

WAGO→I/O→SYSTEM 750

Bibliotheken für die Gebäudeautomation

Bausteinbeschreibungen für Zeitschaltprogramme

Letzte Änderung: 25.04.2019



Copyright © 2019 by WAGO Kontakttechnik GmbH & Co. KG
Alle Rechte vorbehalten.

WAGO Kontakttechnik GmbH & Co. KG

Hansastraße 27
D-32423 Minden

Tel.: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69
E-Mail: info@wago.com
Web: <http://www.wago.com>

Technischer Support

Tel.: +49 (0) 571/8 87 – 4 45 55
Fax: +49 (0) 571/8 87 – 84 45 55
E-Mail: tcba@wago.com

Es wurden alle erdenklichen Maßnahmen getroffen, um die Richtigkeit und Vollständigkeit der vorliegenden Dokumentation zu gewährleisten. Da sich Fehler, trotz aller Sorgfalt, nie vollständig vermeiden lassen, sind wir für Hinweise und Anregungen jederzeit dankbar.

Wir weisen darauf hin, dass die im Dokument verwendeten Soft- und Hardwarebezeichnungen und Markennamen der jeweiligen Firmen im Allgemeinen einem Warenzeichenschutz, Markenschutz oder patentrechtlichem Schutz unterliegen.

WAGO-I/O-PRO CAA Bibliothek für Zeitschaltprogramme

Inhalt

Wichtige Erläuterungen	4
Urheberschutz	4
Personalqualifikation	4
Bestimmungsgemäßer Gebrauch	4
Gültigkeitsbereich	5
Zeitschaltprogramme	6
Wochenzeitschaltprogramm (FbScheduleWeekly)	6
Jahreszeitschaltprogramm (FbScheduleSpecialPeriod)	9
Feiertags-, Jahres- und Wochenzeitschaltprogramm (FbScheduler)	12
Stundenplanprogramm (FbTimetable)	22
Zusatzfunktionen	28
Ermittlung der Feiertage (FbPublicHoliday)	28
Ermittlung der Ferien (FbHoliday)	32
Allgemeine Wertvorgabe (FbSetScheduleValue)	35
Nachbildung der Uhrenfunktion (FbClock)	39
Speichern von Daten im File-System (FbSaveVariable)	41
Sommer- / Winterzeitumstellung (FbSummer_Wintertime)	42
Konvertierungen	43
Aufsplittung des Datentyps DT (Fu_DT_To_DetailTime)	43
Zusammenfassen zum Datentyp DT (FuDetailTime_To_DT)	44
Umwandlung des Datentyps DT zum String (FuDT_TO_String)	45
RTC-Klemme	46
Datum und Zeit mittels der RTC-Klemme (FbTime_RTC_Modul)	46
Zeitumrechnung (FuDT_LocalTime)	47

Wichtige Erläuterungen

Um dem Anwender eine schnelle Installation und Inbetriebnahme der beschriebenen Geräte zu gewährleisten, ist es notwendig, die nachfolgenden Hinweise und Erläuterungen sorgfältig zu lesen und zu beachten.

Urheberschutz

Dieses Dokument, einschließlich aller darin befindlichen Abbildungen, ist urheberrechtlich geschützt. Jede Weiterverwendung dieses Dokumentes, die von den urheberrechtlichen Bestimmungen abweicht, ist nicht gestattet.

Die Reproduktion, Übersetzung in andere Sprachen, sowie die elektronische und fototechnische Archivierung und Veränderung bedarf der schriftlichen Genehmigung der WAGO Kontakttechnik GmbH & Co. KG, Minden. Zuwiderhandlungen ziehen einen Schadenersatzanspruch nach sich.

Die WAGO Kontakttechnik GmbH & Co. KG behält sich Änderungen, die dem technischen Fortschritt dienen, vor.

Alle Rechte für den Fall der Patenterteilung oder des Gebrauchsmusterschutzes sind der WAGO Kontakttechnik GmbH & Co. KG vorbehalten. Fremdprodukte werden stets ohne Vermerk auf Patentrechte genannt. Die Existenz solcher Rechte ist daher nicht auszuschließen.

Personalqualifikation

Der in diesem Dokument beschriebene Produktgebrauch richtet sich ausschließlich an Fachkräfte mit einer Ausbildung in der SPS-Programmierung, Elektrofachkräfte oder von Elektrofachkräften unterwiesene Personen, die außerdem mit den geltenden Normen vertraut sind. Für Fehlhandlungen und Schäden, die an WAGO-Produkten und Fremdprodukten durch Missachtung der Informationen dieses Dokumentes entstehen, übernimmt die WAGO Kontakttechnik GmbH & Co. KG keine Haftung.

Bestimmungsgemäßer Gebrauch

Die Komponenten werden ab Werk für den jeweiligen Anwendungsfall mit einer festen Hard- und Softwarekonfiguration ausgeliefert. Änderungen sind nur im Rahmen der in dem Dokument aufgezeigten Möglichkeiten zulässig. Alle anderen Veränderungen an der Hard- oder Software, sowie der nicht bestimmungsgemäße Gebrauch der Komponenten, bewirken den Haftungsausschluss der WAGO Kontakttechnik GmbH & Co. KG.

Wünsche an eine abgewandelte bzw. neue Hard- oder Softwarekonfiguration richten Sie bitte an WAGO Kontakttechnik GmbH & Co. KG.

Gültigkeitsbereich

Dieser Anwendungshinweis basiert auf die genannte Hard- und Software der jeweiligen Hersteller sowie auf die zugehörige Dokumentation. Daher gilt dieser Anwendungshinweis nur für die beschriebene Installation. Neue Hard- und Softwareversionen erfordern eventuell eine geänderte Handhabung.

Beachten Sie die ausführliche Beschreibung in den jeweiligen Handbüchern.

Zeitschaltprogramme

Wochenzeitschaltprogramm (FbScheduleWeekly)

WAGO-I/O-PRO CAA Elemente der Bibliothek		
Kategorie:	Gebäudetechnik	
Name:	FbScheduleWeekly	
Typ:	Funktion <input type="checkbox"/>	Funktionsblock <input checked="" type="checkbox"/> Programm <input type="checkbox"/>
Name der Bibliothek:	Scheduler_03.lib	
Anwendbar für:	Alle programmierbaren Feldbus-Controller	
Eingangsparameter:	Datentyp:	Kommentar:
xEnable	BOOL	Aktivierung des Funktionsbausteins Voreinstellung: TRUE
dtActualTime	DT	Eingang für das aktuelle Datum und die Uhrzeit
Ein-Ausgangsparameter:	Datentyp:	Kommentar:
typScheduleWeekly	typScheduleWeekly	Datenstruktur für das Wochezeitschaltprogramm
.ON_hour	WORD	Einschaltzeit (Stunden)
.ON_minute	WORD	Einschaltzeit (Minuten)
.OFF_hour	WORD	Ausschaltzeit (Stunden)
.OFF_minute	WORD	Ausschaltzeit (Minuten)
.Weekday	WORD	Wochentag (Bitcodiert) Bit 0 = Montag Bit 1 = Dienstag Bit 2 = Mittwoch Bit 3 = Donnerstag Bit 4 = Freitag Bit 5 = Samstag Bit 6 = Sonntag
.Active	WORD	Zeitschaltbedingung erfüllt
Rückgabewert:	Datentyp:	Kommentar:
xSwitchChannel	BOOL	Schaltausgang des Wochenzeitschaltprogramms
Grafische Darstellung:		
<div><div>FbScheduleWeekly</div><div><div>xEnable</div><div>xSwitchChannel</div></div><div><div>dtActualTime</div><div>typScheduleWeekly ▶</div></div></div>		

Strukturaufbau der Variable „typScheduleWeekly“

```

typScheduleWeekly
  .ON_hour = 0
  .ON_minute = 0
  .OFF_hour = 0
  .OFF_minute = 0
  .Weekday = 0
  .Active = 0
    
```

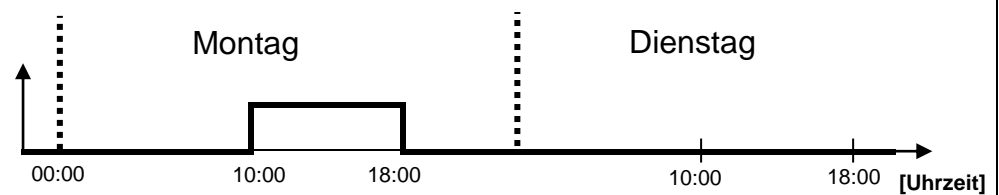
**Datenstruktur des
Wochenzeitschaltprogramms**

Konfigurationsoberfläche:

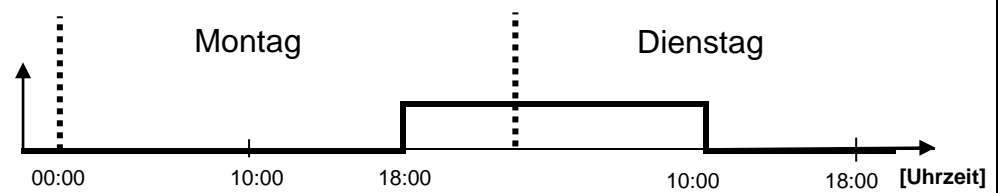
ON	OFF	Weekday							Status
00:00	00:00	MO	TU	WE	TH	FR	SA	SU	<input type="checkbox"/>

Zeitliches Verhalten:

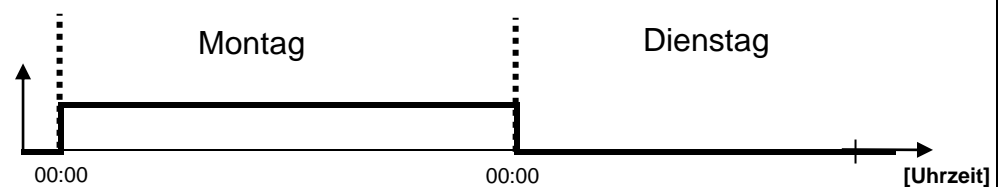
Beispiel 1: Montag aktiv, Ein: 10:00 Uhr, Aus: 18:00 Uhr



Beispiel 2: Montag aktiv, Ein: 18:00 Uhr, Aus: 10:00 Uhr



Beispiel 3: Montag aktiv, Ein: 00:00 Uhr, Aus: 00:00 Uhr



Funktionsbeschreibung:

Der Funktionsbaustein **FbScheduleWeekly** beinhaltet die Funktionalität eines Wochenzeitschaltprogramms.

Der Eingang „**xEnable**“ dient dazu, den Funktionsbaustein zu aktivieren.

Der Eingang „**dtActualTime**“ wird mit der aktuellen Zeit verknüpft. Diese Zeit dient als Grundlage für das Zeitschaltprogramm.

Die Strukturvariable „**typScheduleWeekly**“ beinhaltet die Werte für das Zeitschaltprogramm des Bausteins **FbScheduleWeekly**.

Der Ausgang „**xSwitchChannel**“ wird gesetzt, wenn die Zeitschaltbedingung erfüllt ist.

Hinweise:

- 1.) Die aktuelle Systemzeit kann über die Funktion **SysRtcGetTime** ermittelt werden. Diese Funktion befindet sich in der Bibliothek **SysLibRtc.lib**.
- 2.) Zur komfortablen Konfiguration des Zeitschaltprogramms ist in der Bibliothek die Konfigurationsoberfläche „**visuScheduleWeekly**“ hinterlegt.
- 3.) Die Struktur „**typScheduleWeekly**“ sollte als RETAIN deklariert werden, damit die Zeitschaltbedingung nach einem Controller-Reset erhalten bleibt.
- 4.) Bei den 32 Bit-Controller sollte die Struktur „**typScheduleWeekly**“ zusätzlich als PERISISTENT deklariert werden, damit die Zeitschaltbedingung auch nach dem Download erhalten bleibt.
- 5.) Optional kann zur Speicherung der Zeitschaltbedingung die Struktur „**typScheduleWeekly**“ ins File-System des 32 Bit-Controllers geschrieben werden. Das Speichern der Daten wird über den Funktionsbaustein **FbSaveVariable** realisiert.

Jahreszeitschaltprogramm (FbScheduleSpecialPeriod)

WAGO-I/O-PRO CAA Elemente der Bibliothek		
Kategorie:	Gebäudetechnik	
Name:	FbScheduleSpecialPeriod	
Typ:	Funktion <input type="checkbox"/> Funktionsblock <input checked="" type="checkbox"/> Programm <input type="checkbox"/>	
Name der Bibliothek:	Scheduler_03.lib	
Anwendbar für:	Alle programmierbaren Feldbus-Controller	
Eingangsparameter:		
	Datentyp:	Kommentar:
xEnable	BOOL	Aktivierung des Funktionsbausteins Voreinstellung: TRUE
dtActualTime	DT	Eingang für das aktuelle Datum und die Uhrzeit
Ein-Ausgangsparameter:		
	Datentyp:	Kommentar:
typScheduleSpecialPeriod	typScheduleSpecialPeriod	Datenstruktur für das Jahreszeitschaltprogramm
.FirstDay_day	WORD	Erster Tag (Tag)
.FirstDay_month	WORD	Erster Tag (Monat)
.FirstDay_year	WORD	Erster Tag (Jahr)
.LastDay_day	WORD	Letzter Tag (Tag)
.LastDay_month	WORD	Letzter Tag (Monat)
.LastDay_year	WORD	Letzter Tag (Jahr)
.ON_hour	WORD	Einschaltzeit (Stunden)
.ON_minute	WORD	Einschaltzeit (Minuten)
.OFF_hour	WORD	Ausschaltzeit (Stunden)
.OFF_minute	WORD	Ausschaltzeit (Minuten)
.Weekday	WORD	Wochentag (Bitcodiert) Bit 0 = Montag Bit 1 = Dienstag Bit 2 = Mittwoch Bit 3 = Donnerstag Bit 4 = Freitag Bit 5 = Samstag Bit 6 = Sonntag
.Active	WORD	Zeitschaltbedingung erfüllt
Rückgabewert:		
	Datentyp:	Kommentar:
xSwitchChannel	BOOL	Schaltausgang des Jahreszeitschaltprogramms
Grafische Darstellung:		
<div style="border: 1px solid black; padding: 10px; width: fit-content; margin: 0 auto;"> <div style="text-align: center; margin-bottom: 10px;">FbScheduleSpecialPeriod</div> <div style="display: flex; justify-content: space-between;"> xEnable xSwitchChannel </div> <div style="margin-top: 10px;"> dtActualTime typScheduleSpecialPeriod ▶ </div> </div>		

Strukturaufbau der Variable „typScheduleSpecialPeriod“:

```

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
  
```

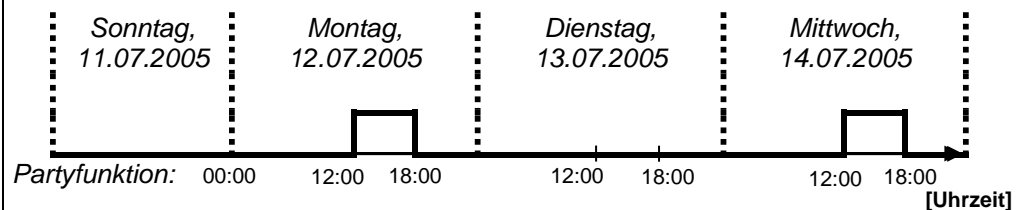
**Datenstruktur des
Jahreszeitschaltprogramms**

Konfigurationsoberfläche:

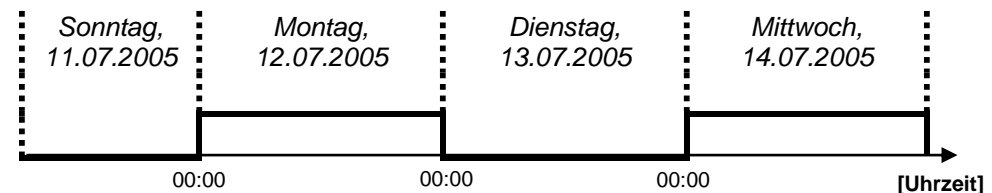
First day	Last day	ON	OFF	Weekday							Status
01-01-2007	01-01-2007	00:00	00:00	MO	TU	WE	TH	FR	SA	SU	<input type="checkbox"/>

Zeitliches Verhalten:

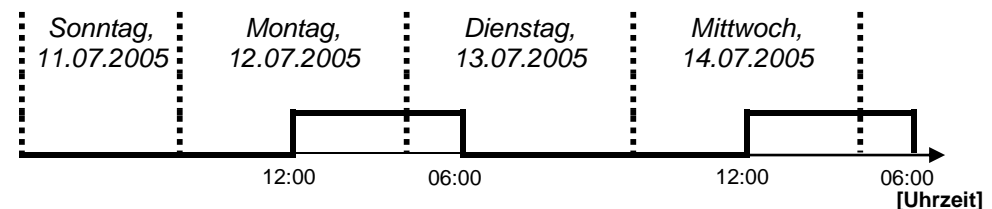
*Beispiel 1: Erster Tag: 12.07.2005, Letzter Tag: 14.07.2005
Montag und Mittwoch sind aktiviert.
Ein: 12:00 Uhr, Aus: 18:00 Uhr*



*Beispiel 2: Erster Tag: 12.07.2005, Letzter Tag: 14.07.2005
Montag und Mittwoch sind aktiviert.
Ein: 00:00 Uhr, Aus: 00:00 Uhr*



*Beispiel 3: Erster Tag: 12.07.2005, Letzter Tag: 14.07.2005
Montag und Mittwoch sind aktiviert.
Ein: 12:00 Uhr, Aus: 06:00 Uhr*



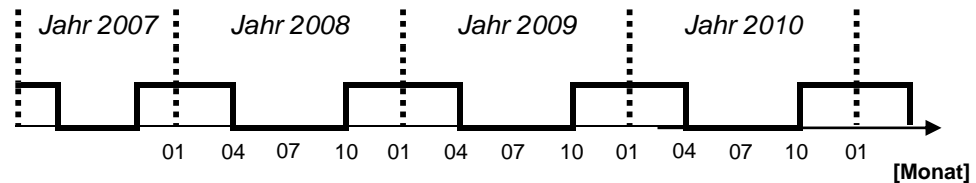
Zeitliches Verhalten:

Beispiel 4: jährlich wiederholender Termin (als Jahresangabe ist dann eine Null einzutragen)

Erster Tag: 01.10.0000, Letzter Tag: 01.04.0000

Alle Wochentage sind aktiviert.

Ein: 00:00 Uhr, Aus: 00:00 Uhr

**Funktionsbeschreibung:**

Der Funktionsbaustein **FbScheduleSpecialPeriod** beinhaltet die Funktionalität eines Jahreszeitschaltprogramms.

Der Eingang „**xEnable**“ dient dazu, das Zeitschaltprogramm zu aktivieren.

Der Eingang „**dtActualTime**“ wird mit der aktuellen Zeit verknüpft. Diese Zeit dient als Grundlage für das Zeitschaltprogramm.

Die Strukturvariable „**typScheduleSpecialPeriod**“ beinhaltet die Zeitschaltbedingung des Jahreszeitschaltprogramms.

Der Ausgang „**xSwitchChannel**“ wird gesetzt, wenn die Zeitschaltbedingung erfüllt ist.

Hinweis:

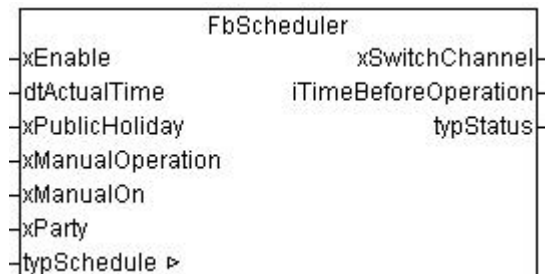
- 1.) Die aktuelle Systemzeit kann über die Funktion **SysRtcGetTime** ermittelt werden. Diese Funktion befindet sich in der Bibliothek **SysLibRtc.lib**.
- 2.) Zur komfortablen Konfiguration des Zeitschaltprogramms ist in der Bibliothek die Konfigurationsoberfläche „**visuScheduleSpecialPeriod**“ hinterlegt.
- 3.) Die Struktur „**typScheduleSpecialPeriod**“ sollte als RETAIN deklariert werden, damit die Zeitschaltbedingung auch nach einem Controller-Reset erhalten bleibt.
- 4.) Bei den 32 Bit-Controllern sollte die Struktur „**typScheduleSpecialPeriod**“ zusätzlich als PERISISTENT deklariert werden, damit die Zeitschaltbedingung auch nach dem Download erhalten bleibt.
- 5.) Optional kann zur Speicherung der Zeitschaltbedingung die Struktur „**typScheduleSpecialPeriod**“ ins File-System des 32 Bit-Controllers geschrieben werden. Das Speichern der Daten wird über den Funktionsbaustein **FbSaveVariable** realisiert.

Feiertags-, Jahres- und Wochenzeitschaltprogramm (FbScheduler)

WAGO-I/O-PRO CAA Elemente der Bibliothek			
Kategorie:	Gebäudetechnik		
Name:	FbScheduler		
Typ:	Funktion <input type="checkbox"/>	Funktionsblock <input checked="" type="checkbox"/>	Programm <input type="checkbox"/>
Name der Bibliothek:	Scheduler_03.lib		
Anwendbar für:	Alle programmierbaren Feldbus-Controller		
Eingangsparameter:	Datentyp:	Kommentar:	
xEnable	BOOL	Aktivierung des Funktionsbausteins Voreinstellung: TRUE	
dtActualTime	DT	Eingang für die aktuelle Uhrzeit	
xPublicHoliday	BOOL	Aktivierung des Feiertagsschaltprogramms	
xManualOperation	BOOL	Handbedienmodus	
xManualOn	BOOL	Schalten im Handbedienmodus	
xParty	BOOL	Start bzw. Neustart der Partyfunktion	
Ein-Ausgangsparameter:	Datentyp:	Kommentar:	
typSchedule	typSchedule	Datenstruktur für die Zeitschaltprogramme	
.ScheduleWeekly	ARRAY[1..10] of typSchedule Weekly	Zehn Wochenzeitschaltprogramme	
.ON_hour	WORD	Einschaltzeit (Stunden)	
.ON_minute	WORD	Einschaltzeit (Minuten)	
.OFF_hour	WORD	Ausschaltzeit (Stunden)	
.OFF_minute	WORD	Ausschaltzeit (Minuten)	
.Weekday	WORD	Wochentag (Bitcodiert) Bit 0 = Montag Bit 1 = Dienstag Bit 2 = Mittwoch Bit 3 = Donnerstag Bit 4 = Freitag Bit 5 = Samstag Bit 6 = Sonntag	
.Active	WORD	Zeitschaltbedingung erfüllt	
.ScheduleSpecialPeriods	ARRAY[1..5] of typSchedule SpecialPeriod	Fünf Jahreszeitschaltprogramme	
.FirstDay_day	WORD	Erster Tag (Tag)	
.FirstDay_month	WORD	Erster Tag (Monat)	
.FirstDay_year	WORD	Erster Tag (Jahr)	
.LastDay_day	WORD	Letzter Tag (Tag)	
.LastDay_month	WORD	Letzter Tag (Monat)	
.LastDay_year	WORD	Letzter Tag (Jahr)	
.ON_hour	WORD	Einschaltzeit (Stunden)	
.ON_minute	WORD	Einschaltzeit (Minuten)	

.OFF_hour	WORD	Ausschaltzeit (Stunden)
.OFF_minute	WORD	Ausschaltzeit (Minuten)
.Weekday	WORD	Wochentag (Bitcodiert) Bit 0 = Montag Bit 1 = Dienstag Bit 2 = Mittwoch Bit 3 = Donnerstag Bit 4 = Freitag Bit 5 = Samstag Bit 6 = Sonntag
.Active	WORD	Zeitschaltbedingung erfüllt
.PublicHolidayON_hour	WORD	Einschaltzeit am Feiertag (Stunden)
.PublicHolidayON_minute	WORD	Einschaltzeit am Feiertag (Minuten)
.PublicHolidayOFF_hour	WORD	Ausschaltzeit am Feiertag (Stunden)
.PublicHolidayOFF_minute	WORD	Ausschaltzeit am Feiertag (Minuten)
.PublicHoliday_active	WORD	Zeitschaltbedingung für Feiertage erfüllt
.Party_length	WORD	Dauer der Partyfunktion [min]
.Party_active	WORD	Partyfunktion aktiv
.SwitchChannel	WORD	Kopie des Schaltausgangs
.StatusScheduler	WORD	Status des Zeitschaltprogramms Bit 0 = Baustein aktiviert Bit 1 = Handbedienung aktiv Bit 2 = Status Handbedienung Bit 3 = Feiertag Bit 4 = Party funktion aktiv Bit 5 = Sonderschaltprogramm aktiv
.TimeBeforeOperation	INT	Anzeige der Zeit vor Nutzungsbeginn(+) Verbleibende Nutzungszeit (-) [min]

Rückgabewert:	Datentyp:	Kommentar:
xSwitchChannel	BOOL	Schaltausgang des Zeitschaltprogramms
iTimeBeforeOperation	INT	Anzeige der Zeit vor Nutzungsbeginn(+) Verbleibende Nutzungszeit (-) [min]
typScheduleStatus	typSchedule Status	Datenstruktur für den Status
.xEnable	BOOL	Freigabe des Bausteins
.xManualOperation	BOOL	Manuell schalten
.xManualOn	BOOL	Manuell ein
.xPublicHoliday	BOOL	Feiertag
.xParty	BOOL	Partyfunktion
.xSwitchChannel	BOOL	Schaltausgang
.axWeekly	Array[1..10] of BOOL	Status der 10 Wochenzeitschaltprogramme
.axSpecialPeriod	Array[1..5] of BOOL	Status der 5 Jahreszeitschaltprogramme

Grafische Darstellung:


Strukturaufbau der Variable „typSchedule“:

```

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]
├── 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]
├── PublicHolidayON_hour = 0
├── PublicHolidayON_minute = 0
├── PublicHolidayOFF_hour = 0
├── PublicHolidayOFF_minute = 0
├── PublicHoliday_active = 0
├── Party_length = 60
├── Party_active = 0
├── SwitchChannel = 0
├── StatusScheduler = 0
└── TimeBeforeOperation = 0
    
```

Wochenzeitschaltprogramme

Jahreszeitschaltprogramme

Feiertagszeitschaltprogramm

Partyfunktion

Statusinformationen

Strukturaufbau der Variable „typScheduleStatus“:

```

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
    
```

Statusanzeige des FbScheduler:

Enable	Channel ON	Manual operation	Manual on	Time before operation
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0 min

Feiertagszeitschaltprogramm und Partyfunktion:

Public holiday	ON	OFF	Status	Party function	Switch-on-time	Status
<input type="checkbox"/>	00 : 00	00 : 00	<input type="checkbox"/>	<input type="checkbox"/>	60	<input type="checkbox"/>

Jahreszeitschaltprogramme:

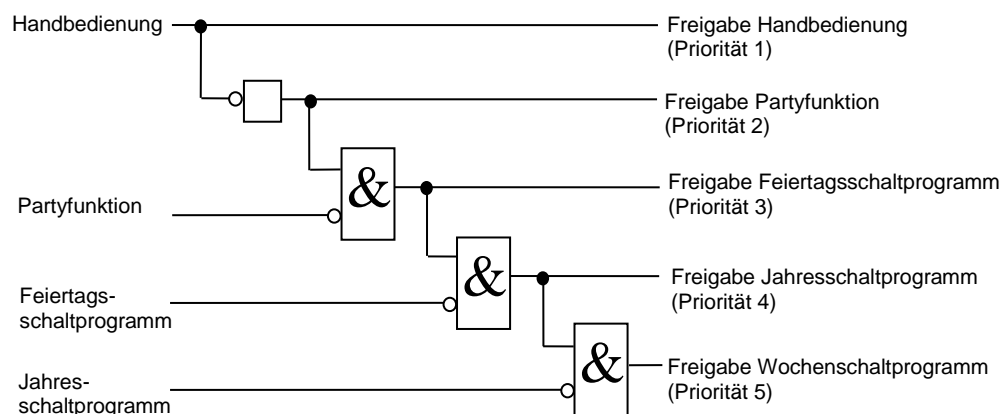
First day	Last day	ON	OFF	Weekday							Status
01-01-2007	01-01-2007	00:00	00:00	MO	TU	WE	TH	FR	SA	SU	<input type="checkbox"/>
01-01-2007	01-01-2007	00:00	00:00	MO	TU	WE	TH	FR	SA	SU	<input type="checkbox"/>
01-01-2007	01-01-2007	00:00	00:00	MO	TU	WE	TH	FR	SA	SU	<input type="checkbox"/>
01-01-2007	01-01-2007	00:00	00:00	MO	TU	WE	TH	FR	SA	SU	<input type="checkbox"/>
01-01-2007	01-01-2007	00:00	00:00	MO	TU	WE	TH	FR	SA	SU	<input type="checkbox"/>

Wochenzeitschaltprogramme:

ON	OFF	Weekday							Status
00 : 00	00 : 00	MO	TU	WE	TH	FR	SA	SU	<input type="checkbox"/>
00 : 00	00 : 00	MO	TU	WE	TH	FR	SA	SU	<input type="checkbox"/>
00 : 00	00 : 00	MO	TU	WE	TH	FR	SA	SU	<input type="checkbox"/>
00 : 00	00 : 00	MO	TU	WE	TH	FR	SA	SU	<input type="checkbox"/>
00 : 00	00 : 00	MO	TU	WE	TH	FR	SA	SU	<input type="checkbox"/>
00 : 00	00 : 00	MO	TU	WE	TH	FR	SA	SU	<input type="checkbox"/>
00 : 00	00 : 00	MO	TU	WE	TH	FR	SA	SU	<input type="checkbox"/>
00 : 00	00 : 00	MO	TU	WE	TH	FR	SA	SU	<input type="checkbox"/>
00 : 00	00 : 00	MO	TU	WE	TH	FR	SA	SU	<input type="checkbox"/>
00 : 00	00 : 00	MO	TU	WE	TH	FR	SA	SU	<input type="checkbox"/>

Prioritätenreihenfolge:

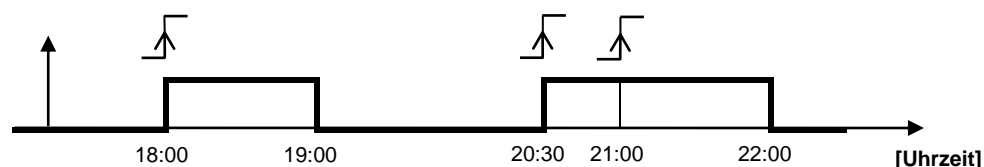
Prioritätenreihenfolge der Zeitschaltfunktionen:



Zeitliches Verhalten:

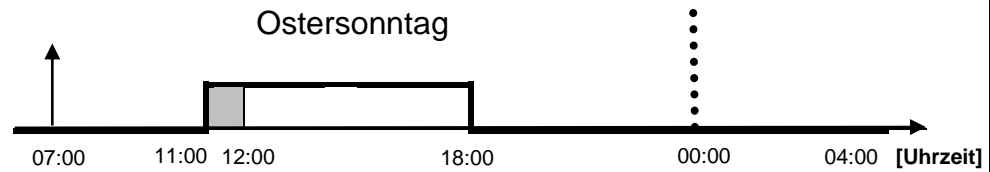
Partyfunktion:

Beispiel: Partyfunktion 60 Minuten.

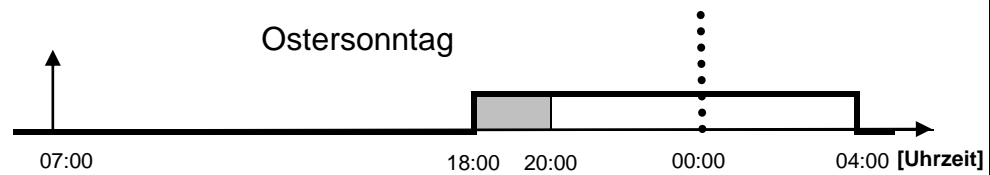


Feiertagszeitschaltprogramm:

*Beispiel 1: Ostersonntag (Aktiv), Ein: 12:00 Uhr, Aus: 18:00 Uhr,
Zeit vor Nutzungsbeginn 60 Minuten.*



*Beispiel 2: Ostersonntag (Aktiv), Ein: 20:00 Uhr, Aus: 4:00 Uhr,
Zeit vor Nutzungsbeginn 120 Minuten.*



Beispiel 3: Ostersonntag (Aktiv), Ein: 00:00 Uhr, Aus: 00:00 Uhr

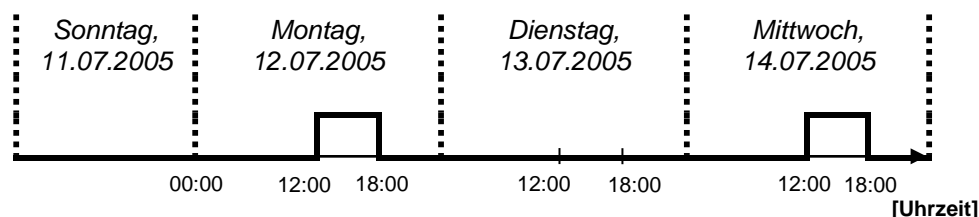


*Beispiel 4: Ostersonntag (Aktiv), Ein: 24:00 Uhr, Aus: 24:00 Uhr
Ostermontag (Aktiv), Ein: 00:00 Uhr, Aus: 00:00 Uhr*

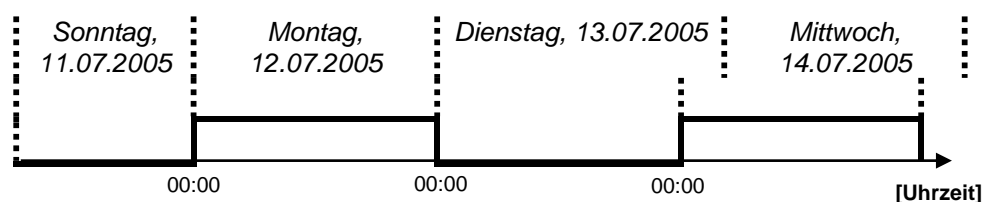


Jahreszeitschaltprogramm:

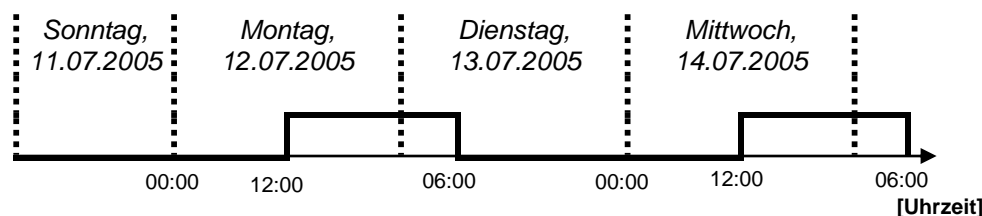
*Beispiel 1: Erster Tag: 12.07.2005, Letzter Tag: 14.07.2005
Montag und Mittwoch sind aktiviert.
Ein: 12:00 Uhr, Aus: 18:00 Uhr*



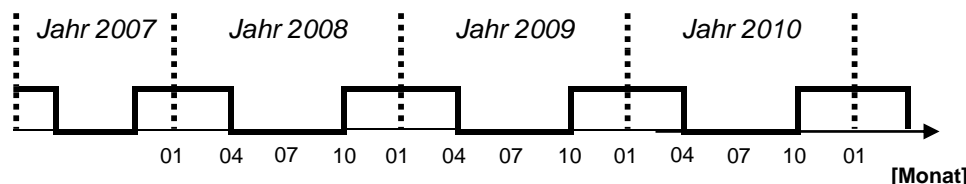
*Beispiel 2: Erster Tag: 12.07.2005, Letzter Tag: 14.07.2005
Montag und Mittwoch sind aktiviert.
Ein: 00:00 Uhr, Aus: 00:00 Uhr*



*Beispiel 3: Erster Tag: 12.07.2005, Letzter Tag: 14.07.2005
Montag und Mittwoch sind aktiviert.
Ein: 12:00 Uhr, Aus: 06:00 Uhr*

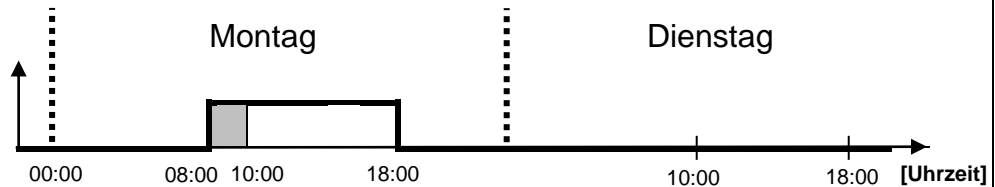


*Beispiel 4: jährlich wiederholender Termin (als Jahresangabe ist dann eine Null einzutragen)
Erster Tag: 01.10.0000, Letzter Tag: 01.04.0000
Alle Wochentage sind aktiviert.
Ein: 00:00 Uhr, Aus: 00:00 Uhr*

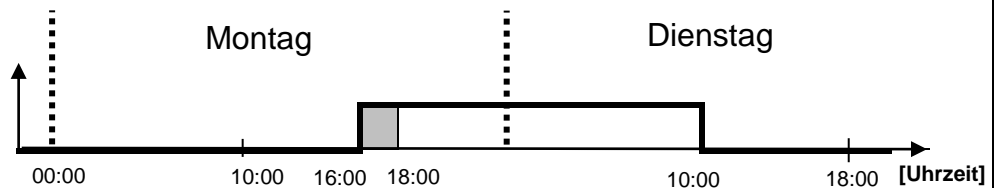


Wochenzeitschaltprogramm:

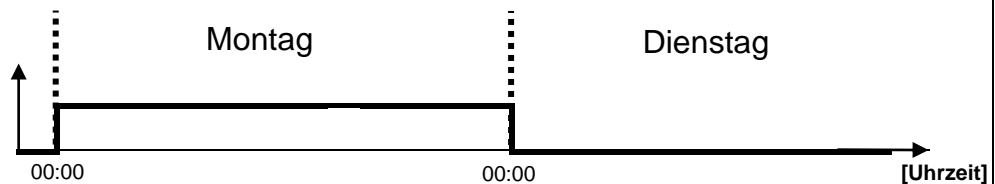
*Beispiel 1: Montag aktiv, Ein: 10:00 Uhr, Aus: 18:00 Uhr,
Zeit vor Nutzungsbeginn 120 Minuten.*



*Beispiel 2: Montag aktiv, Ein: 18:00 Uhr, Aus: 10:00 Uhr,
Zeit vor Nutzungsbeginn 120 Minuten.*



Beispiel 3: Montag aktiv, Ein: 00:00 Uhr, Aus: 00:00 Uhr


Funktionsbeschreibung:

Der Funktionsbaustein **FbScheduler** beinhaltet verschiedene Zeitschaltfunktionen wie z.B. Jahreszeitschaltprogramm, Feiertagszeitschaltprogramm und Wochenzeitschaltprogramm.

Damit es nicht zu Überschneidungen zwischen den Zeitschaltprogrammen kommt, sind den verschiedenen Zeitschaltprogrammen Priorität zugeordnet.

Priorität der Zeitschaltprogramme:

- Handbedienung (Priorität 1)
- Partyfunktion (Priorität 2)
- Feiertagszeitschaltprogramm (Priorität 3)
- Jahreszeitschaltprogramme (Priorität 4)
- Wochenzeitschaltprogramme (Priorität 5)

Der Eingang „**xEnable**“ dient dazu, den Funktionsbaustein **FbScheduler** zu aktivieren.

Der Eingang „**dtActualTime**“ wird mit der aktuellen Zeit verknüpft. Diese Zeit dient als Grundlage für die Zeitschaltprogramme.

Wenn der Eingang „**xPublicHoliday**“ TRUE ist, wird das Feiertagsschaltprogramm ausgeführt. Die Erkennung der Feiertage kann durch den Funktionsbaustein **FbPublicHoliday** realisiert werden.

Der Eingang „**xManualOperation**“ aktiviert den Handbedienmodus. Im Handbedienmodus wird das Zeitschaltprogramm vom Eingang „**xManualOn**“ gesteuert.

Eine positive Flanke am Eingang „**xParty**“ startet die Partyfunktion. Bei der Partyfunktion wird der Ausgang des Zeitschaltprogramms für eine einstellbare Zeit gesetzt. Erkennt der Funktionsblock innerhalb dieser Zeit eine weitere positive Flanke, wird die eingestellte Zeit neu gestartet.

Die Strukturvariable „**typSchedule**“ beinhaltet alle Werte für die Zeitschaltprogramme des Bausteins **FbScheduler**.

Der Ausgang „**xSwitchChannel**“ wird gesetzt, wenn eine Zeitschaltbedingung erfüllt ist.

Der Ausgang „**iTimeBeforeOperation**“ gibt die Zeit vor Nutzungsbeginn (+) bzw. die verbleibende Nutzungszeit (-) der Zeitschaltbedingungen an. Die Berechnung der Zeit vor Nutzungsbeginn beginnt immer um 00:00 Uhr.

Die Strukturvariable „**typScheduleStatus**“ zeigt den aktuellen Status des Bausteins **FbScheduler** und der verschiedenen Zeitschaltprogramme an.

Hinweise:

- 1.) Die aktuelle Systemzeit kann über die Funktion **SysRtcGetTime** ermittelt werden. Diese Funktion befindet sich in der Bibliothek **SysLibRtc.lib**.
- 2.) Zur komfortablen Konfiguration der Zeitschaltprogramme sind in der Bibliothek die Konfigurationsoberflächen „**visuScheduler_Status**“, „**visuScheduler_SpecialPeriods**“, „**visuScheduler_Weekly**“ und „**visuScheduler_HolidayParty**“ hinterlegt.
- 3.) Die Struktur „**typSchedule**“ sollte als RETAIN deklariert werden, damit die Zeitschaltbedingungen nach einem Controller-Reset erhalten bleiben.
- 4.) Bei den 32 Bit-Controllern sollte die Struktur „**typSchedule**“ zusätzlich als PERISISTENT deklariert werden, damit die Zeitschaltbedingungen auch nach dem Download erhalten bleiben.
- 5.) Optional kann zur Speicherung der Zeitschaltbedingungen die Struktur „**typSchedule**“ ins File-System des 32 Bit-Controllers geschrieben werden. Das Speichern der Daten wird über den Funktionsbaustein **FbSaveVariable** realisiert.

Stundenplanprogramm (FbTimetable)

WAGO-I/O-PRO CAA Elemente der Bibliothek			
Kategorie:	Gebäudetechnik		
Name:	FbTimetable		
Typ:	Funktion <input type="checkbox"/>	Funktionsblock <input checked="" type="checkbox"/>	Programm <input type="checkbox"/>
Name der Bibliothek:	Scheduler_03.lib		
Anwendbar für:	Alle programmierbaren Feldbus-Controller		
Eingangsparameter:	Datentyp:	Kommentar:	
xEnable	BOOL	Aktivierung des Funktionsbausteins Voreinstellung: TRUE	
dtActualTime	DT	Eingang für die aktuelle Uhrzeit	
xManualOperation	BOOL	Handbedienmodus	
xManualOn	BOOL	Schalten im Handbedienmodus	
xHoliday	BOOL	Ferien Ja/Nein	
Ein-Ausgangsparameter:	Datentyp:	Kommentar:	
typTimetable	typTimetable	Datenstruktur für das Stundenplanprogramm	
.wDayLesson	ARRAY[1..7] of WORD	Bitweise Aktivierung der Unterrichts- einheiten Bit 0 = Abendeinheit Bit 1 = 1. Unterrichtseinheit Bit 2 = 2. Unterrichtseinheit usw.	
.wDayLessonOnce	ARRAY[1..7] of WORD	Bitweise einmalige Aktivierung der Unterrichts-einheiten Bit 0 = Abendeinheit Bit 1 = 1. Unterrichtseinheit Bit 2 = 2. Unterrichtseinheit usw.	
.wEveningOn_hour	ARRAY[1..7] of WORD	Einschaltzeit abends (Stunden)	
.wEveningOff_hour	ARRAY[1..7] of WORD	Ausschaltzeit abends (Stunden)	
.wEveningOn_minute	ARRAY[1..7] of WORD	Einschaltzeit abends (Minuten)	
.wEveningOff_minute	ARRAY[1..7] of WORD	Ausschaltzeit abends (Minuten)	
.wSpecialFirstDay	ARRAY[1..3] of WORD	Beginn der zusätzlichen Ferienperiode	
.wSpecialLastDay	ARRAY[1..3] of WORD	Ende der zusätzlichen Ferienperiode	
.stRoomNr	STRING(13)	Bezeichnung des Raumes (max. 13 Zeichen)	

WAGO-I/O-PRO CAA Elemente der Bibliothek		
.bStatus	BYTE	Bitweise Anzeige des aktuellen Status Bit 0 = Vor- bzw. Nachlaufzeit in/aktiv Bit 1 = Stundenplan in/aktiv Bit 2 = Ferien in/aktiv Bit 3 = einmal Schaltprogramm in/aktiv
.bConfig	BYTE	Bitweise Anzeige der aktuellen Konfiguration Bit 0 = Unterrichtszeiten während des Tages Bit 1 = Unterrichtszeiten in der Nacht Bit 2 = einmalige Ausführung in/aktiv
.wWeekdayOn_hour	ARRAY[1..15] of WORD	Einschaltzeit wochentags (Stunden)
.wWeekdayOff_hour	ARRAY[1..15] of WORD	Ausschaltzeit wochentags (Stunden)
.wWeekdayOn_minute	ARRAY[1..15] of WORD	Einschaltzeit wochentags (Minuten)
.wWeekdayOff_minute	ARRAY[1..15] of WORD	Ausschaltzeit wochentags (Minuten)
.wWeekendOn_hour	ARRAY[1..15] of WORD	Einschaltzeit Wochenende (Stunden)
.wWeekendOff_hour	ARRAY[1..15] of WORD	Ausschaltzeit Wochenende (Stunden)
.wWeekendOn_minute	ARRAY[1..15] of WORD	Einschaltzeit Wochenende (Minuten)
.wWeekendOff_minute	ARRAY[1..15] of WORD	Ausschaltzeit Wochenende (Minuten)
.wLeadTime	WORD	Vorlaufzeit (Minuten)
.wFollowupTime	WORD	Nachlaufzeit (Minuten)
Rückgabewert:		
xSwitchChannel	BOOL	Schaltausgang des Stundenplanprogramms
bStatus	BYTE	Bitweise Anzeige des aktuellen Status Bit 0 = Vor- bzw. Nachlaufzeit in/aktiv Bit 1 = Stundenplan in/aktiv Bit 2 = Ferien in/aktiv Bit 3 = einmal Schaltprogramm in/aktiv
Grafische Darstellung:		
 <pre> graph LR subgraph FbTimetable direction TB xEnable dtActualTime xManualOperation xManualOn xHoliday typTimetable typLesson end xSwitchChannel bStatus </pre>		

Strukturaufbau der Variable „typTimetable“:

```

└─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
  
```

**Datenstruktur des
Stundenplanprogramms**

Strukturaufbau der Variable „typLesson“:

```

└─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
  
```

**Datenstruktur des
Stundenplanprogramms**

Konfigurationsoberflächen:

Statusanzeige des FbTimetable:

Layout Weekday Room: Classroom 1																
Lesson	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Evening
Starting time	08:00	10:00	12:00	14:00	15:45	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	individual
End time	09:30	11:30	13:30	15:30	17:15	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	
Monday																00:00 - 00:00
Tuesday																19:00 - 21:00
Wednesday																00:00 - 00:00
Thursday																00:00 - 00:00
Friday																18:00 - 20:00

Layout Weekend Room: Classroom 1																
Lesson	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Evening
Starting time	08:00	10:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	individual
End time	09:30	11:30	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	
Saturday																20:00 - 21:00
Sunday																17:00 - 20:00

occupied

unoccupied

once run

Status

Timetable: ■

Holiday: ■

Holiday Switch-Off

manual activate: Day Evening

First Day

additional period: 17.04.2010

Last Day

25.04.2010

Farbe Status: grün = aktiv; rot = inaktiv; orange = Vor-/ Nachlaufzeit aktiv; blue = einmalige Ausführung aktiv

Feiertagszeitschaltprogramm und Partyfunktion:

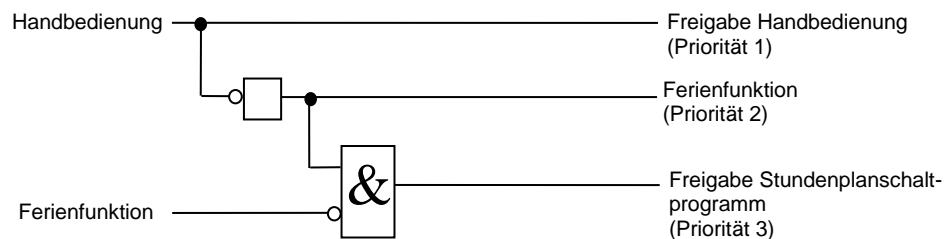
class time Weekday															
Lesson	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Start	08:00	10:00	12:00	14:00	15:45	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00
End	09:30	11:30	13:30	15:30	17:15	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00

class time Weekend															
Lesson	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Start	08:00	10:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00
End	09:30	11:30	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00

follow-up- and lead time	
follow-up time	60 min
lead time	60 min

Prioritätenreihenfolge:

Prioritätenreihenfolge der Zeitschaltfunktionen:



Stundenplanschaltprogramm:

Beispiel 1: beliebiger Wochentag

Stunde 1 (8:00 – 9:30): belegt

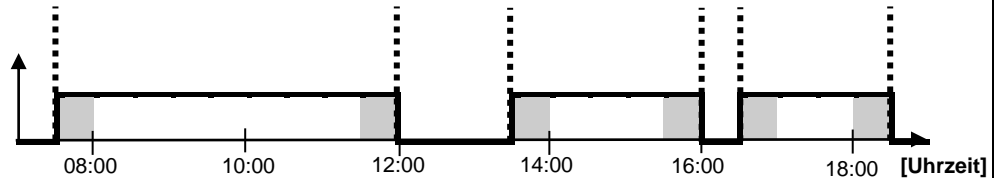
Stunde 2 (10:00 – 11:30): belegt

Stunde 4 (14:00 – 15:30): belegt

individuell abends (17:00 – 18:00): belegt

zu allen anderen Zeiten ist der Raum nicht belegt

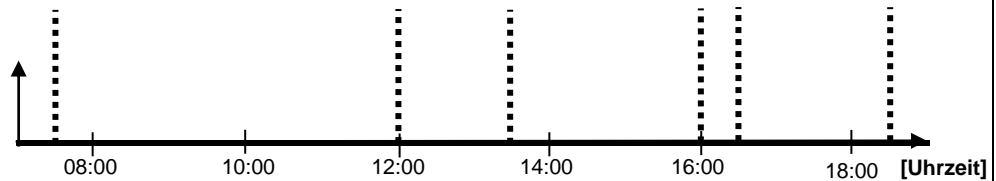
Zeit vor und nach Nutzungsbeginn 30 Minuten.



Beispiel 2: beliebiger Ferientermin aktiv

Zeitplan wie in Beispiel 1

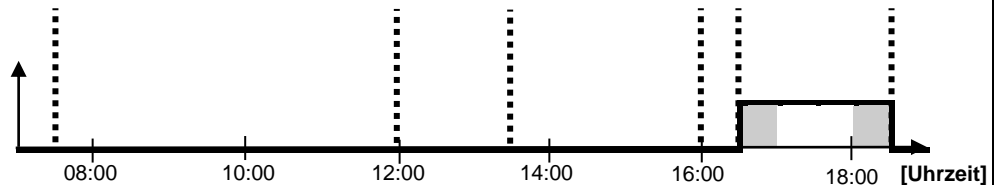
Tag- und Abendaktivierung ausgeschaltet



Beispiel 3: beliebiger Ferientermin aktiv

Zeitplan wie in Beispiel 1

Tagaktivierung ausgeschaltet, Abendaktivierung eingeschaltet



Funktionsbeschreibung:

Der Funktionsbaustein **FbTimetable** beinhaltet die Zeitschaltfunktion für einen Belegungsplan (Stundenplan) eines Raumes.

Damit es nicht zu Überschneidungen zwischen den Zeitschaltprogrammen kommt, sind den verschiedenen Zeitschaltprogrammen Prioritäten zugeordnet.

Priorität der Zeitschaltprogramme:

- Handbedienung (Priorität 1)
- Ferienfunktion (Priorität 2)
- Stundenplanschaltprogramm (Priorität 3)

Der Eingang „**xEnable**“ dient dazu, den Funktionsbaustein **FbTimetable** zu aktivieren.

Der Eingang „**dtActualTime**“ wird mit der aktuellen Zeit verknüpft. Diese Zeit dient als Grundlage für das Zeitschaltprogramm.

Wenn der Eingang „**xHoliday**“ TRUE ist, wird das Ferienschaltprogramm ausgeführt. Die Erkennung der Feiertage kann durch den Funktionsbaustein **FbHoliday** realisiert werden.

Der Eingang „**xManualOperation**“ aktiviert den Handbedienmodus. Im Handbedienmodus wird das Zeitschaltprogramm vom Eingang „**xManualOn**“ gesteuert.

Die Strukturvariable „**typTimetable**“ beinhaltet alle Werte für die Zeitschaltprogramme des Visualisierungselement **FbTimetable**.

Die Strukturvariable „**typLesson**“ beinhaltet alle Werte für die Zeitschaltprogramme des Visualisierungselement **LessonFromFbTimetable**.

Der Ausgang „**xSwitchChannel**“ wird gesetzt, wenn eine Zeitschaltbedingung erfüllt ist.

Der Ausgang „**bStatus**“ zeigt den aktuellen Zustand des Funktionsbausteins an

Hinweise:

- 1.) Die aktuelle Systemzeit kann über die Funktion **SysRtcGetTime** ermittelt werden. Diese Funktion befindet sich in der Bibliothek **SysLibRtc.lib**.
- 2.) Zur komfortablen Konfiguration der Zeitschaltprogramme sind in der Bibliothek die Konfigurationsoberflächen „**visuTimetable**“, „**visuTimetable_Lesson**“. Für die erste Oberfläche sind die Einstellungen der Auflösung anzupassen (Anzeigebreite: 914 und Anzeigehöhe: 567)
- 3.) Die Strukturen „**typTimetable**“ und „**typLesson**“ sollten als RETAIN deklariert werden, damit die Zeitschaltbedingungen nach einem Controller-Reset erhalten bleiben.
- 4.) Bei den 32 Bit-Controllern sollten die Strukturen „**typTimetable**“ und „**typLesson**“ zusätzlich als PERISISTENT deklariert werden, damit die Zeitschaltbedingungen auch nach dem Download erhalten bleiben.
- 5.) Optional kann zur Speicherung der Zeitschaltbedingungen der Strukturen „**typTimetable**“ und „**typLesson**“ ins File-System des 32 Bit-Controllers geschrieben werden. Das Speichern der Daten wird über den Funktionsbaustein **FbSaveVariable** realisiert.

Zusatzfunktionen

Ermittlung der Feiertage (FbPublicHoliday)

WAGO-I/O-PRO CAA Elemente der Bibliothek			
Kategorie:		Gebäudetechnik	
Name:		FbPublicHoliday	
Typ:		Funktion <input type="checkbox"/>	Funktionsblock <input checked="" type="checkbox"/> Programm <input type="checkbox"/>
Name der Bibliothek:		Scheduler_03.lib	
Anwendbar für:		Alle programmierbaren Feldbus-Controller	
Eingangsparameter:		Datentyp:	Kommentar:
xEnable		BOOL	Aktivierung des Funktionsbausteins Voreinstellung: TRUE
dtActualTime		DT	Eingang für die aktuelle Uhrzeit
Ein-Ausgangsparameter:		Datentyp:	Kommentar:
typPublicHoliday		typPublic Holiday	Datenstruktur für die deutschen Feiertage.
.Neujahr		BOOL	Neujahr
.Heilige_Drei_Koenige		BOOL	Heilige Drei Könige
.Rosenmontag		BOOL	Rosenmontag
.Aschermittwoch		BOOL	Aschermittwoch
.Karfreitag		BOOL	Karfreitag
.Ostersonntag		BOOL	Ostersonntag
.Ostermontag		BOOL	Ostermontag
.Maifeiertag		BOOL	Maifeiertag
.Christi_Himmelfahrt		BOOL	Christi Himmelfahrt
.Pfingstsonntag		BOOL	Pfingstsonntag
.Pfingstmontag		BOOL	Pfingstmontag
.Fronleichnam		BOOL	Fronleichnam
.Mariae_Himmelfahrt		BOOL	Mariä Himmelfahrt
.Tag_der_Deutschen_Einheit		BOOL	Tag der Deutschen Einheit
.Reformationstag		BOOL	Reformationstag
.Allerheiligen		BOOL	Allerheiligen
.Buss_und_Betttag		BOOL	Buß- und Betttag
.Heiligabend		BOOL	Heiligabend
.Erster_Weihnachtstag		BOOL	Erster Weihnachtstag
.Zweiter_Weihnachtstag		BOOL	Zweiter Weihnachtstag
.Sylvester		BOOL	Sylvester
.SetSpecialPublic Holiday		WORD	Bitweise Aktivierung der frei definierten Feiertage Bit 0 = Erster frei definierter Feiertag Bit 1 = Zweiter frei definierter Feiertag usw.

[illegible]

Grafische Darstellung:

```

FbPublicHoliday
xEnable          xPublicHoliday-
dtActualTime
typPublicHoliday ▶

```

Konfigurationsoberfläche:

Public Holiday	Active
Neujahr	<input checked="" type="checkbox"/>
Heilige Drei Könige	<input type="checkbox"/>
Rosenmontag	<input type="checkbox"/>
Achermittwoch	<input type="checkbox"/>
Karfreitag	<input checked="" type="checkbox"/>
Ostersonntag	<input type="checkbox"/>
Ostermontag	<input checked="" type="checkbox"/>
Maifeiertag	<input checked="" type="checkbox"/>
Christi Himmelfahrt	<input checked="" type="checkbox"/>
Pfingstsonntag	<input type="checkbox"/>
Pfingstmontag	<input checked="" type="checkbox"/>
Fronleichnam	<input type="checkbox"/>
Mariä Himmelfahrt	<input type="checkbox"/>
Tag der Deutschen Einheit	<input checked="" type="checkbox"/>
Reformationstag	<input type="checkbox"/>
Allerheiligen	<input type="checkbox"/>
Buß- und Bettag	<input type="checkbox"/>
Heiligabend	<input checked="" type="checkbox"/>
Erster Weihnachtstag	<input checked="" type="checkbox"/>
Zweiter Weihnachtstag	<input checked="" type="checkbox"/>
Sylvester	<input checked="" type="checkbox"/>
06 - 12	<input checked="" type="checkbox"/>
01 - 01	<input type="checkbox"/>
01 - 01	<input type="checkbox"/>
01 - 01	<input type="checkbox"/>
01 - 01	<input type="checkbox"/>

Funktionsbeschreibung:

Der Funktionsbaustein **FbPublicHoliday** dient zur Erkennung der Feiertage. Es werden nur die Feiertage erkannt, die in der Struktur „*typPublicHoliday*“ aktiviert sind.

Der Eingang „**xEnable**“ dient dazu, den Funktionsbaustein **FbPublicHoliday** zu aktivieren.

Der Eingang „**dtActualTime**“ wird mit der aktuellen Zeit verknüpft. Diese Zeit dient als Grundlage für die Feiertagsermittlung.

Die Strukturvariable „*typPublicHoliday*“ beinhaltet die deutschen Feiertage. Neben den fest definierten Feiertagen besteht die Möglichkeit bis zu fünf Feiertage frei zu definieren.

Wenn der Funktionsbaustein **FbPublicHoliday** einen Feiertag erkennt, wird der Ausgang „**xPublicHoliday**“ auf TRUE gesetzt.

Hinweise:

- 1.) Die aktuelle Systemzeit kann über die Funktion **SysRtcGetTime** ermittelt werden. Diese Funktion befindet sich in der Bibliothek **SysLibRtc.lib**.
- 2.) Die Struktur „*typPublicHoliday*“ sollte als RETAIN deklariert werden, damit die Auswahl der Feiertage nach einem Controller-Reset erhalten bleibt.
- 3.) Bei den 32 Bit-Controllern sollte die Struktur „*typPublicHoliday*“ zusätzlich als PERISISTENT deklariert werden, damit die Auswahl der Feiertage auch nach dem Download erhalten bleibt.
- 4.) Optional kann zur Speicherung der ausgewählten Feiertage die Struktur „*typPublicHoliday*“ ins File-System des 32 Bit-Controllers geschrieben werden. Das Speichern der Daten wird über den Funktionsbaustein **FbSaveVariable** realisiert.
- 5.) Zur komfortablen Auswahl der Feiertage ist in der Bibliothek die Konfigurationsoberfläche „**visuPublicHoliday**“ hinterlegt.

Ermittlung der Ferien (FbHoliday)

WAGO-I/O-PRO CAA Elemente der Bibliothek		
Kategorie:	Gebäudetechnik	
Name:	FbHoliday	
Typ:	Funktion <input type="checkbox"/> Funktionsblock <input checked="" type="checkbox"/> Programm <input type="checkbox"/>	
Name der Bibliothek:	Scheduler_03.lib	
Anwendbar für:	Alle programmierbaren Feldbus-Controller	
Eingangsparameter:	Datentyp:	Kommentar:
xEnable	BOOL	Aktivierung des Funktionsbausteins Voreinstellung: TRUE
dtActualTime	DT	Eingang für die aktuelle Uhrzeit und Datum
Ein-Ausgangsparameter:	Datentyp:	Kommentar:
typHoliday	typHoliday	Datenstruktur für die Ferien
.wFirstDay	ARRAY[1..3,1..20] of WORD	erster Ferientag
.wLastDay	ARRAY[1..3,1..20] of WORD	letzter Ferientag
.strComment	ARRAY[1..20] of STRING(20)	Kommentar mit max. 20 Zeichen
Rückgabewert:	Datentyp:	Kommentar:
xHoliday	BOOL	Zeigt an, ob das aktuelle Datum ein Ferientag ist
Strukturaufbau der Variable „typHoliday“:		
<pre> 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] = " </pre>		

WAGO-I/O-PRO CAA Elemente der Bibliothek

Grafische Darstellung:



Konfigurationsoberfläche:

	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

Funktionsbeschreibung:

Der Funktionsbaustein **FbHoliday** dient zur Erkennung der Ferien. Ferientermine werden durch die Einträge in der Struktur „**typHoliday**“ vorgegeben.

Der Eingang „**xEnable**“ dient dazu, den Funktionsbaustein **FbHoliday** zu aktivieren.

Der Eingang „**dtActualTime**“ wird mit der aktuellen Zeit verknüpft. Diese Zeit dient als Grundlage zum berechnen der Ferien.

Die Strukturvariable „**typHoliday**“ beinhaltet die Daten der Ferien, sowie deren Beschreibung als Kommentar. Die Ferienzeiten können frei gewählt werden. Es können bis zu 20 unterschiedliche Ferientermine eingetragen werden.

Wenn der Funktionsbaustein **FbHoliday** einen Ferientermin erkennt, wird der Ausgang „**xHoliday**“ auf TRUE gesetzt.

Hinweise:

- 1.) Die aktuelle Systemzeit kann über die Funktion **SysRtcGetTime** ermittelt werden. Diese Funktion befindet sich in der Bibliothek **SysLibRtc.lib**.
- 2.) Die Struktur „**typHoliday**“ sollte als RETAIN deklariert werden, damit die Auswahl der Feiertage nach einem Controller-Reset erhalten bleibt.
- 3.) Bei den 32 Bit-Controllern sollte die Struktur „**typHoliday**“ zusätzlich als PERISISTENT deklariert werden, damit die Auswahl der Feiertage auch nach dem Download erhalten bleibt.
- 4.) Optional kann zur Speicherung der ausgewählten Feiertage die Struktur „**typHoliday**“ ins File-System des 32 Bit-Controllers geschrieben werden. Das Speichern der Daten wird über den Funktionsbaustein **FbSaveVariable** realisiert.
- 5.) Zur komfortablen Auswahl der Feiertage ist in der Bibliothek die Konfigurationsoberfläche „**visuHoliday**“ hinterlegt.

Allgemeine Wertvorgabe (FbSetScheduleValue)

WAGO-I/O-PRO CAA Elemente der Bibliothek			
Kategorie:	Gebäudetechnik		
Name:	FbHoliday		
Typ:	Funktion <input type="checkbox"/>	Funktionsblock <input checked="" type="checkbox"/>	Programm <input type="checkbox"/>
Name der Bibliothek:	Scheduler_03.lib		
Anwendbar für:	Alle programmierbaren Feldbus-Controller		
Eingangsparameter:	Datentyp:	Kommentar:	
typScheduleStatus	typScheduleStatus	Datenstruktur für den Status	
.xEnable	BOOL	Freigabe des Bausteins	
.xManualOperation	BOOL	Manuell schalten	
.xManualOn	BOOL	Manuell ein	
.xPublicHoliday	BOOL	Feiertag	
.xParty	BOOL	Partyfunktion	
.xSwitchChannel	BOOL	Schaltausgang	
.axWeekly	Array[1..10] of BOOL	Status der 10 Wochenzeitschaltprogramme	
.axSpecialPeriod	Array[1..5] of BOOL	Status der 5 Jahreszeitschaltprogramme	
typScheduleValue	typScheduleValue	Datenstruktur für die Werte der Zeitschaltprogramme	
.arWeekly	Array[1..10] of REAL	Werte der 10 Wochenzeitschaltprogramme	
.arSpecial	Array[1..5] of REAL	Werte der 5 Jahreszeitschaltprogramme	
.rHoliday	REAL	Wert für das Ferien/-Feiertagsprogramm	
.rParty	REAL	Wert für die Partyfunktion	
.Default	REAL	Standardwert	
Rückgabewert:	Datentyp:	Kommentar:	
rValue	REAL	Aktuell freigegebener Wert	

Strukturaufbau der Variable „typScheduleValue“:

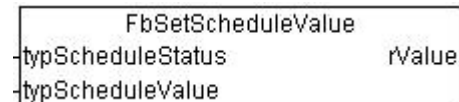
```

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
  
```

Strukturaufbau der Variable „typScheduleStatus“:

```

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
  
```

Grafische Darstellung:

Konfigurationsoberfläche:**Wertvorgabe der Party- und Holidayprogramme**

Public holiday	ON	OFF	Status	Value	Party function	Switch-on-time	Status	Value	Default
<input type="checkbox"/>	00:00	00:00	<input type="checkbox"/>	17.0	<input type="checkbox"/>	60	<input type="checkbox"/>	21.0	17.0

Wertvorgabe für das Jahreszeitschaltprogramm

(zum anfügen an die entsprechende Oberfläche des Bausteins FbScheduler)

Value	Default
21.0	17.0
21.0	
21.0	
21.0	
21.0	

Wertvorgabe für das Wochenzeitschaltprogramm

(zum anfügen an die entsprechende Oberfläche des Bausteins FbScheduler)

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

Funktionsbeschreibung:

Der Funktionsbaustein **FbSetScheduleValue** dient der flexiblen Zuordnung von Werten, des jeweils aktiven Zeitschaltprogramms, auf den Ausgang des Bausteins. Sind mehrere Zeitschaltprogramme aktiv, erfolgt die Wertvorgabe nach Prioritäten. Die Reihenfolge der Prioritäten ist identisch mit der des Bausteins **FbScheduler**. Ist kein Zeitschaltprogramm aktiv, wird der Default-Wert (Variable „typScheduleStatus.rDefault“) am Ausgang ausgegeben.

Die erste Strukturvariable „**typScheduleStatus**“ am Eingang des Bausteins **FbSetScheduleValue** muss mit dem Ausgang „**typScheduleStatus**“ des Bausteins **FbScheduler** verbunden werden. Der Wert am Ausgang des Bausteins wird über die Variablen der Struktur gesteuert.

Die zweite Strukturvariable „**typScheduleValue**“ am Eingang beinhaltet alle Werte der einzelnen Zeitschaltprogramme. Diese Struktur kann mit den entsprechenden Konfigurationsoberflächen verknüpft werden.

Am Ausgang „**rValue**“ des Bausteins **FbSetScheduleValue** wird der aktuelle Wert des entsprechend aktiven Zustandes angezeigt.

Hinweise:

- 1.) Die aktuelle Systemzeit kann über die Funktion **SysRtcGetTime** ermittelt werden. Diese Funktion befindet sich in der Bibliothek **SysLibRtc.lib**.
- 2.) Die Struktur „**typScheduleValue**“ sollte als RETAIN deklariert werden, damit die Auswahl der Feiertage nach einem Controller-Reset erhalten bleibt.
- 3.) Bei den 32 Bit-Controllern sollte die Struktur „**typScheduleValue**“ zusätzlich als PERISISTENT deklariert werden, damit die Auswahl der Feiertage auch nach dem Download erhalten bleibt.
- 4.) Optional kann zur Speicherung der ausgewählten Feiertage die Struktur „**typScheduleValue**“ ins File-System des 32 Bit-Controllers geschrieben werden. Das Speichern der Daten wird über den Funktionsbaustein **FbSaveVariable** realisiert.
- 5.) Zur komfortablen Auswahl der Werte sind in der Bibliothek die Konfigurationsoberflächen „**visuScheduler_HolidayPartyValue**“, „**visuSetScheduleValue_SpecialPeriods**“ und „**visuSetScheduleValue_Weekly**“ hinterlegt.

Nachbildung der Uhrenfunktion (FbClock)

WAGO-I/O-PRO CAA Elemente der Bibliothek		
Kategorie:	Gebäudetechnik	
Name:	FbClock	
Typ:	Funktion <input type="checkbox"/> Funktionsblock <input checked="" type="checkbox"/> Programm <input type="checkbox"/>	
Name der Bibliothek:	Scheduler_03.lib	
Anwendbar für:	Alle programmierbaren Feldbus-Controller	
Eingangsparameter:		
	Datentyp:	Kommentar:
wYear	WORD	Eingangssignal Jahr
bMonth	BYTE	Eingangssignal Monat
bDay	BYTE	Eingangssignal Tag
bHour	BYTE	Eingangssignal Stunden
bMinute	BYTE	Eingangssignal Minuten
bSecond	BYTE	Eingangssignal Sekunden
Ein-Ausgangsparameter:		
	Datentyp:	Kommentar:
dtActualTime	DT	Ausgabe des aktuellen Datums und der Uhrzeit
Rückgabewert:		
	Datentyp:	Kommentar:
bWeekday	BYTE	Anzeige der Wochentage: 1 = Montag 2 = Dienstag 3 = Mittwoch 4 = Donnerstag 5 = Freitag 6 = Samstag 7 = Sonntag
Grafische Darstellung:		
		

Funktionsbeschreibung:

Der Funktionsbaustein **FbClock** ermöglicht die Nachbildung einer Uhrenfunktion. Die zeitliche Abweichung der Uhrzeit beträgt ca. 10 Sekunden pro Tag. Deshalb ist es notwendig, dass der Funktionsblock von einer externen Echtzeituhr synchronisiert wird (Empfehlung 1mal täglich). Die Synchronisation des Datums erfolgt über die Eingänge „**wYear**“, „**bMonth**“ und „**bDay**“ und die Synchronisation der Uhrzeit über die Eingänge „**bHour**“, „**bMinute**“ und „**bSecond**“.

Wenn sich einer der Werte an den Eingängen „**bHour**“, „**bMinute**“ oder „**bSecond**“ verändert, wird die Zeit „**dtActualTime**“ mit allen drei Eingangswerten initialisiert. Eine Wertänderung an den Eingängen „**wYear**“, „**bMonth**“ und „**bDay**“ bewirkt, dass nur der Wert, der sich geändert hat, den aktuellen Wert von „**dtActualTime**“ überschreibt.

Wenn sich kein Eingangswert ändert, läuft die Zeit der Variablen „**dtActualTime**“ kontinuierlich weiter und bildet somit die Uhrenfunktion nach. Bei der Ermittlung des aktuellen Datums werden die Schaltjahre berücksichtigt.

In Abhängigkeit vom Datum wird der Wochentag ermittelt und am Ausgang „**bWeekday**“ angezeigt.

Hinweis:

- 1.) Wenn am Eingang „**wYear**“ eine Null anliegt, wird die Uhrzeit nicht initialisiert.
- 2.) Die Variable „**dtActualTime**“ sollte als RETAIN deklariert werden, damit die Uhrzeit nach einem Controller-Reset mit der zuletzt ermittelten Zeit erhalten bleibt.

Speichern von Daten im File-System (FbSaveVariable)

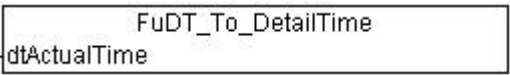
WAGO-I/O-PRO CAA Elemente der Bibliothek			
Kategorie:		Gebäudetechnik	
Name:		FbSaveVariable	
Typ:		Funktion <input type="checkbox"/>	Funktionsblock <input checked="" type="checkbox"/> Programm <input type="checkbox"/>
Name der Bibliothek:		Scheduler_03.lib	
Anwendbar für:		Ethernet Feldbus-Controller 750-841	
Verwendete Bibliothek:		SysLibFile.lib	
Eingangsparameter:		Datentyp:	Kommentar:
stFileName		STRING	Name der Datei.
xLoad		BOOL	Laden der Variable aus dem File-System
xSave		BOOL	Speichern der Variable ins File-System
ptLoadSaveVariable		POINTER TO DWORD	Anfangsadresse der Variable
dwSize		DWORD	Größe der Variable [Byte]
Rückgabewert:		Datentyp:	Kommentar:
dtLastModification		DT	Datum der letzten Änderung
xError		BOOL	Fehler beim Laden der Datei.
Grafische Darstellung:			
<div><div><div>FbSaveVariable</div><div><div>stFileName</div><div>dtLastModification</div><div>xLoad</div><div>xError</div><div>xSave</div><div>ptLoadSaveVariable</div><div>dwSize</div></div></div><div><div>ADR(Data)</div><div>SIZEOF(Data)</div></div></div>			
Funktionsbeschreibung:			
<p>Der Funktionsbaustein FbSaveVariable dient zum Speichern und Auslesen von Daten im File-System des Ethernet Feldbus-Controllers 750-841.</p> <p>Am Eingang „stFileName“ wird der Name der Datei vorgegeben. Die Adresse der zu speichernden Variable wird am Eingang „ptLoadSaveVariable“ vorgegeben. Zusätzlich muss die Anzahl der Bytes zum Schreiben oder Lesen am Eingang „dwSize“ vorgegeben werden. Als Hilfe zur Ermittlung der Datengröße dient der Operator SIZEOF.</p> <p>Bei einer positiven Flanke am Eingang „xLoad“ wird der Wert der Variablen aus dem File-System ausgelesen.</p> <p>Bei einer positiven Flanke am Eingang „xSave“ wird der Wert der Variablen ins File-System geschrieben.</p> <p>Der Ausgang „dtLastModification“ zeigt an, wann die Variable das letzte mal gespeichert wurde.</p> <p>Der Ausgang „xError“ zeigt an, ob ein Fehler beim speichern oder laden der Datei aufgetreten ist.</p>			

Sommer- / Winterzeitumstellung (FbSummer_Wintertime)

WAGO-I/O-PRO CAA Elemente der Bibliothek		
Kategorie:	Gebäudetechnik	
Name:	FbSummer_Wintertime	
Typ:	Funktion <input type="checkbox"/>	Funktionsblock <input checked="" type="checkbox"/> Programm <input type="checkbox"/>
Name der Bibliothek:	Scheduler_03.lib	
Anwendbar für:	Ethernet Feldbus-Controller 750-841	
Verwendete Bibliothek:	SysLibRtc.lib	
Eingangsparameter:	Datentyp:	Kommentar:
xEnable	BOOL	Aktivierung des Funktionsbausteins Voreinstellung: TRUE
Grafische Darstellung:		
<div style="border: 1px solid black; padding: 5px; display: inline-block;"> FbSummer_Wintertime xEnable </div>		
Funktionsbeschreibung:		
<p>Der Funktionsbaustein FbSummer_Wintertime stellt die Uhrzeit am letzten Sonntag im März von 2 Uhr auf 3 Uhr vor und am letzten Sonntag im Oktober von 3 Uhr auf 2 Uhr zurück.</p> <p>Der Eingang „xEnable“ dient dazu, die Sommer- / Winterzeitumstellung zu aktivieren.</p>		

Konvertierungen

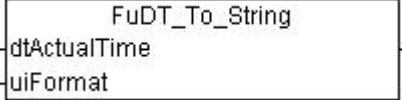
Aufsplittung des Datentyps DT (Fu_DT_To_DetailTime)

WAGO-I/O-PRO CAA Elemente der Bibliothek		
Kategorie:	Gebäudetechnik	
Name:	FuDT_To_DetailTime	
Typ:	Funktion <input checked="" type="checkbox"/> Funktionsblock <input type="checkbox"/> Programm <input type="checkbox"/>	
Name der Bibliothek:	Scheduler_03.lib	
Anwendbar für:	Alle programmierbaren Feldbus-Controller	
Eingangsparameter:	Datentyp:	Kommentar:
dtActualTime	DT	Aktuelle Zeit Datentyp: „Date_and_Time“
Rückgabewert:	Datentyp:	Kommentar:
FuDT_To_DetailTime	typDetailTime	Datenstruktur für die einzelnen Zeiteinheiten
.wYear	WORD	Jahr
.bMonth	BYTE	Monat
.bDay	BYTE	Tag
.bHour	BYTE	Stunden
.bMinute	BYTE	Minuten
.bSecond	BYTE	Sekunden
.bWeekday	BYTE	Wochentag 1 = Montag 2 = Dienstag 3 = Mittwoch 4 = Donnerstag 5 = Freitag 6 = Samstag 7 = Sonntag
Grafische Darstellung:		
		
Funktionsbeschreibung:		
<p>Die Funktion FuDT_To_DetailTime wandelt den Datentyp DT in die Struktur „typDetailTime“. In der Struktur „typDetailTime“ wird jede Zeiteinheit einzeln dargestellt.</p>		

Zusammenfassen zum Datentyp DT (FuDetailTime_To_DT)

WAGO-I/O-PRO CAA Elemente der Bibliothek		
Kategorie:	Gebäudetechnik	
Name:	FuDetailTime_To_DT	
Typ:	Funktion <input checked="" type="checkbox"/> Funktionsblock <input type="checkbox"/> Programm <input type="checkbox"/>	
Name der Bibliothek:	Scheduler_03.lib	
Anwendbar für:	Alle programmierbaren Feldbus-Controller	
Eingangsparameter:	Datentyp:	Kommentar:
wYear	WORD	Eingangssignal Jahr
bMonth	BYTE	Eingangssignal Monat
bDay	BYTE	Eingangssignal Tag
bHour	BYTE	Eingangssignal Stunden
bMinute	BYTE	Eingangssignal Minuten
bSecond	BYTE	Eingangssignal Sekunden
Rückgabewert:	Datentyp:	Kommentar:
FuDetailTime_To_DT	DT	Ausgabe des Datums und der Uhrzeit im Datentyp „ Date_and_Time “
Grafische Darstellung:		
		
Funktionsbeschreibung:		
Die Funktion FuDetailTime_To_DT wandelt die einzelnen Zeiteinheiten in den Datentyp DT um.		

Umwandlung des Datentyps DT zum String (FuDT_TO_String)

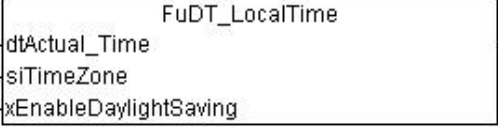
WAGO-I/O-PRO CAA Elemente der Bibliothek		
Kategorie:	Gebäudetechnik	
Name:	FuDT_TO_String	
Typ:	Funktion <input checked="" type="checkbox"/> Funktionsblock <input type="checkbox"/> Programm <input type="checkbox"/>	
Name der Bibliothek:	Scheduler_03.lib	
Anwendbar für:	Alle programmierbaren Feldbus-Controller	
Eingangsparameter:	Datentyp:	Kommentar:
dtActualTime	DT	Aktuelle Zeit Datentyp: „ Date_and_Time “
uiFormat	UINT	Auswahl der Zeitangabe 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
Rückgabewert:	Datentyp:	Kommentar:
FuDT_To_String	DT	Ausgabe des Datums und der Uhrzeit im Datentyp „ String “
Grafische Darstellung:		
		
Funktionsbeschreibung:		
<p>Die Funktion FuDT_To_String wandelt den Eingangsparameter „dtActualTime“ in einen String um, welcher die aktuelle Uhrzeit und die Datumsangabe beinhaltet. Mit Hilfe des Eingangs „uiFormat“ gibt es die Möglichkeit verschiedene Darstellungsformen von Zeitangaben auszuwählen.</p>		

RTC-Klemme

Datum und Zeit mittels der RTC-Klemme (FbTime_RTC_Modul)

WAGO-I/O-PRO CAA Elemente der Bibliothek			
Kategorie:		Gebäudetechnik	
Name:		FbTime_RTC_Modul	
Typ:		Funktion <input type="checkbox"/>	Funktionsblock <input checked="" type="checkbox"/> Programm <input type="checkbox"/>
Name der Bibliothek:		Scheduler_03.lib	
Anwendbar für:		Alle programmierbaren Feldbus-Controller	
Eingangsparameter:		Datentyp:	Kommentar:
dtSetTime	DT	Vorgabe der Zeit Datentyp: „Date_and_Time“	
xSetTime	BOOL	Übernehmen der Zeit vom Eingang dtSetTime	
Ausgangsparameter:		Datentyp:	Kommentar:
dtActualTime	DT	Aktuelle Uhrzeit aus der RTC-Klemme	
bStatus	BYTE	Bitweise Anzeige des Status: Bit 0 = Antennensignalpegel Bit 1 = Pufferspannung zu niedrig oder Uhr ausgefallen Bit 2 = Antennen-Signal nicht vorhanden oder fehlerhaft Bit 3 = Speicher nicht ansprechbar	
Grafische Darstellung:			
<div><div>FbTime_RTC_Modul</div><div><div>dtSetTime</div><div>dtActualTime</div><div>xSetTime</div><div>bStatus</div></div></div>			
Funktionsbeschreibung:			
<p>Der Funktionsbaustein FbTime_RTC_Modul ermittelt das Datum und die Zeit aus der RTC-Klemme 750-640 und stellt die Daten am Ausgang „dtActualTime“ zur Verfügung. Der zweite Ausgang „bStatus“ dient zur Diagnose der Klemme. Mit dem Eingang „dtSetTime“ kann eine Zeit vorgegeben werden, welche mit einer positiven Flanke am Eingang „xSetTime“ übernommen wird.</p>			
Hinweis:			
<p>1) Die Eingänge zur manuellen Vorgabe der Zeit werden nur benötigt, wenn keine Antenne angeschlossen sein sollte.</p>			

Zeitumrechnung (FuDT_LocalTime)

WAGO-I/O-PRO CAA Elemente der Bibliothek		
Kategorie:	Gebäudetechnik	
Name:	FuDT_LocalTime	
Typ:	Funktion <input checked="" type="checkbox"/>	Funktionsblock <input type="checkbox"/> Programm <input type="checkbox"/>
Name der Bibliothek:	Scheduler_03.lib	
Anwendbar für:	Alle programmierbaren Feldbus-Controller	
Eingangsparameter:	Datentyp:	Kommentar:
dtActualTime	DT	Eingabe des Datums und der Uhrzeit im Datentyp „ Date_and_Time “
siTimeZone	SINT	Zeitzone
xEnableDaylightSaving	BOOL	Sommer-/Winterumschaltung in/-aktiv
Rückgabewert:	Datentyp:	Kommentar:
FuDT_LocalTime	DT	Rückgabewert des Datums und der Uhrzeit im Datentyp „ Date_and_Time “
Grafische Darstellung:		
		
Funktionsbeschreibung:		
<p>Die Funktion FuDT_LocalTime dient zur Umrechnung der UTC-Zeit am Eingang „dtActualTime“ auf die lokale Zeit. Dafür kann die Zeitzone (Eingang: „siTimeZone“) und/oder eine Sommer-/Winterumschaltung (Eingang: „xEnableDaylightSaving“) berücksichtigt werden. Als Rückgabewert wird dann die umgerechnete Zeit ausgegeben.</p>		



WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D-32385 Minden
Hansastraße 27 • D-32423 Minden
Telefon: 05 71/8 87 – 0
Telefax: 05 71/8 87 – 1 69
E-Mail: info@wago.com

Internet: <http://www.wago.com>