PSS 4000 Safety Gate with PSENslock Structured Text

Product
Type: FS_SafetyGate, FS_OutputFBL
Name: PSS 4000, Blocks, PAS4000, PLC, STL
Manufacturer: Pilz GmbH & Co. KG, Safe Automation

Document
Release Number: 01
Release Date: 23 April 2012

Application Note - No. 1002518_EN_01

the spirit of safety

Document Revision History

Release	Date	Changes	Chapter
01	2012-04-23	Creation	all

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 know-ledge and belief.
However, we cannot accept liability for the accuracy and entirety of the information provided, except in the case of gross negligence. 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.

April 2012

Support

Technical help round the clock!
Technical support is available from Pilz round the clock.
This service is provided free of charge beyond standard business hours.

Americas
- Brazil
  +55 11 8245-8267
- Mexico
  +52 55 5572 1300
- USA (toll free)
  +1 877-PILZUSA (745-9872)

Asia
- China
  +86 21 62494658-216
- Japan
  +81 45 471-2281
- Korea
  +82 2 2263 9540

Australia
- Australia
  +61 3 95446300

Europe
- Austria
  +43 1 7986263-0
- Belgium, Luxembourg
  +32 9 3217575
- England
  +44 1536 462203
- France
  +33 3 88104000
- Germany
  +49 711 3409-444
- Ireland
  +353 21 4804983
- Italy
  +39 031 789511
- Scandinavia
  +45 74436332
- Spain
  +34 938497433
- Switzerland
  +41 62 88979-30
- The Netherlands
  +31 347 320477
- Turkey
  +90 216 5775552

You can reach our international hotline on:

+49 711 3409-444 or **mailto:support@pilz.com**

Pilz GmbH & Co. KG
Safe Automation
Felix-Wankel-Straße 2
73760 Ostfildern, Germany

Telephone: +49 711 3409-0
Telefax: +49 711 3409-133
E-Mail: pilz.gmbh@pilz.de
Internet: www.pilz.com

Contents

Abbreviations

PAS	**P**ilz **A**utomation **S**uite (software platform)	
PSS	Programmable control system	(DE: **P**rogrammierbares **S**teuerungs**s**ystem)
PNOZ	Pilz E-STOP Positive-guided	(DE: **P**ilz **NO**T-AUS-**Z**wangsgeführt)
POU	**P**rogram **O**rganisation **U**nit	
PRG	**Pr**o**g**ram	
FB	**F**unction **B**lock	
FUN	**Fun**ction	
STL (ST)	**S**tructured **T**ext **L**anguage	

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.
1	Pilz international homepage, download section	www.pilz.com
2	Operating Manual PSSu H PLC1 FS SN SD	21939-EN-xx
3	Operating Manual PSSu E F 4DI	21 311-xx
4	Operating Manual PSSu E F DI OZ 2	21 329-xx
5	Operating Manual PSSu E F 4DO 0.5	21 317-xx
6	Operating Manual PSSu E S 4DI	21 340-EN-xx
7	Operating Manual PSSu E S 4DO 0.5	21 346-EN-xx
8	Operating Manual PSSu E PD	21 305-xx
9	System Description Programmable safety and control system PSS 4000	1001 467-EN-xx
10	Safety Manual Programmable safety and control system PSS 4000	1001 468-EN-xx
11	PAS4000 online help	-
12	Operating Manual PSEN sl-1.0p 2.1	21 910_xx

1.2. Documentation from other sources of information

No.	Description	Item No.
1		
2		

Note
The present example (PSS 4000 Safety Gate with PSENslock) is also available in the programming languages Instruction List and PASmulti.

2. Hardware configuration

2.1. Pilz products

No.	Description	Order number	Version	Number
1	PSSu H PLC1 FS SN SD	312 070	001	1
2	PSSu E F 4DI	312 200	-	1
3	PSSu E F DI OZ 2	312 220	-	1
4	PSSu E F 4DO 0.5	312 210	-	2
5	PSSu E S 4DI	312 400	-	1
6	PSSu E S 4DO 0.5	312 405	-	1
7	PSSu E PD	312 195	-	1
8	PSSu BP 1/8 C	312 601	-	7
9	PSEN sl-1.0p 2.1	570 601	-	1
10	PAS4000	-	v1.5.0	1

2.2. Hardware configuration

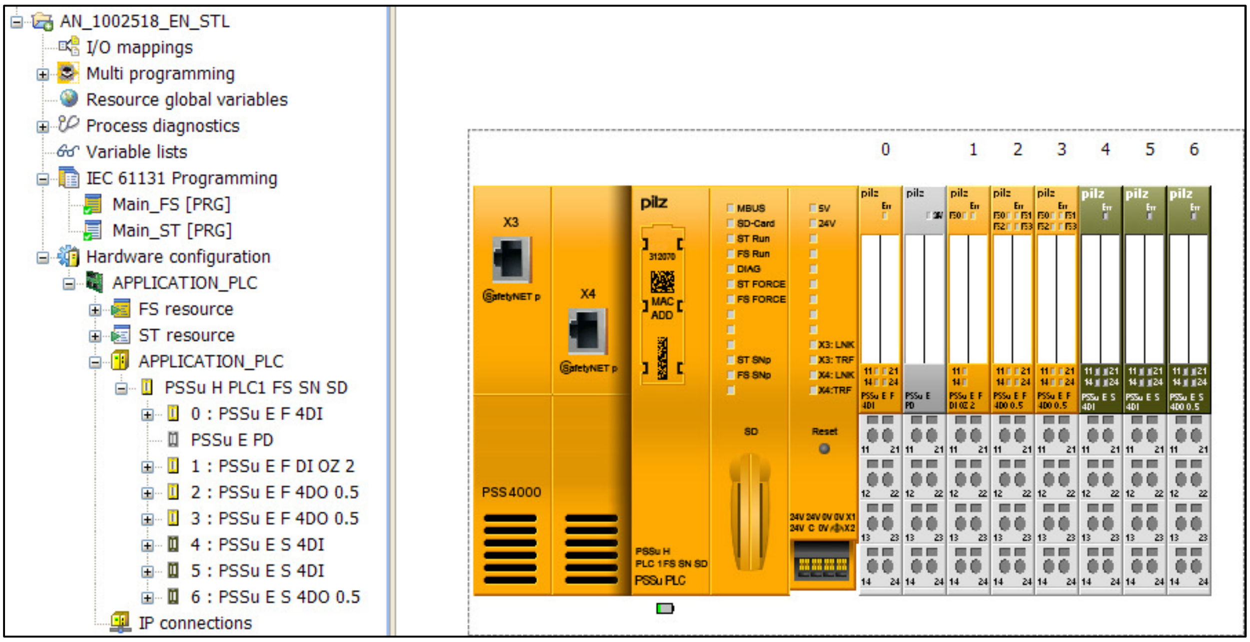


Fig. 1: Hardware configuration

3. Application Task

3.1. Description

The example shows the implementation of a safety gate application with a PSS 4000 PLC.
The safe control and evaluation of the signals is taken over by two Pilz Function blocks (FS-FB) from the library.

- FS_SafetyGate
  CRC ADDB

- FS_OutputFBL
  CRC B3A7

The workflow is divided into the following two main functions:
- Safety Gate and
- Feedback Loop Monitoring

3.1.1. Safety gate monitoring function

The control system monitors the safety gate switch (S1) via the user program.
An instance of the Pilz function block *"FS_SafetyGate"* is assigned to them. This FS-FB detects whether the assigned safety gate switch has been operated, as well as detecting incorrect input signals and whether the contact synchronization time has been exceeded, etc.

If the safety gate switch are operated or an error occurs, the enable output *"Enable"* on the FS-FB will immediately be reset.
The enable output *"Enable"* is also reset when the PSS is stopped and when the PSS is switched on. The signal from the enable output *"Enable"* must be evaluated by the user program and trigger an appropriate reaction.

Based on the diagnostic outputs (*"DiagSwitchError"*, *"DiagInputNotValid"*) it can be determined why "Enable" was reset.
The outputs *"DiagOperated"*, *"DiagReadyForReset"* and *"DiagReadyForTest"* are used as status messages.

A "valid bit" is formed by the system for the respective hardware input to determine whether a process value received from a sensor is valid.
The valid bit is queried in the Function block and indicates whether an error has occurred in the signal transmission between hardware input and processor (such as test clock error, module overheats, etc.).
If the valid bit is FALSE, the process value is invalid and the Pilz function block provides an appropriate diagnostic message. The error signal reset enable.
(For more information, see "Validity process data" in PAS4000 online help)

The way in which the error is reset will depend on the operating mode set on the FS-FB.
In this application example, parameters for FS-FB have been set in such a way that *"Reset"* (S3) is required in order to reset output parameter *"Enable"* when:

- the PSS is cold started (PSS switched from off to on),
- warm started (PSS transferring from STOP to RUN) or
- when the safety gate switches are released.

Although the safety gate functions are configured to reset themselves, a PSS cold start or the closing of the safety gate may not directly enable a machine to start up without further conditions being met.

Input circuit safety assessment

- A short between the input circuits within a multicore cable will be detected as an error by the PSENslock.
- A short between 24 VDC and an input circuit will be detected as an error by the PSENslock.
- If the shutdown occurs via the inputs of the PSENslock, reactivation of the outputs is only possible after both safety inputs have been locked simultaneously (partial operation lock).
- If an operator completely (or even maybe partly) is able to access the dangerous area, a risk analysis should clarify whether an additional, separate "manual reset function" is required.

Caution:
The magnetic guard locking of the PSENslock is only for process protection, but not as a safety guard locking (personal protection).

3.1.2. Feedback loop monitoring function

The control system monitors the feedback circuits (NC contacts) of the motor contactors KM1 and KM2 via the user program.
An instance of the Pilz function block *"FS_OutputFBL"* is assigned to them.
The FS_FB drives the contactors as well as monitoring the feedback loop.
A 1-signal at input parameter *"Input"* of the FS-FB sets the outputs that drive the contactors, *"Output1"* and *"Output2"*, to "1"; a 0-signal sets it to "0".
If an error occurs, the outputs *"Output1"* and *"Output2"* that drive the contactors on FS-FB will immediately be reset. Both outputs are also reset when the PSS is stopped and when the PSS is switched on.

Based on the diagnostic outputs (*"DiagFeedbackLoopError"*, *"DiagFeedbackLoopNotValid"*) it can be determined why the outputs were reset.

A "valid bit" is formed by the system for the respective hardware input to determine whether a process value received from a sensor is valid.
The valid bit is queried in the Function block and indicates whether an error has occurred in the signal transmission between hardware input and processor (such as test clock error, module overheats, etc.).
If the valid bit is FALSE, the process value is invalid and the Pilz function block provides an appropriate diagnostic message. The error signal reset enable.
(For more information, see "Validity process data" in PAS4000 online help)

If an error occurs, a new activity has to take place at the input *"Input"* of the FS-FB once the error has been rectified, so that the outputs *"Output1"* and *"Output2"* will be set again.

Feedback loop monitoring safety assessment

- A short between 24 VDC and a safety output or a feedback loop input will be detected as an error by the programmable safety system. The load can be switched off via the second shutdown route.
- The feedback loop contacts must be installed in a single mounting area (control cabinet).
- To achieve a higher level of safety, 2 actuators must be used.

3.2. Functional safety

3.2.1. Safety-related characteristics in accordance with EN ISO 13849-1

No.	Safety function	PL	Safety-related parts of the control system
1	Machine shut down when a safety gate is opened	PL e	Sensor (PSEN sl-1.0p 2.1 S1) Input (PSSu E F 4DI) Logic (PSSu H PLC1 FS SN) Output (PSSu E F DI OZ 2) Actuator (contactors KM1, KM2)

Prerequisites

No.	Description		Identification
1	Common cause failure (CCF)		Requirements are considered to be met (must be tested on implementation)
2	Mission time		20 years
3	Operating interval (electromechanical components)	Sensor	two operations per hour
		Actuator	two operations per hour
4	Characteristic data of contactors KM1/KM2	B10d	2,000,000

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

3.2.2. Safety-related characteristics in accordance with EN 62061

No.	Safety-related control function (SFCF)	Safety Integrity Level	Subsystems
1	Machine shut down when a safety gate is opened	SIL 3	Sensor (PSEN sl-1.0p 2.1 S1) Input (PSSu E F 4DI) Logic (PSSu H PLC1 FS SN) Output (PSSu E F DI OZ 2) Actuator (contactors KM1, KM2)

Prerequisites

No.	Description		Identification
1	Common cause failure (CCF)		ß = 2% (must be tested on implementation)
2	Proof test interval		20 years
3	Operating interval (electromechanical components)	Sensor	two operations per hour
		Actuator	two operations per hour
4	Characteristic data of contactors KM1/KM2	B10d	2,000,000
		Dangerous failure rate	65%

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

3.3. PAS-Project

To operate a plant with one or more programmable control systems PSS 4000, a project must be created in PAS4000.
A project consists of the hardware configuration and the user program.

3.3.1. IEC 61131 Programming

When programming in accordance with IEC 61131, the user program is structured by three types of program organisation units (POUs):

- Programs (PRG)
- Function blocks (FB)
- Functions (FUN)

The program forms the higher structural level. Functions and function blocks may be called up within a program.
Function blocks and functions undertake specific individual tasks within the program.

Each POU consists of a declaration part and an instruction part.

The variables and type declarations are made in the declaration part, which is shown in text format, irrespective of the programming language.

The instruction part contains the instructions. The instructions can be formulated in one of the IEC 61131 programming languages.

(For more information about programming with PAS4000, look at PAS4000 online help)

3.3.1.1. STL-Editor

The program for the cyclic process is created in a POU of the type "Program".
The Pilz function blocks for safety gate and feedback loop were added from the library. (Right-click in the declaration part of the POU "Add library element")

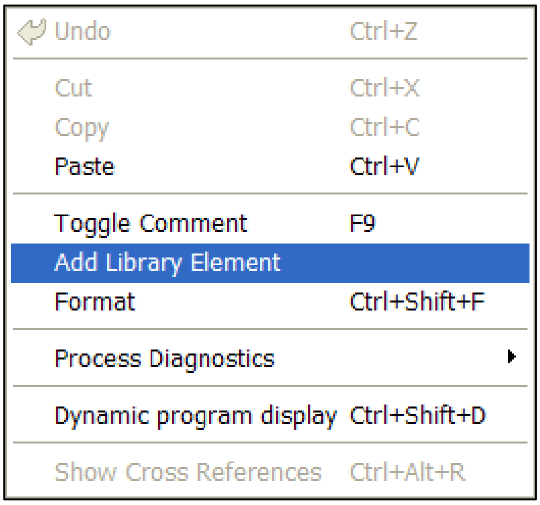


Fig. 2: Add library element

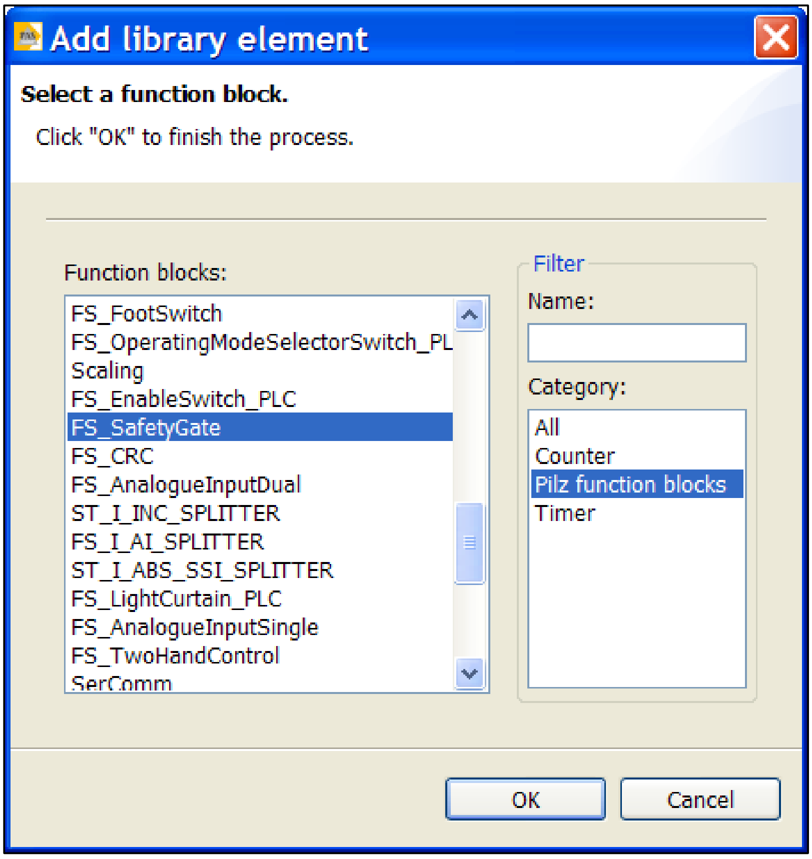


Fig. 3: Selection library element

Failsafe program

Declaration part

```
PROGRAM Main_FS
VAR
        // My SGATE 1
        MySafetyGate1                                : FS_SafetyGate;
        MySGate1_Enable                    AT %Q* : SAFEBOOL;
        MySGate1_DiagOperated              AT %Q* : SAFEBOOL;
        MySGate1_DiagReadyForReset         AT %Q* : SAFEBOOL;
        MySGate1_DiagReadyForTest          AT %Q* : SAFEBOOL;
        MySGate1_DiagSwitchError           AT %Q* : SAFEBOOL;
        MySGate1_DiagInputNotValid         AT %Q* : SAFEBOOL;

        // My FBL
        MyFeedBackLoop1                              : FS_OutputFBL;
        MyFBL1_DiagFblError                AT %Q* : SAFEBOOL;
        MyFBL1_DiagFblNotValid             AT %Q* : SAFEBOOL;
END_VAR

VAR CONSTANT
        // Declaration SwitchType 3 (NCNC)
        MySGATE1_DOUBLE_CH                           : USINT := USINT#3;
END_VAR
```

Instruction part

```
// Safety-Block Emergency-Stop1
MySafetyGate1(
        SwitchType := MySGATE1_DOUBLE_CH,
        AutoStart := FALSE,
        AutoReset := FALSE,
        MonitoredReset := TRUE,
        StartupTest := FALSE,
        SimultaneityTime := T#100ms,
        DelayTime := T#40ms,
        Enable => MySGate1_Enable,
        DiagOperated => MySGate1_DiagOperated,
        DiagReadyForReset => MySGate1_DiagReadyForReset,
        DiagReadyForTest => MySGate1_DiagReadyForTest,
        DiagSwitchError => MySGate1_DiagSwitchError,
        DiagInputNotValid => MySGate1_DiagInputNotValid);

// Safety-Block monitoring FBL1 and release motor1
MyFeedBackLoop1(
        Input := MySGate1_Enable,
        FeedbackLoopTime := T#200ms,
        DiagFeedbackLoopError => MyFBL1_DiagFblError,
        DiagFeedbackLoopNotValid => MyFBL1_DiagFblNotValid);

END_PROGRAM
```

Standard program

The signals from the start- and stop switch and those the control switch for the magnetic guard locking are imported from standard modules PSSu E S 4DI (1A4, 1A5). These signals belong to the standard control functions of the machine.
The program code for the evaluation and processing of these signals is not processed within the FS resource (safety-related part) of the control, but in a ST resource in a separate task as independent application (additional POU of type program).

Declaration part

```
PROGRAM Main_ST
VAR
        // Start / Stop
        MyRisingEdge1                          : R_TRIG;
        Start                       AT %I* : BOOL;
        Stop                        AT %I* : BOOL;
        Motor1_On                   AT %Q* : BOOL;
        Other_Stop_Conditions       AT %I* : BOOL;
        FF_Motor1On                            : RS;

        // Lock SGate1
        MyRisingEdge2                          : R_TRIG;
        MySGate1_Lock               AT %I* : BOOL;
        MySGate1_Unlock             AT %I* : BOOL;
        MySGate1_SignalOutputLock AT %I* : BOOL;
        MySGate1_LockUnlock         AT %Q* : BOOL;
        MySGate1_Lock_FLR                      : BOOL;
        MySGate1_Request_Unlock                : BOOL;
END_VAR
```

Instruction part

```
//--- Motor --------------------------------------

// Rising Edge 1 - Monitoring Start
MyRisingEdge1(clk := Start);

// Flip-Flop Motor1 On
FF_Motor1On(set := MyRisingEdge1.q (* AND Other_Start_Conditions {e.g.
                                    Operating Mode} *),
            reset1 := NOT Stop OR NOT Other_Stop_Conditions (* e.g.
                                           Enable Signal SGate1 *),
            q1 => Motor1_On);

//--- Lock ---------------------------------------

// Rising Edge 2 - Monitoring Lock
MyRisingEdge2(
       clk := MySGate1_Lock,
       q => MySGate1_Lock_FLR);

// Request "Unlock" In Process
IF MySGate1_Unlock = FALSE THEN MySGate1_Request_Unlock := TRUE;
END_IF;

// Open/Close Magnetic Guard Locking Device
IF MySGate1_Lock_FLR = TRUE AND MySGate1_SignalOutputLock = TRUE
       THEN // Close Magnetic Guard Locking Device "Lock"
              MySGate1_LockUnlock := TRUE;
ELSIF MySGate1_Request_Unlock = TRUE (* AND Other_Lock_Conditions = TRUE
                                      // e.g. Machine Standstill *)
       THEN // Open Magnetic Guard Locking Device "Unlock"
              MySGate1_LockUnlock := FALSE;
              MySGate1_Request_Unlock := FALSE;
END_IF;

END_PROGRAM
```

3.3.2. I/O Mapping

In PAS4000, variables can be created and the user program can be programmed without the need of the mapping to the hardware being present at the beginning of the project.
After identification of the used I/O from the variable declaration, the required hardware can be determined.
The I/O mapping editor forms the connecting between the user program and the hardware and coordinates the available I/O and existing PI-variables.

3.3.2.1. I/O Mapping Editor

The PI variables declared in the user-program can be assigned in the I/O mapping editor to the hardware configuration.

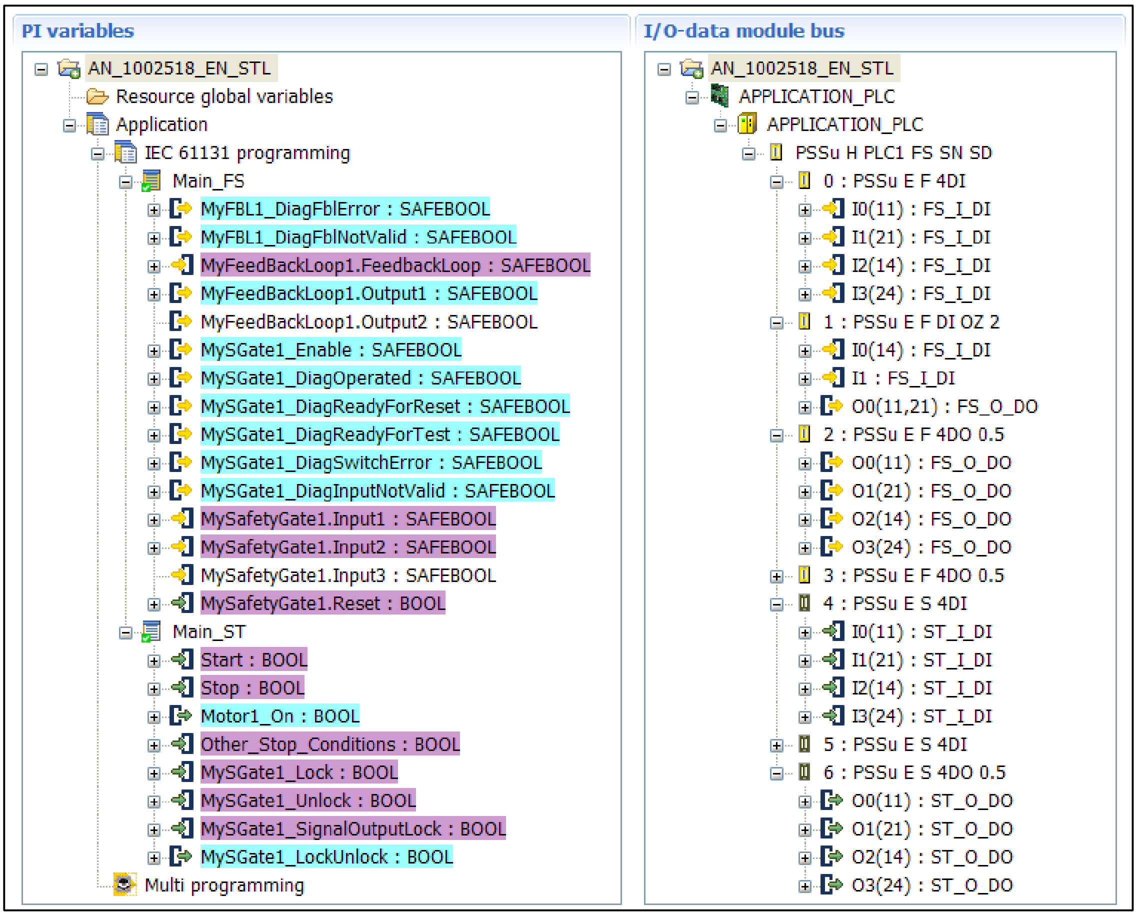


Fig. 4: Mapping Editor – STL-program

3.3.3. Process PAS Project

- Step 1: In the STL editor, PI variables and the logical sequence will be generated as a program.

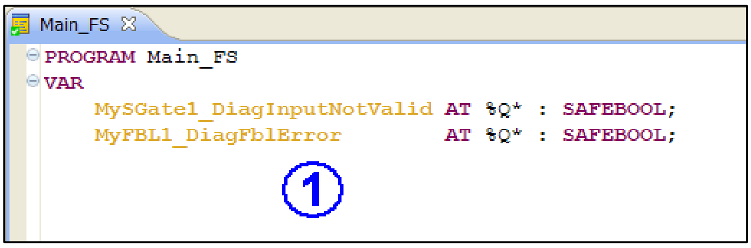


Fig. 5: PI variables

- Step 2: The design of the hardware (control, I/O, sensor, actor) will created as a circuit diagram. (parallel possible to Step 1)

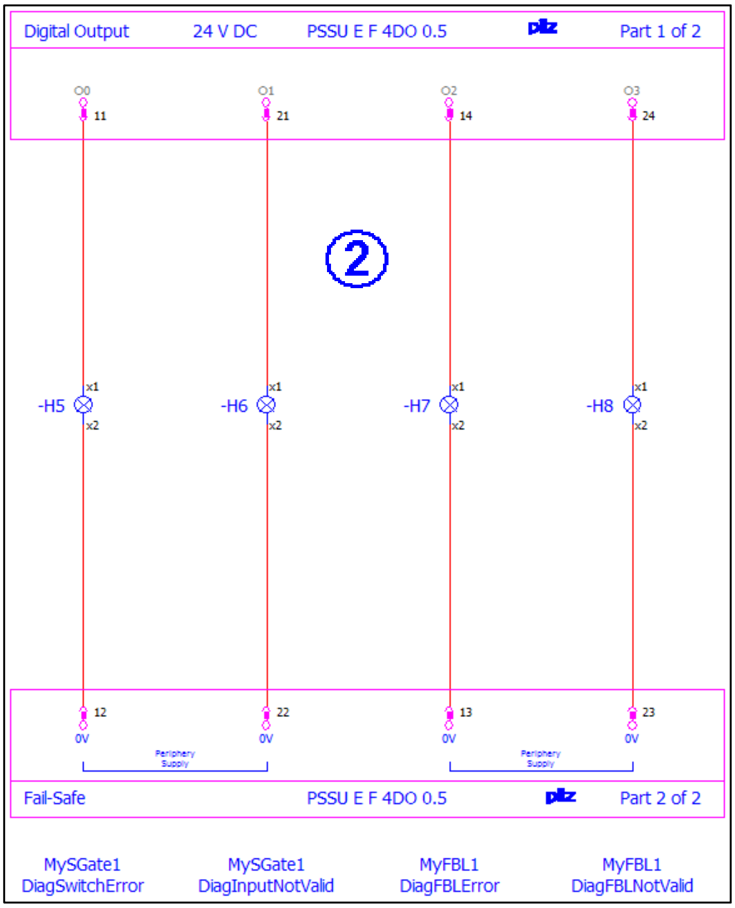


Fig. 6: Circuit diagram (extract)

- Step 3: Based on the PI variables (I/O), the required power of control (PLC, Multi) is selected. The implementation of the I/O modules in the PAS system occurs in the PSSu module editor.

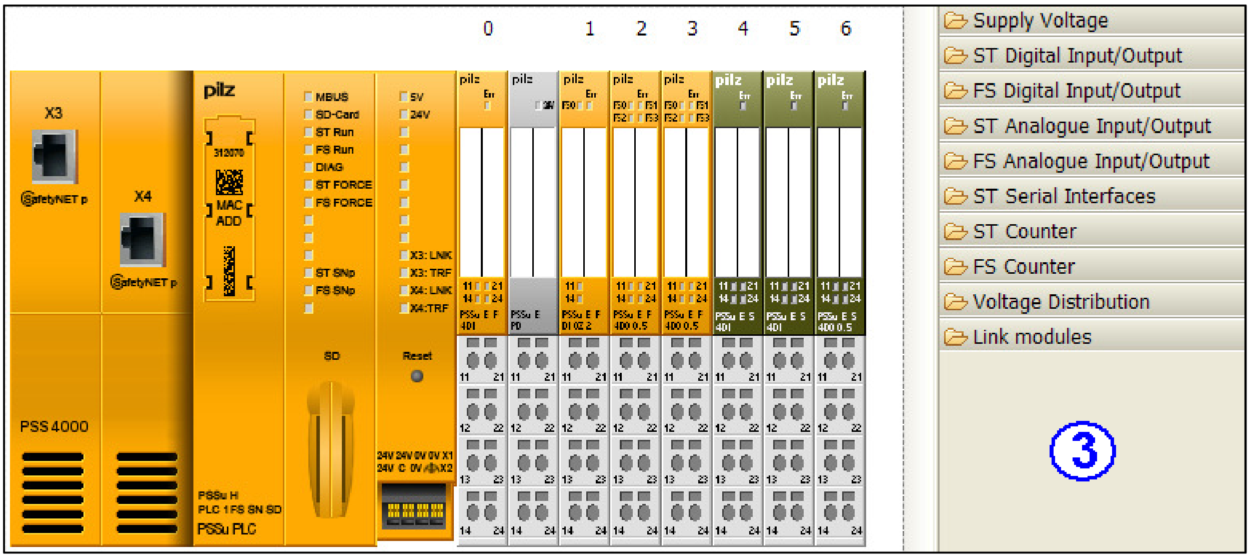


Fig. 7: PSSu Module Editor

- Step 4: Assignment of the PI variables in the I/O Mapping Editor.

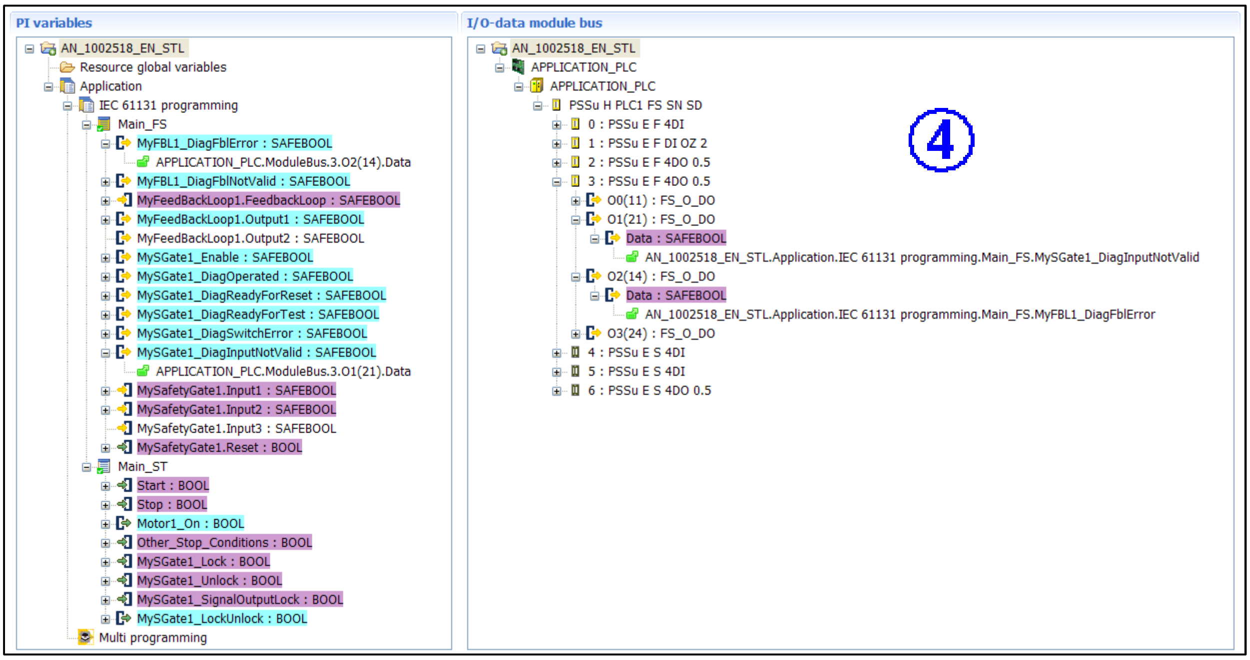


Fig. 8: I/O Mapping Editor

▸ Overview process PAS Project (Steps 1-4)

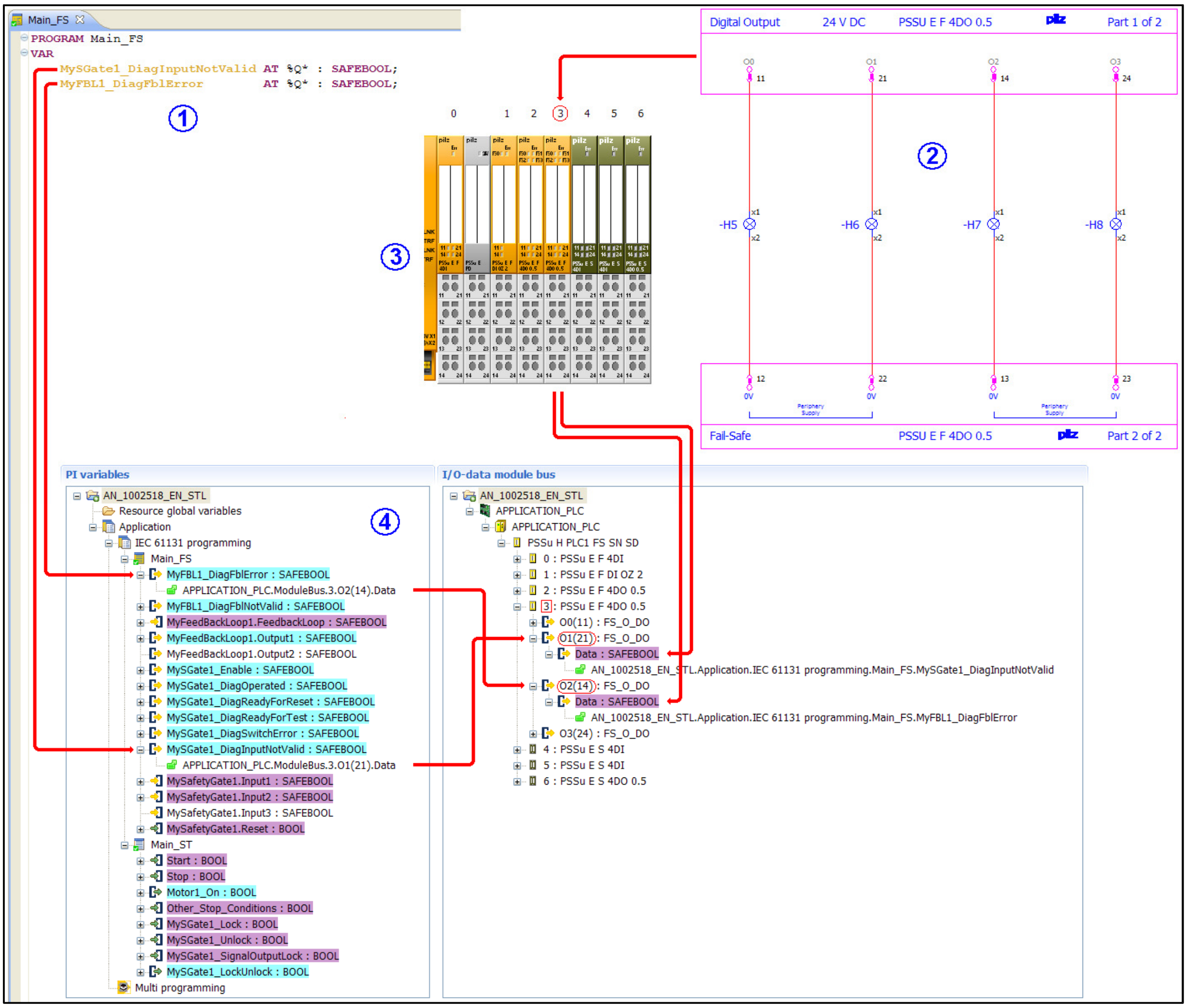
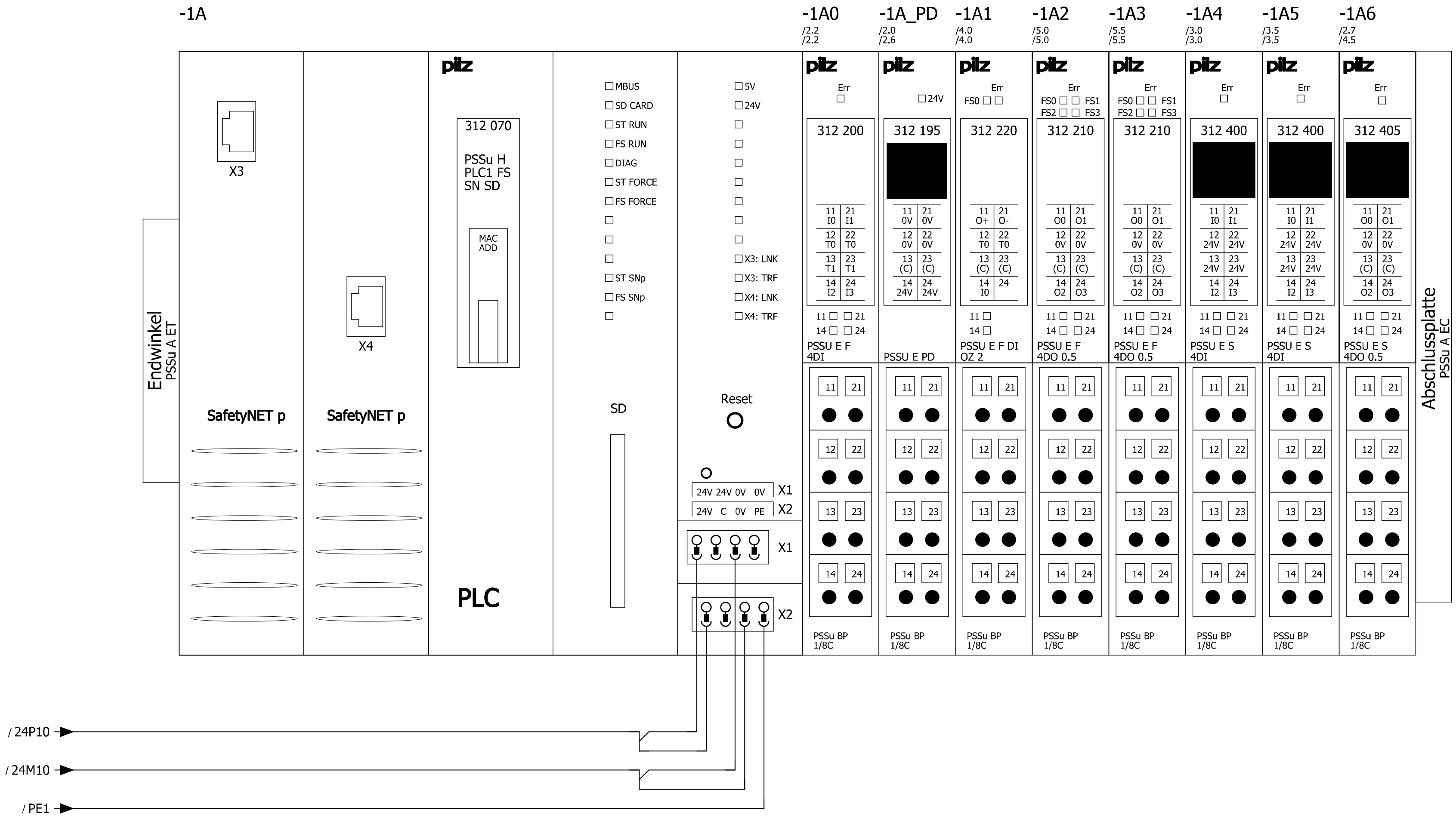


Fig. 9: Process PAS Project

0	1	2	3	4	5	6	7	8	9



PSSu E PD
312 195

PSSu E F 4DO 0.5
312 210

PSSu E S 4DI 0.5
312 400

PSSu E S 4DO 0.5
312 405

PSSu E F 4DI
312 200

PSSu E F DI OZ 2
312 220

PSSu E F 4DO 0.5
312 210

PSSu E S 4DI 0.5
312 400

PSSu BP 1/8C
312 601

PSSu BP 1/8C
312 601

PSSu BP 1/8C
312 601

PSSu BP 1/8C
312 601

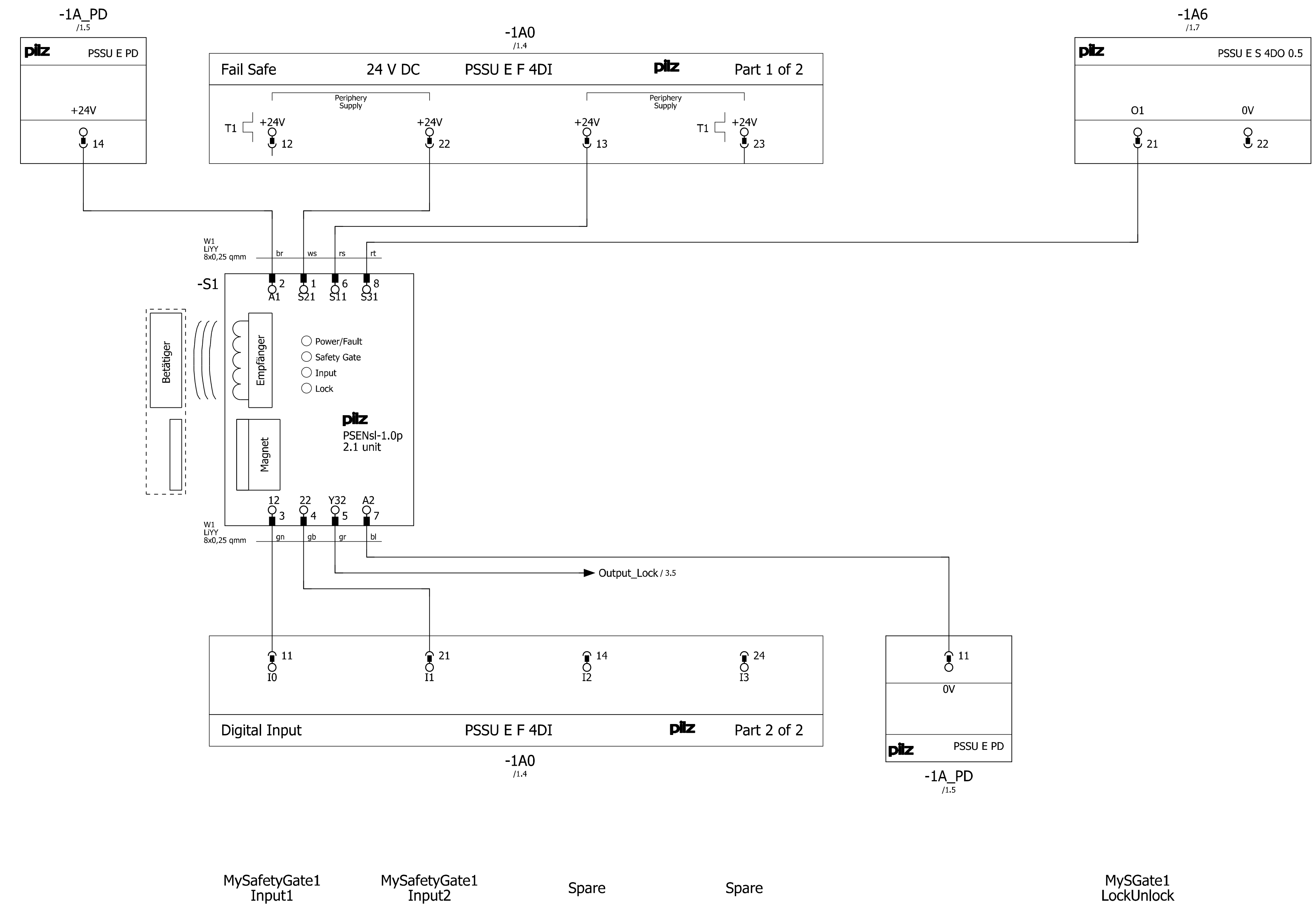
PSSu BP 1/8C
312 601

PSSu BP 1/8C
312 601

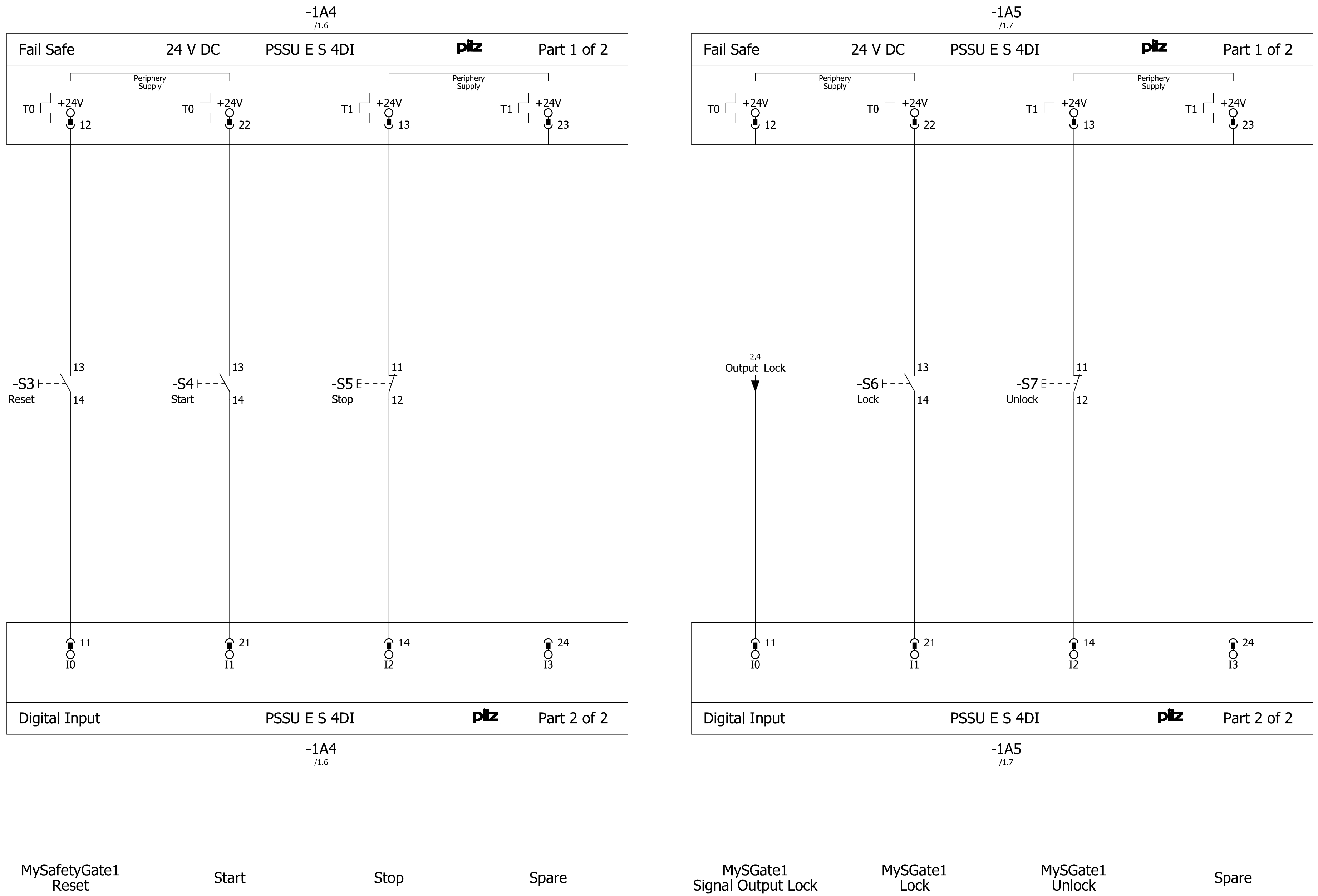
PSSu BP 1/8C
312 601

PSSu BP 1/8C
312 601

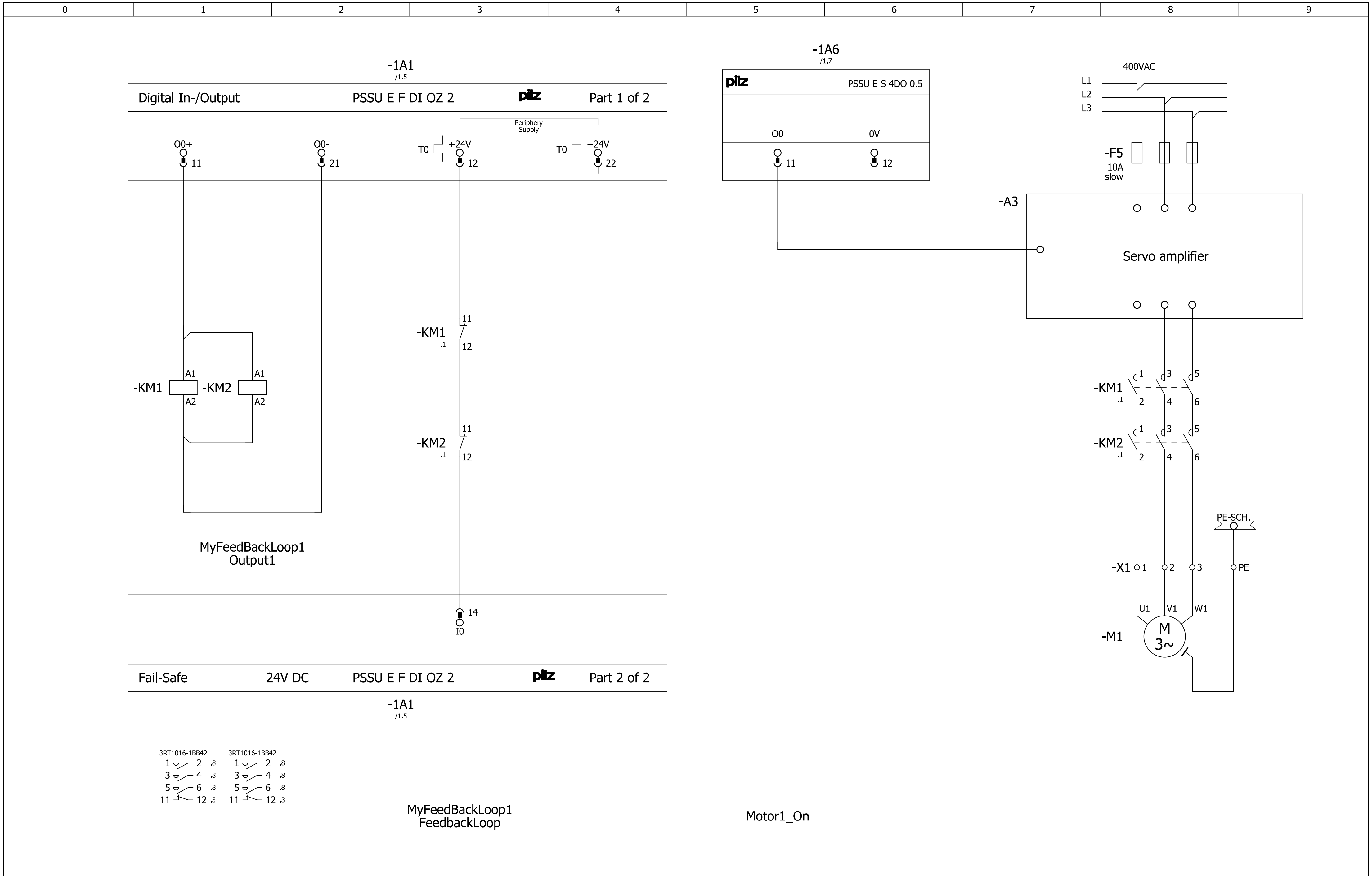
Revision	08.03.2012	Date	19.01.2005	EN ISO 13849-1:2006 PL e	Pilz GmbH & Co.KG, Felix-Wankel-Str.2, 73760 Ostfildern	PSS 4000 - Safety Gate with PSENslock / Power supply PSS 4000
Name	RDS	Name	RDS			Mounting place: + AN_1002026_02
				EN 62061:2005 SIL 3		
		Dep.	CS			Page: 1 / 5



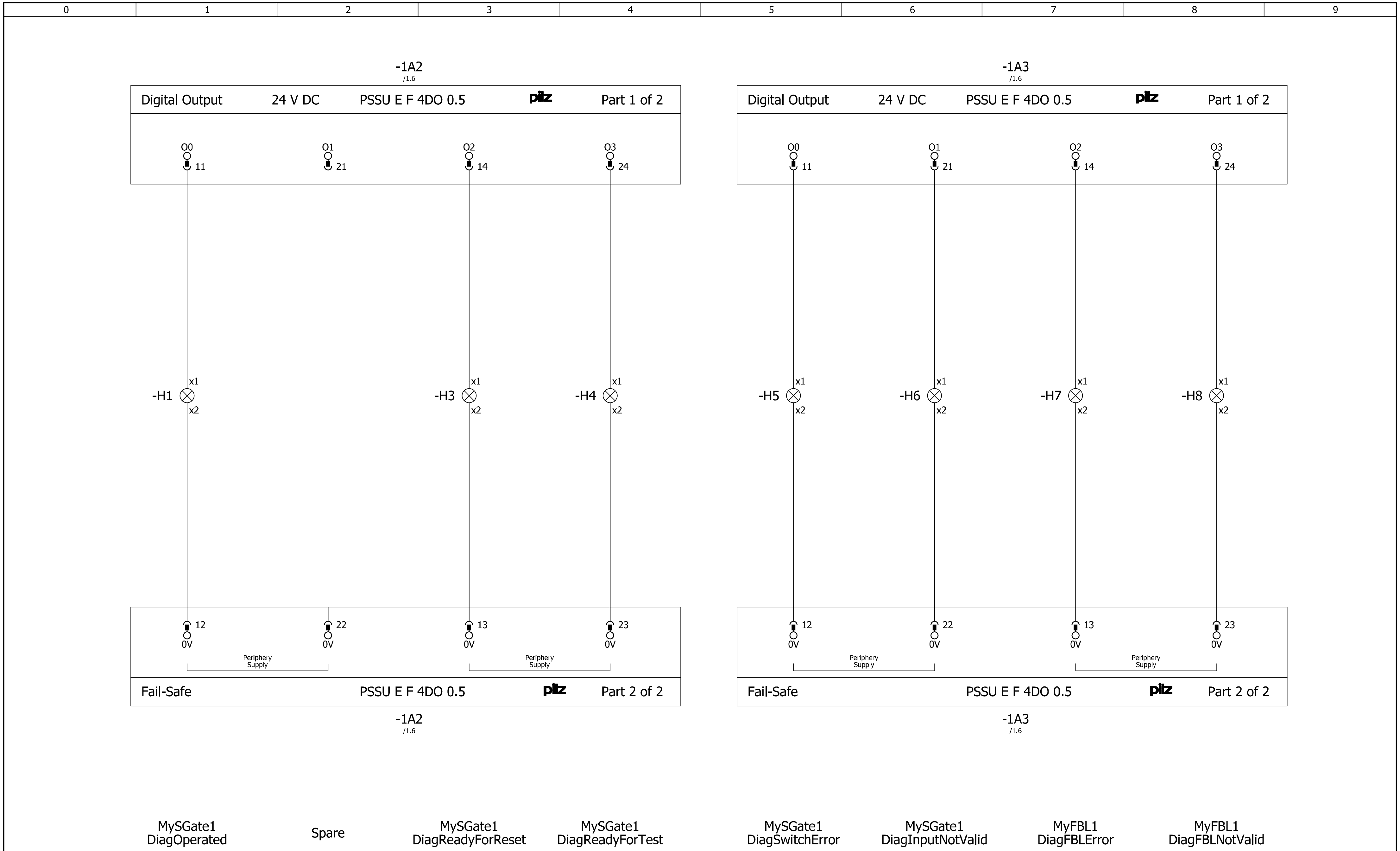
-1A_PD
/1.5
pilz
PSSU E PD
+24V
14
-1A0
/1.4
Fail Safe
24 V DC
PSSU E F 4DI
pilz
Part 1 of 2
Periphery Supply
Periphery Supply
T1
+24V
12
+24V
22
+24V
13
T1
+24V
23
-1A6
/1.7
pilz
PSSU E S 4DO 0.5
O1
0V
21
22
W1
LiYY
8x0,25 qmm
br
ws
rs
rt
-S1
2
1
6
8
A1
S21
S11
S31
Betätiger
Empfänger
Power/Fault
Safety Gate
Input
Lock
pilz
PSENsl-1.0p
2.1 unit
Magnet
12
22
Y32
A2
3
4
5
7
W1
LiYY
8x0,25 qmm
gn
gb
gr
bl
Output_Lock / 3.5
11
21
14
24
I0
I1
I2
I3
Digital Input
PSSU E F 4DI
pilz
Part 2 of 2
-1A0
/1.4
11
0V
pilz
PSSU E PD
-1A_PD
/1.5
MySafetyGate1
Input1
MySafetyGate1
Input2
Spare
Spare
MySGate1
LockUnlock



Revision	08.03.2012	Date	19.01.2005	EN ISO 13849-1:2006 PL e	Pilz GmbH & Co.KG Felix-Wankel-Str.2 73760 Ostfildern	PSS 4000 - Safety Gate with PSENslock Inputs 2	Mounting place + AN_1002026_02
Name	RDS	Name	RDS	EN 62061:2005 SIL 3			Page: 3 / 5
		Dep.	CS				



Revision	08.03.2012	Date	19.01.2005	EN ISO 13849-1:2006 PL e	pilz	PSS 4000 - Safety Gate with PSENslock Drive	Mounting place + AN_1002026_02
Name	RDS	Name	RDS		Pilz GmbH & Co.KG Felix-Wankel-Str.2 73760 Ostfildern		
		Dep.	CS	EN 62061:2005 SIL 3			Page: 4 / 5

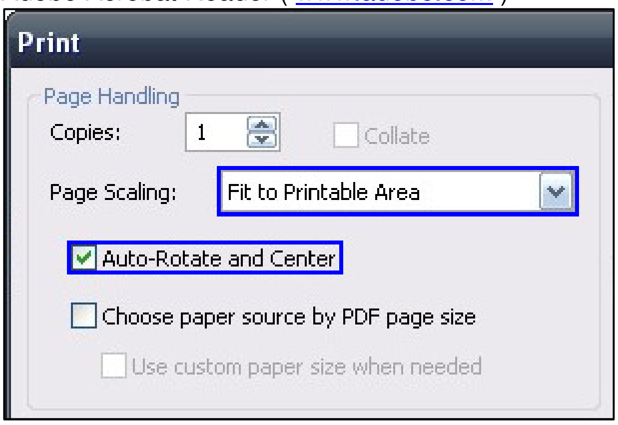


Revision	08.03.2012	Date	19.01.2005	EN ISO 13849-1:2006 PL e	Pilz GmbH & Co.KG Felix-Wankel-Str.2 73760 Ostfildern	PSS 4000 - Safety Gate with PSENslock Status/Error message	Mounting place + AN_1002026_02
Name	RDS	Name	RDS				
				EN 62061:2005 SIL 3			Page: 5 / 5
		Dep.	CS				

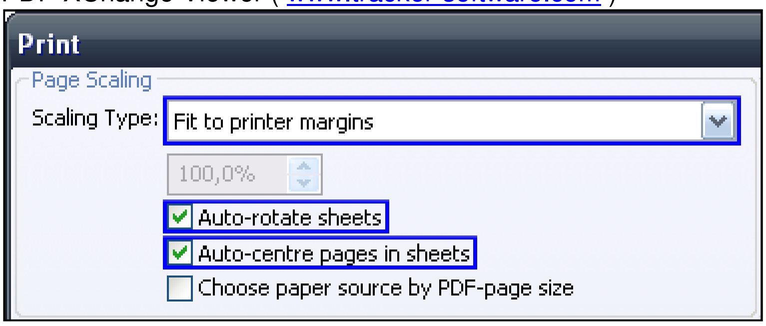
4. Table of figures

Recommended printer settings

Adobe Acrobat Reader (www.adobe.com)



PDF-XChange Viewer (www.tracker-software.com)



...

In many countries we are represented by our subsidiaries and sales partners.

Please refer to our homepage for further details or contact our headquarters.

Technical support

+49 711 3409-444
support@pilz.com

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