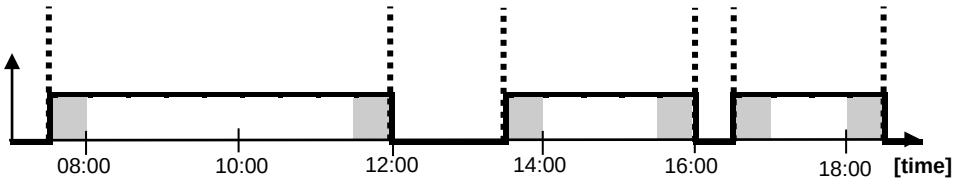
Priority order of the time switching functions:



Timetable switching program:

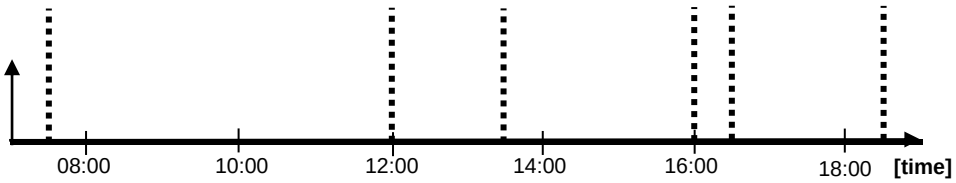
Example 1: Any weekday

Hour 1 (8:00 – 9:30): occupied
Hour 2 (10:00 – 11:30): occupied
Hour 4 (14:00 – 15:30): occupied
individual evenings (17:00 – 18:00): occupied
For all other times, the room is not occupied
Time before and after use 30 minutes.



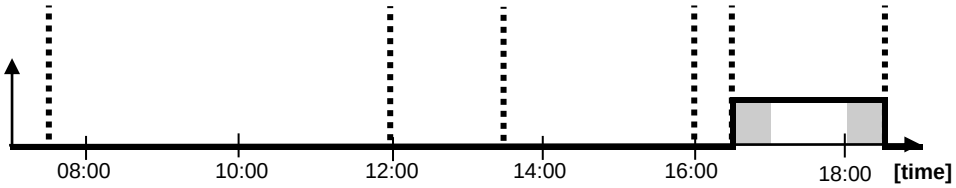
Example 2: Any holiday date active

Time schedule as in example 1
Day and evening activation OFF



Example 3: Any holiday date active

Time schedule as in example 1
Day activation OFF, evening activation ON



Function description:

The **FbTimetable** function block contains the time switching function for an occupancy schedule (timetable) of a room.

To prevent overlapping between time switching programs, priorities are assigned to the various time switching programs.

Priority of the time switching programs:

- Manual operation (priority 1)
- Holiday function (priority 2)
- Timetable switching program (priority 3)

The ***"xEnable"*** input is used to enable the **FbTimetable** function block.

The ***"dtActualTime"*** input is linked to the actual time. This time is used as the basis for the time switching program.

If the ***"xHoliday"*** input is TRUE, the holiday switching program is executed. The **FbHoliday** function block is used to identify holidays.

The ***"xManualOperation"*** input enables the manual mode. In manual mode, the time switching program is controlled by the ***"xManualOn"*** input.

The ***"typTimetable"*** structure variable contains all values for the time switching programs of the **FbTimetable** visualization element.

The ***"typLesson"*** structure variable contains all values for the time switching programs of the **LessonFromFbTimetable** visualization element.

The ***"xSwitchChannel"*** output is set when a time switching condition is met.

The ***"bStatus"*** output displays the current status of the function block.

Notes:

1.) The **SysRtcGetTime** function can be used to determine the current system time. This function is found in the **SysLibRtc.lib** library.

2.) For convenient configuration of the time switching programs, the ***"visuTimetable", "visuTimetable_Lesson"*** configuration interfaces are in the library. For the first interface the settings of the resolution are adjusted (display size: 914 Display Height: 567)

3.) The *"typTimetable"* and *"typLesson"* structures should be declared as RETAIN, so that the time switching conditions are retained after a controller reset.

4.) For 32-bit controllers, the *"typTimetable"* and *"typLesson"* structures should also be declared as PERISISTENT, so that time switching conditions are also retained after the download.

5.) As an option, the *"typTimetable"* and *"typLesson"* structures can be written to the file system of the 32-bit controller to save the time switching conditions. The **FbSaveVariable** function block is used to save the data.

Additional functions

Identification of Public Holidays (FbPublicHoliday)

WAGO-I/O-*PRO* CAA Library Elements		
Category:	Building Automation	
Name:	FbPublicHoliday	
Type:	Function ☐	Function block ☒ / Program ☐
Library name:	Scheduler_03.lib	
Applicable to:	Programmable fieldbus controllers	
Input parameter:	**Data type:**	**Comment:**
xEnable	BOOL	Activation of the function block Default value: TRUE
dtActualTime	DT	Input for current date and time
Input/output parameter:	**Data type:**	**Comment:**
typPublicHoliday	typPublic Holiday	Data structure for German public holidays.
.Neujahr	BOOL	New Year's Day
.Heilige_Drei_Koenige	BOOL	Epiphany, Three Kings
.Rosenmontag	BOOL	Rose Monday
.Aschermittwoch	BOOL	Ash Wednesday
.Karfreitag	BOOL	Good Day
.Ostersonntag	BOOL	Easter Sunday
.Ostermontag	BOOL	Easter Monday
.Maifeiertag	BOOL	May Day, Labor Day
.Christi_Himmelfahrt	BOOL	Ascension Day
.Pfingstsonntag	BOOL	Whit Sunday
.Pfingstmontag	BOOL	Whit Monday
.Fronleichnam	BOOL	Corpus Christi
.Mariae_Himmelfahrt	BOOL	Assumption
.Tag_der_Deutschen_ Einheit	BOOL	Day of German Unity
.Reformationstag	BOOL	Reformation Day
.Allerheiligen	BOOL	All Saints' Day
.Buss_und_Bettag	BOOL	Day of Prayer and Repentance
.Heiligabend	BOOL	Christmas Eve
.Erster_Weihnachtstag	BOOL	Christmas Day
.Zweiter_Weihnachtstag	BOOL	Second Day of Christmas
.Sylvester	BOOL	New Year's Eve

.SetSpecialPublic Holiday	WORD	Activation of the special public holidays bit by bit. Bit 0 = First free configurable public holiday Bit 1 = Second free configurable public holiday etc.
.SpecialPublicHoliday1 _day	WORD	First free configurable public holiday (day) Default value = 1
.SpecialPublicHoliday1 _month	WORD	First free configurable public holiday (month) Default value = 1
: :	WORD	etc.
.SpecialPublicHoliday5 _day	WORD	Fifth free configurable public holiday (day) Default value = 1
.SpecialPublicHoliday5 _month	WORD	Fifth free configurable public holiday (month) Default value = 1
Feedback value:	**Data type:**	**Comment:**
xPublicHoliday	BOOL	Indicates whether the current date is a selected public holiday

Structure of the "typPublicHoliday" variable:

```
⊟ typPublicHoliday
    .Neujahr = TRUE
    .Heilige_Drei_Koenige = FALSE
    .Rosenmontag = FALSE
    .Aschermittwoch = FALSE
    .Karfreitag = TRUE
    .Ostersonntag = FALSE
    .Ostermontag = TRUE
    .Maifeiertag = TRUE
    .Christi_Himmelfahrt = TRUE
    .Pfingstsonntag = FALSE
    .Pfingstmontag = TRUE
    .Fronleichnam = FALSE
    .Mariae_Himmelfahrt = FALSE
    .Tag_der_Deutschen_Einheit = TRUE
    .Reformationstag = FALSE
    .Allerheiligen = FALSE
    .Buss_und_Bettag = FALSE
    .Heiligabend = TRUE
    .Erster_Weihnachtstag = TRUE
    .Zweiter_Weihnachtstag = TRUE
    .Sylvester = TRUE
    .dummy = FALSE
    .SetSpecialPublicHoliday = 1
    .SpecialPublicHoliday1_day = 6
    .SpecialPublicHoliday1_month = 12
    .SpecialPublicHoliday2_day = 1
    .SpecialPublicHoliday2_month = 1
    .SpecialPublicHoliday3_day = 1
    .SpecialPublicHoliday3_month = 1
    .SpecialPublicHoliday4_day = 1
    .SpecialPublicHoliday4_month = 1
    .SpecialPublicHoliday5_day = 1
    .SpecialPublicHoliday5_month = 1
```

General public holidays in Germany

Freely selectable public holidays

Graphical display:

Configuration interface:

Public Holiday	Active
Neujahr	■
Heilige Drei Könige	□
Rosenmontag	□
Achermittwoch	□
Karfreitag	■
Ostersonntag	□
Ostermontag	■
Maifeiertag	■
Christi Himmelfahrt	■
Pfingstsonntag	□
Pfingstmontag	■
Fronleichnam	□
Mariä Himmelfahrt	□
Tag der Deutschen Einheit	■
Reformationstag	□
Allerheiligen	□
Buß- und Bettag	□
Heiligabend	■
Erster Weihnachtstag	■
Zweiter Weihnachtstag	■
Sylvester	■
06 - 12	■
01 - 01	□
01 - 01	□
01 - 01	□
01 - 01	□

Function description:
The **FbPublicHoliday** function block is used for detecting public holidays. Only those public holidays are detected, which are activated in the ***"typPublicHoliday"*** structure. The ***"xEnable"*** input is used for activating the **FbPublicHoliday** function block. The ***"dtActualTime"*** input is linked to the current time. This time is used as the basis for determining the public holidays. The *"typPublicHoliday"* structure variable contains the German public holidays. As well as the fixed defined public holidays, there is also a facility for defining up to five public holidays. When the **FbPublicHoliday** function block detects a public holiday, the ***"xPublicHoliday"*** output is set to TRUE. **Notes:** 1.) The current system time of the Ethernet controller 750-841 can be determined by means of the **SysRtcGetTime** function. This function can be found in the **SysLibRtc** library. 2.) The "*typPublicHoliday*" structure should be declared as RETAIN so that the selected public holidays are retained following a controller reset. 3.) In the case of the Ethernet controller 750-841, the "*typPublicHoliday*" structure should also be declared as PERSISTENT, so that the selected public holidays are also retained after a download. 4.) As an option, the "*typPublicHoliday*" structure can be written to the file system of the Ethernet controller 750-841 in order to back-up the selected public holidays. The data are saved by means of the **FbSaveVariable** function block. 5.) The "***visuPublicHoliday***" configuration interface for conveniently selecting public holidays is also provided in the library.

Holiday Determination (FbHoliday)

WAGO-I/O-PRO CAA Library Elements			
Category:	Building Automation		
Name:	FbHoliday		
Type:	Function ☐	Function block ☒	Program ☐
Name of library:	Scheduler_03.lib		
Applicable to:	All programmable fieldbus controllers		

Input parameter:	**Data type:**	**Comment:**
xEnable	BOOL	Activation of the function block Default setting: TRUE
dtActualTime	DT	Input for the actual date and time

Input/output parameter:	**Data type:**	**Comment:**
typHoliday	typHoliday	Data structure for the holidays
.wFirstDay	ARRAY[1..3,1..20] of WORD	First day of the holiday
.wLastDay	ARRAY[1..3,1..20] of WORD	Last day of the holiday
.strComment	ARRAY[1..20] of STRING(20)	Comments with max. 20 characters

Return value:	**Data type:**	**Comment:**
xHoliday	BOOL	Displays whether the current data is a holiday

Structure of the "typHoliday" variable:

```
⊟····.typHoliday
    ⊞····.wFirstDay
    ⊞····.wLastDay
    ⊟····.strComment
        ·····.strComment[1] = 'spring'
        ·····.strComment[2] = 'summer'
        ·····.strComment[3] = 'autumn'
        ·····.strComment[4] = 'winter'
        ·····.strComment[5] = ''
        ·····.strComment[6] = ''
        ·····.strComment[7] = ''
        ·····.strComment[8] = ''
        ·····.strComment[9] = ''
        ·····.strComment[10] = ''
        ·····.strComment[11] = ''
        ·····.strComment[12] = ''
        ·····.strComment[13] = ''
        ·····.strComment[14] = ''
        ·····.strComment[15] = ''
        ·····.strComment[16] = ''
        ·····.strComment[17] = ''
        ·····.strComment[18] = ''
        ·····.strComment[19] = ''
        ·····.strComment[20] = ''
```

WAGO-I/O-PRO CAA Library Elements

Graphical illustration:

Configuration interface:

	Comment	First Day	Last Day		Comment	First Day	Last Day
1.	spring	05-03-2010	15-03-2010	11.		00-00-0000	00-00-0000
2.	summer	03-07-2010	09-08-2010	12.		00-00-0000	00-00-0000
3.	autumn	21-10-2010	28-10-2010	13.		00-00-0000	00-00-0000
4.	winter	22-12-2010	07-01-2011	14.		00-00-0000	00-00-0000
5.		00-00-0000	00-00-0000	15.		00-00-0000	00-00-0000
6.		00-00-0000	00-00-0000	16.		00-00-0000	00-00-0000
7.		00-00-0000	00-00-0000	17.		00-00-0000	00-00-0000
8.		00-00-0000	00-00-0000	18.		00-00-0000	00-00-0000
9.		00-00-0000	00-00-0000	19.		00-00-0000	00-00-0000
10.		00-00-0000	00-00-0000	20.		00-00-0000	00-00-0000

Function description:

The **FbHoliday** function block is used to identify holidays. Holiday dates are specified by the entries in the ***"typHoliday"*** structure.

The ***"xEnable"*** input is used to enable the **FbHoliday** function block.

The ***"dtActualTime"*** input is linked to the actual time. This time is used as the basis for computing the holidays.

The ***"typHoliday"*** structure variable contains the holiday data and its description as comments. The holiday times can be freely selected. Up to 20 different holiday dates can be entered.

If the **FbHoliday** function block identifies a holiday date, the ***"xHoliday"*** output is set to TRUE.

Notes:

1.) The **SysRtcGetTime** function can be used to determine the current system time. This function is found in the **SysLibRtc.lib** library.

2.) The *"typHoliday"* structure should be declared as RETAIN, so that the selection of holidays is retained after a controller reset.

3.) For 32-bit controllers, the *"typHoliday"* structure should also be declared as PERSISTENT, so that the selection of holidays is also retained after the download.

4.) As an option, the *"typHoliday"* structure can be written to the file system of the 32-bit controller to save the selected holidays. The **FbSaveVariable** function block is used to save the data.

5.) For convenient selection of the holidays, the ***"visuHoliday"*** configuration interface is saved in the library.

General Value Specifications (FbSetScheduleValue)

WAGO-I/O-PRO CAA Library Elements		
Category:	Building Automation	
Name:	FbHoliday	
Type:	Function ☐	Function block ☒ / Program ☐
Name of library:	Scheduler_03.lib	
Applicable to:	All programmable fieldbus controllers	
Input parameter:	**Data type:**	**Comment:**
typScheduleStatus	typSchedule Status	Data structure for the status
.xEnable	BOOL	Module enable
.xManualOperation	BOOL	Switch manually
.xManualOn	BOOL	Manual ON
.xPublicHoliday	BOOL	Holiday
.xParty	BOOL	Party function
.xSwitchChannel	BOOL	Switching Output
.axWeekly	Array[1..10] of BOOL	Status for the ten weekly time-switching programs
.axSpecialPeriod	Array[1..5] of BOOL	Status for the five period time-switching programs
typScheduleValue	typSchedule Value	Data structure for the values of the time switching programs
.arWeekly	Array[1..10] of REAL	alues of the 10 weekly time switching programs
.arSpecial	Array[1..5] of REAL	Values of the 5 12-month time switching programs
.rHoliday	REAL	Value for the holiday/holiday program
.rParty	REAL	Value for the party function
.Default	REAL	Default
Return value:	**Data type:**	**Comment:**
rValue	REAL	Currently released value

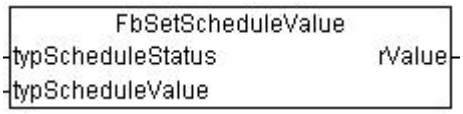
Structure of the "typScheduleValue" variable:

```
typScheduleValue
    .arWeekly
        .arWeekly[1] = 21
        .arWeekly[2] = 21
        .arWeekly[3] = 21
        .arWeekly[4] = 21
        .arWeekly[5] = 21
        .arWeekly[6] = 21
        .arWeekly[7] = 21
        .arWeekly[8] = 21
        .arWeekly[9] = 21
        .arWeekly[10] = 21
    .arSpecial
        .arSpecial[1] = 21
        .arSpecial[2] = 21
        .arSpecial[3] = 21
        .arSpecial[4] = 21
        .arSpecial[5] = 21
    .rHoliday = 17
    .rParty = 21
    .rDefault = 17
```

Structure of the "typScheduleStatus" variable:

```
typScheduleStatus
    .xEnable = TRUE
    .xManualOperation = FALSE
    .xManualOn = FALSE
    .xPublicHoliday = FALSE
    .xParty = FALSE
    .xSwitchChannel = FALSE
    .axWeekly
        .axWeekly[0] = FALSE
        .axWeekly[1] = FALSE
        .axWeekly[2] = FALSE
        .axWeekly[3] = FALSE
        .axWeekly[4] = FALSE
        .axWeekly[5] = FALSE
        .axWeekly[6] = FALSE
        .axWeekly[7] = FALSE
        .axWeekly[8] = FALSE
        .axWeekly[9] = FALSE
        .axWeekly[10] = FALSE
    .axSpecialPeriod
        .axSpecialPeriod[0] = FALSE
        .axSpecialPeriod[1] = FALSE
        .axSpecialPeriod[2] = FALSE
        .axSpecialPeriod[3] = FALSE
        .axSpecialPeriod[4] = FALSE
        .axSpecialPeriod[5] = FALSE
```

Graphical illustration:



Configuration interface:

Value specification of the party and holiday programs

Public holiday	ON	OFF	Status	Value	Party function	Switch-on-time	Status	Value	Default
☐	00 : 00	00 : 00	☐	17.0	☐	60	☐	21.0	17.0

Value specification for the 12-month time switching program

(for adding to the respective interface of the FbScheduler module)

Value	Default
21.0	17.0
21.0	
21.0	
21.0	
21.0	

Value specification for the weekly time switching program

(for adding to the respective interface of the FbScheduler module)

Value	Default
21.0	17.0
21.0	
21.0	
21.0	
21.0	
21.0	
21.0	
21.0	
21.0	

Function description:

The **FbSetScheduleValue** function block is used to flexibly assign values of the active time switching program to the outputs of the module. If several time switching programs are active, the values are assigned by priority. The order of the priorities is identical to that of the **FbScheduler** module. If no time switching program is active, the default value ("typScheduleStatus.rDefault" variable) is released to the output.

The first structure variable ***"typScheduleStatus"*** on the input of the **FbSetScheduleValue** module must be connected to the ***"typScheduleStatus"*** output of the **FbScheduler** module. The value on the output of the module is controlled by the variables of the structure.

The second structure variable ***"typScheduleValue"*** on the input contains all values of the individual time switching programs. This structure can be linked to the respective configuration interfaces.

The current value of the respective active status is displayed on the ***"rValue"*** output of the **FbSetScheduleValue** module .

Notes:

1.) The **SysRtcGetTime** function can be used to determine the current system time. This function is found in the **SysLibRtc.lib** library.

2.) The *"typScheduleValue"* structure should be declared as RETAIN, so that the selection of holidays is retained after a controller reset.

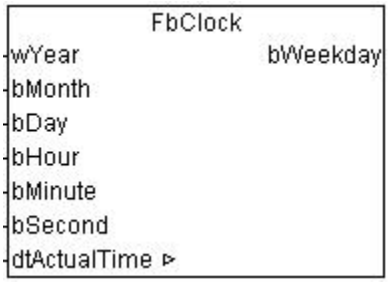
3.) For 32-bit controllers, the *"typScheduleValue"* structure should also be declared as PERSISTENT, so that the selection of holidays is also retained after the download.

4.) As an option, the *"typScheduleValue"* structure can be written to the file system of the 32-bit controller to save the selected holidays. The **FbSaveVariable** function block is used to save the data.

5.) For convenient selection of the values, the ***visuScheduler_Holiday-PartyValue", „visuSetScheduleValue_SpecialPeriods"*** and ***„visuSetScheduleValue_Weekly"*** configuration interfaces are saved in the library.

Simulating the clock function (FbClock)

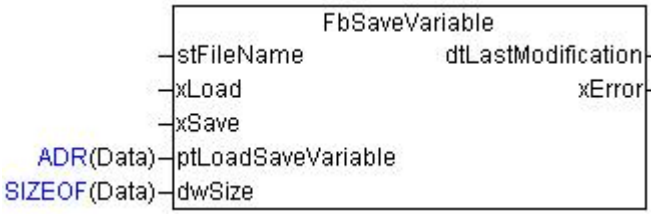
WAGO-I/O-*PRO* CAA Library Elements			
Category:	Building Automation		
Name:	FbClock		
Type:	Function ☐	Function block ☒	Program ☐
Library name:	Scheduler_03.lib		
Applicable to:	Programmable fieldbus controllers		
Input parameter:	**Data type:**	**Comment:**	
wYear	WORD	Input signal year	
bMonth	BYTE	Input signal month	
bDay	BYTE	Input signal day	
bHour	BYTE	Input signal hour	
bMinute	BYTE	Input signal minute	
bSecond	BYTE	Input signal second	
Input/output parameter:	**Data type:**	**Comment:**	
dtActualTime	DT	Output of current date and time	
Feedback Value:	**Data type:**	**Comment:**	
bWeekday	BYTE	Indication of the day of the week: 1 = Monday 2 = Tuesday 3 = Wednesday 4 = Thursday 5 = Friday 6 = Saturday 7 = Sunday	
Graphical display:			



Function description:
The **FbClock** function block enables a clock function to be simulated. The time deviation is ca. 10 seconds per day. It is therefore necessary to synchronize the function block from an external real-time clock (recommended once a day). The date is synchronized by means of the inputs ***"wYear"***, ***"bMonth"*** and ***"bDay"***, and the time is synchronized by means of the inputs ***"bHour"***, ***"bMinute"*** and ***"bSecond"***. When one of the values on the inputs *"bHour"*, *"bMinute"* or *"bSecond"* changes, the time "*dtActualTime"* is initialized with all three input values. A change of value at the inputs *"wYear", "bMonth"* and *"bDay"* has the effect that only the value that has changed overwrites the current value of *"dtActualTime"*. If no input value changes, the time of the *"dtActualTime"* variables continues to run and thus replicates the clock function. Leap years are taken into account when determining the current date. The day of the week is determined based on the date, and is indicated at the output ***"bWeekday"***. **Note:** 1) If zero appears at the *"wYear"* input, the time is not initialized. 2) The "*dtActualTime*" variable should be declared as RETAIN so that the time is maintained at the last determined time following a controller reset.

Storing data in the file system (FbSaveVariable)

WAGO-I/O-*PRO* CAA Library Elements			
Category:	Building Automation		
Name:	FbSaveVariable		
Type:	Function ☐	Function block ☒	Program ☐
Library name:	Scheduler_03.lib		
Applicable to:	Ethernet fieldbus-controller 750-841		
Library used:	SysLibFile.lib		
Input parameter:	**Data type:**	**Comment:**	
stFileName	STRING (80)	Name of file	
xLoad	BOOL	Load the variable from the file system	
xSave	BOOL	Save the variable in the file system	
ptLoadSaveVariable	POINTER TO DWORD	Start address of the variable	
dwSize	DWORD	Size of variable [bytes]	
Feedback value:	**Data type:**	**Comment:**	
dtLastModification	DT	Date of last modification	
xError	BOOL	Error when loading the file.	
Graphical display:			



Function description:

The **FbSaveVariable** function module is used for saving and reading out data in the file system of the Ethernet field bus controller 750-841.

The name of the file is specified at the ***"stFileName"*** input. The address of the variable to be saved is specified at the ***"ptLoadSaveVariable"*** input. The number of bytes to be written or read must also be specified at the ***"dwSize"*** input. The SIZEOF operator can be used as an aid to determining the size of the data.

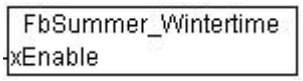
The value of the variables is read from the file system with a positive edge on the ***"xLoad"*** input.

The value of the variables is written to the file system with a positive edge on the ***"xSave"*** input.

The ***"dtLastModification"*** output indicates when the variable was last saved.

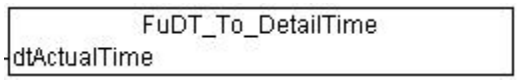
The "***xError"*** output indicates whether an error has occurred when saving or loading the file.

Daylight saving (FbSummer_Wintertime)

WAGO-I/O-*PRO* CAA Library Elements			
Category:	Building Automation		
Name:	FbSummer_Wintertime		
Type:	Function ☐	Function block ☒	Program ☐
Library name:	Scheduler_03.lib		
Applicable to:	Ethernet fieldbus controller 750-841		
Library used:	SysLibRtc.lib		
Input parameter:	**Data type:**	**Comment:**	
xEnable	BOOL	Activation of the function block Default Value: TRUE	
Graphical display:			
			
Function description:			
The **FbSummer_Wintertime** function block puts the time forward from 2 to 3 am on the last Sunday in March, and puts it back from 3 to 2 am on the last Sunday in October. The ***"xEnable"*** input is used to activate the summer / winter time changeover.			

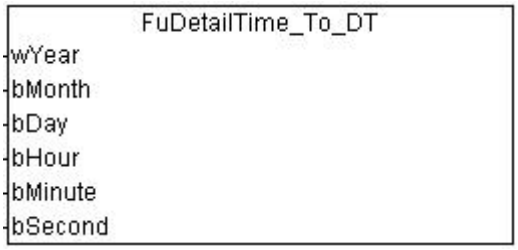
Conversions

Split-up data type DT (FuDT_To_DetailTime)

WAGO-I/O-*PRO* CAA Library Elements			
Category:	Building Automation		
Name:	FuDT_To_DetailTime		
Type:	Function ☒	Function block ☐	Program ☐
Library name:	Scheduler_03.lib		
Applicable to:	Programmable fieldbus controllers		
Input parameter:	**Data type:**	**Comment:**	
dtActualTime	DT	Actual time Data type: **„Date_and_Time“**	
Feedback value:	**Data type:**	**Comment:**	
FuDT_To_DetailTime	typDetailTime	Data structure for the individual time elements	
.wYear	WORD	Year	
.bMonth	BYTE	Month	
.bDay	BYTE	Day	
.bHour	BYTE	Hour	
.bMinute	BYTE	Minute	
.bSecond	BYTE	Second	
.bWeekday	BYTE	Weekday 1 = Monday 2 = Tuesday 3 = Wednesday 4 = Thursday 5 = Friday 6 = Saturday 7 = Sunday	
Graphical display:			
			
Function description:			
The **FuDT_To_DetailTime** function converts data type DT to the ***"typDetailTime"*** structure. Each time element is shown individually in the *"typDetailTime"* structure.			

Put together data type DT (FuDetailTime_To_DT)

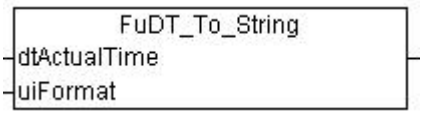
WAGO-I/O-*PRO* CAA Library Elements		
Category:	Building Automation	
Name:	FuDetailTime_To_DT	
Type:	Function ☒	Function block ☐ Program ☐
Library name:	Scheduler_03.lib	
Applicable to:	Programmable fieldbus controllers	
Input parameter:	**Data type:**	**Comment:**
wYear	WORD	Input signal year
bMonth	BYTE	Input signal month
bDay	BYTE	Input signal day
bHour	BYTE	Input signal hour
bMinute	BYTE	Input signal minute
bSecond	BYTE	Input signal second
Feedback value:	**Data type:**	**Comment:**
FuDetailTime_To_DT	DT	Output of date and time in the ***"Date_and_Time"*** data type
Graphical display:		



Function description:

The **FuDetailTime_To_DT** function converts the individual time elements to data type DT.

Conversion of the data type DT to string (FuDT_TO_String)

WAGO-I/O-*PRO* CAA Library Elements			
Category:	Building Automation		
Name:	FuDT_To_String		
Type:	Function ☒	Function block ☐	Program ☐
Library name:	Scheduler_03.lib		
Applicable to:	Programmable fieldbus controllers		
Input parameter:	**Data type:**	**Comment:**	
dtActualTime	DT	Actual time Data type: **„Date_and_Time“**	
uiFormat	UINT	Selection of the time format 0 = DD.MM.YYYY – hh:mm:ss 1 = DD/MM/YYYY – hh:mm:ss 2 = DD-MM-YYYY – hh:mm:ss 3 = MM/DD/YYYY – hh:mm:ss 4 = MM-DD-YYYY – hh:mm:ss 5 = YYYY/MM/DD – hh:mm:ss 6 = YYYY-MM-DD – hh:mm:ss	
Feedback value:	**Data type:**	**Comment:**	
FuDT_To_String	DT	Output of date and time in the ***„String“*** data type	
Graphical display::			
			
Function description:			
The function **FuDT_To_String** converts the input parameter ***„dtActualTime“*** into string. This variable contains the current time and the date. With the input ***“uiFormat"*** it is possible to choose different forms of representation.			

RTC Module

Date and Time via the RTC Module (FbTime_RTC_Modul)

WAGO-I/O-PRO CAA Library Elements		
Category:	Building Automation	
Name:	FbTime_RTC_Modul	
Type:	Function ☐	Function block X ☐ / Program ☐
Name of library:	Scheduler_03.lib	
Applicable to:	All programmable fieldbus controllers	
Input parameter:	**Data type:**	**Comment:**
dtSetTime	DT	Time default Data type: **"Date_and_Time"**
xSetTime	BOOL	Applying the time from the input dtSetTime
Output parameters:	**Data type:**	**Comment:**
dtActualTime	DT	Actual time from the RTC module
bStatus	BYTE	Bit-by-bit status display: Bit 0 = Antenna signal level Bit 1 = Backup voltage too low or clock failed Bit 2 = Antenna signal not available or faulty Bit 3 = Memory not addressable
Graphical illustration:		

Function description:

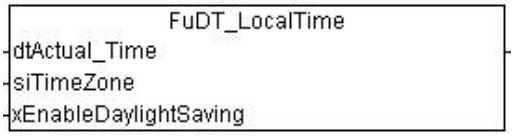
The **FbTime_RTC_Modul** function block determines the date and time from the RTC module 750-640 and makes the data available to the ***"dtActualTime"*** output. The second ***"bStatus"*** output is used to diagnose the module. The ***"dtSetTime"*** input is used to define a time at which a positive edge is applied to the ***"xSetTime"*** input.

Note:

1) The inputs to manually set the time are only needed if no antenna is connected.

Time Conversion (FuDT_LocalTime)

WAGO-I/O-PRO CAA Library Elements		
Category:	Building Automation	
Name:	FuDT_LocalTime	
Type:	Function ☒	Function block ☐ Program ☐
Name of library:	Scheduler_03.lib	
Applicable to:	All programmable fieldbus controllers	
Input parameter:	**Data type:**	**Comment:**
dtActualTime	DT	Input of the date and time in the ***"Date_and_Time"*** data type
siTimeZone	SINT	Time zone
xEnableDaylightSaving	BOOL	Summer/winter switching disabled/enabled
Return value:	**Data type:**	**Comment:**
FuDT_LocalTime	DT	Return value of the date and time in the ***"Date_and_Time"*** data type
Graphical illustration:		



Function description:
The function **FuDT_LocalTime** is used to convert the UTC time at the **"dtActualTime"** input on the local time. Therefore, the time zone (input: **"siTimeZone"**) and/or summer/winter time changeover (input: **"xEnableDaylightSaving"**) are considered. The return value supplies the converted time.

WAGO Kontakttechnik GmbH & Co. KG
P.O.box 2880 • D-32385 Minden, Germany
Hansastraße 27 • D-32423 Minden, Germany
Phone: 05 71/8 87 – 0
Fax: 05 71/8 87 – 1 69
E-Mail: info@wago.com

Internet: http://www.wago.com