YASKAWA

VIPA SPEED7 Library

OPL_SP7-LIB | SW90ES0MA V10.003 | Manual

HB00 | OPL_SP7-LIB | SW90ES0MA V10.003 | en | 18-45

Block library - Building Control

VIPA CONTROLS

VIPA GmbH

Ohmstr. 4

91074 Herzogenaurach

Telephone: +49 9132 744-0

Fax: +49 9132 744-1864

Email: info@vipa.com

Internet: www.vipa.com

Table of contents

1   General

1.1 Copyright © VIPA GmbH

All Rights Reserved

This document contains proprietary information of VIPA and is not to be disclosed or used except in accordance with applicable agreements.

This material is protected by the copyright laws. It may not be reproduced, distributed, or altered in any fashion by any entity (either internal or external to VIPA), except in accordance with applicable agreements, contracts or licensing, without the express written consent of VIPA and the business management owner of the material.

For permission to reproduce or distribute, please contact: VIPA, Gesellschaft für Visualisierung und Prozessautomatisierung mbH Ohmstraße 4, D-91074 Herzogenaurach, Germany

Tel.: +49 9132 744 -0

Fax.: +49 9132 744-1864

EMail: info@vipa.de

http://www.vipa.com

> *Every effort has been made to ensure that the information contained in this document was complete and accurate at the time of publishing. Nevertheless, the authors retain the right to modify the information.*
>
> *This customer document describes all the hardware units and functions known at the present time. Descriptions may be included for units which are not present at the customer site. The exact scope of delivery is described in the respective purchase contract.*

CE Conformity Declaration

Hereby, VIPA GmbH declares that the products and systems are in compliance with the essential requirements and other relevant provisions. Conformity is indicated by the CE marking affixed to the product.

Conformity Information

For more information regarding CE marking and Declaration of Conformity (DoC), please contact your local VIPA customer service organization.

Trademarks

VIPA, SLIO, System 100V, System 200V, System 300V, System 300S, System 400V, System 500S and Commander Compact are registered trademarks of VIPA Gesellschaft für Visualisierung und Prozessautomatisierung mbH.

SPEED7 is a registered trademark of profichip GmbH.

SIMATIC, STEP, SINEC, TIA Portal, S7-300, S7-400 and S7-1500 are registered trademarks of Siemens AG.

Microsoft and Windows are registered trademarks of Microsoft Inc., USA.

Portable Document Format (PDF) and Postscript are registered trademarks of Adobe Systems, Inc.

All other trademarks, logos and service or product marks specified herein are owned by their respective companies.

Information product support

Contact your local VIPA Customer Service Organization representative if you wish to report errors or questions regarding the contents of this document. If you are unable to locate a customer service centre, contact VIPA as follows:

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telefax: +49 9132 744-1204

EMail: documentation@vipa.de

Technical support

Contact your local VIPA Customer Service Organization representative if you encounter problems with the product or have questions regarding the product. If you are unable to locate a customer service centre, contact VIPA as follows:

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Tel.: +49 9132 744-1150 (Hotline)

EMail: support@vipa.de

1.2 About this manual

Objective and contents

The manual describes the block library *'Building Control'* from VIPA:

■ It contains a description of the structure, project implementation and usage in several programming systems.
■ The manual is targeted at users who have a background in automation technology.
■ The manual is available in electronic form as PDF file. This requires Adobe Acrobat Reader.
■ The manual consists of chapters. Every chapter provides a self-contained description of a specific topic.
■ The following guides are available in the manual:
  – An overall table of contents at the beginning of the manual
  – References with pages numbers

Icons Headings

Important passages in the text are highlighted by following icons and headings:

> **DANGER!**
> Immediate or likely danger. Personal injury is possible.

> **CAUTION!**
> Damages to property is likely if these warnings are not heeded.

> *Supplementary information and useful tips.*

2   Important notes

2.1   General

> *In the following, you will find important notes, which must always be observed when using the blocks.*

2.2   Internally used blocks

> **CAUTION!**
> The following blocks are used internally and must not be overwritten! The direct call of an internal block leads to errors in the corresponding instance DB! Please always use the corresponding function for the call.

FC/SFC	Designation	Description
FC/SFC 131	TSEND_	is used internally for FB 63
FC/SFC 132	TRECV_	is used internally for FB 64
FC/SFC 133	TCON_	is used internally for FB 65
FC/SFC 134	TDISCON_	is used internally for FB 66
FC/SFC 135	TUSEND_	is used internally for FB 67
FC/SFC 136	TURECV_	is used internally for FB 68
FC/SFC 192	CP_S_R	is used internally for FB 7 and FB 8
FC/SFC 196	AG_CNTRL	is used internally for FC 10
FC/SFC 198	USEND_	is used internally for FB 8
FC/SFC 198	URCV_	is used internally for FB 9
FC/SFC 200	AG_GET	is used internally for FB/SFB 14
FC/SFC 201	AG_PUT	is used internally for FB/SFB 15
FC/SFC 202	AG_BSEND	is used internally for FB/SFB 12
FC/SFC 203	AG_BRCV	is used internally for FB/SFB 13
FC/SFC 204	IP_CONF	is used internally for FB 55 IP_CONF
FC/SFC 205	AG_SEND	is used internally for FC 5 AG_SEND
FC/SFC 206	AG_RECV	is used internally for FC 6 AG_RECV
FC/SFC 253	IBS_ACCESS	is used internally for SPEED bus INTERBUS masters
SFB 238	EC_RWOD	is used internally for EtherCAT Communication
SFB 239	FUNC	is used internally for FB 240, FB 241

3   Include library

Block library *'Building Control'*

The block library can be found for download in the *'Service/Support'* area of www.vipa.com at *'Downloads* ➔ *VIPA Lib'* as *'Block library Building Control - SW90ES0MA'*. The library is available as packed zip file. As soon as you want to use these blocks you have to import them into your project.

> *Please always use the manual associated with your library. As long as there are no description-relevant changes, the version information in the manual can differ from those of the library and its files.*

The following block libraries are available

File	Description
BuildingControl_S7_V0003.zip	■ Block library for Siemens SIMATIC Manager. ■ For use in CPUs from VIPA or S7-300 CPUs from Siemens.
BuildingControl_TIA_V0005.zip	■ Block library for Siemens TIA Portal V15. ■ For use in CPUs from VIPA or S7-300 CPUs from Siemens.

3.1   Integration into Siemens SIMATIC Manager

Overview

The integration into the Siemens SIMATIC Manager requires the following steps:

1. ▶ Load ZIP file
2. ▶ "Retrieve" the library
3. ▶ Open library and transfer blocks into the project

Load ZIP file

▶ Navigate on the web page to the desired ZIP file, load and store it in your work directory.

Retrieve library

1. ▶ Start the Siemens SIMATIC Manager with your project.
2. ▶ Open the dialog window for ZIP file selection via *'File* ➔ *Retrieve'*.
3. ▶ Select the according ZIP file and click at [Open].
4. ▶ Select a destination folder where the blocks are to be stored.
5. ▶ Start the extraction with [OK].

Open library and transfer blocks into the project

1. ▶ Open the library after the extraction.
2. ▶ Open your project and copy the necessary blocks from the library into the directory "blocks" of your project.
   ⇨ Now you have access to the VIPA specific blocks via your user application.

> *Are FCs used instead of SFCs, so they are supported by the VIPA CPUs starting from firmware 3.6.0.*

3.2 Integration into Siemens TIA Portal

Overview

The integration into the Siemens TIA Portal requires the following steps:

1. ▶ Load ZIP file
2. ▶ Unzip the Zip file
3. ▶ "Retrieve" the library
4. ▶ Open library and transfer blocks into the project

Load ZIP file

1. ▶ Navigate on the web page to the ZIP file, that matches your version of the program.
2. ▶ Load and store it in your work directory.

Unzip the Zip file

▶ Unzip the zip file to a work directory of the Siemens TIA Portal with your unzip application.

Open library and transfer blocks into the project

1. ▶ Start the Siemens TIA Portal with your project.
2. ▶ Switch to the *Project view*.
3. ▶ Choose "Libraries" from the task cards on the right side.
4. ▶ Click at "Global libraries".
5. ▶ Click at "Open global libraries".
6. ▶ Navigate to your work directory and load the file ..._TIA.al1x.
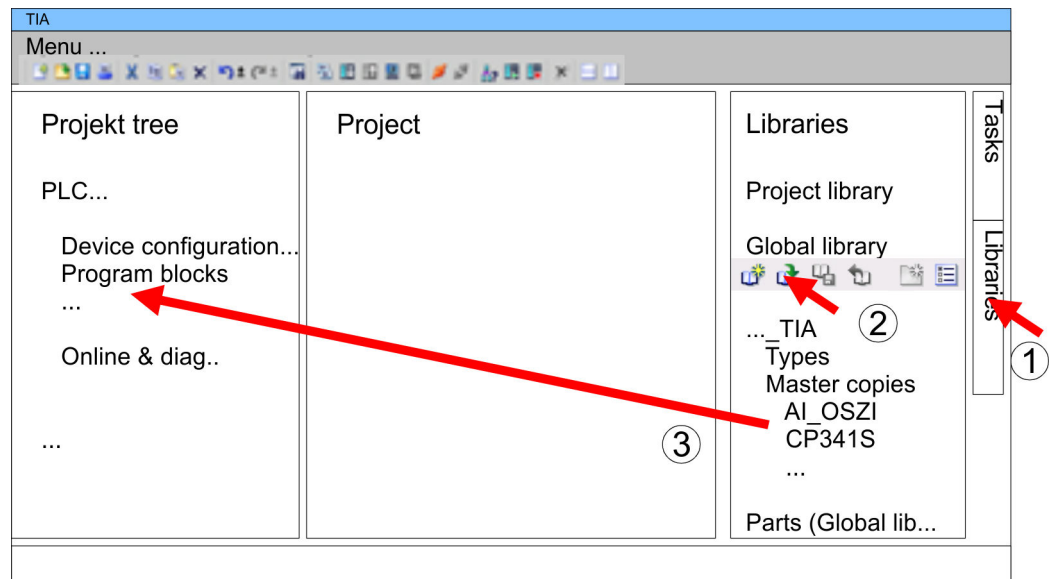
7. ▶ Copy the necessary blocks from the library into the "Program blocks" of the *Project tree* of your project. Now you have access to the VIPA specific blocks via your user application.

4 Building Control

4.1 Overview

In this chapter the function blocks (FB45 ... FB50) for building control (GLT) can be found. The blocks use the system time of the CPU. There are no S7 timers required. You have the option to use for each block an instance data block or multiple instances. There are the following blocks:

FB		Description
FB 45	LAMP	Controlling a lamp or socket
FB 46	BLIND	Controlling blind
FB 47	DSTRIKE	Controlling an electric door opener
FB 48	ACONTROL	Access control
FB 49	KEYPAD	Requesting a keypad with external power supply
FB 50	KEYPAD2	Requesting a keypad without external power supply

4.1.1 Call example - instance DB

Network 1

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

4.1.2 Call example - multi instances DB

Content of: "Environment \Interface\Stat"

In the following there is a STL call example of the usage of multiple lights and a blind with multiple instances.

Name	Data type	Address
Ceiling lamp	LAMP	0.0
Floor lamp	LAMP	46.0
Mirror lamp	LAMP	92.0
Blind	BLIND	138.0

Network 1

```
CALL #Ceiling lamp
  ON        :=M20.0
  OFF       :=20.1
  ONOFF     :=20.2
```

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

Network 2
```
                    CALL #Blind
                      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 Room

4.2.1 FB 45 - LAMP - Controlling lamp / socket

Description

With this block you can control load relays for lamps and sockets. It can be controlled via On/Off button or via separate On and Off button. Additionally with *Duration* you have the possibility to set a duration for the automatic switching-off. With *TimeDebounce* you can specify a debounce time for the input signals.

■ When driving a monostable relay the output remains set as long as the relay is to be activated. With an edge change 0-1 at *OnOff* respectively *On* the static output *Output* is set. It remains set until you reset it with an edge change 0-1 at *OnOff* respectively *Off* or the time of *Duration* has expired.

■ When controlling a bistable relay 2 outputs are used. Here *PulseOn* controls the switch on and *PulseOff* the switch off procedure. With *TimePulse* the pulse duration and with *TimePause* the switch time of the two outputs can be specified.

Parameters

Parameter	Declaration	Data type	Description
OnOff	INPUT	BOOL	With an edge change 0-1 *Output* is activated respectively deactivated and *PulseOn* or *PulseOff* is activated. Default: FALSE
On	INPUT	BOOL	With an edge change 0-1 *Output* is activated respectively deactivated and *PulseOn* is activated. Default: FALSE
Off	INPUT	BOOL	With an edge change 0-1 *Output* is deactivated and *PulseOff* is activated. Default: FALSE
Duration	INPUT	TIME	Time for the duration the *Output* is deactivated respectively *PulseOff* is activated. With 0ms the automatic switch off is deactivated. Default: 0ms
Output	OUTPUT	BOOL	Static output to drive a monostable relay.

Parameter	Declaration	Data type	Description
PulseOn	OUTPUT	BOOL	Pulse output to control the bistable relay (On signal).
PulseOff	OUTPUT	BOOL	Pulse output to control the bistable relay (Off signal).
TimeDebounce	CONSTANT	TIME	Time for debounce of the inputs. Default: 100ms
TimePulse	CONSTANT	TIME	Time for the pulse duration of *PulseOn* respectively *PulseOff*. Default: 100ms
TimePause	CONSTANT	TIME	Time for the break between resetting and setting of *PulseOn* respectively *PulseOff*. Default: 100ms

4.2.2 FB 46 - BLIND - Controlling blind

Description

With this block a motorized blind can be controlled. For this you have to release the drive with *Enable*.

■ The controlling for "lifting" *BlindUp* and "sinking" *BlindDown* happens by 2 buttons (*Up*/*Down* respectively *CentralUp*/*CentralDown*).
  – *CentralUp*/*CentralDown*: Used for central control of all blinds in a building.
  – *Up*/*Down*: Used for local control of a blind. Here a pending *CentralUp*/*Central-Down* signal is ignored.
■ If the corresponding button is pressed longer as the specified *TimeShortLong* the blend drive moves to the respective end position. By pressing on of the two buttons (*Up/Down* respectively *CentralUp/CentralDown*) you can stop the movement and reverse, it if necessary.
■ With *TimeMaxDuration* you can specify the maximum run time of the motor and with *TimePause* you can specify the pause for the change of direction.
■ By jogging the blend drive shortly moves. With this function you can adjust the blind slats fine.
■ With *TimeDebounce* you can specify a debounce time for the input signals.
■ With Status you can check the position of the blend.
  – 0: Upper limit position
  – 50: Unknown position between the two limit positions
  – 100: Lowest limit position

> **CAUTION!**
> The blend drive must have its own limit switches that turn off power automatically!

Parameters

Parameter	Declaration	Data type	Description
Up	INPUT	BOOL	With an edge change 0-1 the output *BlindUp* is activated. Depending on the input signal the blend drives to the upper limit position or is shortly moved. As long as the signal is pending the signals *CentralUp/CentralDown* are ignored. Default: FALSE
Down	INPUT	BOOL	With an edge change 0-1 the output *BlindDown* is activated. Depending on the input signal the blend drives to the lower limit position or is shortly moved. As long as the signal is pending the signals *CentralUp/CentralDown* are ignored. Default: FALSE
CentralUp	INPUT	BOOL	With an edge change 0-1 the output *BlindUp* is activated. Here the blind moves to the upper limit position. Default: FALSE
CentralDown	INPUT	BOOL	With an edge change 0-1 the output *BlindDown* is activated. Here the blind moves to the lowest limit position. Default: FALSE
TimeMaxDuration	INPUT	TIME	Maximum drive time to reach the respective end position. Default: 30s
TimePause	INPUT	TIME	Break between a direction change. Default: 2s
TimeShortLong	INPUT	TIME	Time for the distinction between jog mode and continuous mode. Default: 1s
Enable	INPUT	BOOL	Release for the drive (static) Default: TRUE
BlindUp	OUTPUT	BOOL	Static output blind "lifting"
BlindDown	OUTPUT	BOOL	Static output blind "sinking"

Status	OUTPUT	INT	■ Status - Blind position
			  – 0: Upper limit position
			  – 50: Unknown position between the two limit positions
			  – 100: Lowest limit position

TimeDebounce	CONSTANT	TIME	Time for debounce of the inputs. Default: 100ms

4.2.3 FB 47 - DSTRIKE - Electric door opener

Description

With this block an electric door opener can be controlled, if its not locked with *DoorIsLocked*.

■ With an edge change 0-1 at the input *Open* for the duration *'TimeOpening' 'Output'* is controlled.

■ With an edge change 0-1 at the input *EnableAlwaysOpen* respectively *DisableAlwaysOpen Open* is static activated respectively deactivated. Additionally with set *EnableAlwaysOpen* the static output *AlwaysOpen* is set.

■ You can connect your door contacts at *DoorIsClosed* and *DoorIsLocked*. *DoorIsClosed* is set, as soon as the door is closed. *DoorIsLocked* is set as soon as the door is locked, i.e. the contact is controlled by the locking mechanism of the door and opening of the door by means of the electric door opener is disabled.

Parameters

Parameter	Declaration	Data type	Description
Open	INPUT	BOOL	With an edge change 0-1 *Output* is activated for the duration of *TimeOpening*. Default: FALSE
EnableAlwaysOpen	INPUT	BOOL	With an edge change 0-1 *Output* is static set. Default: FALSE
DisableAlwaysOpen	INPUT	BOOL	With an edge change 0-1 *Output* is static reset. Default: FALSE
TimeOpening	INPUT	TIME	Time for the duration of the activation of *Output*. Default: 3s
DoorIsClosed	INPUT	BOOL	■ Optional - Position door TRUE: Door is closed FALSE: Door is open Default: FALSE
DoorIsLocked	INPUT	BOOL	■ Optional - Lock state of the door – TRUE: Door is locked – FALSE: Door is not locked Default: FALSE
Output	OUTPUT	BOOL	Static output to drive a monostable relay.
AlwaysOpen	OUTPUT	BOOL	Static output to indicate "Door is static open".

4.3 Access Control

4.3.1 FB 48 - ACONTROL - Access control

Description

With this block a access control can be implemented. After getting a code from an external keypad, panel or RFID reader, the code is compared with a list. Depending on the result, then the relative outputs are controlled.

■ The access codes are to be applied in a data block, which is specified by *ACLBlock*. Here you can also specify which outputs *Access1...6* are to be controlled and how (pulse/static) are they controlled. With the data block up to 16 access codes can be treaded.

■ Via *AccessCode1...4* the code of the corresponding input device is specified.

■ Via *CheckCode1...4* the code is compared with the code in your data block *ACLBlock*.

– If the access code in the data block exists, the corresponding outputs are controlled according to the specifications. With configured pulse output you can specify the pulse duration with *TimePulse*.

– If the access code does not exist in the data block, the output *Error* is set for the duration *TimeError*.

■ With an edge change 0-1 of *CentralLock* all the access codes are disabled. Here the output *CentralLocked* is set.

■ With an edge change 0-1 of *CentralUnlock* all the access codes are enabled and the output *CentralLocked* is reset.

Parameters

Parameter	Declaration	Data type	Description
AccessCode1	INPUT	STRING[16]	Access code, e.g. from keypad.
CheckCode1	INPUT	BOOL	With an edge change 0-1, the *AccessCode1* is compared with the access code in the data block *ACLBlock*. Default: 0
AccessCode2	INPUT	STRING[16]	Access code, e.g. from panel.
CheckCode2	INPUT	BOOL	With an edge change 0-1, the *AccessCode2* is compared with the access code in the data block *ACLBlock*. Default: 0
AccessCode3	INPUT	STRING[16]	Access code, e.g. RFID reader.
CheckCode3	INPUT	BOOL	With an edge change 0-1, the *AccessCode3* is compared with the access code in the data block *ACLBlock*. Default: 0
AccessCode4	INPUT	STRING[16]	Access code, e.g. from an other system
CheckCode4	INPUT	BOOL	With an edge change 0-1, the *AccessCode4* is compared with the access code in the data block *ACLBlock*. Default: 0
CentralLock	INPUT	BOOL	With an edge change 0-1 all the access codes are disabled. Here the output *CentralLocked* is set.
CentralUnlock	INPUT	BOOL	With an edge change 0-1 of *CentralUnlock* all the access codes are enabled and the output *CentralLocked* is reset.

Parameter	Declaration	Data type	Description
ACLBlock	INPUT	BLOCK	Data block with the access codes. ✋ *Chap. 4.3.3 'UDT 4 - ACL - Data structure for FB 48' page 16*
Access1	OUTPUT	BOOL	Output 1, can be controlled as pulse or static.
Access2	OUTPUT	BOOL	Output 2, can be controlled as pulse or static.
Access3	OUTPUT	BOOL	Output 3, can be controlled as pulse or static.
Access4	OUTPUT	BOOL	Output 4, can be controlled as pulse or static.
Access5	OUTPUT	BOOL	Output 5, can be controlled as pulse or static.
Access6	OUTPUT	BOOL	Output 6, can be controlled as pulse or static.
Error	OUTPUT	BOOL	If the access code does not exist in the data block, the output *Error* is set for the duration *Time-Error*.
CentralLocked	OUTPUT	BOOL	■ Access – TRUE: locked - access not possible – FALSE: not locked - access possible Default: TRUE
TimePulse	CONSTANT	Time	Time for the pulse duration at an output. Default: 3s
TimeError	CONSTANT	Time	Time for the duration of the error signal. Default: 500ms

4.3.2 UDT 3 - ACLREC - Data structure for FB 48

Description

Address	Name	Type	Start value	Comment
0.0		**STRUCT**		
+0.0	Code	STRING[16]	' '	Byte 0 ... 17: Access code S7String with max. 16 ASCII characters for access code
+18.0	EnableOutput1	BOOL	FALSE	Byte 18: Signal for the outputs to be controlled TRUE: activate output, FALSE: deactivate output
+18.1	EnableOutput2	BOOL	FALSE	
+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	

Address	Name	Type	Start value	Comment
0.0		**STRUCT**		
+19.0	SignalOutput1	BOOL	FALSE	Byte 19: Signal type FALSE: Pulse, TRUE: static 1, deactivation with additional 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 - Data structure for FB 48

Description

Address	Name	Type	Start value	Comment
0.0		**STRUCT**		
+0.0	RecordCount	INT	16	DBW0: Number valid record sets (0 ... n)
+2.0	RecordLen	INT	20	DBW2: Length of one record set in bytes (20)
+4.0	Record	ARRAY[0...15]		The first record set starts from DBB4
*20.0		"UDT 3 - ACLREC"		♦ *Chap. 4.3.2 'UDT 3 - ACLREC - Data structure for FB 48' page 15*
=324.0		BOOL		

> ⚠ **CAUTION!**
>
> A code must only occur 1 x in the whole list. Duplicate Codes are not allowed.
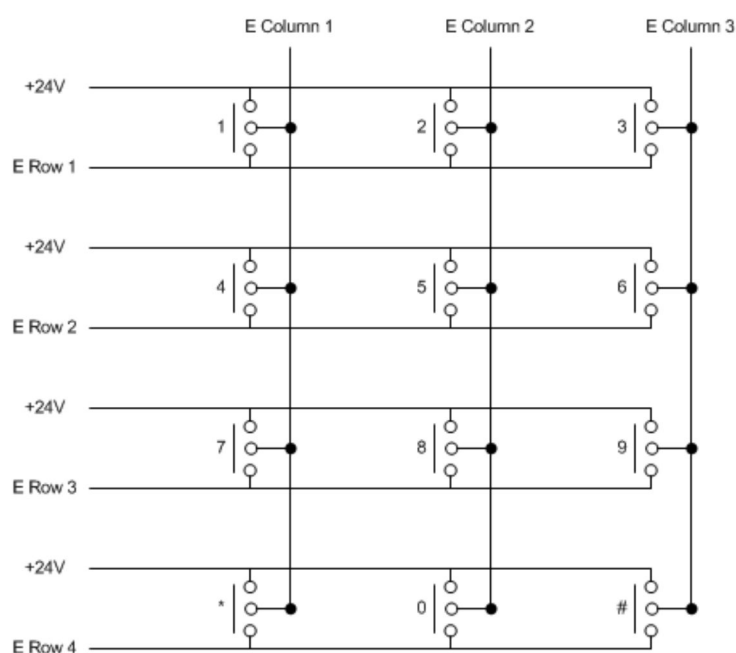
4.3.4 FB 49 - KEYPAD - Keyboard

Description

This block is used to connect an external keypad (0...9,*,#) with external DC 24V power supply. Depending on the pressed key, the keyboard provides the row and column signals (24V). The block evaluates the signals internally by means of a bit pattern table and transfers the determined ASCII code into the keyboard buffer. If necessary, or automatically the keyboard buffer is output as max. 16byte character string.

- Via *Row 1...4* the rows 1...4 of the keyboard matrix are connected.
- Via *Column 1...3* the columns 1...3 of the keyboard matrix are connected.
- Via *ClearCode* you can specify a key code to clear the keyboard buffer.

- Via *EnterCode* you can specify a key code to output the keyboard buffer at *Output* for one cycle. During this time the output *Valid* is enabled.
- Via edge change 0-1 of *Clear* the keyboard buffer cleared.
- Via *TimeAutoClear* you specify the max. duration for pressing the keys. Otherwise the keyboard buffer is cleared.
- Via *CountCharAutoEnter* you can specify the number of characters, which are automatically output as keyboard buffer at *Output* for one cycle. During this time the output *Valid* is enabled.
- *Error* is activated for the time *TimeError* when a key is pressed, but the keyboard buffer is full.
- With *TimeDebounce* you can specify a debounce time for the input signals.

Parameters

Parameter	Declaration	Data type	Description
Row1	INPUT	BOOL	Row 1 of the keyboard matrix. Default: FALSE
Row2	INPUT	BOOL	Row 2 of the keyboard matrix. Default: FALSE
Row3	INPUT	BOOL	Row 3 of the keyboard matrix. Default: FALSE
Row4	INPUT	BOOL	Row 4 of the keyboard matrix. Default: FALSE
Column1	INPUT	BOOL	Column 1 of the keyboard matrix. Default: FALSE
Column2	INPUT	BOOL	Column 2 of the keyboard matrix. Default: FALSE

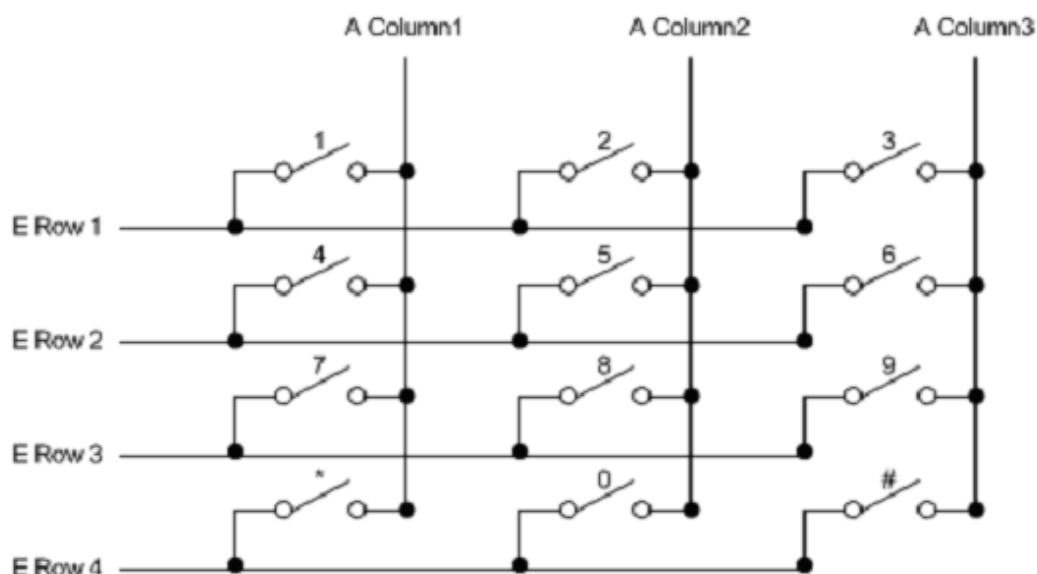| Parameter	Declaration	Data type	Description
Column3	INPUT	BOOL	Column 3 of the keyboard matrix. Default: FALSE
ClearCode	INPUT	BYTE	The value at which the keyboard buffer is to be cleared. 0: deactivated Default: 42 = *
EnterCode	INPUT	BYTE	The value at which the keyboard buffer is to be output. 0: deactivated Default: 35 = #
Clear	INPUT	BOOL	Edge change 0-1 clears the keyboard buffer. Default: FALSE
TimeAutoClear	INPUT	TIME	Duration within a further key must be pressed. Otherwise the keyboard buffer is cleared. 0: Buffer is not cleared Default: 10s
CountCharAu-toEnter	INPUT	INT	Number of characters, which are automatically output as keyboard buffer. 0: deactivated Default: 0
Output	OUTPUT	STRING[16]	Contents of the keyboard buffer as max. 16 byte string.
Valid	OUTPUT	BOOL	The static output indicates that the string at *Output* is valid. The signal is pending for one cycle.
Error	OUTPUT	BOOL	*Error* is activated for the time *TimeError* when a key is pressed, but the keyboard buffer is full.
TimeDebounce	CONSTANT	TIME	Time for debounce of the inputs. Default: 100ms
TimeError	CONSTANT	TIME	Time for the duration of the error signal. Default: 500ms

4.3.5 FB 50 - KEYPAD2 - Keyboard

Description

This block is used to connect an external keypad (0...9,*,#) without an own power supply. The block provides output column signals. Depending on the pressed key, the keyboard provides the according row signals. The block evaluates the signals internally by means of a bit pattern table and transfers the determined ASCII code into the keyboard buffer. If necessary, or automatically the keyboard buffer is output as max. 16byte character string.

- Via *Row 1...4* the rows 1...4 of the keyboard matrix are connected.
- Via *Column 1...3* the columns 1...3 of the keyboard matrix are connected.
- Via *TimeDelay* you can specify a waiting time after setting the column outputs up to reading the corresponding row inputs. This time must be greater than the delay time of the used module.
- Via *ClearCode* you can specify a key code to clear the keyboard buffer.

- Via *EnterCode* you can specify a key code to output the keyboard buffer at *Output* for one cycle. During this time the output *Valid* is enabled.
- Via edge change 0-1 of *Clear* the keyboard buffer cleared.
- Via *TimeAutoClear* you specify the max. duration for pressing the keys. Otherwise the keyboard buffer is cleared.
- Via *CountCharAutoEnter* you can specify the number of characters, which are automatically output as keyboard buffer at *Output* for one cycle. During this time the output *Valid* is enabled.
- *Error* is activated for the time *TimeError* when a key is pressed, but the keyboard buffer is full.
- With *TimeDebounce* you can specify a debounce time for the input signals.

Parameters

Parameter	Declaration	Data type	Description
Row1	INPUT	BOOL	Row 1 of the keyboard matrix. Default: FALSE
Row2	INPUT	BOOL	Row 2 of the keyboard matrix. Default: FALSE
Row3	INPUT	BOOL	Row 3 of the keyboard matrix. Default: FALSE
Row4	INPUT	BOOL	Row 4 of the keyboard matrix. Default: FALSE
ClearCode	INPUT	BYTE	The value at which the keyboard buffer is to be cleared. 0: deactivated Default: 42 = *
EnterCode	INPUT	BYTE	The value at which the keyboard buffer is to be output. 0: deactivated Default: 35 = #

Parameter	Declaration	Data type	Description
Clear	INPUT	BOOL	Edge change 0-1 clears the keyboard buffer. Default: FALSE
TimeAutoClear	INPUT	TIME	Duration within a further key must be pressed. Otherwise the keyboard buffer is cleared. 0: Buffer is not cleared Default: 10s
CountCharAutoEnter	INPUT	INT	Number of characters, which are automatically output as keyboard buffer. 0: deactivated Default: 0
Column1	OUTPUT	BOOL	Column 1 of the keyboard matrix. Default: FALSE
Column2	OUTPUT	BOOL	Column 2 of the keyboard matrix. Default: FALSE
Column3	OUTPUT	BOOL	Column 3 of the keyboard matrix. Default: FALSE
Output	OUTPUT	BYTE	Contents of the keyboard buffer as max. 16 byte string.
Valid	OUTPUT	BOOL	The static output indicates that the string at *Output* is valid. The signal is pending for one cycle.
Error	OUTPUT	BOOL	*Error* is activated for the time *TimeError* when a key is pressed, but the keyboard buffer is full.
TimeDebounce	CONSTANT	TIME	Time for debounce of the inputs. Default: 100ms
TimeError	CONSTANT	TIME	Time for the duration of the error signal. Default: 500ms
TimeDelay	CONSTANT	TIME	Duration after setting the column outputs up to reading the corresponding row inputs. This time must be greater than the delay time of the used module. Default: 10ms