

VIPA SPEED7 Library

OPL_SP7-LIB | SW90ES0MA V10.002 | Handbuch

HB00 | OPL_SP7-LIB | SW90ES0MA V10.002 | de | 18-28

Baustein Bibliothek - Building Control



VIPA GmbH
Ohmstr. 4
91074 Herzogenaurach
Telefon: +49 9132 744-0
Telefax: +49 9132 744-1864
E-Mail: info@vipa.com
Internet: www.vipa.com

Inhaltsverzeichnis

1	Allgemeines	4
	1.1 Copyright © VIPA GmbH	4
	1.2 Über dieses Handbuch.....	5
2	Wichtige Hinweise	6
	2.1 Allgemein.....	6
	2.2 Intern verwendete Bausteine.....	6
3	Bibliothek einbinden	7
	3.1 Einbinden in Siemens SIMATIC Manager.....	7
	3.2 Einbinden in Siemens TIA Portal.....	8
4	Gebäude-Automatisierung - "Building Control"	9
	4.1 Übersicht.....	9
	4.1.1 Aufruf-Beispiel - Instanz-DB.....	9
	4.1.2 Aufruf-Beispiel - Multiinstanzen-DB.....	9
	4.2 Raumsteuerung - "Room"	10
	4.2.1 FB 45 - LAMP - Leuchte / Steckdose steuern.....	10
	4.2.2 FB 46 - BLIND - Jalousie steuern.....	11
	4.2.3 FB 47 - DSTRIKE - Elektrischer Türöffner	13
	4.3 Zugangskontrolle - "Access Control"	14
	4.3.1 FB 48 - ACONTROL - Zutrittssteuerung.....	14
	4.3.2 UDT 3 - ACLREC - Datenstruktur für FB 48.....	15
	4.3.3 UDT 4 - ACL - Datenstruktur für FB 48.....	16
	4.3.4 FB 49 - KEYPAD - Tastatur.....	17
	4.3.5 FB 50 - KEYPAD2 - Tastatur.....	19

1 Allgemeines

1.1 Copyright © VIPA GmbH

All Rights Reserved

Dieses Dokument enthält geschützte Informationen von VIPA und darf außer in Übereinstimmung mit anwendbaren Vereinbarungen weder offengelegt noch benutzt werden.

Dieses Material ist durch Urheberrechtsgesetze geschützt. Ohne schriftliches Einverständnis von VIPA und dem Besitzer dieses Materials darf dieses Material weder reproduziert, verteilt, noch in keiner Form von keiner Einheit (sowohl VIPA-intern als auch -extern) geändert werden, es sei denn in Übereinstimmung mit anwendbaren Vereinbarungen, Verträgen oder Lizenzen.

Zur Genehmigung von Vervielfältigung oder Verteilung wenden Sie sich bitte an: VIPA, Gesellschaft für Visualisierung und Prozessautomatisierung mbH Ohmstraße 4, D-91074 Herzogenaurach, Germany

Tel.: +49 9132 744 -0

Fax.: +49 9132 744-1864

E-Mail: info@vipa.de

<http://www.vipa.com>



Es wurden alle Anstrengungen unternommen, um sicherzustellen, dass die in diesem Dokument enthaltenen Informationen zum Zeitpunkt der Veröffentlichung vollständig und richtig sind. Das Recht auf Änderungen der Informationen bleibt jedoch vorbehalten.

Die vorliegende Kundendokumentation beschreibt alle heute bekannten Hardware-Einheiten und Funktionen. Es ist möglich, dass Einheiten beschrieben sind, die beim Kunden nicht vorhanden sind. Der genaue Lieferumfang ist im jeweiligen Kaufvertrag beschrieben.

EG-Konformitätserklärung

Hiermit erklärt VIPA GmbH, dass die Produkte und Systeme mit den grundlegenden Anforderungen und den anderen relevanten Vorschriften übereinstimmen. Die Übereinstimmung ist durch CE-Zeichen gekennzeichnet.

Informationen zur Konformitätserklärung

Für weitere Informationen zur CE-Kennzeichnung und Konformitätserklärung wenden Sie sich bitte an Ihre Landesvertretung der VIPA GmbH.

Warenzeichen

VIPA, SLIO, System 100V, System 200V, System 300V, System 300S, System 400V, System 500S und Commander Compact sind eingetragene Warenzeichen der VIPA Gesellschaft für Visualisierung und Prozessautomatisierung mbH.

SPEED7 ist ein eingetragenes Warenzeichen der profichip GmbH.

SIMATIC, STEP, SINEC, TIA Portal, S7-300, S7-400 und S7-1500 sind eingetragene Warenzeichen der Siemens AG.

Microsoft und Windows sind eingetragene Warenzeichen von Microsoft Inc., USA.

Portable Document Format (PDF) und Postscript sind eingetragene Warenzeichen von Adobe Systems, Inc.

Alle anderen erwähnten Firmennamen und Logos sowie Marken- oder Produktnamen sind Warenzeichen oder eingetragene Warenzeichen ihrer jeweiligen Eigentümer.

Dokument-Support

Wenden Sie sich an Ihre Landesvertretung der VIPA GmbH, wenn Sie Fehler anzeigen oder inhaltliche Fragen zu diesem Dokument stellen möchten. Ist eine solche Stelle nicht erreichbar, können Sie VIPA über folgenden Kontakt erreichen:

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telefax: +49 9132 744-1204

E-Mail: documentation@vipa.de

Technischer Support

Wenden Sie sich an Ihre Landesvertretung der VIPA GmbH, wenn Sie Probleme mit dem Produkt haben oder Fragen zum Produkt stellen möchten. Ist eine solche Stelle nicht erreichbar, können Sie VIPA über folgenden Kontakt erreichen:

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telefon: +49 9132 744-1150 (Hotline)

E-Mail: support@vipa.de

1.2 Über dieses Handbuch

Zielsetzung und Inhalt

Das Handbuch beschreibt die Baustein-Bibliothek *"Building Control"* von VIPA:

- Beschrieben wird Aufbau, Projektierung und Anwendung in verschiedenen Programiersystemen.
- Das Handbuch ist geschrieben für Anwender mit Grundkenntnissen in der Automatisierungstechnik.
- Das Handbuch ist in elektronischer Form als PDF-Datei verfügbar. Hierzu ist der Adobe Acrobat Reader erforderlich.
- Das Handbuch ist in Kapitel gegliedert. Jedes Kapitel beschreibt eine abgeschlossene Thematik.
- Als Orientierungshilfe stehen im Handbuch zur Verfügung:
 - Gesamt-Inhaltsverzeichnis am Anfang des Handbuchs
 - Verweise mit Seitenangabe

Piktogramme Signalwörter

Besonders wichtige Textteile sind mit folgenden Piktogrammen und Signalworten ausgezeichnet:

**GEFAHR!**

Unmittelbar drohende oder mögliche Gefahr. Personenschäden sind möglich.

**VORSICHT!**

Bei Nichtbefolgen sind Sachschäden möglich.



Zusätzliche Informationen und nützliche Tipps.

2 Wichtige Hinweise

2.1 Allgemein



Nachfolgend finden Sie wichtige Hinweise, die grundsätzlich beim Einsatz der Bausteine zu beachten sind.

2.2 Intern verwendete Bausteine



VORSICHT!

Folgende Bausteine werden intern verwendet und dürfen nicht überschrieben werden! Der direkte Aufruf eines internen Bausteins führt zu Fehler im entsprechenden Instanz-DB! Bitte verwenden Sie für den Aufruf immer die zugehörige Funktion.

FC/SFC	Bezeichnung	Beschreibung
FC/SFC 192	CP_S_R	wird intern für FB 7 und FB 8 verwendet
FC/SFC 196	AG_CNTRL	wird intern für FC 10 verwendet
FC/SFC 200	AG_GET	wird intern für FB/SFB 14 verwendet
FC/SFC 201	AG_PUT	wird intern für FB/SFB 15 verwendet
FC/SFC 202	AG_BSEND	wird intern für FB/SFB 12 verwendet
FC/SFC 203	AG_BRCV	wird intern für FB/SFB 13 verwendet
FC/SFC 204	IP_CONF	wird intern für FB 55 IP_CONF verwendet
FC/SFC 205	AG_SEND	wird intern für FC 5 AG_SEND verwendet
FC/SFC 206	AG_RECV	wird intern für FC 6 AG_RECV verwendet
FC/SFC 253	IBS_ACCESS	wird intern für SPEED-Bus-INTERBUS-Master verwendet
SFB 238	EC_RWOD	wird intern für EtherCAT-Kommunikation verwendet
SFB 239	FUNC	wird intern für FB 240, FB 241 verwendet

3 Bibliothek einbinden

Baustein-Bibliothek "Building Control"

Die Baustein-Bibliothek finden Sie im "Service/Support"-Bereich auf www.vipa.com unter "Downloads → VIPA Lib" als "Baustein-Bibliothek Building Control - SW90ES0MA" zum Download. Die Bibliothek liegt als gepackte zip-Dateien vor. Sobald Sie die Bausteine verwenden möchten, müssen Sie diese in Ihr Projekt importieren.




Folgende Bausteinbibliotheken stehen zur Verfügung

Datei	Beschreibung
BuildingControl_S7_V0003.zip	<ul style="list-style-type: none"> ■ Bausteinbibliothek für Siemens SIMATIC Manager. ■ Für den Einsatz in CPUs von VIPA bzw. S7-300 CPUs von Siemens.
BuildingControl_TIA_V0004.zip	<ul style="list-style-type: none"> ■ Bausteinbibliothek für Siemens TIA Portal V14. ■ Für den Einsatz in CPUs von VIPA bzw. S7-300 CPUs von Siemens.


3.1 Einbinden in Siemens SIMATIC Manager

Übersicht






Die Einbindung in den Siemens SIMATIC Manager erfolgt nach folgenden Schritten:

1.  ZIP-Datei laden
2.  Bibliothek "dearchivieren"
3.  Bibliothek öffnen und Bausteine in Projekt übertragen



ZIP-Datei laden

 Navigieren Sie auf der Webseite zu der gewünschten ZIP-Datei, laden und speichern Sie diese in Ihrem Arbeitsverzeichnis.

Bibliothek dearchivieren

1.  Starten Sie den Siemens SIMATIC Manager mit Ihrem Projekt.
2.  Öffnen Sie mit "Datei → Dearchivieren" das Dialogfenster zur Auswahl der ZIP-Datei.
3.  Wählen Sie die entsprechende ZIP-Datei an und klicken Sie auf [Öffnen].
4.  Geben Sie ein Zielverzeichnis an, in dem die Bausteine abzulegen sind.
5.  Starten Sie den Entpackvorgang mit [OK].

Bibliothek öffnen und Bausteine in Projekt übertragen

1.  Öffnen Sie die Bibliothek nach dem Entpackvorgang.
2.  Öffnen Sie Ihr Projekt und kopieren Sie die erforderlichen Bausteine aus der Bibliothek in das Verzeichnis "Bausteine" Ihres Projekts.
 - ⇒ Nun haben Sie in Ihrem Anwenderprogramm Zugriff auf die VIPA-spezifischen Bausteine.







Werden anstelle der SFCs FCs verwendet, so werden diese von den VIPA CPUs ab Firmware 3.6.0 unterstützt.



3.2 Einbinden in Siemens TIA Portal

Übersicht


Die Einbindung in das Siemens TIA Portal erfolgt nach folgenden Schritten:

1.  ZIP-Datei laden
2.  ZIP-Datei entpacken
3.  Bibliothek "dearchivieren"
4.  Bibliothek öffnen und Bausteine in Projekt übertragen







ZIP-Datei laden

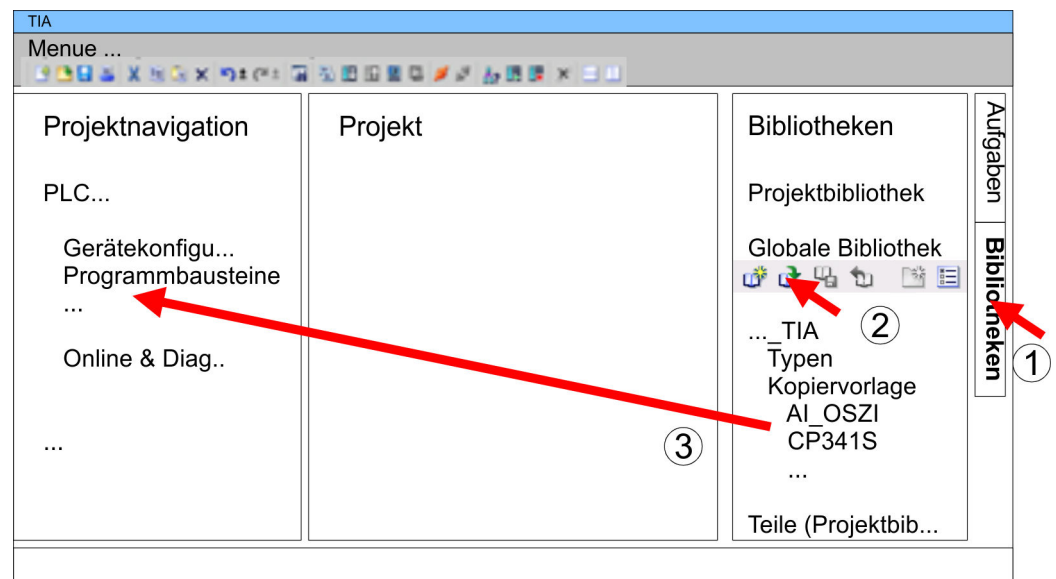
1.  Navigieren Sie auf der Webseite zu der ZIP-Datei, welche zu Ihrer Programmversion passt.
2.  Laden und speichern Sie diese in Ihrem Arbeitsverzeichnis.


ZIP-Datei entpacken

-  Entpacken Sie die ZIP-Datei mit Ihrem Entpackprogramm in ein Arbeitsverzeichnis für das Siemens TIA Portal.

Bibliothek öffnen und Bausteine in Projekt übertragen

1.  Starten Sie das Siemens TIA Portal mit Ihrem Projekt.
2.  Wechseln sie in die *Projektansicht*.
3.  Wählen Sie auf der rechten Seite die Task-Card "Bibliotheken".
4.  Klicken Sie auf "Globale Bibliothek".
5.  Klicken Sie auf "Globale Bibliothek öffnen".
6.  Navigieren Sie zu ihrem Arbeitsverzeichnis und laden Sie die Datei ..._TIA.al1x.



7.  Kopieren Sie die erforderlichen Bausteine aus der Bibliothek in das Verzeichnis "Programmbausteine" in der *Projektnavigation* Ihres Projekts. Nun haben Sie in Ihrem Anwenderprogramm Zugriff auf die VIPA-spezifischen Bausteine.

4 Gebäude-Automatisierung - "Building Control"

4.1 Übersicht

In diesem Kapitel finden Sie die Funktionsbausteine (FB45 ... FB50) für die Gebäudeleittechnik (GLT). Die Bausteine verwenden die Systemzeit der CPU. Es sind keine S7-Timer erforderlich. Sie haben die Möglichkeit für jeden Baustein einen Instanz-Datenbaustein oder Multiinstanzen zu verwenden. Es gibt folgende Bausteine:

FB		Beschreibung
FB 45	LAMP	Steuern einer Leuchte oder Steckdose
FB 46	BLIND	Steuern einer Jalousie
FB 47	DSTRIKE	Steuern eines elektrischen Türöffners
FB 48	ACONTROL	Zutrittssteuerung
FB 49	KEYPAD	Abfrage eines Tastenfelds mit externer Spannungsversorgung
FB 50	KEYPAD2	Abfrage eines Tastenfelds ohne externe Spannungsversorgung

4.1.1 Aufruf-Beispiel - Instanz-DB

```

Netzwerk 1          CALL "Deckenleuchte", DB 1
                      ON           :=M20.0
                      OFF          :=20.1
                      ONOFF        :=20.2
                      Duration     :=T#5M
                      Output       :=M20.3
                      PulseOn      :=
                      PulseOff     :=

```

4.1.2 Aufruf-Beispiel - Multiinstanzen-DB

Inhalt von: "Umgebung \Schnittstelle\Stat" Nachfolgend sehen Sie einen Beispielaufruf für mehrere Leuchten und eine Jalousie in AWL mit Multiinstanzen.

Name	Datentyp	Adresse
Deckenleuchte	LAMP	0.0
Stehleuchte	LAMP	46.0
Spiegelleuchte	LAMP	92.0
Jalousie	BLIND	138.0

```

Netzwerk 1          CALL #Deckenleuchte
                      ON           :=M20.0
                      OFF          :=20.1
                      ONOFF        :=20.2

```

Raumsteuerung - Room > FB 45 - LAMP - Leuchte / Steckdose steuern

```

Duration :=T#5M
Output   :=M20.3
PulseOn  :=
PulseOff :=

```

Netzwerk 2

```

CALL #Jalousie
Up      :=M30.0
Down    :=M30.1
CentralUp   :=
CentralDown :=
TimeMaxDuration :=T#10S
TimePause      :=T#1S
TimeShortLong  :=T#2S
Endable        :=
BlindUp        :=M30.6
BlindDown      :=M30.7

```

4.2 Raumsteuerung - "Room"**4.2.1 FB 45 - LAMP - Leuchte / Steckdose steuern****Beschreibung**

Mit diesem Baustein können Sie Lastrelais für Leuchten und Steckdosen ansteuern. Die Ansteuerung kann über einen Ein/Aus-Taster oder über getrennte Ein- und Aus-Taster erfolgen. Zusätzlich haben Sie die Möglichkeit über *Duration* eine Zeit für das automatische Abschalten vorzugeben. Über *TimeDebounce* können Sie für die Eingangssignale eine Entprellzeit vorgeben.

- Bei Ansteuerung eines monostabilen Relais bleibt ein Ausgang solange gesetzt, solange das Relais aktiv sein soll. Mit einem Flankenwechsel 0-1 an *OnOff* bzw. *On* wird der statische Ausgang *Output* gesetzt. Dieser bleibt solange gesetzt, bis Sie diesen mit Flankenwechsel 0-1 an *OnOff* bzw. *Off* wieder zurücksetzen oder die über *Duration* vorgegebene Zeit abgelaufen ist.
- Bei Ansteuerung eines bistabilen Relais werden 2 Ausgänge verwendet. Hierbei steuert *PulseOn* den Einschaltvorgang und *PulseOff* den Ausschaltvorgang. Über *TimePulse* geben Sie die Impulsdauer und über *TimePause* die Umschaltzeit der beiden Ausgänge vor.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
OnOff	INPUT	BOOL	Mit Flankenwechsel 0-1 wird der Ausgang <i>Output</i> aktiviert bzw. deaktiviert und <i>PulseOn</i> oder <i>PulseOff</i> aktiviert. Default: FALSE
On	INPUT	BOOL	Mit Flankenwechsel 0-1 wird der Ausgang <i>Output</i> bzw. <i>PulseOn</i> aktiviert. Default: FALSE
Off	INPUT	BOOL	Mit Flankenwechsel 0-1 wird der Ausgang <i>Output</i> deaktiviert und <i>PulseOff</i> aktiviert. Default: FALSE

Parameter	Deklaration	Datentyp	Beschreibung
Duration	INPUT	TIME	Zeit für die Dauer, nach der der Ausgang <i>Output</i> deaktiviert bzw. <i>PulseOff</i> aktiviert wird. Mit Angabe von 0ms ist das automatische Abschalten deaktiviert. Default: 0ms
Output	OUTPUT	BOOL	Statischer Ausgang zur Ansteuerung eines monostabilen Relais.
PulseOn	OUTPUT	BOOL	Impulsausgang zur Ansteuerung eines bistabilen Relais (Ein-Signal).
PulseOff	OUTPUT	BOOL	Impulsausgang zur Ansteuerung eines bistabilen Relais (Aus-Signal).
TimeDebounce	CONSTANT	TIME	Zeitvorgabe zum Entprellen der Eingänge. Default: 100ms
TimePulse	CONSTANT	TIME	Zeitvorgabe für die Impulsdauer von <i>PulseOn</i> bzw. <i>PulseOff</i> . Default: 100ms
TimePause	CONSTANT	TIME	Zeitvorgabe für die Pause zwischen dem Rücksetzen und Setzen von <i>PulseOn</i> bzw. <i>PulseOff</i> . Default: 100ms

4.2.2 FB 46 - BLIND - Jalousie steuern

Beschreibung

Mit diesem Baustein können Sie eine motorisch angetriebene Jalousie ansteuern. Hierzu müssen Sie den Antrieb mit *Enable* freigeben.

- Die Ansteuerung für "Heben" *BlindUp* und "Senken" *BlindDown* hat über 2 Taster (*Up/Down* bzw. *CentralUp/CentralDown*) zu erfolgen.
 - *CentralUp/CentralDown*: Dient zur zentralen Ansteuerung aller Jalousien eines Gebäudes
 - *Up/Down*: Dient zur lokalen Ansteuerung einer Jalousie. Hier wird ein anstehendes *CentralUp/CentralDown*-Signal ignoriert.
- Wird der entsprechende Taster länger als die Zeit *TimeShortLong* gedrückt, fährt der Jalousie-Motor in die entsprechende Endlage. Durch Tippen eines der beiden Taster (*Up/Down* bzw. *CentralUp/CentralDown*) können Sie die Bewegung stoppen und diese gegebenenfalls umkehren.
- Mit *TimeMaxDuration* geben Sie die Maximal Laufzeit des Motors vor und mit *TimePause* die Pause für den Richtungswechsel.
- Durch Tippen fährt der Jalousie-Motor nur kurz an. Mit dieser Funktion können Sie die Jalousie-Lamellen fein justieren.
- Über *TimeDebounce* können Sie für die Eingangssignale eine Entprellzeit vorgeben.
- Über Status können Sie die Stellung der Jalousie abfragen
 - 0: Endposition oben
 - 50: Unbestimmte Position zwischen den beiden Endpositionen
 - 100: Endposition unten



VORSICHT!

Der Jalousie-Motor muss über eigene Endlagenschalter verfügen, die diesen selbsttätig abschalten!

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Up	INPUT	BOOL	Mit Flankenwechsel 0-1 wird der Ausgang <i>BlindUp</i> aktiviert. Abhängig vom Eingangs-Signal fährt die Jalousie in die Endlage oben oder wird nur kurz bewegt. Solange dieses Signal ansteht werden die Signale <i>CentralUp/CentralDown</i> ignoriert. Default: FALSE
Down	INPUT	BOOL	Mit Flankenwechsel 0-1 wird der Ausgang <i>Blind-Down</i> aktiviert. Abhängig vom Eingangs-Signal fährt die Jalousie in die Endlage unten oder wird nur kurz bewegt. Solange dieses Signal ansteht werden die Signale <i>CentralUp/CentralDown</i> ignoriert. Default: FALSE
CentralUp	INPUT	BOOL	Mit Flankenwechsel 0-1 wird der Ausgang <i>BlindUp</i> aktiviert. Hierbei fährt die Jalousie in die Endlage oben. Default: FALSE
CentralDown	INPUT	BOOL	Mit Flankenwechsel 0-1 wird der Ausgang <i>Blind-Down</i> aktiviert. Hierbei fährt die Jalousie in die Endlage unten. Default: FALSE
TimeMaxDuration	INPUT	TIME	Maximale Laufzeit des Motors für die Fahrt in die jeweilige Endlage. Default: 30s
TimePause	INPUT	TIME	Zeitvorgabe für die Pause für den Richtungswechsel. Default: 2s
TimeShortLong	INPUT	TIME	Zeitvorgabe für die Unterscheidung zwischen Tipp- und Dauer-Betrieb. Default: 1s
Enable	INPUT	BOOL	Freigabe für den Antrieb (statisch) Default: TRUE
BlindUp	OUTPUT	BOOL	Statischer Ausgang Jalousie "heben".
BlindDown	OUTPUT	BOOL	Statischer Ausgang Jalousie "senken".
Status	OUTPUT	INT	<ul style="list-style-type: none"> ■ Status - Position der Jalousie <ul style="list-style-type: none"> – 0: Endposition oben – 50: Unbestimmte Position zwischen den beiden Endpositionen – 100: Endposition unten
TimeDebounce	CONSTANT	TIME	Zeitvorgabe zum Entprellen der Eingänge. Default: 100ms

4.2.3 FB 47 - DSTRIKE - Elektrischer Türöffner

Beschreibung

Mit diesem Baustein können Sie einen elektrischen Türöffner ansteuern, sofern dieser nicht mit *DoorIsLocked* "verriegelt" ist.

- Mit einem Flankenwechsel 0-1 am Eingang *Open* wird für die Zeit "*TimeOpening*" der Ausgang "*Output*" angesteuert.
- Mit einem Flankenwechsel 0-1 des Eingangssignals *EnableAlwaysOpen* bzw. *DisableAlwaysOpen* wird der Ausgang *Open* dauerhaft aktiviert bzw. deaktiviert. Zusätzlich wird bei gesetztem *EnableAlwaysOpen* der statische Ausgang *AlwaysOpen* gesetzt.
- An die Eingangssignale *DoorIsClosed* und *DoorIsLocked* können Sie Ihre Tür-Kontakte anbinden. *DoorIsClosed* wird gesetzt, sobald Ihre Türe geschlossen ist. *DoorIsLocked* ist bei verriegelter Tür aktiv, d.h. der Kontakt wird über den Schließmechanismus ausgelöst und das Öffnen mittels Türöffner ist nicht möglich.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Open	INPUT	BOOL	Mit Flankenwechsel 0-1 wird der Ausgang <i>Output</i> für die Zeit <i>TimeOpening</i> aktiviert. Default: FALSE
EnableAlwaysOpen	INPUT	BOOL	Mit Flankenwechsel 0-1 wird der Ausgang <i>Output</i> dauerhaft angesteuert. Default: FALSE
DisableAlwaysOpen	INPUT	BOOL	Mit Flankenwechsel 0-1 wird der Ausgang <i>Output</i> dauerhaft deaktiviert. Default: FALSE
TimeOpening	INPUT	TIME	Zeit für die Dauer der Ansteuerung von <i>Output</i> . Default: 3s
DoorIsClosed	INPUT	BOOL	<ul style="list-style-type: none"> ■ Optional - Position der Tür TRUE: Tür ist geschlossen FALSE: Tür ist geöffnet Default: FALSE
DoorIsLocked	INPUT	BOOL	<ul style="list-style-type: none"> ■ Optional - Schließzustand der Tür - TRUE: Tür ist verriegelt - FALSE: Tür ist nicht verriegelt Default: FALSE
Output	OUTPUT	BOOL	Statischer Ausgang zur Ansteuerung eines monostabilen Relais.
AlwaysOpen	OUTPUT	BOOL	Statischer Ausgang für Anzeige "Tür ist dauerhaft geöffnet".

4.3 Zugangskontrolle - "Access Control"

4.3.1 FB 48 - ACONTROL - Zutrittssteuerung

Beschreibung

Mit diesem Baustein können Sie eine Zugangssteuerung realisieren. Nach Vorgabe eines Codes von einem externen Tastenfeld, Panel oder RFID-Lesers wird dieser Code mit einer Liste verglichen. Je nach Ergebnis werden daraufhin entsprechende Ausgänge angesteuert.

- Die ZugangsCodes sind in einem Datenbaustein anzulegen, welchen Sie über *ACL-Block* angeben. Hier bestimmen Sie auch welche Ausgänge *Access1...6* angesteuert werden und wie (Impuls/statisch) diese angesteuert werden sollen. Mit dem Datenbaustein können Sie bis zu 16 ZugangsCodes verwalten.
- Über *AccessCode1...4* geben Sie den Code des entsprechenden Eingabegeräts vor.
- Mit *CheckCode1...4* wird der Code mit den Codes in Ihrem Datenbaustein *ACLBlock* verglichen.
 - Ist der Zugangscode im Datenbaustein vorhanden, werden die entsprechenden Ausgänge nach Vorgabe angesteuert. Bei projektierte Impuls-Ausgabe können Sie über *TimePulse* die Impulsdauer vorgeben.
 - Ist der Zugangscode im Datenbaustein nicht vorhanden, wird für die Zeit *TimeError* der Ausgang *Error* gesetzt.
- Mit einem Flankenwechsel 0-1 an *CentralLock* werden alle ZugangsCodes deaktiviert. Hierbei wird der Ausgang *CentralLocked* gesetzt.
- Mit einem Flankenwechsel 0-1 an *CentralUnlock* werden alle ZugangsCodes aktiviert und der Ausgang *CentralLocked* zurückgesetzt.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
AccessCode1	INPUT	STRING[16]	Zutrittscode, z.B. von Keypad.
CheckCode1	INPUT	BOOL	Mit Flankenwechsel 0-1 wird der <i>AccessCode1</i> mit dem Zugangscode im Datenbaustein <i>ACL-Block</i> verglichen. Default: 0
AccessCode2	INPUT	STRING[16]	Zutrittscode, z.B. von Panel
CheckCode2	INPUT	BOOL	Mit Flankenwechsel 0-1 wird der <i>AccessCode2</i> mit dem Zugangscode im Datenbaustein <i>ACL-Block</i> verglichen. Default: 0
AccessCode3	INPUT	STRING[16]	Zutrittscode, z.B. von RFID-Lesegerät
CheckCode3	INPUT	BOOL	Mit Flankenwechsel 0-1 wird der <i>AccessCode3</i> mit dem Zugangscode im Datenbaustein <i>ACL-Block</i> verglichen. Default: 0
AccessCode4	INPUT	STRING[16]	Zutrittscode, z.B. von sonstigem System
CheckCode4	INPUT	BOOL	Mit Flankenwechsel 0-1 wird der <i>AccessCode4</i> mit dem Zugangscode im Datenbaustein <i>ACL-Block</i> verglichen. Default: 0
CentralLock	INPUT	BOOL	Mit Flankenwechsel 0-1 werden alle ZugangsCodes deaktiviert. Hierbei wird der Ausgang <i>CentralLocked</i> gesetzt.

Parameter	Deklaration	Datentyp	Beschreibung
CentralUnlock	INPUT	BOOL	Mit Flankenwechsel 0-1 an <i>CentralUnlock</i> werden alle Zugangscodes aktiviert und der Ausgang <i>CentralLocked</i> zurückgesetzt.
ACLBlock	INPUT	BLOCK	Datenbaustein, in dem die Zugangscodes hinterlegt sind vom Typ UDT 4 - ACL. ↪ <i>Kapitel 4.3.3 "UDT 4 - ACL - Datenstruktur für FB 48" auf Seite 16</i>
Access1	OUTPUT	BOOL	Ausgang 1, kann als Impuls oder statisch angesteuert werden.
Access2	OUTPUT	BOOL	Ausgang 2, kann als Impuls oder statisch angesteuert werden.
Access3	OUTPUT	BOOL	Ausgang 3, kann als Impuls oder statisch angesteuert werden.
Access4	OUTPUT	BOOL	Ausgang 4, kann als Impuls oder statisch angesteuert werden.
Access5	OUTPUT	BOOL	Ausgang 5, kann als Impuls oder statisch angesteuert werden.
Access6	OUTPUT	BOOL	Ausgang 6, kann als Impuls oder statisch angesteuert werden.
Error	OUTPUT	BOOL	Ist der Zugangscodes im Datenbaustein nicht vorhanden, wird für die Zeit <i>TimeError</i> der Ausgang <i>Error</i> gesetzt.
CentralLocked	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Zugang <ul style="list-style-type: none"> – TRUE: verriegelt - Zugang nicht möglich – FALSE: nicht verriegelt - Zugang möglich Default: TRUE
TimePulse	CONSTANT	Time	Zeit für die Impulsdauer an einem Ausgang. Default: 3s
TimeError	CONSTANT	Time	Zeit für die Dauer des Error-Signals. Default: 500ms

4.3.2 UDT 3 - ACLREC - Datenstruktur für FB 48

Beschreibung

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	Code	STRING[16]	''	Byte 0 ... 17: Zugangscodes S7String mit max. 16 ASCII-Zeichen für Zugriffscodes
+18.0	EnableOutput1	BOOL	FALSE	Byte 18: Signal für anzusteuernde Ausgänge TRUE: aktiviere Ausgang, FALSE: deaktiviere Ausgang
+18.1	EnableOutput2	BOOL	FALSE	

Zugangskontrolle - Access Control > UDT 4 - ACL - Datenstruktur für FB 48

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+18.2	EnableOutput3	BOOL	FALSE	
+18.3	EnableOutput4	BOOL	FALSE	
+18.4	EnableOutput5	BOOL	FALSE	
+18.5	EnableOutput6	BOOL	FALSE	
+18.6	EnableRes7	BOOL	FALSE	
+18.7	EnableRes8	BOOL	FALSE	
+19.0	SignalOutput1	BOOL	FALSE	Byte 19: Signaltyp FALSE: Impuls, TRUE: statische 1, Deaktivierung mit weiterem Code
+19.1	SignalOutput2	BOOL	FALSE	
+19.2	SignalOutput3	BOOL	FALSE	
+19.3	SignalOutput4	BOOL	FALSE	
+19.4	SignalOutput5	BOOL	FALSE	
+19.5	SignalOutput6	BOOL	FALSE	
+19.6	SignalRes7	BOOL	FALSE	
+19.7	SignalRes8	BOOL	FALSE	
=20.0				

4.3.3 UDT 4 - ACL - Datenstruktur für FB 48

Beschreibung

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	RecordCount	INT	16	DBW0: Anzahl gültiger Datensätze (0 ... n)
+2.0	RecordLen	INT	20	DBW2: Länge eines Datensatzes in Bytes (20)
+4.0	Record	ARRAY[0...15]		Ab DBB4 beginnt der erste Datensatz
*20.0		"UDT 3 - ACLREC"		↪ Kapitel 4.3.2 "UDT 3 - ACLREC - Datenstruktur für FB 48" auf Seite 15
=324.0		BOOL		



VORSICHT!

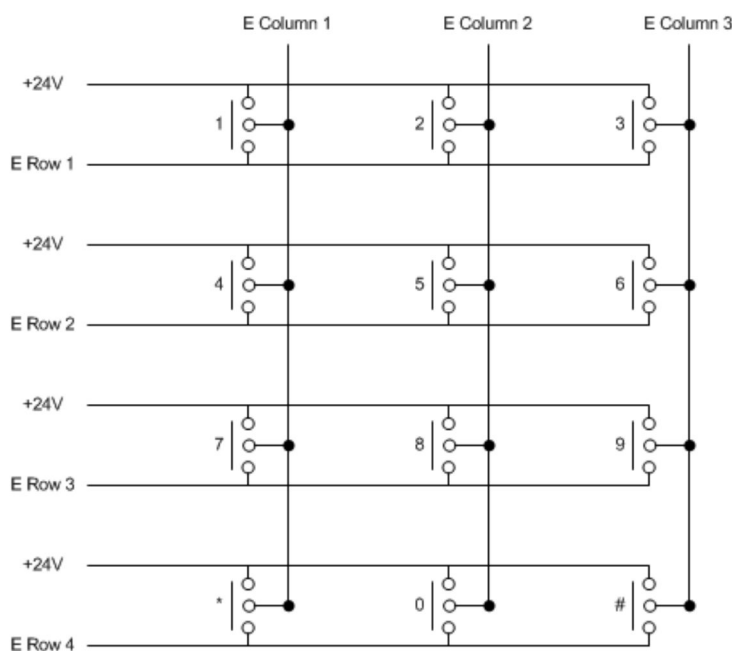
Ein Code darf nur 1 x in der ganzen Liste vorkommen. Doppelte Codes sind unzulässig.

4.3.4 FB 49 - KEYPAD - Tastatur

Beschreibung

Dieser Baustein dient zur Anbindung eines externen Tastenfelds (0...9,*,#) mit externer DC 24V Spannungsversorgung. Abhängig von der betätigten Taste liefert die Tastatur Zeilen und Spaltensignale (24V). Der Baustein wertet die Signale intern mittels einer Bitmuster-Tabelle aus und übergibt den ermittelten ASCII-Code in den Tastaturpuffer. Bei Bedarf oder automatisch wird der Tastaturpuffer als max. 16Byte lange Zeichenkette ausgegeben.

- Über *Row 1...4* werden die Zeilen 1...4 der Tastaturmatrix angebinden.
- Über *Column 1...3* werden die Spalten 1...3 der Tastaturmatrix angebinden.
- Über *ClearCode* können Sie einen Tastencode vorgeben, mit dem der Eingabepuffer gelöscht wird.
- Über *EnterCode* können Sie einen Tastencode vorgeben, mit dem der Tastaturpuffer für die Länge eines Zyklus an *Output* ausgegeben wird. Während dieser Zeit wird der Ausgang *Valid* aktiv.
- Über Flankenwechsel 0-1 an *Clear* wird der Tastaturpuffer gelöscht.
- Über *TimeAutoClear* bestimmen Sie Zeit, innerhalb der eine Taste gedrückt werden muss. Ansonsten wird der Tastaturpuffer wieder gelöscht.
- Über *CountCharAutoEnter* können Sie die Anzahl der Zeichen vorgeben, nach deren Eingabe der Tastaturpuffer für die Länge eines Zyklus an *Output* ausgegeben wird. Während dieser Zeit wird der Ausgang *Valid* aktiv.
- *Error* wird für die Zeit *TimeError* aktiviert, wenn eine Taste betätigt wird, der Tastaturpuffer aber voll ist.
- Über *TimeDebounce* können Sie für die Eingangssignale eine Entprellzeit vorgeben.



Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Row1	INPUT	BOOL	Zeile 1 der Tastaturmatrix. Default: FALSE
Row2	INPUT	BOOL	Zeile 2 der Tastaturmatrix. Default: FALSE

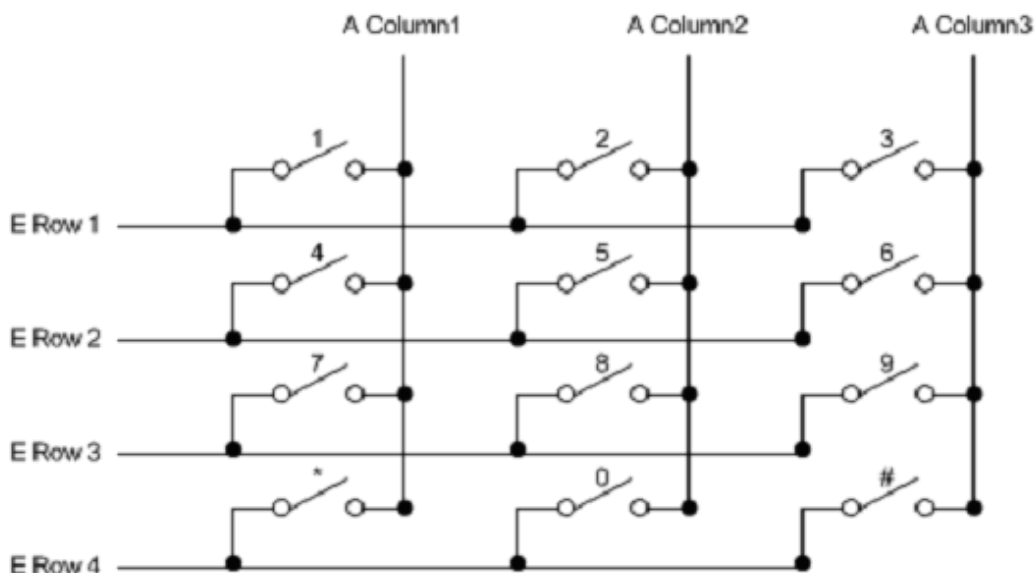
Parameter	Deklaration	Datentyp	Beschreibung
Row3	INPUT	BOOL	Zeile 3 der Tastaturmatrix. Default: FALSE
Row4	INPUT	BOOL	Zeile 4 der Tastaturmatrix. Default: FALSE
Column1	INPUT	BOOL	Spalte 1 der Tastaturmatrix. Default: FALSE
Column2	INPUT	BOOL	Spalte 2 der Tastaturmatrix. Default: FALSE
Column3	INPUT	BOOL	Spalte 3 der Tastaturmatrix. Default: FALSE
ClearCode	INPUT	BYTE	Der Wert, bei dem der Tastaturpuffer gelöscht werden soll. 0: deaktiviert Default: 42 = *
EnterCode	INPUT	BYTE	Der Wert, bei dem der Tastaturpuffer ausgegeben werden soll. 0: deaktiviert Default: 35 = #
Clear	INPUT	BOOL	Flankenwechsel 0-1 löscht den Tastaturpuffer. Default: FALSE
TimeAutoClear	INPUT	TIME	Zeitvorgabe für die Dauer innerhalb der eine Taste gedrückt werden muss. Ansonsten wird der Tastaturpuffer wieder gelöscht. 0: keine Löschung Default: 10s
CountCharAutoEnter	INPUT	INT	Anzahl der Zeichen nach deren Eingabe der Tastaturpuffer automatisch ausgegeben wird. 0: deaktiviert Default: 0
Output	OUTPUT	STRING[16]	Inhalt des Tastaturpuffers als max. 16 Byte lange Zeichenkette.
Valid	OUTPUT	BOOL	Statischer Ausgang, zeigt an, dass der über <i>Output</i> ausgegebene String gültig ist. Das Signal steht für einen Zyklus an.
Error	OUTPUT	BOOL	<i>Error</i> wird für die Zeit <i>TimeError</i> aktiviert, wenn eine Taste betätigt wird, der Tastaturpuffer aber voll ist.
TimeDebounce	CONSTANT	TIME	Zeitvorgabe zum Entprellen der Eingänge. Default: 100ms
TimeError	CONSTANT	TIME	Zeit für die Dauer des Error-Signals Default: 500ms

4.3.5 FB 50 - KEYPAD2 - Tastatur

Beschreibung

Dieser Baustein dient zur Anbindung eines externen Tastenfelds (0...9,*,#) ohne eigene Spannungsversorgung. Der Baustein liefert Ausgangs-Spalten-Signale. Abhängig von der betätigten Taste liefert die Tastatur das entsprechende Zeilen-Signal. Der Baustein wertet das Signal intern mittels einer Bitmuster-Tabelle aus und übergibt den ermittelten ASCII-Code in den Tastaturpuffer. Bei Bedarf oder automatisch wird der Tastaturpuffer als max. 16Byte lange Zeichenkette ausgegeben.

- Über die Eingänge *Row 1...4* werden die Zeilen 1...4 der Tastaturmatrix angeben.
- Über die Ausgänge *Column 1...3* werden die Spalten 1...3 der Tastaturmatrix angeben.
- Über *TimeDelay* geben Sie eine Wartezeit nach dem Setzen der Spalten-Ausgänge bis zum Einlesen der entsprechenden Zeilen-Eingänge vor. Diese Zeit muss größer sein als die Verzögerungszeit der verwendeten Baugruppe.
- Über *ClearCode* können Sie einen Tastencode vorgeben, mit dem der Eingabepuffer gelöscht wird.
- Über *EnterCode* können Sie einen Tastencode vorgeben, mit dem der Tastaturpuffer für die Länge eines Zyklus an *Output* ausgegeben wird. Während dieser Zeit wird der Ausgang *Valid* aktiv.
- Über Flankenwechsel 0-1 an *Clear* wird der Tastaturpuffer gelöscht.
- Über *TimeAutoClear* bestimmen Sie Zeit, innerhalb der eine Taste gedrückt werden muss. Ansonsten wird der Tastaturpuffer wieder gelöscht.
- Über *CountCharAutoEnter* können Sie die Anzahl der Zeichen vorgeben, nach deren Eingabe der Tastaturpuffer für die Länge eines Zyklus an *Output* ausgegeben wird. Während dieser Zeit wird der Ausgang *Valid* aktiv.
- *Error* wird für die Zeit *TimeError* aktiviert, wenn eine Taste betätigt wird, der Tastaturpuffer aber voll ist.
- Über *TimeDebounce* können Sie für die Eingangssignale eine Entprellzeit vorgeben.



Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Row1	INPUT	BOOL	Zeile 1 der Tastaturmatrix. Default: FALSE
Row2	INPUT	BOOL	Zeile 2 der Tastaturmatrix. Default: FALSE

Parameter	Deklaration	Datentyp	Beschreibung
Row3	INPUT	BOOL	Zeile 3 der Tastaturmatrix. Default: FALSE
Row4	INPUT	BOOL	Zeile 4 der Tastaturmatrix. Default: FALSE
ClearCode	INPUT	BYTE	Der Wert, bei dem der Tastaturpuffer gelöscht werden soll. 0: deaktiviert Default: 42 = *
EnterCode	INPUT	BYTE	Der Wert, bei dem der Tastaturpuffer ausgegeben werden soll. 0: deaktiviert Default: 35 = #
Clear	INPUT	BOOL	Flankenwechsel 0-1 löscht den Tastaturpuffer. Default: FALSE
TimeAutoClear	INPUT	TIME	Zeitvorgabe für die Dauer innerhalb der eine Taste gedrückt werden muss. Ansonsten wird der Tastaturpuffer wieder gelöscht. 0: keine Löschung Default: 10s
CountCharAutoEnter	INPUT	INT	Anzahl der Zeichen nach deren Eingabe der Tastaturpuffer automatisch ausgegeben wird. 0: deaktiviert Default: 0
Column1	OUTPUT	BOOL	Spalte 1 der Tastaturmatrix. Default: FALSE
Column2	OUTPUT	BOOL	Spalte 2 der Tastaturmatrix. Default: FALSE
Column3	OUTPUT	BOOL	Spalte 3 der Tastaturmatrix. Default: FALSE
Output	OUTPUT	BYTE	Inhalt des Tastaturpuffers als max. 16 Byte lange Zeichenkette.
Valid	OUTPUT	BOOL	Statischer Ausgang, zeigt an, dass der über <i>Output</i> ausgegebene String gültig ist. Das Signal steht für einen Zyklus an.
Error	OUTPUT	BOOL	<i>Error</i> wird für die Zeit <i>TimeError</i> aktiviert, wenn eine Taste betätigt wird, der Tastaturpuffer aber voll ist.
TimeDebounce	CONSTANT	TIME	Zeitvorgabe zum Entprellen der Eingänge. Default: 100ms

Parameter	Deklaration	Datentyp	Beschreibung
TimeError	CONSTANT	TIME	Zeit für die Dauer des Error-Signals Default: 500ms
TimeDelay	CONSTANT	TIME	Zeitvorgabe für die Dauer nach dem nach dem Setzen der Spalten-Ausgänge bis zum Einlesen der entsprechenden Zeilen-Eingänge vor. Diese Zeit muss größer sein als die Verzögerungszeit der verwendeten Baugruppe. Default: 10ms