

Functionblocks for reading the key ID from PITmode



Product

Type: PITmode
Name: PIT m3.1p, PIT m3.2p, PIT m3.3p, PIT m4SEU
Manufacturer: Pilz GmbH & Co. KG, Safe Automation

Document

Release Number: 02
Release Date: 31 May 2019

Document Revision History

Release	Date	Changes	Chapter
01	2018-10-30	Creation	all
02	2019-05-31	Editorial	2

Validity of Application Note

This present Application Note is valid until a new version of the document is published. This and other Application Notes can be downloaded in the latest version and for free from www.pilz.com. For a simple search, use our [content document \(1002400\)](#) or the [direct search function](#) in the download area.

Exclusion of liability

We have taken great care in compiling our application note. It contains information about our company and our products. All statements are made in accordance with the current status of technology and to the best of our knowledge and belief. While every effort has been made to ensure the information provided is accurate, we cannot accept liability for the accuracy and entirety of the information provided, except in the case of gross negligence. In particular, all information on applicable standards, safety-related classifications and time characteristics should be viewed as provisional. In particular it should be noted that statements do not have the legal quality of assurances or assured properties. We are grateful for any feedback on the contents.

May 2019

All rights to this publication are reserved by Pilz GmbH & Co. KG. We reserve the right to amend specifications without prior notice. Copies may be made for the user's internal purposes. The names of products, goods and technologies used in this manual are trademarks of the respective companies.

Abbreviations

Abbreviation / term	Description	Source
AN	Application Note	 AN content (1002400)">www.pilz.com > AN content (1002400)
PNOZ	Pilz E-STOP positive-guided (DE: Pilz NOT-AUS-Zwangsgef ührt)	 PNOZ">www.pilz.com > PNOZ
PSS	Programmable control system (DE: Programmierbares Steuerungssystem)	 PSS">www.pilz.com > PSS
PSS u2	PSS universal, 2 nd generation	 PSS u2">www.pilz.com > PSS u2
POU	Program Organisation Unit	
NC	Normally Closed	
NO	Normally Open	
NOP	No Operation	

Definition of symbols

- Information that is particularly important is identified as follows:



CAUTION!

This refers to a hazard that can lead to a less serious or minor injury plus material damage, and also provides information on preventive measures that can be taken.



NOTICE

This describes a situation in which the product or devices could be damaged and also provides information on preventive measures that can be taken. It also highlights areas within the text that are of particular importance.



INFORMATION

This gives advice on applications and provides information on special features.

Contents

1. Useful documentation	5
1.1. Documentation from Pilz GmbH & Co. KG	5
2. Introduction	6
3. Overview	7
3.1. Known behaviours	7
4. ST_PIT_m3_1p_Key_ID	8
4.1. Schematic representation	8
4.2. Input Interface	8
4.3. Output Interface	8
4.4. Description	8
4.5. Example with a PIT m3.1p	8
4.5.1. Wiring diagram	9
4.5.2. POU_Cycle	9
4.5.3. I/O Mapping	10
5. PITMode_V32_ID	11
5.1. Schematic representation	11
5.2. Input Interface	11
5.3. Output Interface	11
5.4. Description	11
5.5. Example with a PIT m3.2p	12
5.5.1. Wiring diagram	12
5.5.2. POU_1	12
5.5.3. I/O Mapping	13
6. PITMode_V32	14
6.1. Schematic representation	14
6.2. Input Interface	14
6.3. Output Interface	14
6.4. Description	14
6.5. State information (extract from PIT m3.2p operating manual)	15
6.6. Example with a PIT m3.2p in TIA Portal	16
6.6.1. Steps to create function block from external source file	16
6.6.2. Wiring diagram	18
6.6.3. Main (OB1)	18
7. Table of figures	19

1. Useful documentation

Reading the documentation listed below is necessary for understanding this Application Note. The availability of the indicated tools and safe handling are also presupposed with the user.

1.1. Documentation from Pilz GmbH & Co. KG

No.	Description	Item No. /Download
1	Pilz international homepage, download section	www.pilz.com
2	PIT m3.1p	1002259-EN-02
3	PIT m3.2p	1003176-EN-08
4	PIT m3.3p	1003190-EN-08

2. Introduction

This application note provides the description of a specific function or function block which may be useful for writing applications for the Pilz Automation System PSS4000 using the programming tool PAS4000.

Caution:

The block is intended to be used in the standard part of the PSS4000 automation system. The use of the function/functionblock or code fragments (snippets) of it for another PLC system can lead to other behaviors.

Please make sure that you are using the latest version of the application note and the PAS4000 project with the function/-blocks. Alternatively, download the latest versions from the PILZ website:

[Link to this application note on the PILZ website](#)

3. Overview

Project/Archive	PIT_m3_1p_V1_1.zip	Pitmode_3_2_v01.zip	PITMode_V32.scl
Functionblock	ST_PIT_m3_1p_Key_ID	PITMode_V32_ID	PITMode_V32
Controller	PSS4000	PSS4000	Siemens
Supported Key ID communication modes¹	Transmitter-controlled	Handshake-controlled	Handshake-controlled
PIT m3.1p	X		
PIT m3.2p	X	X	X
PIT m3.3p	X	X	X

3.1. Known behaviours

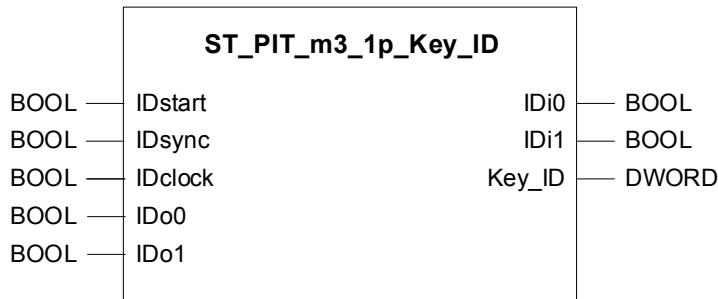
All functions have the following known behaviours, which must be considered or adapted according to the application:

- If a key is removed, the key ID is still available.
- If an error occurs (e.g. the key is removed during the reading process), the previous key ID is still pending.
- Key ID is displayed without leading zeros (integer value).
- No valid output to ensure that the key ID read so far is consistent.

¹ A detailed description of the different communication modes and how they work can be found in the PITmode operating manuals.

4. ST_PIT_m3_1p_Key_ID

4.1. Schematic representation



4.2. Input Interface

Variable	Datatype	Value range	Comment
IDstart	BOOL	TRUE / FALSE	NOP
IDsync	BOOL	TRUE / FALSE	Indicates the start of the transmission.
IDclock	BOOL	TRUE / FALSE	Clock signal used for data transmission.
IDo0	BOOL	TRUE / FALSE	Bits 27 to 14 of the key ID are sent over this data line.
IDo1	BOOL	TRUE / FALSE	Bits 13 to 0 of the key ID are sent over this data line.

4.3. Output Interface

Variable	Datatype	Value range	Comment
IDI0	BOOL	TRUE / FALSE	The received bits 27 to 14 are reported back via this data line from the controller.
IDI1	BOOL	TRUE / FALSE	The received bits 13 to 0 are reported back via this data line from the controller.
Key_ID	DWORD	0 ... DWORD_MAX	Key-ID number.

4.4. Description

Typename: ST_PIT_m3_1p_Key_ID

Block type: Functionblock

Dependencies: DWordOfBits

This function block receives the key ID only in the mode: Transmitter-controlled. The device must be configured accordingly. See the necessary settings in the operating manual of the PITmode.

More remarks are described in the declaration part of the function block.

4.5. Example with a PIT m3.1p

This example describes the use of the PIT m3.1p in automatic mode. As soon as a key is inserted in the device, the key ID is read.

4.5.1. Wiring diagram

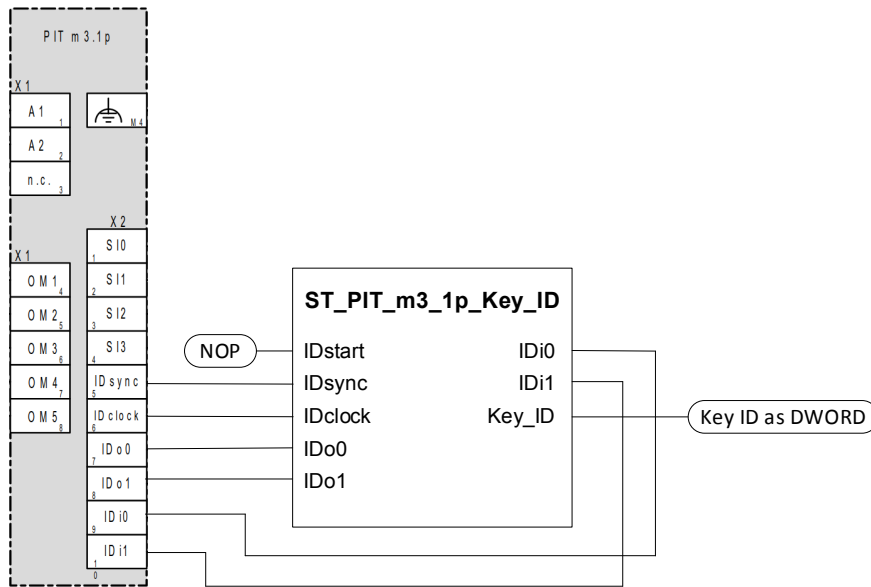


Fig. 1: Wiring diagram PIT m3.1p with ST_PIT_m3_1p_Key_ID

4.5.2. POU_Cycle

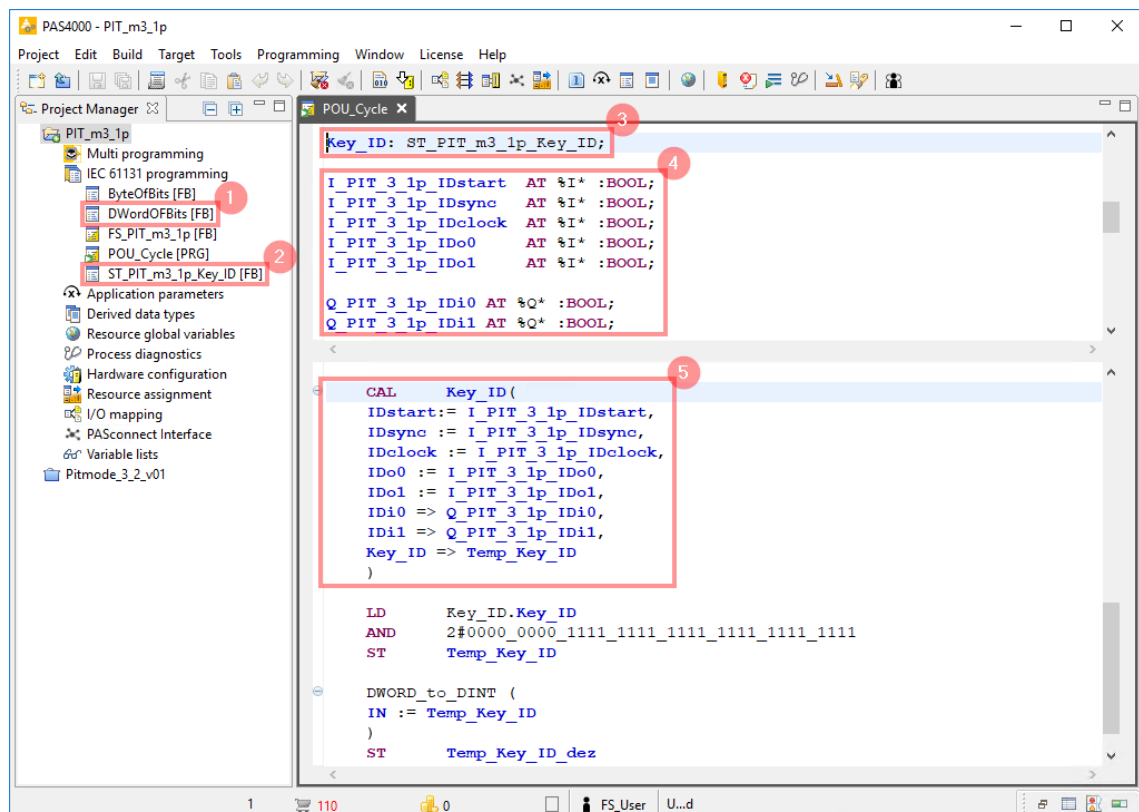


Fig. 2: ST_PIT_m3_1p_Key / POU_Cycle

1. Helper function block necessary for reading key ID with ST_PIT_m3_1p_Key.
2. The ST_PIT_m3_1p_Key function block for reading key ID.
3. Instance of the function block ST_PIT_m3_1p_Key with the name Key_ID.
4. Define process image variables to communicate with the PITmode.
5. Cyclic call of the Key_ID instance for reading the key ID.

4.5.3. I/O Mapping

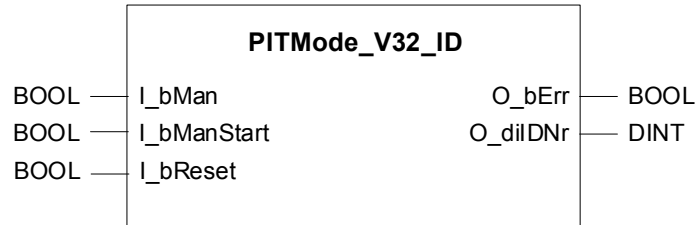
Overview of I/O mappings

Block	Data type	Direction	PI variable	...	Device	Slot	Module	Terminal	Data type	Na...	Direction
POU_Cycle	BOOL	Data sink	POU_Cycle.I_PIT_3_1p_IDclock	<--	Blocksteuerung	4	PSSu E F 4DI	21	SAFEBOOL	Data	Data source
POU_Cycle	BOOL	Data sink	POU_Cycle.I_PIT_3_1p_IDo0	<--	Blocksteuerung	4	PSSu E F 4DI	14	SAFEBOOL	Data	Data source
POU_Cycle	BOOL	Data sink	POU_Cycle.I_PIT_3_1p_IDo1	<--	Blocksteuerung	4	PSSu E F 4DI	24	SAFEBOOL	Data	Data source
POU_Cycle	BOOL	Data sink	POU_Cycle.I_PIT_3_1p_IDstart	<--	Blocksteuerung	0	PSSu E F 4DI	14	SAFEBOOL	Data	Data source
POU_Cycle	BOOL	Data sink	POU_Cycle.I_PIT_3_1p_IDsync	<--	Blocksteuerung	4	PSSu E F 4DI	11	SAFEBOOL	Data	Data source
POU_Cycle	BOOL	Data source	POU_Cycle.Q_PIT_3_1p_IDi0	-->	Blocksteuerung	5	PSSu E F 4DO 0.5	11	SAFEBOOL	Data	Data sink
POU_Cycle	BOOL	Data source	POU_Cycle.Q_PIT_3_1p_IDi1	-->	Blocksteuerung	5	PSSu E F 4DO 0.5	21	SAFEBOOL	Data	Data sink

Fig. 3: ST_PIT_m3_1p_Key / I/O Mapping

5. PITMode_V32_ID

5.1. Schematic representation



5.2. Input Interface

Variable	Datatype	Value range	Comment
I_bMan	BOOL	TRUE / FALSE	Operation mode manual.
I_bManStart	BOOL	TRUE / FALSE	Start for transmitting ID in manual mode.
I_bReset	BOOL	TRUE / FALSE	Error acknowledgment.

5.3. Output Interface

Variable	Datatype	Value range	Comment
O_bErr	BOOL	TRUE / FALSE	Error available.
O_dilDnr	DINT	0 ... DINT_MAX	Key-ID number.

5.4. Description

Typename: PITMode_V32_ID

Block type: Functionblock

This function block receives the key ID only in the mode: Handshake-controlled. The device must be configured accordingly. See the necessary settings in the operating manual of the PITmode.

This function block uses mapping of internal process image variables, which has to be mapped to the PITmode hardware interface:

Variable	Datatype	I/O-PI	Value range	Comment
Input_bIDsync	BOOL	%I*	TRUE / FALSE	Control signal for valid key is available.
Input_bIDresponse	BOOL	%I*	TRUE / FALSE	Handshake signal, valid signals are present.
Input_bIDdata	BOOL	%I*	TRUE / FALSE	Data output PITmode.
Output_bIDrequest	BOOL	%Q*	TRUE / FALSE	Handshake signal, request bit to receive.

In manual mode, (I_bMan is set to true), the application can trigger the start command for reading the key ID by a rising edge at I_bManStart.

If manual mode is not set (I_bMan is set to false), the function block starts automatically the reading process of the key ID as soon as the PITmode reports an inserted key at Input_bIDsync. This behavior is comparable to the mode: Transmitter-Controlled communication, only with the difference that the controller starts the reading process.

If the function reports an error (for ex. key has been removed during the reading process), this error must be reset with a rising edge at I_bReset before a new read operation can be started.

5.5. Example with a PIT m3.2p

This example describes the use of the PIT m3.2p in automatic mode. As soon as a key is inserted in the device, the key ID is read.

5.5.1. Wiring diagram

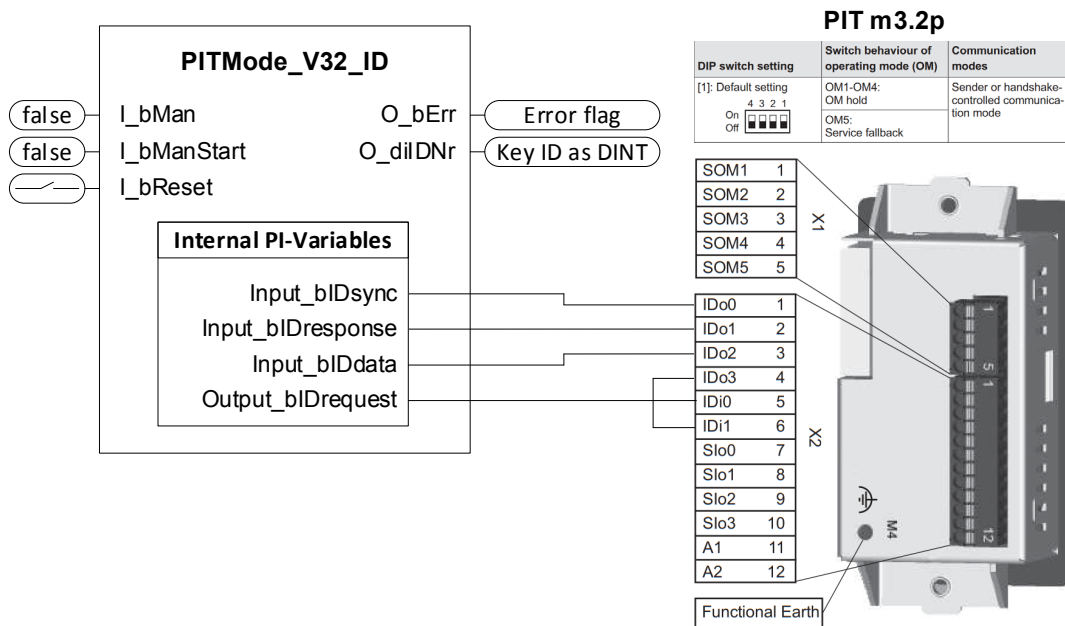


Fig. 4: Wiring diagram PIT m3.2p with PITMode_V32_ID

5.5.2. POU_1

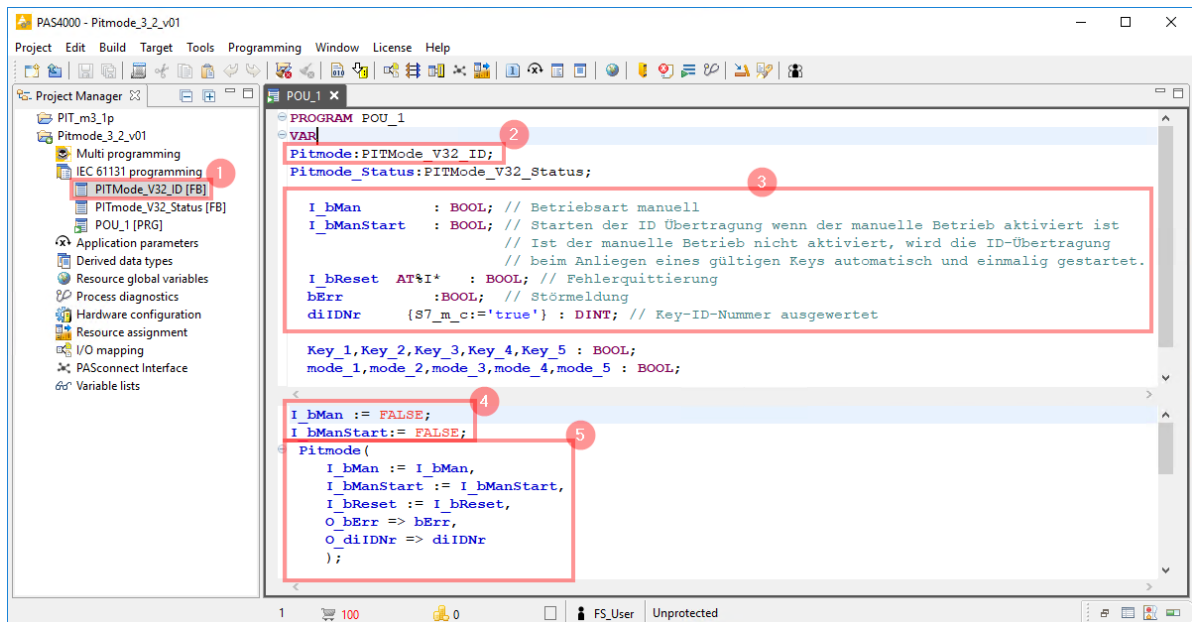


Fig. 5: PITMode_V32_ID / POU_1

1. The PITMode_V32_ID function block for reading key ID.
2. Instance of the function block PITMode_V32_ID with the name PITmode.
3. Define variables to control the PITmode instance.
4. Parameterizing the PITmode instance for automatically reading the key ID.
5. Cyclic call of the PITmode instance for reading the key ID.

5.5.3. I/O Mapping

Overview of I/O mappings

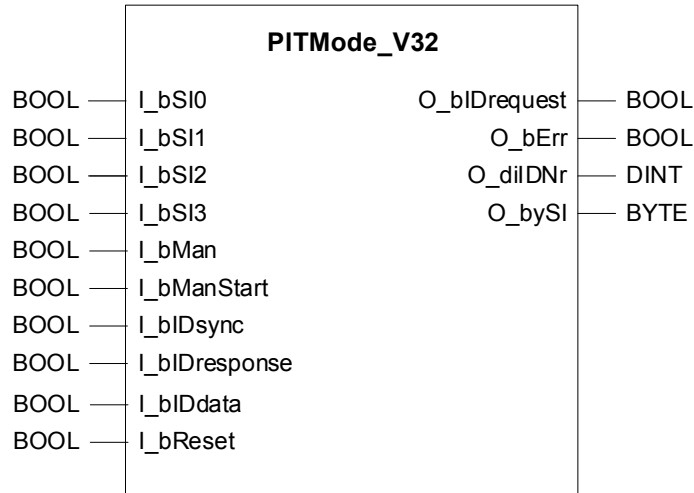
_bl

Block	Data type	Direction	PI variable	...	Device	Slot	Module	Terminal	Data...	Na...	Direction
POU_1	BOOL	Data sink	POU_1.Pitmode.Input_blDsync	<--	Pitmode_3_2	3	PSSu E S 4DI	11	BOOL	Data	Data source
POU_1	BOOL	Data sink	POU_1.Pitmode.Input_blDresponse	<--	Pitmode_3_2	3	PSSu E S 4DI	21	BOOL	Data	Data source
POU_1	BOOL	Data sink	POU_1.Pitmode.Input_blDdata	<--	Pitmode_3_2	3	PSSu E S 4DI	14	BOOL	Data	Data source
POU_1	BOOL	Data source	POU_1.Pitmode.Output_blDrequest	-->	Pitmode_3_2	5	PSSu E S 4DO 0.5	11	BOOL	Data	Data sink

Fig. 6: PITMode_V32_ID / I/O Mapping

6. PITMode_V32

6.1. Schematic representation



6.2. Input Interface

Variable	Datatype	Value range	Comment
I_bSI0	BOOL	TRUE / FALSE	Signal output for state information.
I_bSI1	BOOL	TRUE / FALSE	Signal output for state information.
I_bSI2	BOOL	TRUE / FALSE	Signal output for state information.
I_bSI3	BOOL	TRUE / FALSE	Signal output for state information.
I_bMan	BOOL	TRUE / FALSE	Operation mode manual.
I_bManStart	BOOL	TRUE / FALSE	Start for transmitting ID in manual mode.
I_bIDSync	BOOL	TRUE / FALSE	Control signal for valid key is available.
I_bIDresponse	BOOL	TRUE / FALSE	Handshake signal, valid signals are present.
I_bIDdata	BOOL	TRUE / FALSE	Data output PITmode.
I_bReset	BOOL	TRUE / FALSE	Error acknowledgment.

6.3. Output Interface

Variable	Datatype	Value range	Comment
O_bIDrequest	BOOL	TRUE / FALSE	Handshake signal, request bit to receive.
O_bErr	BOOL	TRUE / FALSE	Error available.
O_dilDNr	DINT	0 ... DINT_MAX	Key-ID number.
O_bySI	BYTE	0 ... Fh	State information (see chapter 6.5).

6.4. Description

Typename: PITMode_V32

Block type: Functionblock (Siemens SCL Source code)

This function block receives the key ID only in the mode: Handshake-controlled. The device must be configured accordingly. See the necessary settings in the operating manual of the PITmode.

In manual mode (I_bMan is set to true), the application can trigger the start command for reading the key ID by a rising edge at I_bManStart.

If manual mode is not set (I_bMan is set to false), the function block starts automatically the reading process of the key ID as soon as the PITmode reports an inserted key at Input_bIDsync. This behavior is comparable to the mode: Transmitter-Controlled communication, only with the difference that the controller starts the reading process.

If the function reports an error (for ex. key has been removed during the reading process), this error must be reset with a rising edge at I_bReset before a new read operation can be started.

In addition, this function can output the state information of the device as a byte value at O_bySI.

6.5. State information (extract from PIT m3.2p operating manual)

State information	Description
0h	Reserve
1h	Reserve
2h	Key 5 inserted.
3h	SOM5 chosen
4h	Key 1 inserted.
5h	Key 2 inserted.
6h	Key 3 inserted.
7h	Key 4 inserted.
8h	SOM1 chosen.
9h	SOM2 chosen.
Ah	SOM3 chosen.
Bh	SOM4 chosen.
Ch	No authorization.
Dh	Device error.
Eh	Key operator error.
Fh	Key released.

6.6. Example with a PIT m3.2p in TIA Portal

This example describes the use of the PIT m3.2p in automatic mode. As soon as a key is inserted in the device, the key ID is read.

6.6.1. Steps to create function block from external source file

1. Add new external file into an existing project.

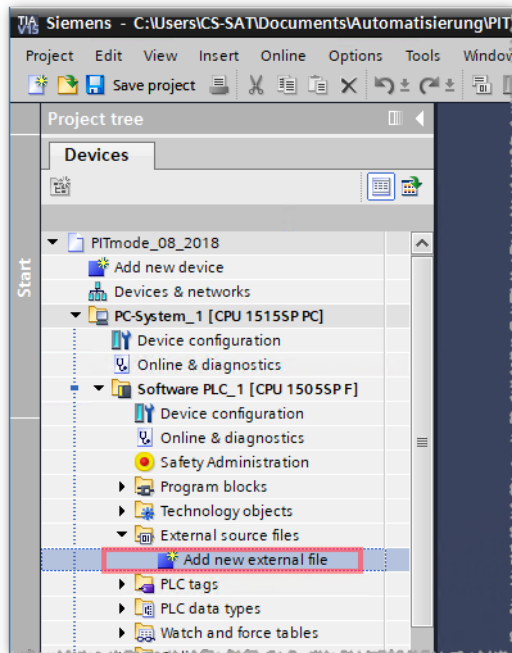


Fig. 7: TIA Portal / Add new external file

2. Select the source file "PITMode_V32.scl" from the file explorer. The source file is now added to the project.

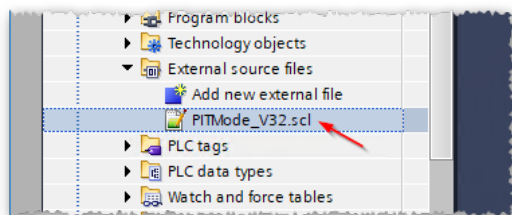


Fig. 8: TIA Portal / external file added to project

- Right-Click on the added source file and select “Generate blocks from source” from context menu.

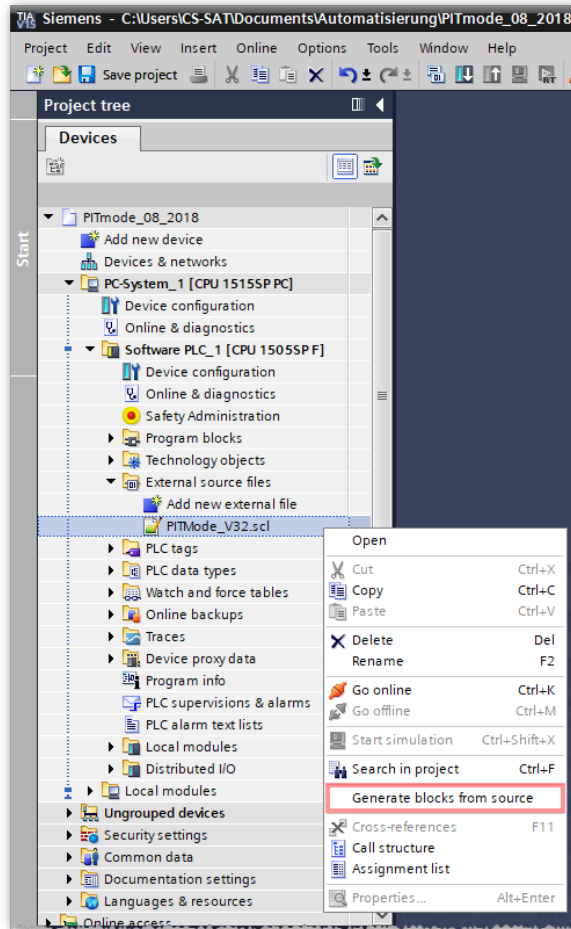


Fig. 9: TIA Portal / Context menu to create blocks from source

- Generated function block appears in the project tree under “Program blocks” and is ready to use.

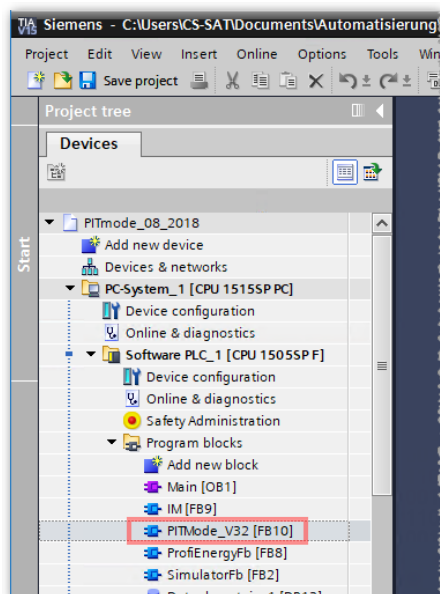


Fig. 10: TIA Portal / Generated function block from source

6.6.2. Wiring diagram

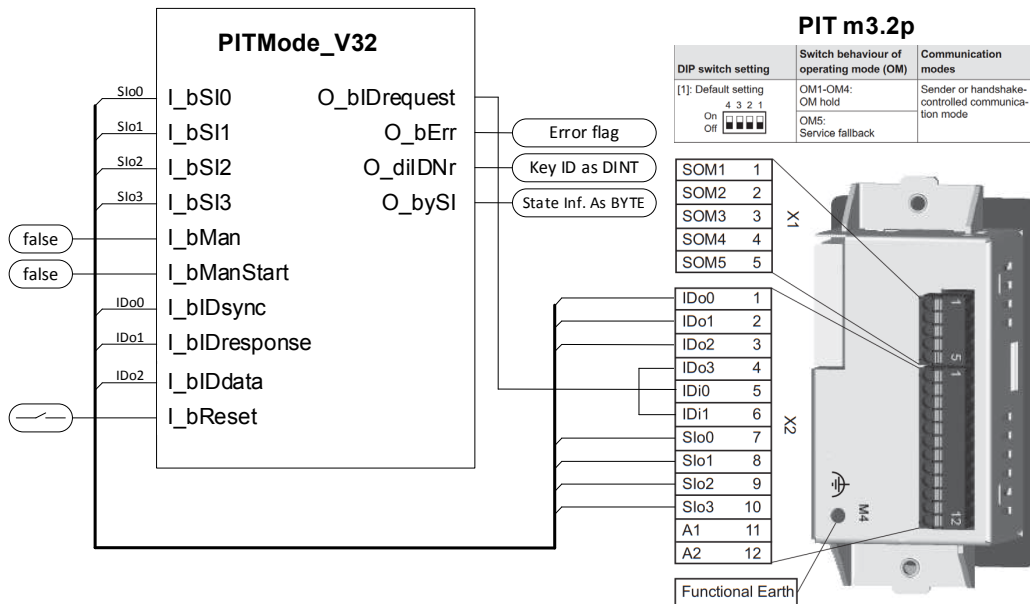


Fig. 11: Wiring diagram PIT m3.2p with PITMode_V32

6.6.3. Main (OB1)

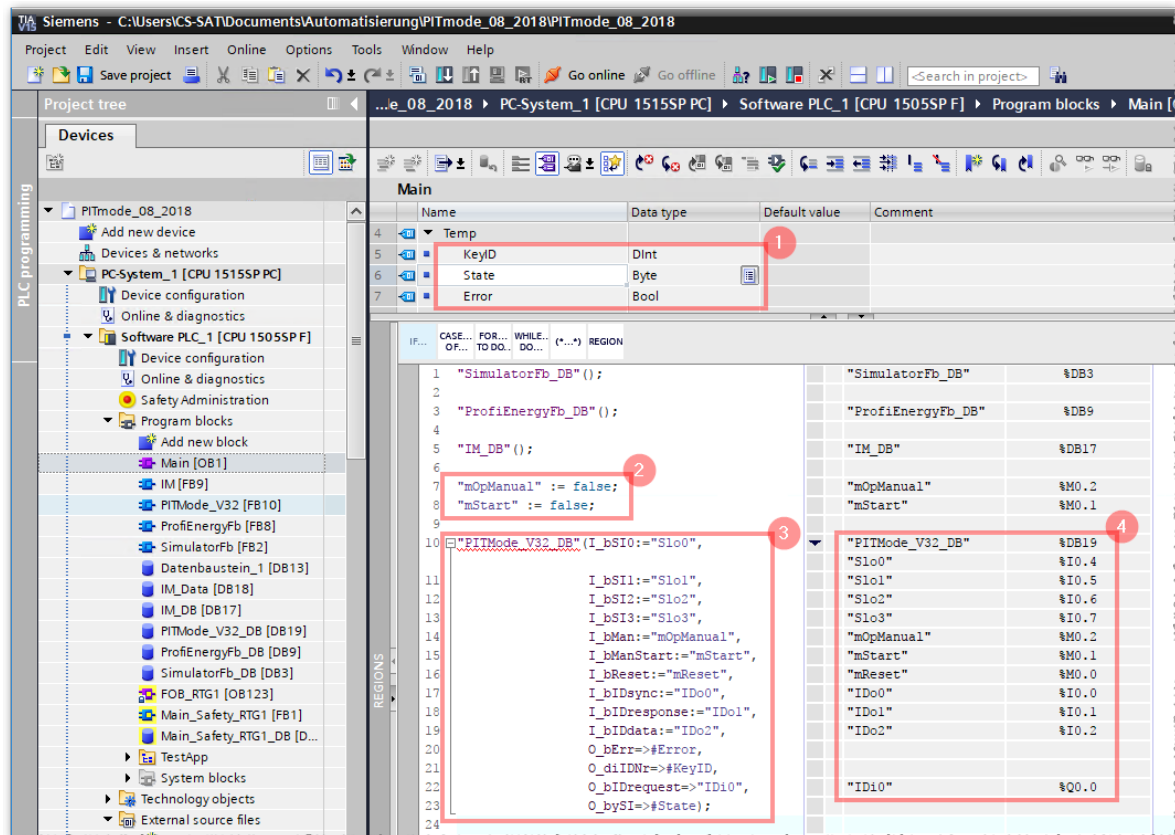


Fig. 12: TIA Portal / Main program (OB1)

1. Defined variables to hold the key ID, information state and error flag.
2. Marker variables to parameterize the function block.
3. Cyclic call of the function block instance for reading the key ID.
4. Mapping of the variables used by the function block.

7. Table of figures

Fig. 1: Wiring diagram PIT m3.1p with ST_PIT_m3_1p_Key_ID.....	9
Fig. 2: ST_PIT_m3_1p_Key / POU_Cycle	9
Fig. 3: ST_PIT_m3_1p_Key / I/O Mapping	10
Fig. 4: Wiring diagram PIT m3.2p with PITMode_V32_ID	12
Fig. 5: PITMode_V32_ID / POU_1	12
Fig. 6: PITMode_V32_ID / I/O Mapping	13
Fig. 7: TIA Portal / Add new external file	16
Fig. 8: TIA Portal / external file added to project	16
Fig. 9: TIA Portal / Context menu to create blocks from source	17
Fig. 10: TIA Portal / Generated function block from source	17
Fig. 11: Wiring diagram PIT m3.2p with PITMode_V32	18
Fig. 12: TIA Portal / Main program (OB1)	18

► Support

Technical support is available from Pilz round the clock.

Americas

Brazil
+55 11 97569-2804
Canada
+1 888-315-PILZ (315-7459)
Mexico
+52 55 5572 1300
USA (toll-free)
+1 877-PILZUSA (745-9872)

Asia

China
+86 21 60880878-216
Japan
+81 45 471-2281
South Korea
+82 31 450 0680

Australia

+61 3 95600621

Europe

Austria
+43 1 7986263-0
Belgium, Luxembourg
+32 9 3217575
France
+33 3 88104000
Germany
+49 711 3409-444
Ireland
+353 21 4804983
Italy, Malta
+39 0362 1826711

Scandinavia

+45 74436332

Spain

+34 938497433

Switzerland

+41 62 88979-30

The Netherlands

+31 347 320477

Turkey

+90 216 5775552

United Kingdom

+44 1536 462203

You can reach our international hotline on:

+49 711 3409-444
support@pilz.com

Pilz develops environmentally-friendly products using ecological materials and energy-saving technologies. Offices and production facilities are ecologically designed, environmentally-aware and energy-saving. So Pilz offers sustainability, plus the security of using energy-efficient products and environmentally-friendly solutions.

Energy
saving by Pilz



Pilz GmbH & Co. KG
Felix-Wankel-Straße 2
73760 Ostfildern, Germany
Tel.: +49 711 3409-0
Fax: +49 711 3409-133
info@pilz.com
www.pilz.com

PILZ
THE SPIRIT OF SAFETY

CMSE®, InduNET® g®, PAS4000®, PASconfig®, Piz®, PIT®, PLID®, PMCPirimo®, PMClendo®, PMD®, PMP®, PNCZ®, Pimo®, PSEN®, PSS®, PVS®, SafetyBUS i
SafetyEYE®, SafetyNET® g®, THE SPIRIT OF SAFETY® are registered and protected trademarks of Pilz GmbH & Co. KG in some countries. We would point out that product features m
from the details stated in this document, depending on the status at the time of publication and the scope of the equipment. We accept no responsibility for the validity, accuracy
and entirety of the text and graphics presented in this information. Please contact our Technical Support if you have any questions.