PSSu Key-in-Pocket Extended

PILZ
THE SPIRIT OF SAFETY

Product

Type:	FS_KeyInPocket_SignInOut, FS_KeyInPocket_Manager, FS_KeyInPocket_BlindSpotCheck
Name:	PSS 4000, PITreader, PITgatebox, PSENmlock
Manufacturer:	Pilz GmbH & Co. KG, Safe Automation

Document

Release Number:	01
Release Date:	16 May 2023

Application Note – 1006505-EN-01

Document Revision History

Release	Date	Changes	Chapter
01	2023-05-16	Creation	all

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.

The Pilz newsletter is free of charge and keeps you up-to-date on all the latest issues and trends in safe automation.

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 2023

Industrial Security

To secure plants, systems, machines and networks against cyberthreats it is necessary to implement (and continuously maintain) an overall Industrial Security concept that is state of the art.
Perform a risk assessment in accordance with VDI/VDE 2182 or IEC 62443-3-2 and plan the security measures with care. If necessary, seek advice from Pilz Customer Support.

Abbreviations

Abbreviation / term	Description	Source
AN	**A**pplication **N**ote	www.pilz.com > AN content (1002400)
PNOZ	Pilz E-STOP positive-guided (DE: **P**ilz **NO**T-AUS-**Z**wangsgeführt)	www.pilz.com > PNOZ
PSS	Programmable control system (DE: **P**rogrammierbares **S**teuerungs**s**ystem)	www.pilz.com > PSS
PSS u2	**PSSu**niversal, **2**nd generation	www.pilz.com > PSS u2
POU	**P**rogram **O**rganisation **U**nit	
NC	**N**ormally **C**losed	
NO	**N**ormally **O**pen	
STO	**S**afe **T**orque **O**ff	

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 Preface

This Application Note provides a basic description of the commissioning of the key-in-pocket system from Pilz, using as an example a manufacturing cell with two access points and two blind spots.

With the help of the access permission system PITreader and a Pilz safety controller, the key-in-pocket system from Pilz guarantees that a plant cannot (re-)start until the last person has left the danger zone. A PSSu PLC from the automation system PSS 4000 is used as the safety controller.

The general procedure for a successful basic configuration is shown.

NOTICE

A detailed explanation of the safety functions employed in the failsafe application and their evaluation in terms of functional safety are not part of this document.

2 Useful documentation

Reading the documentation listed below is necessary for understanding this Application Note. The availability of the software used and its safe handling are also presupposed for the user.

2.1 Documentation from Pilz GmbH & Co. KG

No.	Description	Item No. /Download
1	Pilz international homepage, download section	www.pilz.com
2	System description Key-in-Pocket	1006613-EN-xx
3	System description PSS 4000	1001467-EN-xx
4	Safety manual PSS 4000	1001468-EN-xx
5	Operating manual PSSu H PLC2 FS SN SD	1005195-EN-xx
6	Operating manual PSSu E F 4DI	21310-EN-xx
7	Operating manual PSSu E F 4DO 0.5	21316-EN-xx
8	Operating manual PITreader	1004806-EN-xx
9	Operating manual PIT gb RLLE y ETH	1005249-EN-xx
10	Operating manual PSEN ml b 1.1/2.1/2.2 / PSEN ml DHM	1005444-EN-xx
11	PAS4000 Online help	-

2.2 Documentation from other sources of information

No.	Description	Item No. / Download
1	EN ISO 13849-1:2015 Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design	European standard
2		
3		
4		

3 Used hardware and software

3.1 Pilz products

No.	Descriptions	Order number	Version	Number
1	PSSu H PLC2 FS SN SD	312077	FW 1.25	1
2	PSSu E F 4DI	312200	-	6
3	PSSu E F 4DO 0.5	312210	-	4
4	PSSu BP 1/8C	312601	-	10
5	PSSu A Con 2/8 C (connector set, spring-loaded connection)	313111	-	1
6	PIT gb RLLE y up ETH (PITgatebox – pushbutton unit)	G1000020	-	2
7	PITreader S base unit	402256	-	4
8	PITreader key Adapter h	402308	-	4
9	PITreader key ye g (Transponder key, authorisations freely configurable)	402260	-	1 (min.)
10	PSEN ml b 2.1 switch (Safety gate system with mechanical guard locking, basic version with power reset, fully coded)	570403	-	2
11	PSEN ml DHM down l 2.1 (Door handle module for fully coded PSENmlock switches, left-hand gate end stop)	6O000006	-	2
12	Software platform PAS4000	-	V 1.25	1

The Pilz product portfolio also includes servo amplifiers and drives. However, they are not described in detail here, as they are not central to this Application Note.

3.2 Structure of the application (schematic)

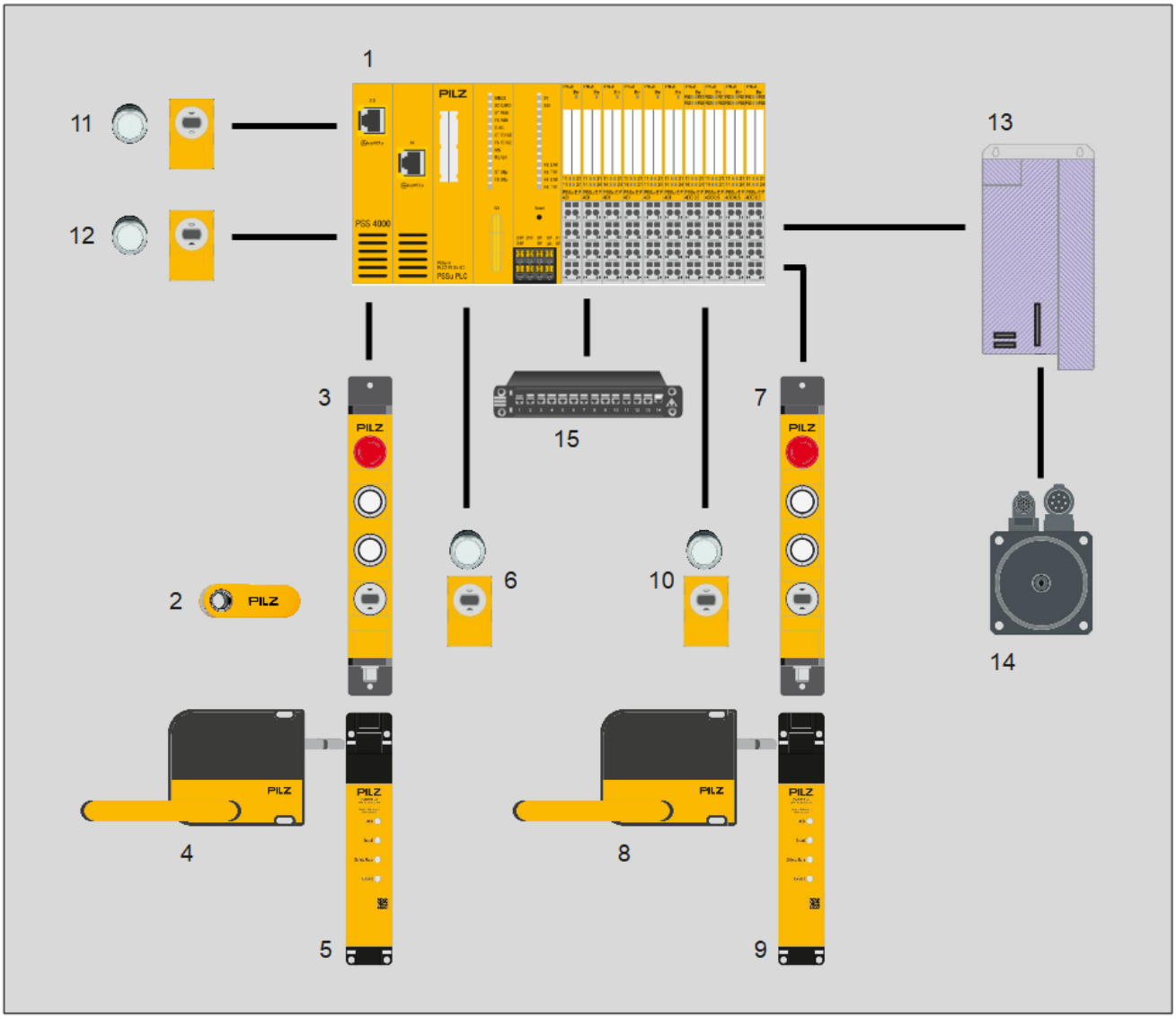


Figure 1: Application – Structure of the hardware (schematic)

1. Failsafe controller PSSu PLC
2. Transponder key
3. PITgatebox – pushbutton unit with PITreader (Safety gate 1 outside)
4. Door handle module for PSENmlock (Safety gate 1 outside)
5. Safety gate system PSENmlock (Safety gate 1 outside)
6. PITreader and pushbutton (Safety gate 1 inside)
7. PITgatebox – pushbutton unit with PITreader (Safety gate 2 outside)
8. Door handle module for PSENmlock (Safety gate 2 outside)
9. Safety gate system PSENmlock (Safety gate 2 outside)
10. PITreader and pushbutton (Safety gate 2 inside)
11. PITreader and pushbutton (Blind spot check 1)
12. PITreader and pushbutton (Blind spot check 2)
13. Servo amplifier
14. Drive
15. Ethernet switch

4 Application description

4.1 Introduction

On a plant in which the danger zone is accessible via safety gates, "key-in-pocket" is a system that guarantees that the plant cannot (re-)start until the last person has left the danger zone. Each person who accesses the danger zone has to sign in to the key-in-pocket system's internal sign in list. When leaving the danger zone, each person must sign out of the internal sign in list. Transponders from the authentication system PITreader from Pilz are used to sign in and out.

The example shows the application of the key-in-pocket system on a machine that executes hazardous movements within a protective enclosure (protective grille and safety gate).

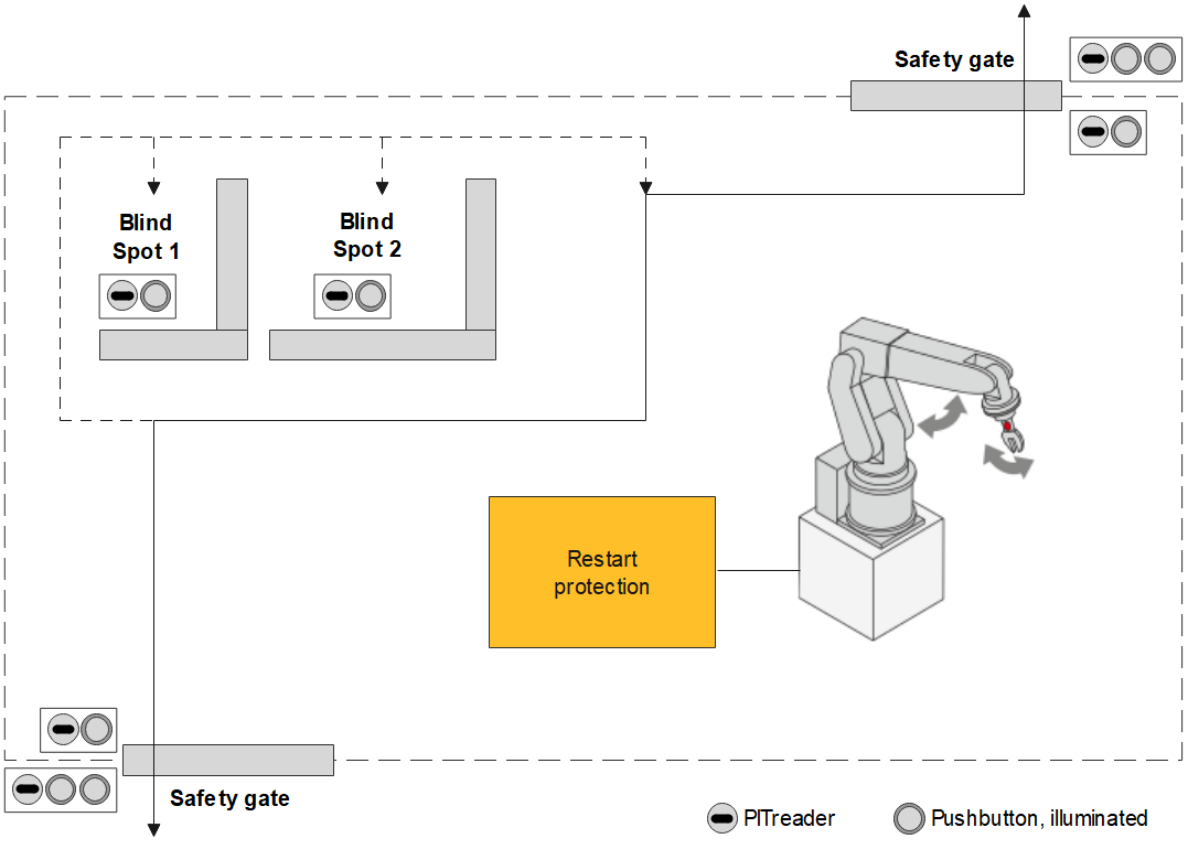


Figure 2: Danger zone

It is possible to enter or leave the danger zone via two access points, which are physically separate. Within the danger zone there are two blind spots, which cannot be seen from the outside. On the inside (within the danger zone), the gates also have devices to activate/deactivate guard locking, so that it is possible to enter the danger via one of the gates and leave via the other. In this example, the blind spots must be checked in a defined sequence.

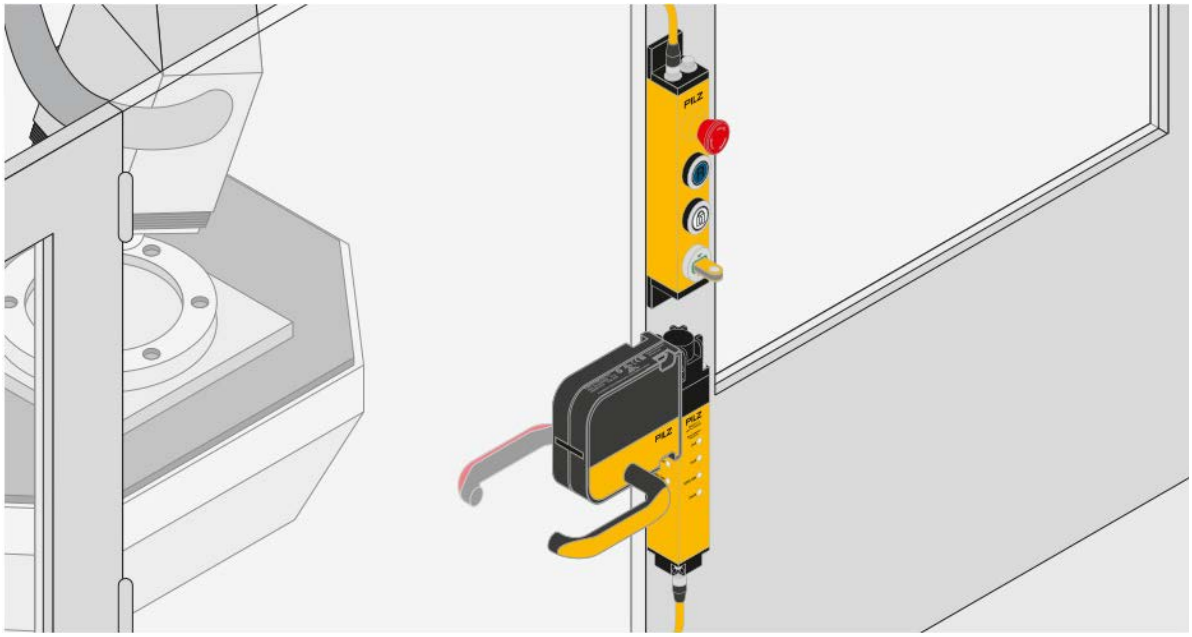


Figure 3: Access to the danger zone

It is only possible to enter the danger zone if:
- A person has used their personal transponder to sign in to the key-in-pocket system's sign in list via the PITreader integrated within the PITgatebox
- All hazardous machine movements are stopped
- The guard locking on a safety gate has been deactivated.

Additional persons wishing to enter the danger zone must use their transponder to sign in to the key-in-pocket system's sign in list.

All persons carry their transponders with them while they are in the danger zone.

It is only possible to restart the machine if:
- All except one person have left the danger zone and signed out of the sign in list
- The last remaining person checks and confirms that there is nobody in the blind spots
- Both safety gates are closed and the guard locking devices are activated
- The last remaining person has used their transponder to sign out of the key-in-pocket system's sign in list.

To operate the key-in-pocket system, each gate uses a PITgatebox, which has two illuminated pushbuttons in addition to an E-STOP and the PITreader.

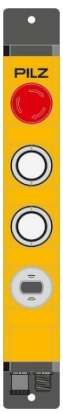


Figure 4: PITgatebox with pushbutton assignment

Activation (lock) and deactivation (unlock) of guard locking are assigned to the upper pushbutton. The status of guard locking (activated/deactivated) is displayed via the integrated indicator lamp.

The lower pushbutton is used to sign a transponder in to the sign in list and to sign a transponder out of the sign in list. The status of a transponder (signed in/signed out) is displayed via the integrated indicator lamp.

In each case, a PITreader and an illuminated pushbutton are provided inside the danger zone for operating the guard locking devices and checking the blind spots.

Two further pushbuttons are required to start and stop the machine.

A *PSEN ml b 2.1* (PSENmlock) is used in conjunction with a handle module *PSEN ml DHM down l 2.1* for safety gate monitoring and guard locking.

4.2 Key-in-Pocket

The basic function of the key-in-pocket system in this application is described below, based on the machine in operation.

The operator wishes to enter the danger zone. To do this, on a gate he positions his transponder in the PITreader, presses the lower pushbutton (sign in/sign out) and then releases it (0.5 … 5 seconds).

The operator ensures that the indicator lamp on the lower pushbutton switches to a continuous light; i.e. the transponder is signed in to the sign in list.

By signing in to the sign in list, a machine stop is requested automatically. Also, signing in to the sign in list activates the display for the status of guard locking. The indicator lamp on the upper pushbutton (lock/unlock) switches to continuous light, so indicating that guard locking is activated.

When the machine is at standstill, the operator can deactivate guard locking by pressing the upper pushbutton (lock/unlock). The indicator lamp goes out, so indicating that the gate can be opened.

The operator removes his transponder (the indicator lamp (sign in/sign out) goes out), opens the gate and enters the danger zone.

Additional persons can sign in to the sign in list and enter the danger zone, including via the second gate if necessary.

Once their work is complete, people gradually leave the danger zone and sign out of the sign in list. As soon as there is only transponder signed in to the sign in list, the blind spots must be checked before the last remaining transponder can be signed out. This is displayed by a continuous light on the indicator lamp at the first blind spot (ReadyForCheck).

First, the person with the last remaining transponder closes one of the gates and activates its guard locking by pressing the pushbutton after the transponder is positioned at the relevant control point (inside or outside). The status of guard locking is displayed via the indicator lamp that is integrated within the pushbutton.

Then the person checks both blind spots consecutively and confirms each check by positioning the transponder at the PITreader installed there and pressing the pushbutton (0.5 … 5 seconds). The indicator lamp that is integrated within the pushbutton (ReadyForCheck) goes out.

Finally, the person leaves the danger zone, closes the gate and positions their transponder in the PITreader (outside). The indicator lamp in the lower pushbutton (sign in/sign out) switches to a continuous light.

Pressing the upper pushbutton (lock/unlock) activates guard locking. The activated status is displayed via the indicator lamp staying continuously lit.

The operator presses the lower pushbutton (sign in/sign out) and then releases it (0.5 … 5 seconds). The indicator lamp goes out, so indicating that the transponder has been signed out of the sign in list. The operator removes his transponder from the PITreader.

If nobody else has signed in to the sign in list in the meantime, then the enable to start the machine is now present (separate Start button). Signing out the last transponder from the sign in list resets the restart interlock.

Any person who enters the danger zone must use their transponder to sign in to the sign in list and then sign out again after leaving the danger zone. A maximum of 21 people can sign in to the sign in list.

The following blocks are used for the key-in-pocket system: “FS_KeyInPocket_SignInOut”, “FS_KeyInPocket_Manager” and “FS_KeyInPocket_BlindSpotCheck”.

Where people have signed in to the sign in list before entering the danger zone but have not signed out of the sign in list after leaving the danger zone and are also no longer available, the block “FS_KeyInPocket_Manager” has the option to delete the list using a transponder with a higher permission and a separate pushbutton (“Delete”). The data on the signed in transponder (security ID and serial number) can be read on the block "FS_KeyInPocket_Manager".

4.3 Monitoring of safety gate and guard locking

The safety gate system PSENmlock signals the status of the safety gate and guard locking via 2 OSSD outputs.

A block with automatic reset, “FS_SafetyGate”, is used to evaluate the OSSD outputs.

Guard locking can only be activated/de-activated when the transponder is positioned and signed in.

Guard locking can only be deactivated once the hazardous movement has ended (safe standstill monitoring).

The functional state of guard locking (locked/unlocked) remains active even after the supply voltage is removed.

4.4 Drive

With the request to stop the machine (sign in to the sign in list, stop button), the motor is stopped and shut down via the servo amplifier, via a preset ramp.

To ensure that the operator does not access the danger zone until the danger has passed, a delay device is started when the machine is stopped. This ensures that any movement has come to a standstill before guard locking on the safety gate can be opened (delay time > maximum stopping time of the motor).

Once the set delay time has elapsed, two safe outputs are shut down, whereby the servo amplifier’s pulse inhibitor safely removes the power to the motor (STO).

The safe standstill monitoring implemented here in the form of a delay time is only one example and should be defined and implemented by the user to suit the specific application.

This example does not describe the design and functionality of the servo amplifier in any detail. The user must select an appropriate drive to suit their application and the safety level it requires.

4.5 Safety assessments

4.5.1 Key-in-pocket system

The system provides protection against an unintended and unauthorised restart. The restart interlock is set (“Enable” = FALSE on the block “FS_KeyInPocket_Manager”), as soon as one person uses their transponder to sign in to the sign in list, and reset (“Enable” = TRUE), when a person has signed out the list’s last remaining transponder from the list.

The safety concept is based on each action (signing in/signing out of the sign in list) requiring operator action on two logically, technologically and physically independent components:

- PITreader with transponders and transfer via network protocol
- Pushbutton on a safe hardware input

Both function elements are considered as independent channels.
Measures to detect and manage single errors are implemented in both channels.

The following measures are used in the channel with the PITreader:
- Data transfer is monitored through data dynamisation
- Data is secured via CRC
- Data on a positioned transponder is valid for a limited time
- Limited to one action while a transponder is positioned
- Data connection is monitored

The following measures are used in the channel with pushbutton:
- A rising and falling edge must be detected to trigger an action.
- An operation is classed as valid if the falling edge occurs between 500 ms and 5 s after the rising edge.

When deleting the list via the block “FS_KeyInPocket_Manager”, the transponder used for deletion remains in the sign in list, so that the delete operation does not reset the restart interlock.

The sign in list is empty when the safety controller starts up (no non-volatile memory). In order to set the enable for the restart, a transponder must be signed in once to the sign in list and then signed out again.

For safe application of the key-in-pocket system, please note the guidelines stated in chapter 7 Application conditions [📖 32].

4.5.2 Safety gate system

The safety gate system PSENmlock meets the following safety requirements:

- Safe guard locking for swing gates and sliding gates
  (the safety switch may only be used with the corresponding actuator.)
- Errors on the OSSD signals (no synchronous switching, OSSD failing to switch from TRUE to FALSE when guard locking is deactivated) are detected by the block “FS_SafetyGate”. A short or cross-short between the OSSDs is detected by the PSENmlock and leads to the OSSDs being shut down. (The safety outputs (OSSDs) must not be connected to 24V.)
- Errors on the 2-channel operation of the guard locking device or servo amplifier (stuck-at-high, cross-short) are detected by the safety controller (on and off test). In the event of an error, guard locking is still guaranteed.

- After the drive has stopped or when the safety gate is opened, the energy supply to the motor is forcibly removed (activation of STO). As a result, the drive can no longer generate a rotational torque and therefore no braking torque either.
  Additional hazards may therefore arise, which must be taken into consideration. For example:
  – Increased time to standstill (overrun)
  – Uncontrolled falling (e.g. on vertical axes)
  – Positional change due to mass, pressure or voltage

Please comply with the safety guidelines and installation and wiring instructions in the PSENmlock operating manual.

4.5.3 Drive system

Please comply with the safety guidelines stated in the operating manual for the relevant servo amplifier.

4.6 Functional safety

4.6.1 Introduction

The restart interlock implemented through the key-in-pocket system is treated as a separate safety function. The safety function depends equally on both input functions (pushbutton und PITreader). PITreader and pushbutton can therefore be regarded as channels of the "Input" subsystem. The key-in-pocket system can be used for applications up to Category 3 PL d of EN ISO 13849-1 or up to SIL 2 of EN 62061.

The safety gate system PSENmlock achieves classification to PL e of EN ISO 13849-1 and SIL 3 of EN 62061, both for safety gate monitoring and mechanical guard locking.
Fault exclusion is assumed for the single-channel mechanical actuator (F_{max} = 2x F_{ZH}).

All the units used within a safety function must be considered when calculating the safety characteristic data.

4.6.2 Safety-related characteristic data in accordance with EN ISO 13849-1

No.	Safety function	Performance level	Safety-related parts of the control system
1	Restart is prevented as long as one person is signed in to the sign in list with their transponder. (KEY-IN-POCKET).	PL d	Sensor (Pushbutton/PITreader PITgatebox) Input (PSSu E F 4DI) Logic (PSSu H PLC2 FS SN SD) Output (PSSu E F 4DO 0.5) Actuator (Servo amplifier)
2	Hazardous movements are shut down when the safety gate is opened; a restart is prevented while the safety gate is open. (SAFETY GATE)	PL e	Sensor (PSENmlock (OSSD)) Input (PSSu E F 4DI) Logic (PSSu H PLC2 FS SN SD) Output (PSSu E F 4DO 0.5) Actuator (Servo amplifier)
3	Restart is prevented if guard locking is not activated. (GUARD LOCKING – start-up)	PL e	Sensor (PSENmlock (OSSD)) Input (PSSu E F 4DI) Logic (PSSu H PLC2 FS SN SD) Output (PSSu E F 4DO 0.5) Actuator (Servo amplifier)
4	Guard locking cannot be deactivated until a safe state is achieved. (GUARD LOCKING – standstill)	PL e	Logic (PSSu H PLC2 FS SN SD) Output (PSSu E F 4DO 0.5) Actuator (PSENmlock (Guard locking))

Requirements:

No.	Description		Identification
1	Common cause failure (CCF)		Requirements are considered to be met (must be checked when implemented)
2	Mission time		20 years
3	Operation interval (electromechanical components) (according to application-related assumption from this example)	Sensor	4 operations per hour
4	Characteristic data of servo amplifier - STO (assuming):	Actuator	PFH_D = 5E-09
			PL e

Please note the further requirements of EN ISO 13849-1, e.g. requirements for avoiding systematic failures.

4.6.3 Safety-related characteristic data in accordance with EN 62061

No.	Safety-related control function (SRCF)	Safety Integrity Level	Subsystems
1	Restart is prevented as long as one person is signed in to the sign in list with their transponder. (KEY-IN-POCKET).	SIL 2	Sensor (Pushbutton/PITreader PITgatebox) Input (PSSu E F 4DI) Logic (PSSu H PLC2 FS SN SD) Output (PSSu E F 4DO 0.5) Actuator (Servo amplifier)
2	Hazardous movements are shut down when the safety gate is opened; a restart is prevented while the safety gate is open. (SAFETY GATE)	SIL 3	Sensor (PSENmlock (OSSD)) Input (PSSu E F 4DI) Logic (PSSu H PLC2 FS SN SD) Output (PSSu E F 4DO 0.5) Actuator (Servo amplifier)
3	Restart is prevented if guard locking is not activated. (GUARD LOCKING – start-up)	SIL 3	Sensor (PSENmlock (OSSD)) Input (PSSu E F 4DI) Logic (PSSu H PLC2 FS SN SD) Output (PSSu E F 4DO 0.5) Actuator (Servo amplifier)
4	Guard locking cannot be deactivated until a safe state is achieved. (GUARD LOCKING – standstill)	SIL 3	Logic (PSSu H PLC2 FS SN SD) Output (PSSu E F 4DO 0.5) Actuator (PSENmlock (Guard locking))

Requirements:

No.	Description		Identification
1	Common cause failure (CCF)		ß = 2 % (must be checked when implemented)
2	Proof test interval		20 years
3	Operation interval (electromechanical components) (according to application-related assumption from this example)	Sensor	4 operations per hour
4	Characteristic data of servo amplifier - STO (assuming):	Actuator	PFH_D = 5E-09
			SIL 3

Please note the further requirements of EN 62061, e.g. requirements for systematic safety integrity.

5 Hardware configuration

5.1 PITgatebox with PITreader

The PITreader is configured via a web application, which is called up via a standard browser. Settings on the PITreader as well as changes to transponders can generally be made via the web application. A detailed description can be found in the operating manual PITreader.

When delivered, the PITreader in the PITgatebox has the IP address 192.168.0.12.

This example uses transponders that are freely configurable. Before use, they must be assigned a permission via the web application.

5.2 PSSu PLC

5.2.1 Overview

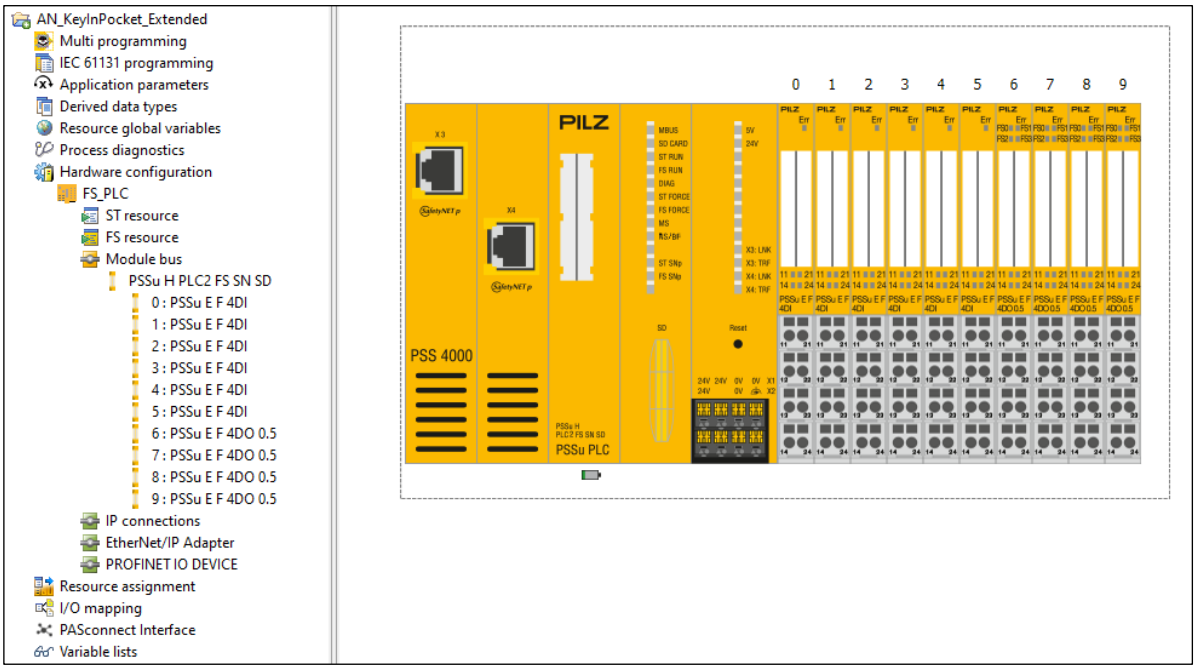


Figure 5: PSS 4000 hardware configuration

The PSSu PLC is assigned the IP address 192.168.0.11.

With the E-modules PSSu E F 4DO 0.5, the off tests must be activated (default setting). The on tests for the outputs that activate guard locking and the servo amplifier must also be activated.

All inputs are configured for a 24 V supply, with the exception of the emergency stop inputs, which are supplied via pulse signals.

5.2.2 IP connections

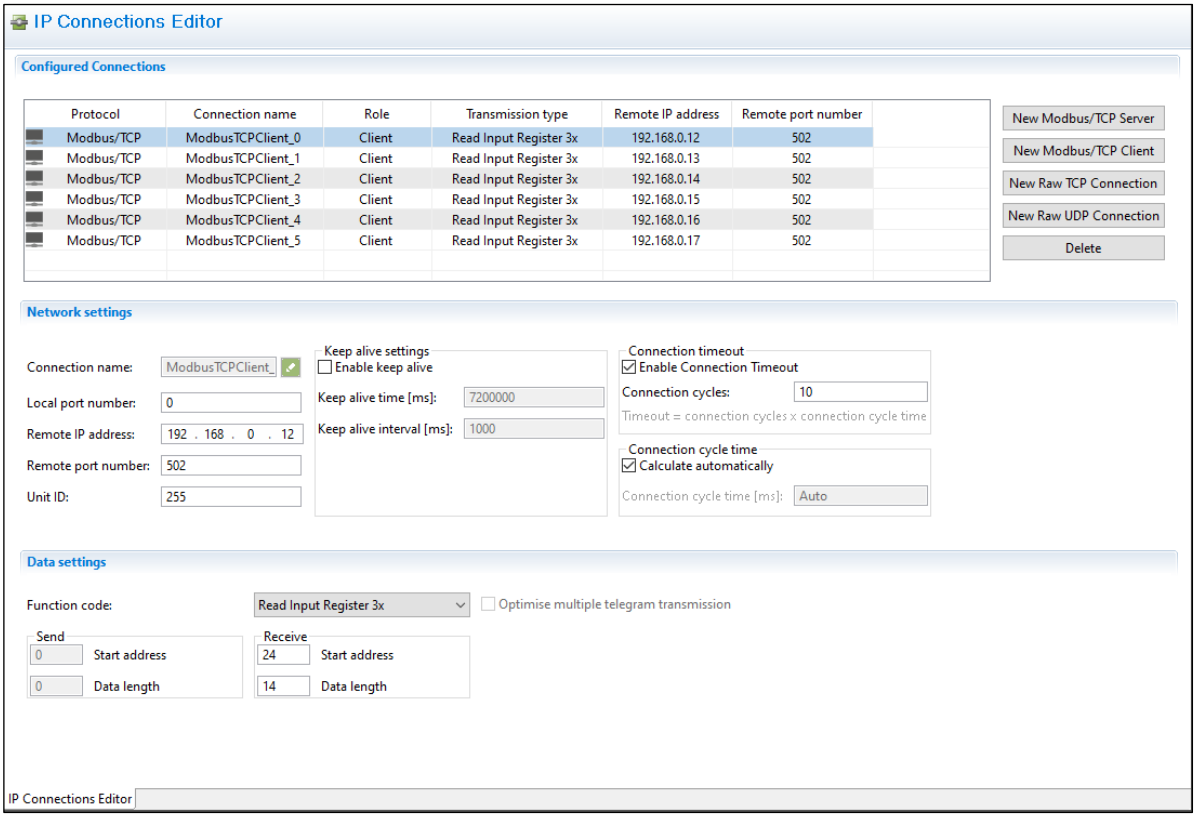


Figure 6: Modbus/TCP client connections

The 6 PITreader units supply data to the PSSu PLC via the IP connections configured above.

6 Software configuration

The software development is shown in the alternative programming types “Multi programming” and “IEC 61131 programming”.

6.1 Multi programming

The program can be divided into 3 functional areas:

- Key-in-pocket system
- Activation and monitoring of the safety gate with guard locking
- Activation of the machine

To keep the illustration understandable, some internal signals have been assigned to PI points and have been connected via PI-PI mapping. These connections can also be created directly without PI points, of course, via lines.

The blue elements are simple component blocks for converting connection points into PI points and vice-versa. The “Button” type elements also supply the valid information from the input signal.

Note: For reasons of clarity, the emergency stop function is not represented here.

6.1.1 Key-in-Pocket

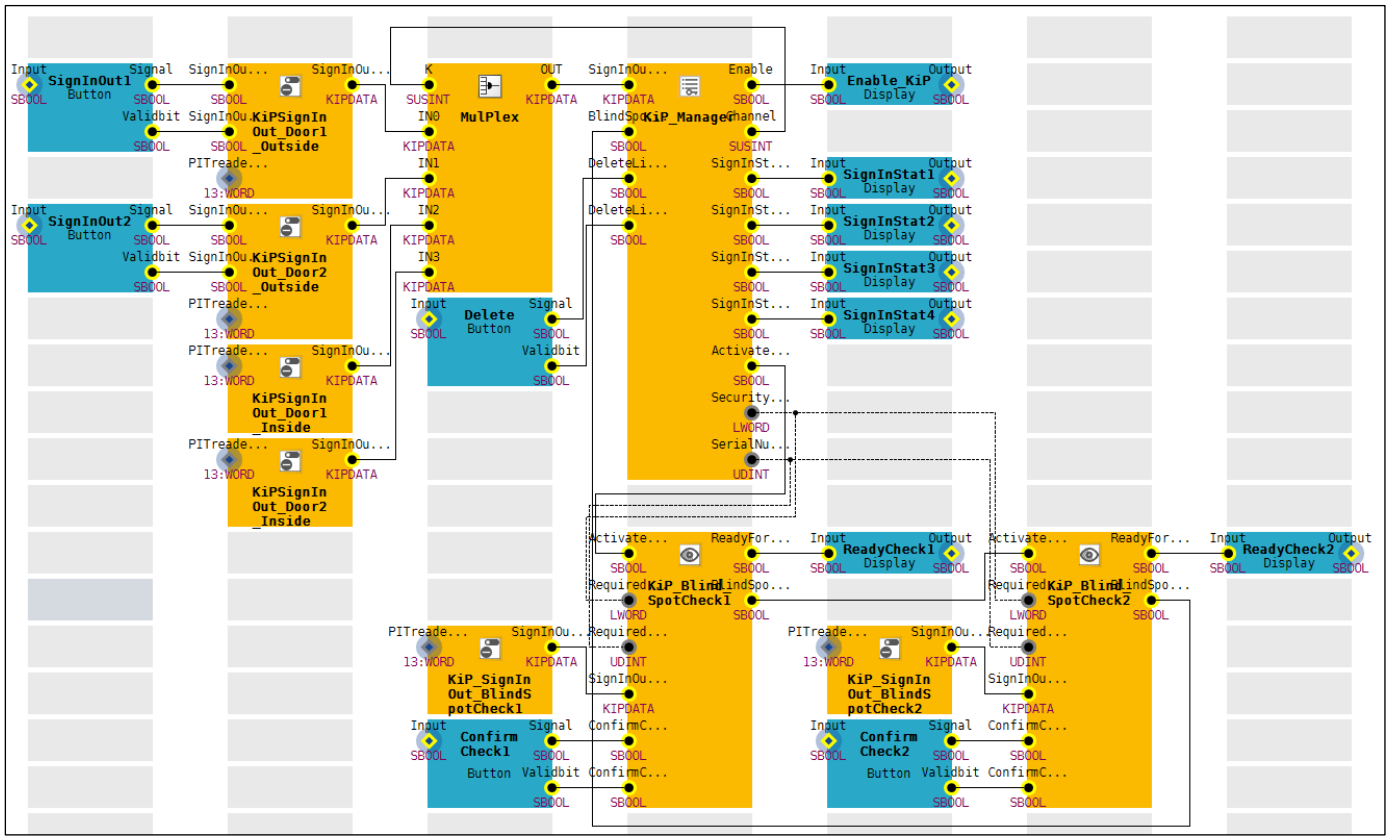


Figure 7: Multi program part “Key-in-Pocket”

The following blocks are used for the key-in-pocket system: “FS_KeyInPocket_SignInOut” (6 instances), “FS_KeyInPocket_Manager” (1 instance) and “FS_KeyInPocket_BlindSpotCheck” (2 instances).

"KiPSignInOut_Door1_Outside" and "KiPSignInOut_Door2_Outside" process the data from the PITreader and the commands from the pushbuttons "SignInOut1" and "SignInOut2" from the gateboxes installed outside beside the gates. The parameter settings are used to activate sign in/sign out and to specify a minimum permission for the transponders:

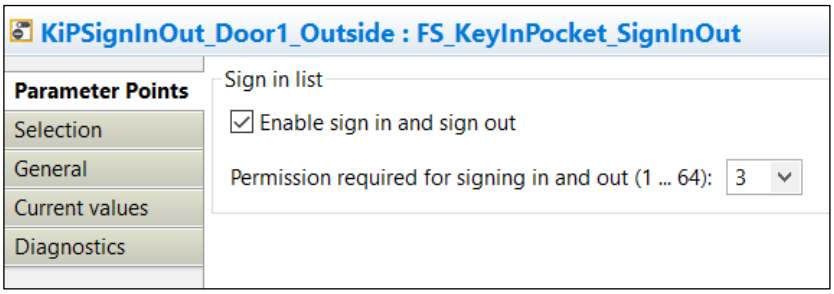


Figure 8: Parameter settings "FS_KeyInPocket_SignInOut" for signing in and out of the sign in list

"KiPSignInOut_Door1_Inside" and "KiPSignInOut_Door2_Inside" process the data from the PITreaders installed in the inside area next to the gates. The parameter settings are used to deactivate sign in/sign out:

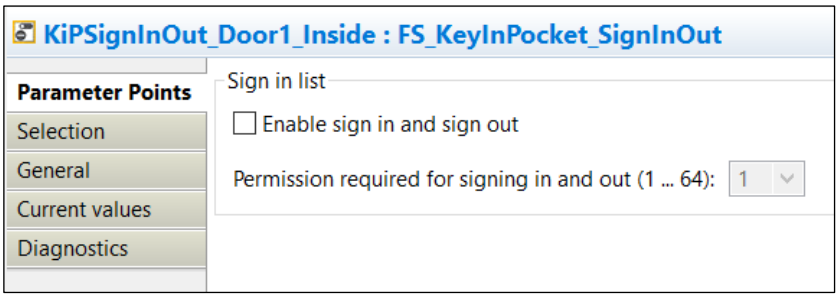


Figure 9: Parameter settings "FS_KeyInPocket_SignInOut" for use without signing in and out of the sign in list

"KiP_Manager" manages the sign in list. The block controls the upstream multiplexer block "MulPlex", receives the data from the "FS_KeyInPocket_SignInOut" instances and signs this in to or out of the internal sign in list. Using a signed-in transponder with a higher permission, the internal list can be deleted apart from one entry, via the "Delete" button. The parameter settings are used to select the blind spot check and to specify the number of channels to be processed, along with the channel and the permission to delete:

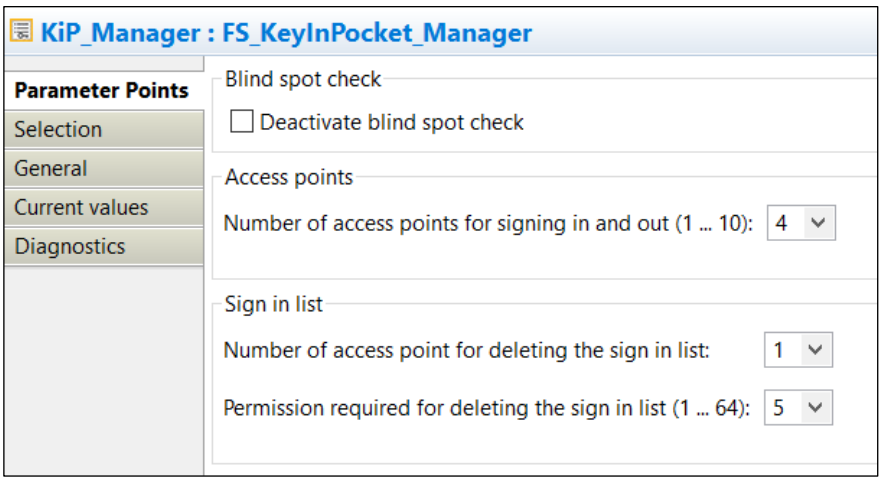


Figure 10: Parameter settings "FS_KeyInPocket_Manager"

The status of a positioned transponder (signed in/signed out) is displayed via the outputs “SignInStat1” ... “SignInStat4”.

The enable output “Enable_KiP” from the key-in-pocket system is subsequently incorporated into the machine’s start and stop conditions (see chapter 6.1.3 Activating the machine [📖 24]).

The blocks “KiP_Blind_SpotCheck1” and “KiP_Blind_SpotCheck2” are used for the serial check of the two blind spots. The parameter settings are used to specify the timeout for the validity of the blind spot check:

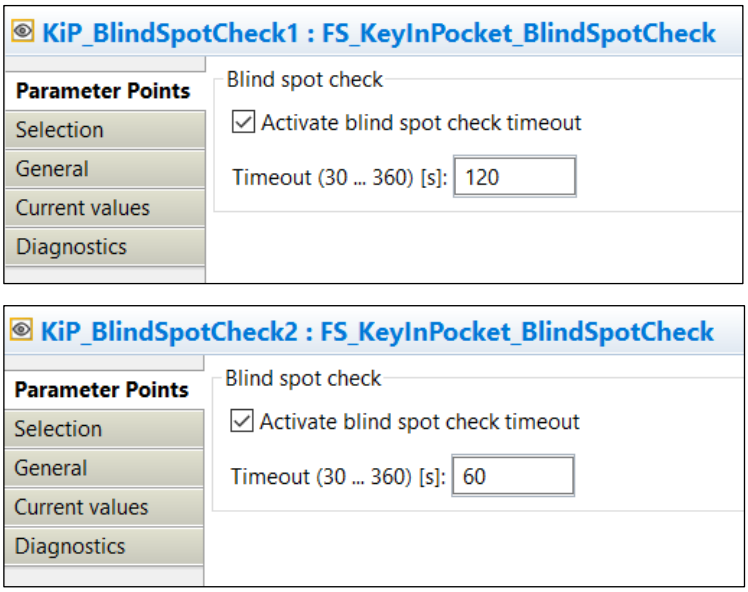


Figure 11: Parameter settings “FS_KeyInPocket_BlindSpotCheck”

The upstream blocks “KiP_SignInOut_BlindSpotCheck1” and “KiP_SignInOut_BlindSpotCheck2” provide the data from the PITreader devices that are installed next to the blind spots. The required transponder data (from the last transponder remaining in the sign in list) is specified by the block “KiP_Manager”.

The first blind spot check is activated via the block “KiP_Manager”; readiness to check is displayed via “ReadyCheck1”. When the blind spot check is confirmed via the pushbutton “ConfirmCheck1”, the second blind spot check is activated via the output “BlindSpotCheckOK” on the block “KiP_Blind_SpotCheck1”. Readiness to check is displayed via “ReadyCheck2”. When the blind spot check is confirmed via the pushbutton “ConfirmCheck2”, feedback is provided to the block “KiP_Manager” via the output “BlindSpotCheckOK” on the block “KiP_Blind_SpotCheck2”.

Via additional outputs, blocks for the key-in-pocket system provide information such as diagnostic messages, a counter for the entries in the sign in list or a list of the signed in transponders. For greater clarity, these outputs are deselected in the current example:

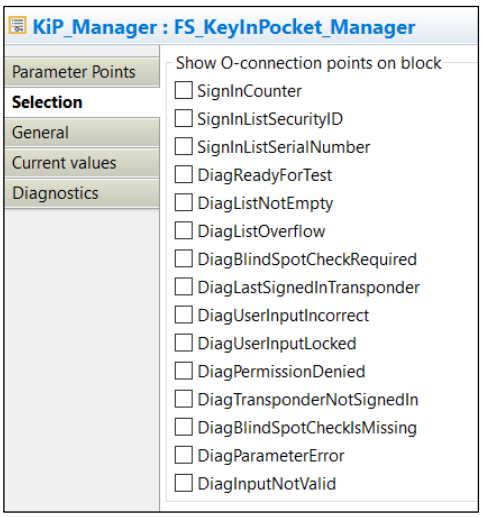


Figure 12: Selection settings "FS_KeyInPocket_Manager"

6.1.2 Activation and monitoring of the safety gate with guard locking

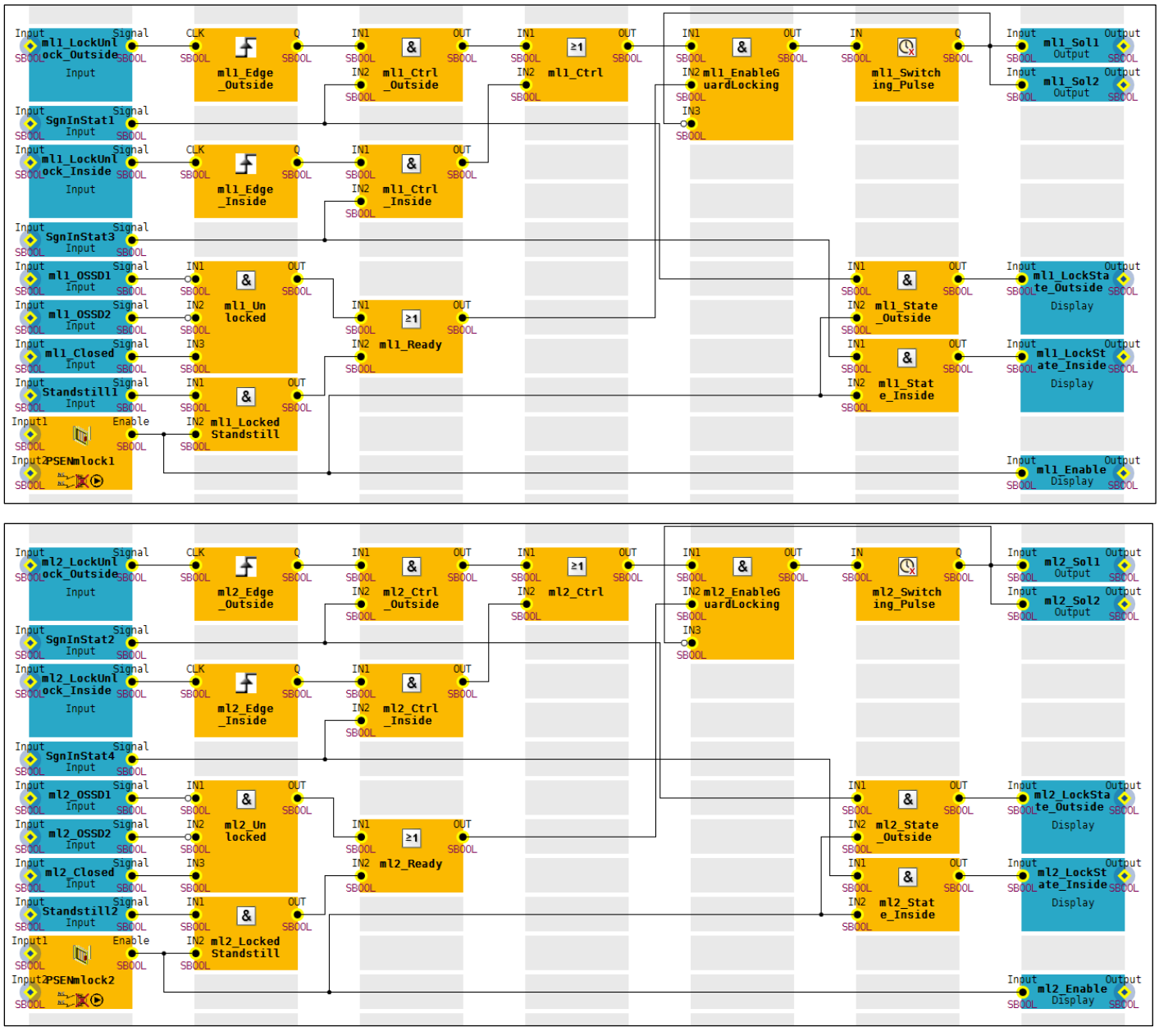


Figure 13: Multi program part "Activating and monitoring the safety gate with guard locking"

The following description refers to the upper section (gate 1), but also applies for the lower section (gate 2).

With a rising edge at the pushbutton "ml1_LockUnlock_Outside" or "ml1_LockUnlock_Inside", guard locking on the PSENmlock1 is activated or deactivated with a pulse at the outputs "ml1_Sol1" and "ml1_Sol2". "ml1_Switching_Pulse" is a switch-off delay "TOF" of 400 ms.

The activation conditions are linked on the block "ml1_EnableGuardLocking".

To **activate** guard locking, the following conditions must be met:

- Safety gate 1 closed, guard locking deactivated: "ml1_Closed" = TRUE, ml1_OSSD1" = FALSE, "ml1_OSSD2" = FALSE (block "ml1_Unlocked")
- Transponder positioned and signed in to the sign in list: "SignInStat1" = TRUE (block "ml1_Ctrl_Outside") or "SignInStat3" = TRUE (block "ml1_Ctrl_Inside")
- The switch-off delay must have elapsed: Q = FALSE (block "ml1_Switching_Pulse")

To **deactivate** guard locking, the following conditions must be met:

- Safety gate 1 closed, guard locking activated: "Enable" = TRUE ("FS_SafetyGate" block "PSENmlock1")
- Machine standstill: "Standstill1" = TRUE (block "ml1_LockedStandstill")
- Transponder positioned and signed in to the sign in list: "SignInStat1" = TRUE (block "ml1_Ctrl_Outside") or "SignInStat3" = TRUE (block "ml1_Ctrl_Inside")
- The switch-off delay must have elapsed: Q = FALSE (block "ml1_Switching_Pulse")

The OSSDs of the PSENmlock on gate 1 are monitored using the block "FS_SafetyGate" ("PSENmlock1"). Simultaneity monitoring of the OSSDs and the reset behaviour are configured via the parameter settings:

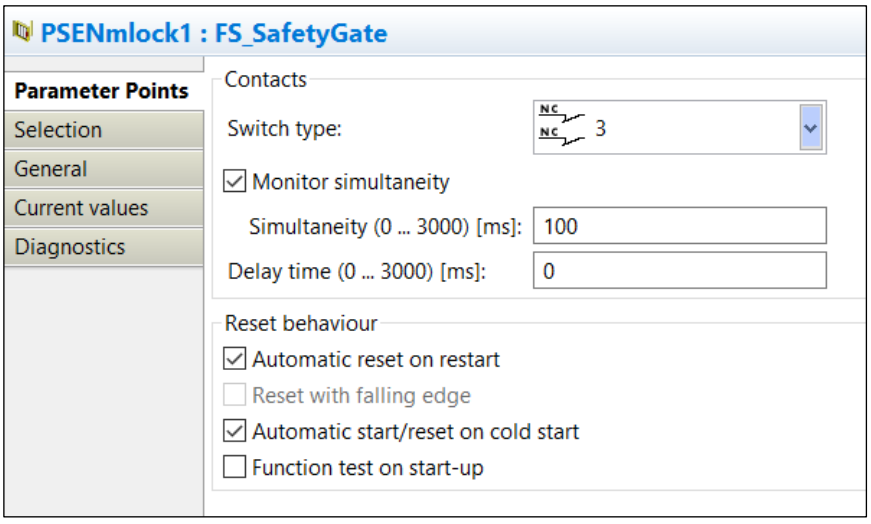


Figure 14: Parameter settings "PSENmlock"

The status of guard locking is forwarded via the output "ml1_Enable" and signalled via the outputs "ml1_LockState_Outside" or "ml1_LockState_Inside", if a transponder is positioned and is signed in to the sign in list ("SignInStat1" = TRUE and block "ml1_State_Outside" or "SignInStat3" = TRUE and block "ml1_State_Inside").

6.1.3 Activating the machine

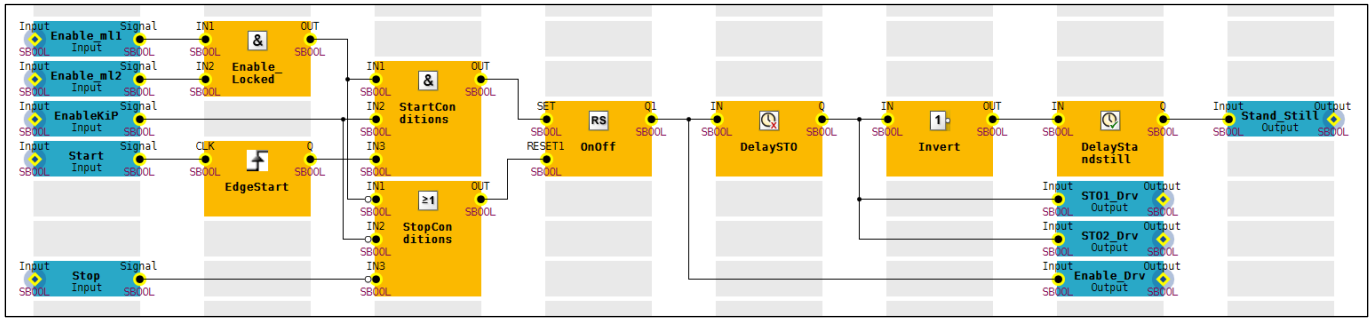


Figure 15: Multi program part "Activating the machine"

With a rising edge at the "Start" button, the machine can be started under the following conditions (block "StartConditions"):

- The sign in list is empty, the key-in-pocket system gives the enable: "EnableKiP" = TRUE
- Both safety gates are closed and the guard locking devices are activated: "Enable_ml1" = TRUE and "Enable_ml2" = TRUE (block "Enable_Locked")

With the start signal, the RS-FlipFlop "OnOff" and the outputs "Enable_Drv", "STO1_Drv" and "STO2_Drv" are set to TRUE.

A machine stop is requested when either (block "StopConditions"):

- A person signs in to the key-in-pocket system's sign in list: "EnableKiP" = FALSE

or

- One of the safety gates with guard locking does not supply an enable: "Enable_Locked" = FALSE

or

- The stop button is operated: "Stop" = FALSE

If a machine stop is triggered (RS-FlipFlop "OnOff" = FALSE), the "Enable_Drv" output for the servo amplifier is switched off immediately, activating the set deceleration ramp in the servo amplifier. Both outputs "STO1_Drv" and "STO2_Drv" are switched off with a time delay, triggering the STO function in the servo amplifier once the drive has come to a standstill. In this case, "DelaySTO" is a switch-off delay "TOF" of 1000 ms.

In this example, standstill detection is implemented via a time delay. By switching off the STO outputs, the switch-on delay "TON" "DelayStandStill" is started via the negation "Invert". The output "Stand_Still" = TRUE is issued once 5000 ms have elapsed.

6.2 IEC 61131 programming (programming language STL: Structured Text)

6.2.1 Declaration part

```
001  PROGRAM FS_Program
002  VAR
003     // Blocks
004     KiP_SignInOut_Door1_Outside                 : FS_KeyInPocket_SignInOut;
005     KiP_SignInOut_Door1_Inside                  : FS_KeyInPocket_SignInOut;
006     KiP_SignInOut_Door2_Outside                 : FS_KeyInPocket_SignInOut;
007     KiP_SignInOut_Door2_Inside                  : FS_KeyInPocket_SignInOut;
008     KiP_SignInOut_BlindSpotCheck1               : FS_KeyInPocket_SignInOut;
009     KiP_SignInOut_BlindSpotCheck2               : FS_KeyInPocket_SignInOut;
010     KiP_BlindSpotCheck1                         : FS_KeyInPocket_BlindSpotCheck;
011     KiP_BlindSpotCheck2                         : FS_KeyInPocket_BlindSpotCheck;
012     KiP_Manager                                 : FS_KeyInPocket_Manager;
013     PSENmlock_Door1                             : FS_SafetyGate_PLC;
014     PSENmlock_Door2                             : FS_SafetyGate_PLC;
015     PSENml1_SwitchingPulse                      : TOF;
016     PSENml2_SwitchingPulse                      : TOF;
```

```
017     PSENml1_LockUnlockEdge_Outside                  : R_TRIG;
018     PSENml1_LockUnlockEdge_Inside                   : R_TRIG;
019     PSENml2_LockUnlockEdge_Outside                  : R_TRIG;
020     PSENml2_LockUnlockEdge_Inside                   : R_TRIG;
021     StartEdge                                       : R_TRIG;
022     OnOff                                           : RS;
023     DelaySTO                                        : TOF;
024     DelayStandstill                                 : TON;
025
026  // PI variables for pushbuttons, display elements and actuator
027     SignInOut1 WITH VALID                       AT %I*  : SAFEBOOL;
028     SignInOut1_Valid EXTENSION VALID OF SignInOut1  AT %I*  : SAFEBOOL;
029     SignInOut2 WITH VALID                       AT %I*  : SAFEBOOL;
030     SignInOut2_Valid EXTENSION VALID OF SignInOut2  AT %I*  : SAFEBOOL;
031     DeleteList WITH VALID                       AT %I*  : SAFEBOOL;
032     DeleteList_Valid EXTENSION VALID OF DeleteList  AT %I*  : SAFEBOOL;
033     BlindSpot1_ConfirmCheck WITH VALID          AT %I*  : SAFEBOOL;
034     BlindSpot1_ConfirmCheck_Valid EXTENSION VALID OF BlindSpot1_ConfirmCheck : SAFEBOOL;
035     BlindSpot2_ConfirmCheck WITH VALID          AT %I*  : SAFEBOOL;
036     BlindSpot2_ConfirmCheck_Valid EXTENSION VALID OF BlindSpot2_ConfirmCheck : SAFEBOOL;
037     SignInState1                                AT %Q*  : SAFEBOOL;
038     SignInState2                                AT %Q*  : SAFEBOOL;
039     BlindSpot1_ReadyForCheck                    AT %Q*  : SAFEBOOL;
040     BlindSpot2_ReadyForCheck                    AT %Q*  : SAFEBOOL;
041     PSENml1_LockUnlock_Outside                  AT %I*  : SAFEBOOL;
042     PSENml1_LockUnlock_Inside                   AT %I*  : SAFEBOOL;
043     PSENml1_Closed                              AT %I*  : SAFEBOOL;
044     PSENml1_OSSD1 WITH VALID                    AT %I*  : SAFEBOOL;
045     PSENml1_OSSD1_Valid EXTENSION VALID OF PSENml1_OSSD1 : SAFEBOOL;
046     PSENml1_OSSD2 WITH VALID                    AT %I*  : SAFEBOOL;
047     PSENml1_OSSD2_Valid EXTENSION VALID OF PSENml1_OSSD2 : SAFEBOOL;
048     PSENml1_Solenoid1                           AT %Q*  : SAFEBOOL;
049     PSENml1_Solenoid2                           AT %Q*  : SAFEBOOL;
050     PSENml2_LockUnlock_Outside                  AT %I*  : SAFEBOOL;
051     PSENml2_LockUnlock_Inside                   AT %I*  : SAFEBOOL;
052     PSENml2_Closed                              AT %I*  : SAFEBOOL;
053     PSENml2_OSSD1 WITH VALID                    AT %I*  : SAFEBOOL;
054     PSENml2_OSSD1_Valid EXTENSION VALID OF PSENml2_OSSD1 : SAFEBOOL;
055     PSENml2_OSSD2 WITH VALID                    AT %I*  : SAFEBOOL;
056     PSENml2_OSSD2_Valid EXTENSION VALID OF PSENml2_OSSD2 : SAFEBOOL;
057     PSENml2_Solenoid1                           AT %Q*  : SAFEBOOL;
058     PSENml2_Solenoid2                           AT %Q*  : SAFEBOOL;
059     PSENml1_LockState_Outside                   AT %Q*  : SAFEBOOL;
060     PSENml1_LockState_Inside                    AT %Q*  : SAFEBOOL;
061     PSENml2_LockState_Outside                   AT %Q*  : SAFEBOOL;
062     PSENml2_LockState_Inside                    AT %Q*  : SAFEBOOL;
063     Start                                       AT %I*  : SAFEBOOL;
064     Stop                                        AT %I*  : SAFEBOOL;
065     Enable_Drv                                  AT %Q*  : SAFEBOOL;
066     STO1_Drv                                    AT %Q*  : SAFEBOOL;
067     STO2_Drv                                    AT %Q*  : SAFEBOOL;
068
069  // Internal variables
070     SignInOutData                                       : KeyInPocketData;
071     SignInState3                                        : SAFEBOOL;
072     SignInState4                                        : SAFEBOOL;
073     PSENml1_Enable                                      : SAFEBOOL;
074     PSENml2_Enable                                      : SAFEBOOL;
075     Enable_KeyInPocket                                  : SAFEBOOL;
076     Enable_Locked                                       : SAFEBOOL;
077     PSENml1_EnableGuardLocking                          : SAFEBOOL;
078     PSENml2_EnableGuardLocking                          : SAFEBOOL;
079     OnOffState                                          : SAFEBOOL;
080     Standstill                                          : SAFEBOOL;
081     StartConditions                                     : SAFEBOOL;
082     StopConditions                                      : SAFEBOOL;
083  END_VAR
```

6.2.2 Instruction part

Note: For reasons of clarity, the emergency stop function is not represented here.
For program description see chapter 6.1 Multi programming [📖 19].

```
// Key in Pocket - Access safety gate 1 outside (sign in/sign out of sign in list)
KiP_SignInOut_Door1_Outside(
   ActivateSignInOut                := TRUE,
   SignInOutPermission              := USINT#3,
   SignInOut                        := SignInOut1,
   SignInOut_Valid                  := SignInOut1_Valid
   );

// Key in Pocket - Access safety gate 2 outside (sign in/sign out of sign in list)
KiP_SignInOut_Door2_Outside(
   ActivateSignInOut                := TRUE,
   SignInOutPermission              := USINT#3,
   SignInOut                        := SignInOut2,
   SignInOut_Valid                  := SignInOut2_Valid
   );

// Key in Pocket - Access safety gate 1 inside (provision of transponder data)
KiP_SignInOut_Door1_Inside(
   ActivateSignInOut                := FALSE,
   SignInOutPermission              := USINT#3
   );

// Key in Pocket - Access safety gate 2 inside (provision of transponder data)
KiP_SignInOut_Door2_Inside(
   ActivateSignInOut                := FALSE,
   SignInOutPermission              := USINT#3
   );

// Key in Pocket - Blind spot 1 (provision of transponder data)
KiP_SignInOut_BlindSpotCheck1(
   ActivateSignInOut                := FALSE
   );

// Key in Pocket - Blind spot 2 (provision of transponder data)
KiP_SignInOut_BlindSpotCheck2(
   ActivateSignInOut                := FALSE
   );

// Multiplexer for transponder data
SignInOutData := MUX(
   K := KiP_Manager.Channel,
   IN0 := KiP_SignInOut_Door1_Outside.SignInOutData,
   IN1 := KiP_SignInOut_Door2_Outside.SignInOutData,
   IN2 := KiP_SignInOut_Door1_Inside.SignInOutData,
   IN3 := KiP_SignInOut_Door2_Inside.SignInOutData
   );

// Key in Pocket - Management of sign in list
KiP_Manager(
   NumberOfChannels                 := USINT#4,
   SignInOutData                    := SignInOutData,
   DeactivateBlindSpotCheck         := FALSE,
   BlindSpotCheckOK                 := KiP_BlindSpotCheck2.BlindSpotCheckOK,
   ChannelDeleteList                := USINT#1,
   PermissionDeleteList             := USINT#5,
   DeleteList                       := DeleteList,
   DeleteList_Valid                 := DeleteList_Valid,
   Enable                           => Enable_KeyInPocket,
   SignInStatus1                    => SignInState1,
   SignInStatus2                    => SignInState2,
   SignInStatus3                    => SignInState3,
   SignInStatus4                    => SignInState4
   );

// Blind spot check 1
// (activated by the last remaining transponder in the sign in list)
KiP_BlindSpotCheck1(
   ActivateBlindSpotCheck           := KiP_Manager.ActivateBlindSpotCheck,
   DeactivateTimeout                := FALSE,
   Timeout                          := T#120s,
   RequiredSecurityID               := KiP_Manager.SecurityID_BlindSpotCheck,
   RequiredSerialNumber             := KiP_Manager.SerialNumberBlindSpotCheck,
```

```
    SignInOutData                   := KiP_SignInOut_BlindSpotCheck1.SignInOutData,
    ConfirmCheck                    := BlindSpot1_ConfirmCheck,
    ConfirmCheck_Valid              := BlindSpot1_ConfirmCheck_Valid,
    ReadyForCheck                   => BlindSpot1_ReadyForCheck
    );

// Blind spot check 2
// (activated by confirmation of blind spot check 1)
KiP_BlindSpotCheck2(
    ActivateBlindSpotCheck          := KiP_BlindSpotCheck1.BlindSpotCheckOK,
    DeactivateTimeout               := FALSE,
    Timeout                         := T#60s,
    RequiredSecurityID              := KiP_Manager.SecurityID_BlindSpotCheck,
    RequiredSerialNumber            := KiP_Manager.SerialNumberBlindSpotCheck,
    SignInOutData                   := KiP_SignInOut_BlindSpotCheck2.SignInOutData,
    ConfirmCheck                    := BlindSpot2_ConfirmCheck,
    ConfirmCheck_Valid              := BlindSpot2_ConfirmCheck_Valid,
    ReadyForCheck                   => BlindSpot2_ReadyForCheck
    );

// Safety gate system PSENmlock - Monitoring of safety gate 1
PSENmlock_Door1(
    SwitchType                      := USINT#3,
    Input1                          := PSENml1_OSSD1,
    Input1_Valid                    := PSENml1_OSSD1_Valid,
    Input2                          := PSENml1_OSSD2,
    Input2_Valid                    := PSENml1_OSSD2_Valid,
    AutoStart                       := TRUE,
    AutoReset                       := TRUE,
    MonitoredReset                  := FALSE,
    StartupTest                     := FALSE,
    SimultaneityTime                := T#100ms,
    DelayTime                       := T#0ms,
    Reset                           := FALSE,
    Enable                          => PSENml1_Enable
    );

// Safety gate system PSENmlock - Monitoring of safety gate 2
PSENmlock_Door2(
    SwitchType                      := USINT#3,
    Input1                          := PSENml2_OSSD1,
    Input1_Valid                    := PSENml2_OSSD1_Valid,
    Input2                          := PSENml2_OSSD2,
    Input2_Valid                    := PSENml2_OSSD2_Valid,
    AutoStart                       := TRUE,
    AutoReset                       := TRUE,
    MonitoredReset                  := FALSE,
    StartupTest                     := FALSE,
    SimultaneityTime                := T#100ms,
    DelayTime                       := T#0ms,
    Reset                           := FALSE,
    Enable                          => PSENml2_Enable
    );

// Enable signal: all safety gates closed and locked
Enable_Locked := PSENml1_Enable AND PSENml2_Enable;

// Safety gate system PSENmlock safety gate 1
// Activate/deactivate guard locking from outside via rising edge
PSENml1_LockUnlockEdge_Outside(
    CLK := PSENml1_LockUnlock_Outside
    );

// Safety gate system PSENmlock safety gate 1
// Activate/deactivate guard locking from inside via rising edge
PSENml1_LockUnlockEdge_Inside(
    CLK := PSENml1_LockUnlock_Inside
    );

// Conditions for activating/deactivating guard locking safety gate 1
PSENml1_EnableGuardLocking :=
    ((SignInState1 AND PSENml1_LockUnlockEdge_Outside.Q) OR
     (SignInState3 AND PSENml1_LockUnlockEdge_Inside.Q))
    AND
    ((PSENml1_Closed AND NOT PSENml1_OSSD1 AND NOT PSENml1_OSSD2) OR
     (PSENml1_Enable AND Standstill))
    AND NOT PSENml1_SwitchingPulse.Q;

// Safety gate system PSENmlock safety gate 1 - Control pulse for guard locking
```

```
PSENml1_SwitchingPulse(
   IN := PSENml1_EnableGuardLocking,
   PT := T#400ms
   );
PSENml1_Solenoid1 := PSENml1_SwitchingPulse.Q;
PSENml1_Solenoid2 := PSENml1_SwitchingPulse.Q;

// Safety gate system PSENmlock safety gate 2 - Activate/deactivate guard locking from
outside via edge
PSENml2_LockUnlockEdge_Outside(
   CLK := PSENml2_LockUnlock_Outside
   );

// Safety gate system PSENmlock safety gate 2 - Activate/deactivate guard locking from
inside via edge
PSENml2_LockUnlockEdge_Inside(
   CLK := PSENml2_LockUnlock_Inside
   );

// Conditions for activating/deactivating guard locking safety gate 2
PSENml2_EnableGuardLocking :=
   ((SignInState2 AND PSENml2_LockUnlockEdge_Outside.Q) OR
    (SignInState4 AND PSENml2_LockUnlockEdge_Inside.Q))
   AND
   ((PSENml2_Closed AND NOT PSENml2_OSSD1 AND NOT PSENml2_OSSD2) OR
    (PSENml2_Enable AND Standstill))
   AND NOT PSENml2_SwitchingPulse.Q;

// Safety gate system PSENmlock safety gate 2 - Control pulse for guard locking
PSENml2_SwitchingPulse(
   IN := PSENml2_EnableGuardLocking,
   PT := T#400ms
   );
PSENml2_Solenoid1 := PSENml2_SwitchingPulse.Q;
PSENml2_Solenoid2 := PSENml2_SwitchingPulse.Q;

// Safety gate system PSENmlock - Display of guard locking status
PSENml1_LockState_Outside := SignInState1 AND PSENml1_Enable;     // s-gate 1 outside
PSENml2_LockState_Outside := SignInState2 AND PSENml2_Enable;     // s-gate 2 outside
PSENml1_LockState_Inside  := SignInState3 AND PSENml1_Enable;     // s-gate 1 inside
PSENml2_LockState_Inside  := SignInState4 AND PSENml2_Enable;     // s-gate 2 inside

// Drive started via edge (start button normally open)
StartEdge(
   CLK := Start
   );

// Criteria for starting the drive
StartConditions := StartEdge.Q AND Enable_Locked AND Enable_KeyInPocket;

// Criteria for stopping the drive (stop button normally closed)
StopConditions := NOT Stop OR NOT Enable_Locked OR NOT Enable_KeyInPocket;

// RS-Flipflop for storing the status of the drive
OnOff(
   SET    := StartConditions,
   RESET1 := StopConditions,
   Q1     => OnOffState
   );

// Switch off delay for shutting down via STO
DelaySTO(
   IN := OnOffState,
   PT := T#1000ms
   );

// Activating the drive
Enable_Drv := OnOffState;
STO1_Drv   := DelaySTO.Q;
STO2_Drv   := DelaySTO.Q;

// Standstill detection via switch on delay after overrun ends
DelayStandstill(
   IN := NOT DelaySTO.Q,
   PT := T#5000ms,
   Q => Standstill
   );

END_PROGRAM
```

6.3 Resource assignment

The program “FS_Program” must be assigned to a task on the FS resource.

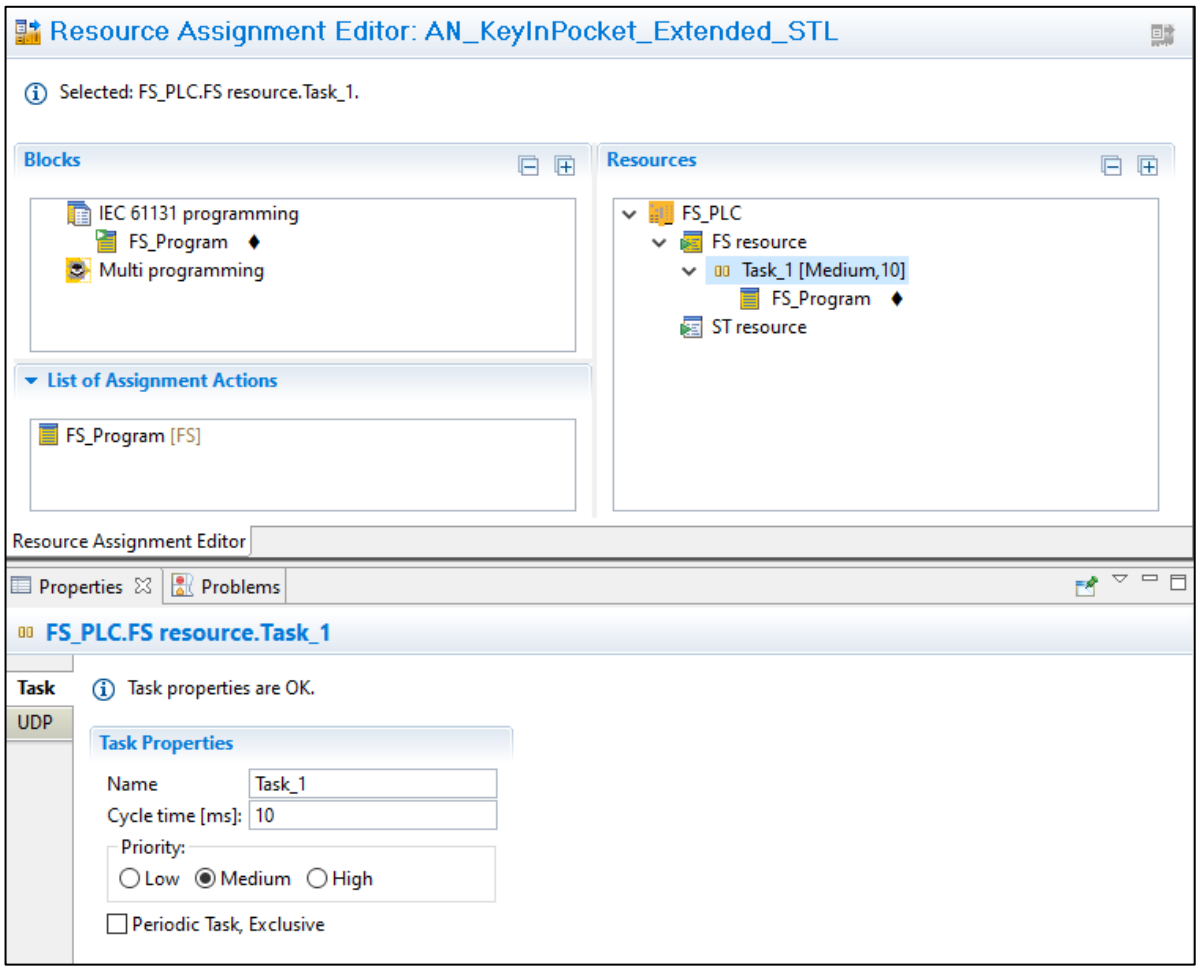


Figure 16: FS resource assignment

6.4 I/O mapping

The PI variables available in the user program can be mapped in the hardware configuration's I/O Mapping Editor. The following illustrations refer to IEC 61131 programming.

The following I/O mappings are to be made with the filter "PI variables <-> Module bus":

Figure 17: I/O mapping PI variables <-> Module bus

The following I/O mappings are to be made with the filter “PI variables <-> IP connections”:

Figure 18: I/O mapping PI variables <-> IP connections

7 Application conditions

NOTICE

When using the key-in-pocket system, make sure that the following training takes place:

- Each person who comes into contact with the plant must be trained to know that they may only access the danger zone if they have previously signed in to the key-in-pocket system's internal sign in list.

- Each user must have an expectation regarding the dynamics of the signal lamp that displays the sign in status. This expectation must be provided through training:

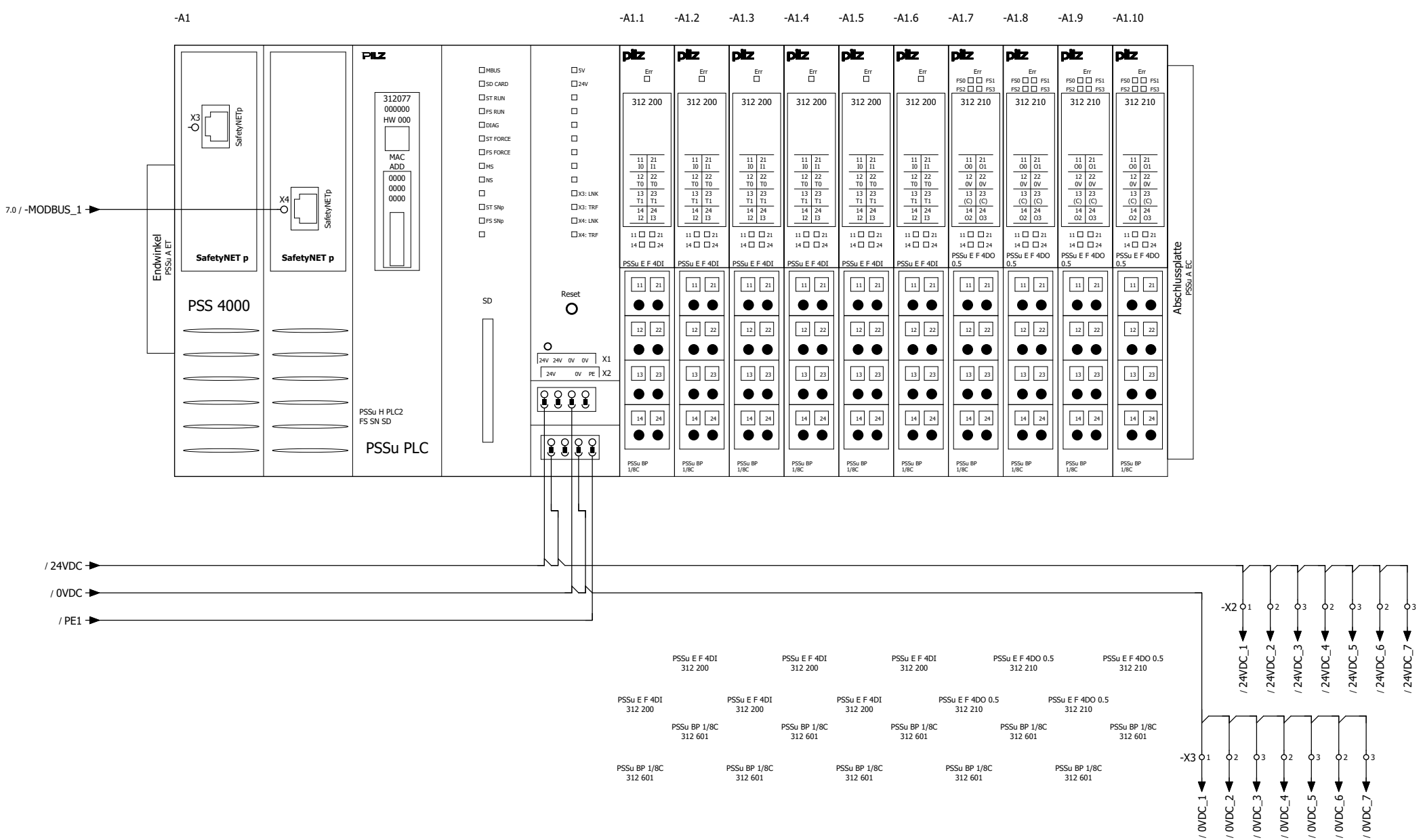
  - If the display element comes on when the transponder is positioned, the transponder is signed in to the sign in list. The display element goes off when the transponder is removed. The plant can be accessed if the person carries the transponder with them.
  - If the display element does not come on when the transponder is positioned, the transponder is not signed in to the sign in list. The plant may not be accessed. Before accessing the plant, the transponder must be signed in to the sign in list.
  - When signing in to the sign in list (transponder is positioned -> display element remains off -> press and release pushbutton), the display element must come on. If the display element does not come on, there is an error (evaluate diagnostic messages). The plant may not be accessed.
  - When signing out of the sign in list (transponder is positioned -> display element comes on -> press and release pushbutton), the display element must go off. If the display element does not go off, there is an error (evaluate diagnostic messages). The plant cannot be put into service.

NOTICE

When using the key-in-pocket system, make sure that the following requirements are met:

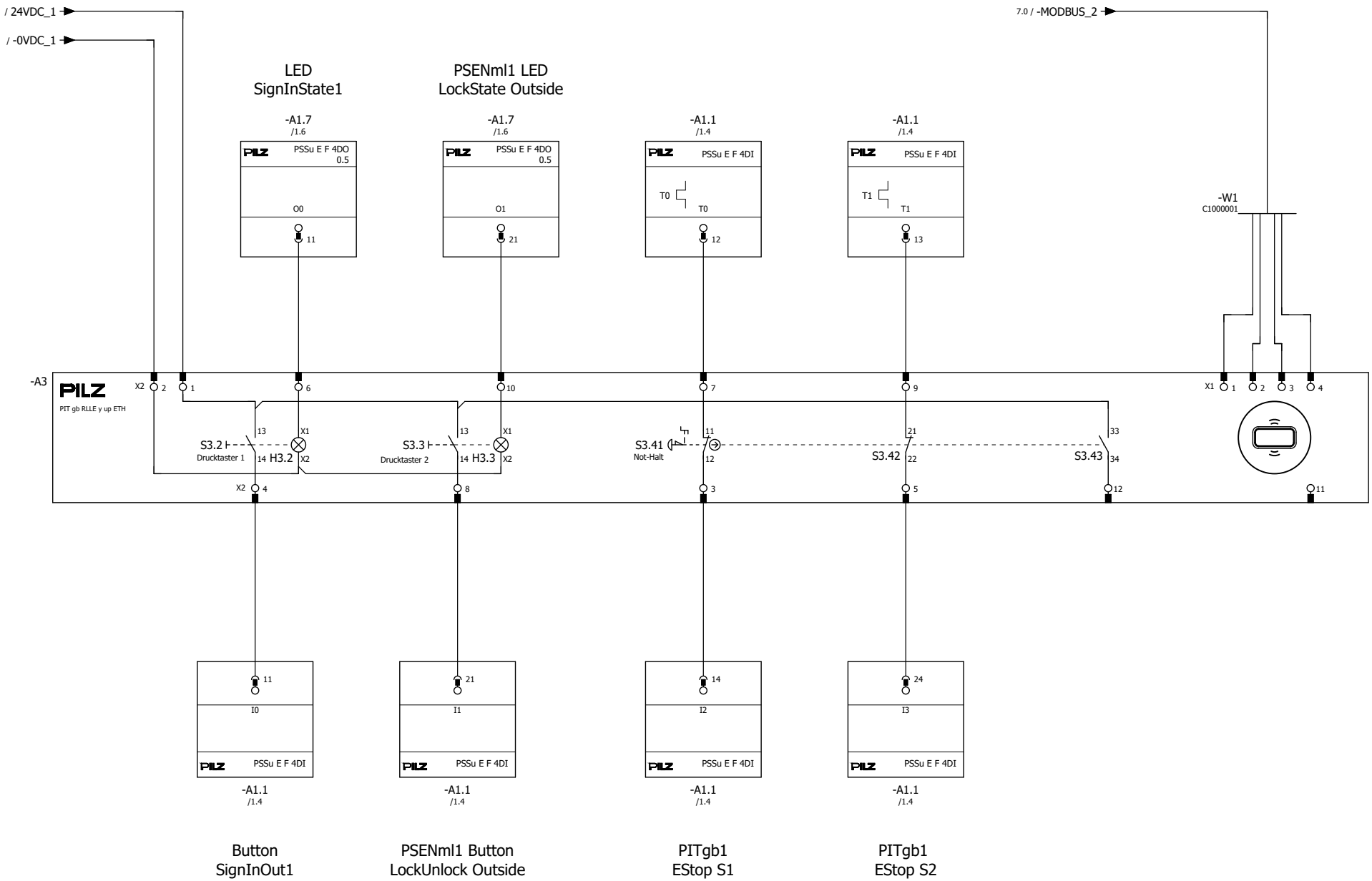
- Where there is a movable guard that prevents access to the danger zone, a person may only open it if they have already signed in successfully to the key-in-pocket system's internal sign in list.
- FS outputs (off tests) must be used to control the display elements that signal the sign in status.
- The operator must use appropriate measures to ensure that the manual deletion of entries from the internal sign in list is restricted to those with appropriate training and is not regarded as a routine operation. This can be supported, for example, by restricting a transponder to certain permissions or by using a dedicated PITreader.
- After intervening in the internal sign in list (deleting entries), appropriate checks in the form of organisational measures must be used to ensure that nobody whose transponder is not included in the internal sign in list is left in the danger zone.
- After the power has failed and then been restored, in addition to the required function test, appropriate checks in the form of organisational measures must be used to ensure that nobody whose transponder is not included in the internal sign in list is left in the danger zone.
- In the following cases, PITreader units may only be used in areas that are accessible after signing into the key-in-pocket system's internal sign in list:
  - If used to check the sign in status of persons in the internal sign in list, with the objective of locking further processes
  - If used on stations for checking blind spots
- If additional functions are to be carried out depending on a transponder's sign in status, then these operations must be confirmed through an additional control element and may not be initiated purely as a result of checking the sign in status.

Please comply with the instructions for installation, wiring, commissioning and operation in the operating manuals for the individual components and in the system descriptions.

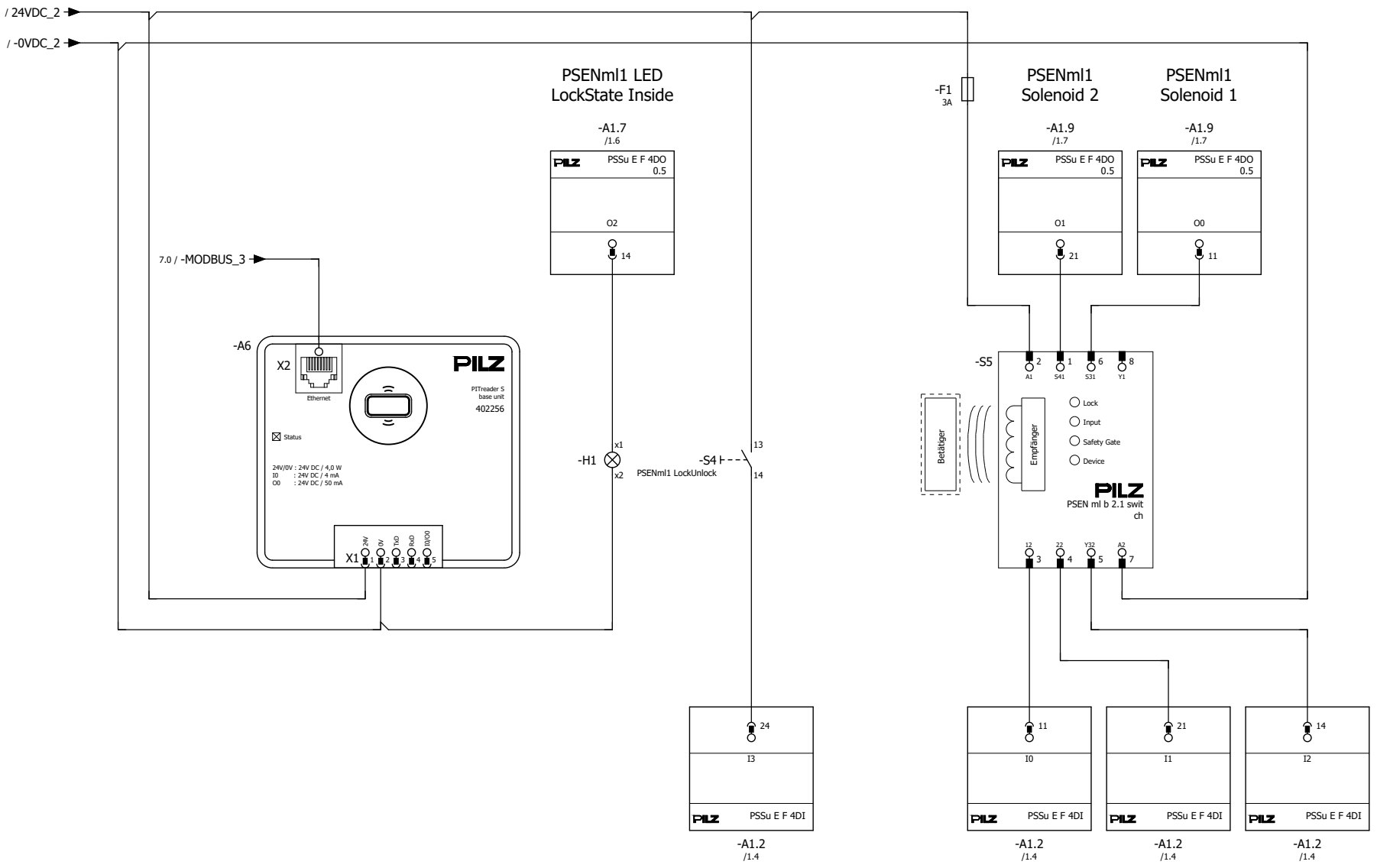


Revision	08.03.2023	Date	27.02.2023	EN ISO 13849-1 PL d	Pilz GmbH & Co. KG Felix-Wankel-Strasse 2 D-73760 Ostfildern	PSSu Basic
Name	Pilz	Name	Pilz			
				EN 62061 SIL 2		
		Dep.	CSI			

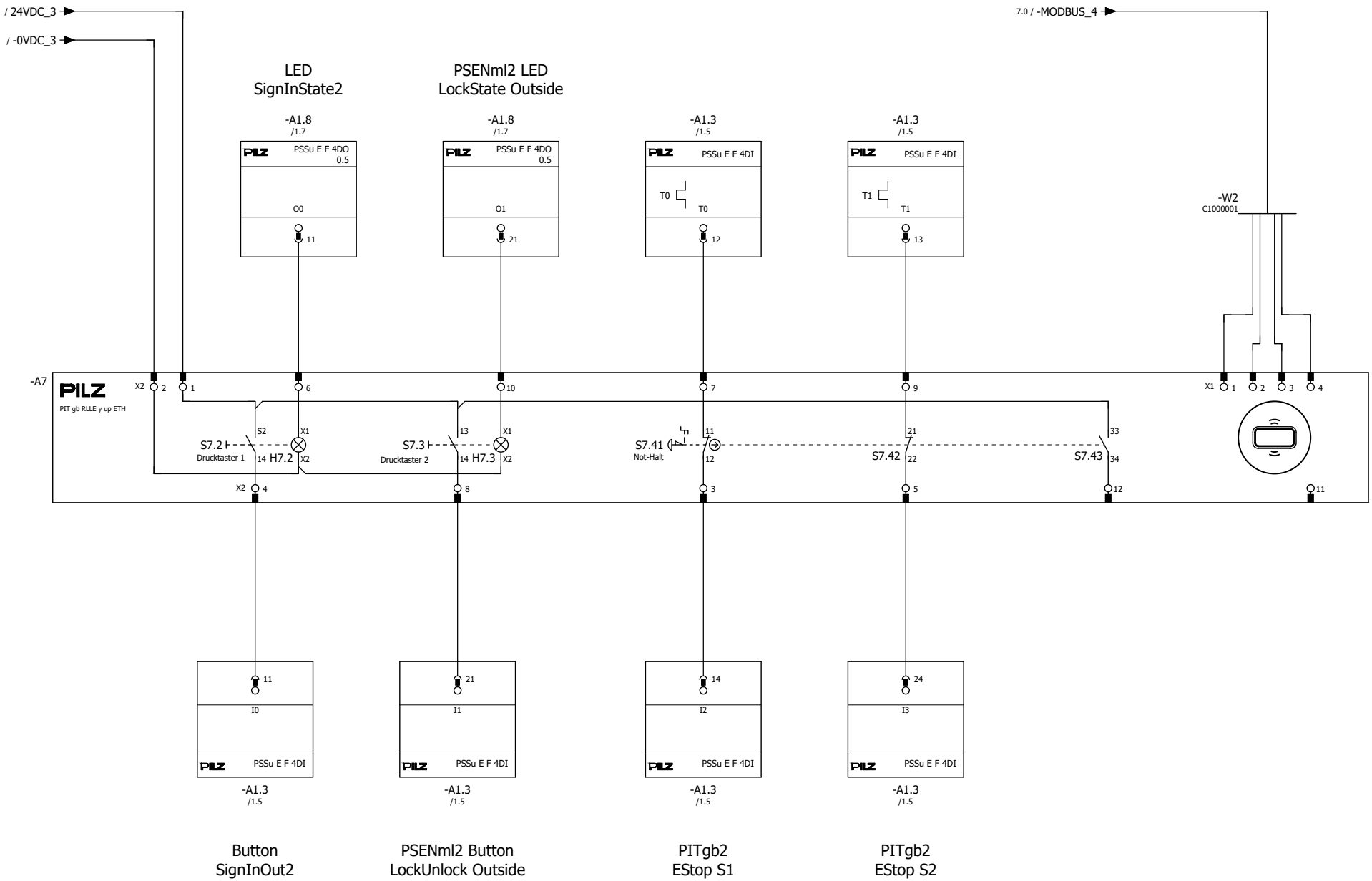
Mounting place
+ AN_1006505_01



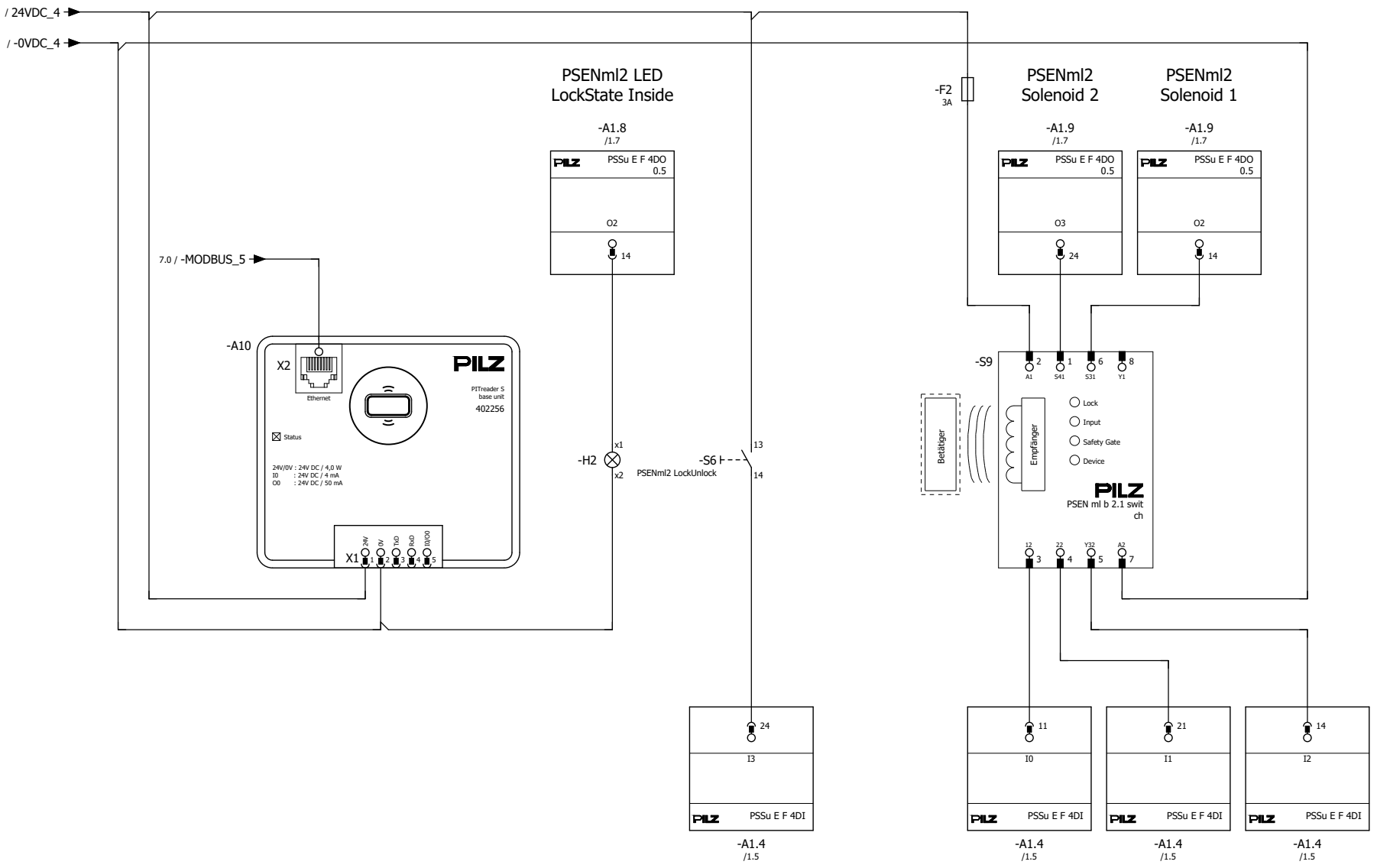
Revision	08.03.2023	Date	27.02.2023	EN ISO 13849-1 PL d	PILZ Pilz GmbH & Co. KG Felix-Wankel-Strasse 2 D-73760 Ostfildern	PSSu PIT gb RLLE 1	Mounting place + AN_1006505_01
Name	Pilz	Name	Pilz				
				EN 62061 SIL 2			Page: 2 / 8
		Dep.	CSI				



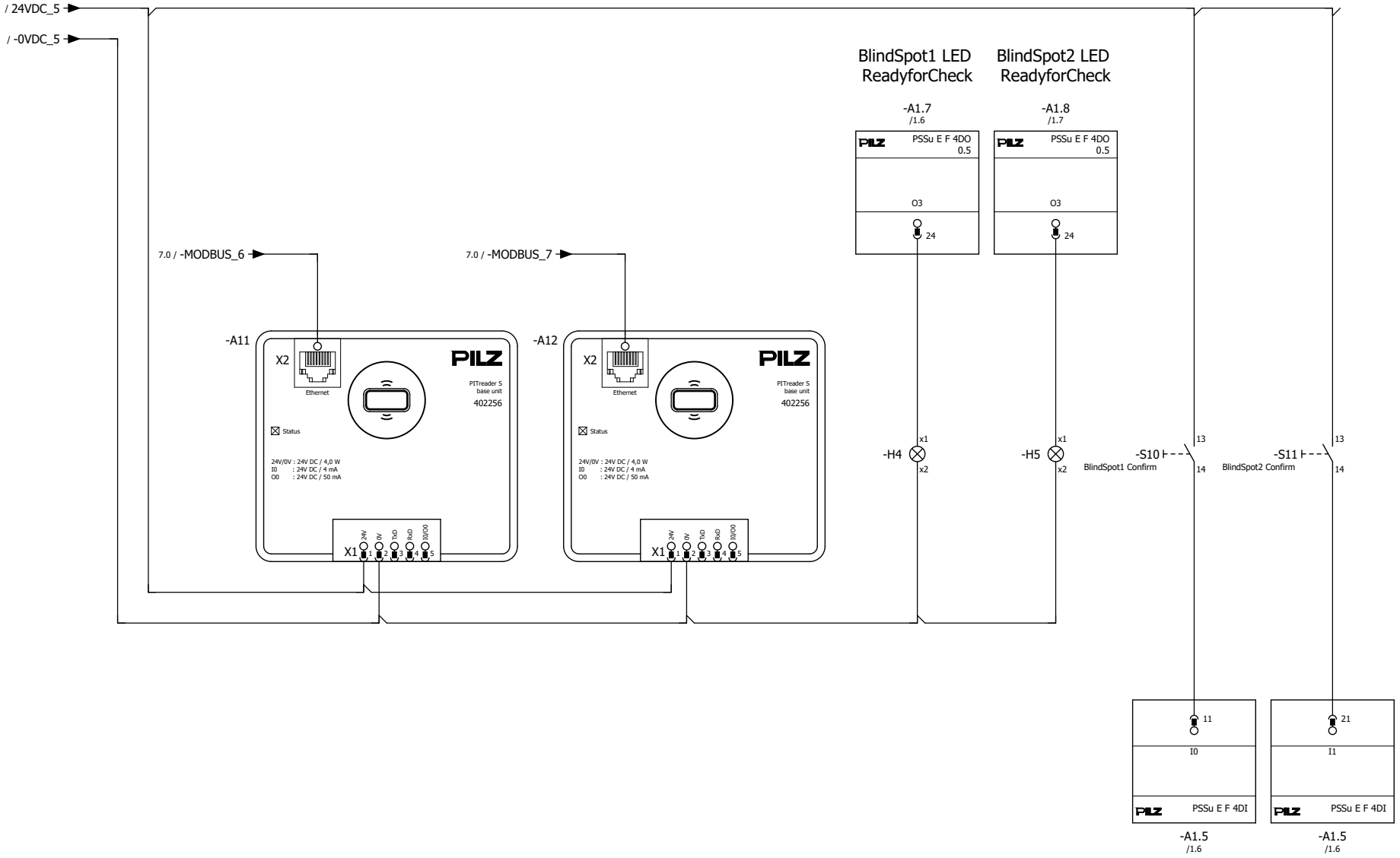
Revision	08.03.2023	Date	27.02.2023	EN ISO 13849-1 PL d	PILZ Pilz GmbH & Co. KG Felix-Wankel-Strasse 2 D-73760 Ostfildern	PSSu PSEN ml b 2.1	Mounting place + AN_1006505_01
Name	Pilz	Name	Pilz				
				EN 62061 SIL 2			Page: 3 / 8
		Dep.	CSI				



Revision	08.03.2023	Date	27.02.2023	EN ISO 13849-1 PL d	Pilz GmbH & Co. KG Felix-Wankel-Strasse 2 D-73760 Ostfildern	PSSu PIT gb RLLE 2	Mounting place + AN_1006505_01
Name	Pilz	Name	Pilz				
				EN 62061 SIL 2			Page: 4 / 8
		Dep.	CSI				



Revision	08.03.2023	Date	27.02.2023	EN ISO 13849-1	PL d	PILZ Pilz GmbH & Co. KG Felix-Wankel-Strasse 2 D-73760 Ostfildern	PSSu PSEN ml b 2.1	Mounting place + AN_1006505_01
Name	Pilz	Name	Pilz					
				EN 62061	SIL 2			Page: 5 / 8
		Dep.	CSI					



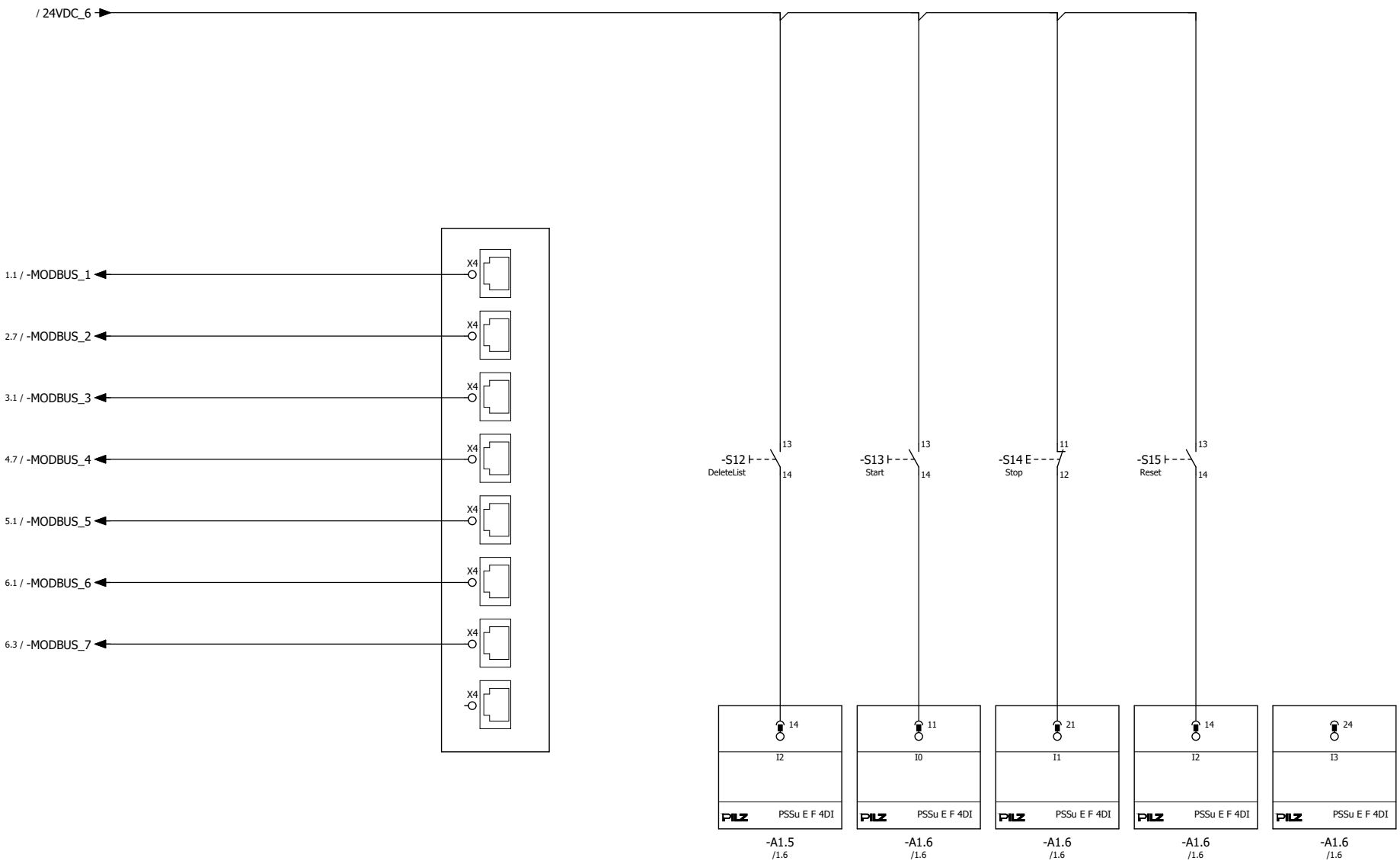
PITreader BlindSpot1

PITreader BlindSpot2

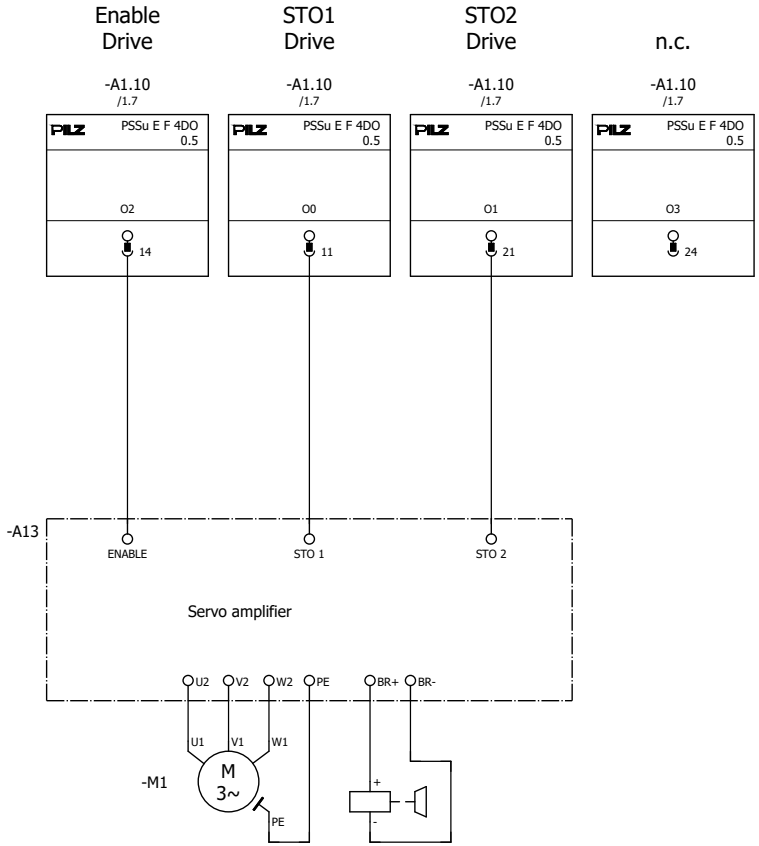
BlindSpot1 Button ConfirmCheck

BlindSpot2 Button ConfirmCheck

Revision	08.03.2023	Date	27.02.2023	EN ISO 13849-1 PL d	Pilz GmbH & Co. KG Felix-Wankel-Strasse 2 D-73760 Ostfildern	PSSu PITreaderBlindSpot	Mounting place + AN_1006505_01
Name	Pilz	Name	Pilz				
				EN 62061 SIL 2			Page: 6 / 8
		Dep.	CSI				



Revision	08.03.2023	Date	27.02.2023	EN ISO 13849-1 PL d	PILZ Pilz GmbH & Co. KG Felix-Wankel-Strasse 2 D-73760 Ostfildern	PSSu Switches	Mounting place + AN_1006505_01
Name	Pilz	Name	Pilz				
				EN 62061 SIL 2			Page: 7 / 8
		Dep.	CSI				



Revision	08.03.2023	Date	27.02.2023	EN ISO 13849-1 PL d	PILZ Pilz GmbH & Co. KG Felix-Wankel-Strasse 2 D-73760 Ostfildern	PSSu Drive	Mounting place + AN_1006505_01
Name	Pilz	Name	Pilz				
				EN 62061 SIL 2			Page: 8 / 8
		Dep.	CSI				

9 Table of figures

Support

Technical support is available from Pilz round the clock.

Americas
Brazil
+55 11 97569-2804
Canada
+1 888 315 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 31 778 3300

Australia
+61 3 95600621

Europe
Austria
+43 1 7986263-0
Belgium, Luxembourg
+32 9 3217570
France
+33 3 88104003
Germany
+49 711 3409-444
Ireland
+353 21 4804983
Italy, Malta
+39 0362 1826711
Scandinavia
+45 74436332
Spain
+34 938497433
Switzerland
+41 62 88979-32
The Netherlands
+31 347 320477
Turkey
+90 216 5775552
United Kingdom
+44 1536 462203

You can reach our international hotline on:
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

BLUECOMPETENCE Alliance Member
Partner of the Engineering Industry Sustainability Initiative

pilz
Partner of:
The Best of German Engineering

1005173-EN-01 Draft 01, 2019-06 Printed in Germany

We are represented internationally. Please refer to our homepage www.pilz.com for further details or contact our headquarters.

Headquarters: Pilz GmbH & Co. KG, Felix-Wankel-Straße 2, 73760 Ostfildern, Germany
Telephone: +49 711 3409-0, Telefax: +49 711 3409-133, E-Mail: info@pilz.com, Internet: www.pilz.com

PILZ
THE SPIRIT OF SAFETY