EUCHNER

## Application


# Integration of MGB2 *Modular* in Beckhoff TwinCAT 3

EN

## From V1.5.8

# Contents

EN

# 1. About this document

## 1.1. Version

| Version | Date | Change/addition | Chapter |
|---|---|---|---|
| 01-02/21 | 9/16/2019 | Prepared | All |
| | | | |
| | | | |

## 1.2. Scope

This document is used for integration and configuration of MGB2 *Modular* using BECKHOFF TwinCAT 3.

## 1.3. Target group

Design engineers and installation planners for safety systems on machines, as well as setup and servicing staff possessing special expertise in handling safety components as well as expertise in the installation, setup, programming and diagnostics of programmable logic controllers (PLCs) and bus systems.

## 1.4. Supplementary documents

The overall documentation for this application consists of the following documents:

| Document title (document number) | Contents | |
|---|---|---|
| Operating instructions (2500235) | System and configuration manual for the modular bus module |  |
| Safety Information and Maintenance (2500232) | Information sheet with important safety information |  |
| Operating instructions for the connected modules and their submodules | Device-specific information for the related module and the installed submodules |  |
| Possibly enclosed data sheets | Item-specific information about deviations or additions |  |

## 1.5. Notice

This application is based on the MGB2 *Modular* operating instructions. Please refer to the operating instructions for technical details and other information.

# 2. Components/modules used

## 2.1. EUCHNER

| Description | Order number / item number |
|---|---|
| Safety system MGB2 *Modular* with modular bus module MBM (PROFINET interface), guard locking with guard lock monitoring | 156386 / MGB2-L1HB-PN-U-S4-D-R-156386 |
| | 156387 / MGB2-L1HB-PN-U-S4-D-L-156387 |
| | 156388 / MGB2-L1HB-PN-U-S3-D-R-156388 |
| | 156389 / MGB2-L1HB-PN-U-S3-D-L-156389 |
| | 156390 / MGB2-L2HB-PN-U-S3-D-R-156390 |
| | 156391 / MGB2-L2HB-PN-U-S3-D-L-156391 |

### 2.1.1. Items included in the MGB2 *Modular* set

| Description | Order number/item number | MGB2 *Modular* set | | | | | |
|---|---|---|---|---|---|---|---|
| | | 156386 | 156387 | 156388 | 156389 | 156390 | 156391 |
| Modular bus module MBM | 156310 / MBM-PN-S3-MLI-3B-156310 | - | - | ● | ● | ● | ● |
| | 156312 / MBM-PN-S4-MLI-3B-156312 | ● | ● | - | - | - | - |
| Locking module MGB2-L | 136776 / MGB2-L1-MLI-U-Y0000-BJ-136776 | ● | ● | ● | ● | - | - |
| | 156392 / MGB2-L2-MLI-U-Y0000-BJ-156392 | - | - | - | - | ● | ● |
| Submodule: emergency stop + two pushbuttons | 136687 / MSM-1-P-CA-BPP-A1-136687 | ● | ● | ● | ● | ● | ● |
| Submodule: three slide-in labels | 137610 / MSM-1-N-AA-QQQ-B1-137610 | ● | ● | ● | ● | ● | ● |
| Handle module | 136691 / MGB2-H-BA1A3-R-136691 | ● | - | ● | - | ● | - |
| | 156394 / MGB2-H-BA1A3-L-156394 | - | ● | - | ● | - | ● |
| Module connector MLI | 157024 / AC-MC-SB-M-A-157024 | ● | ● | ● | ● | ● | ● |
| Blanking cover MLI | 156718 / AC-MC-00-0-B-156718 | ● | ● | ● | ● | ● | ● |

| Key to symbols | | |
|---|---|---|
| | ● | Included in the MGB2 *Modular* set |
| | - | Not included in the MGB2 *Modular* set |

Tip: More information and downloads about the aforementioned EUCHNER products can be found at www.euchner.com. Simply enter the order number in the search box.

## 2.2. Others

| Description | Order number / item number |
|---|---|
| Basic BECKHOFF CPU module with PROFINET RT controller | CX9020-0110-M930 |
| BECKHOFF TwinSAFE Logic | EL6910 |
| BECKHOFF 4-channel digital output terminal, TwinSAFE, 24 V DC | EL2904 |

EN

## 2.3. Software

| Description | Version |
|---|---|
| Microsoft Visual Studio 2013 Shell (Integrated) | Version 12.0.21005.1 REL |
| Microsoft .NET Framework | Version 4.7.03062 |
| TcMeasurement | 1.0 |
| TcProjectCompare | 1.0.0.9 |
| TcTargetBrowserPackage Extension | 1.0 |
| TcXaeDebuggerLiveWatch | 1.0 |
| TcXaeHelper | 1.0 |
| TcXaeModules | 1.0 |
| TwinCAT XAE Base | 3.1.0.0 |
| TwinCAT XAE EventLogger | 1.0 |
| TwinCAT XAE PLC | 3.1.0.0 |

# 3. Functional description

The MGB2-L1HB-PN-.. is a guard locking device in accordance with EN ISO 14119 according to the closed-circuit current principle, the MGB2-L2HB-PN-.. is a guard locking device in accordance with EN ISO 14119 according to the open-circuit current principle. In this example, all safety functions are processed via the PROFIsafe protocol. The MGB2 *Modular* is connected to a CX9020-0110-M930 from BECKHOFF.

# 4. Overview of the communication data

## 4.1. Input

| PROFINET | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| 1st byte | BM.E_G | - | BM.E_SYS | - | - | BM.E_ML2 | BM.E_ML1 | BM.D_RUN |
| 2nd byte | Diagnostics byte (pluggable) | | | | | | | |

| PROFINET | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| 1st byte | LM.E_G | LM.E_ER | LM.E_SM1 | LM.E_SM0 | - | LM.I_UK | LM.I_SK | LM.D_RUN |
| 2nd byte | - | - | - | - | - | LM.I_OL | LM.I_OT | LM.I_OD |
| 3rd byte | Diagnostics byte (pluggable) | | | | | | | |

| PROFINET | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| 1st byte | - | - | SM.E_S1 | - | - | SM.I_S3 | SM.I_S2 | SM.I_S1 |
| 2nd byte | Diagnostics byte (pluggable) | | | | | | | |

| PROFIsafe | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| 1st byte | - | - | - | - | - | SM.FI_ES | LM.FI_UK | LM.FI_SK |
| 2nd byte | - | - | - | - | - | - | - | - |
| Bytes 3-6 | PROFIsafe intern genutzt (Steuerbyte, CRC, usw.)<br>Used within PROFIsafe (control byte, CRC, etc.) | | | | | | | |

## 4.2. Output

| PROFINET | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| 1st byte | BM.ACK_G | - | - | - | - | - | - | - |

| PROFINET | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| 1st byte | LM.ACK_G | LM.ACK_ER | - | - | - | - | - | LM.O_CL |

| PROFINET | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| 1st byte | - | - | SM.O_H3_B | SM.O_H2_B | SM.O_H1_B | SM.O_H3 | SM.O_H2 | SM.O_H1 |

| PROFIsafe | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| 1st byte | - | - | - | - | - | - | - | LM.FO_CL |
| 2nd byte | - | - | - | - | - | - | - | - |
| Bytes 3-6 | PROFIsafe intern genutzt (Statusbyte, CRC, usw.)<br>Used within PROFIsafe (status byte, CRC, etc.) | | | | | | | |

Tip: The individual abbreviations are explained in the operating instructions

**NOTE!**

While PROFINET data are always incorporated by bytes, the data for PROFIsafe are always incorporated by individual bits. The *PROFIsafe 2 Bytes module* was therefore used for this application example. It offers a sufficiently large safe memory area for the bits: *LM.FI_SK, LM.FI_UK* and *SM.FI_ES*.

EN

## 5. Installing the GSD file

You will require the corresponding GSD file in GSDML format to integrate the MGB2 *Modular* into the TwinCAT 3 hardware configuration:

- GSDML-V2.33-EUCHNER-MBM_2512512_T14-YYYYMMDD.xml

You will find the GSD files in the download area at www.euchner.com. Always use the latest GSD file.

Unzip the content of the GSDML file into the following directory:

- C:/TwinCAT/3.1/Config/Io/Profinet

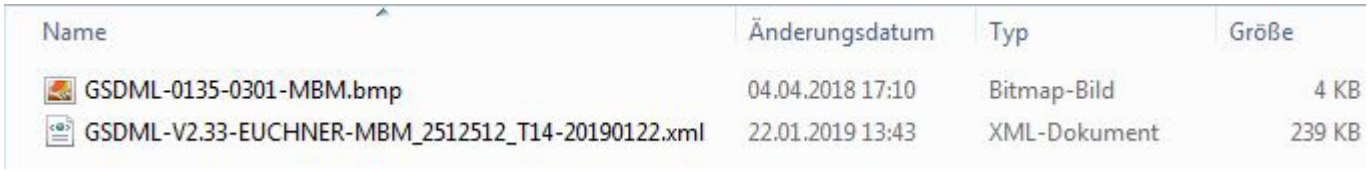



Figure 1: Content of the ZIP file


Figure 2: GSDML file path for TwinCAT 3

## 6. Setting the control system parameters

Specify the cycle time for the *PlcTask*. The value 2 must be set for a PROFINET application.

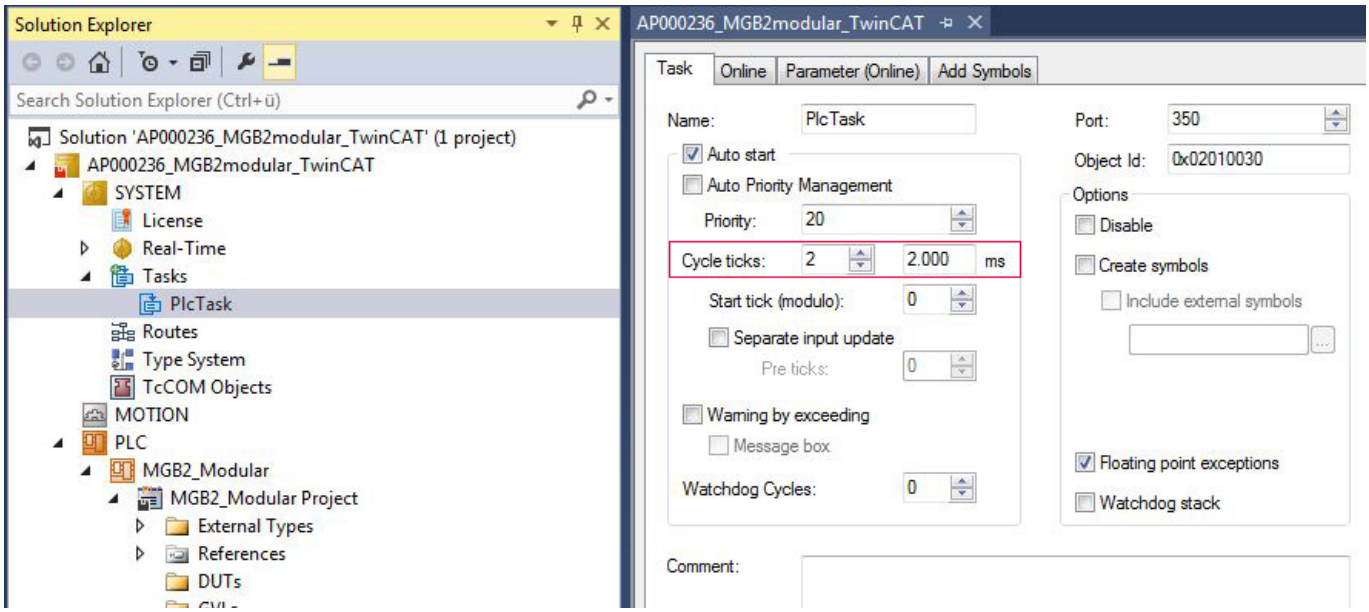



*Figure 3: PlcTask* parameters

# 7. TwinSAFE and PROFIsafe hardware addressing

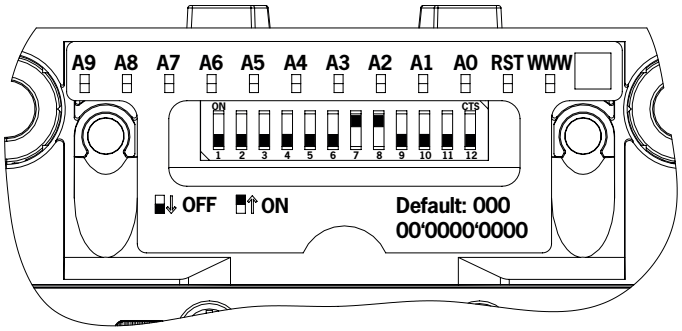
## 7.1. TwinSAFE

The TwinSAFE address must be set for the TwinSAFE logic module EL6910 and the fail-safe output module EL2904. It is set using the DIP switches on the left side of the TwinSAFE terminals.

| TwinSafe terminal | TwinSAFE address in the example |
|---|---|
| EL6910 | 1 |
| EL2904 | 2 |

## 7.2. PROFIsafe

The PROFIsafe address (F_Dest_Add) is set on the bus module MBM using the DIP switches. The PROFIsafe address must be set to the value configured.




| Switch | Description |
|---|---|
| A0 ... A9 | Address switch, bits zero to nine<br>For binary setting of the PROFIsafe address<br>(factory setting: 0000000000) |
| RST | Factory reset<br>(factory setting: off) |
| WWW | Activate device web interface with extended diagnostic options.<br>(factory setting: off) |

The DIP switch setting is as follows from the F_Dest_Add 12 as configured in the hardware configurator:

| Switch | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
|---|---|---|---|---|---|---|---|---|---|---|
| DIP switch position MBM | off | off | off | off | off | off | on | on | off | off |
| Significance | $2^9$ | $2^8$ | $2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
| Decimal value | 512 | 256 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |

Table 1: DIP switch settings

**ATTENTION!**

- The set PROFIsafe address of the MBM and the address configured in the hardware configurator must match.
- The PROFIsafe address set using the DIP switches is applied only after the MBM is restarted.

EN

# 8. Configuration of the MBM and the I/O peripherals

## 8.1. Adding the I/O devices in the project

**NOTE!**

To perform scans, TwinCAT must be in *Config Mode*.

Add the devices as follows:

1. Open *Solution Explorer*, click *I/O*, then right-click *Devices* and select *Scan* from the context-sensitive menu.
2. Select the PROFINET and EtherCAT controllers.

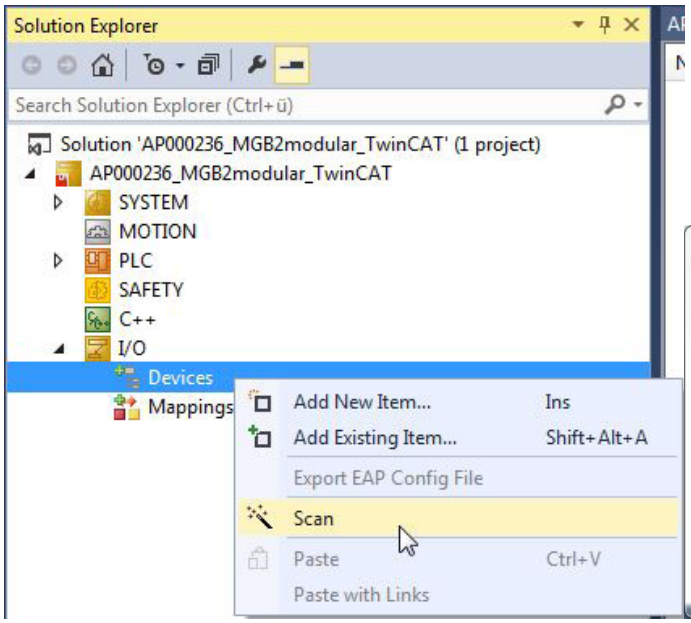



Figure 4: Solution Explorer

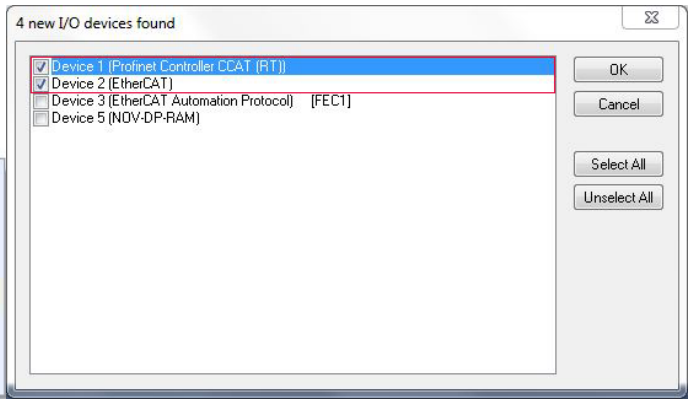



Figure 5: Selecting the controllers

3. Activate the search for PROFINET devices in the following pop-up window, *Scan for Boxes*

4. Compare the MAC address on the type label with the MAC address of the devices available in the network, and add the MBM to PROFINET with *Add Devices*.

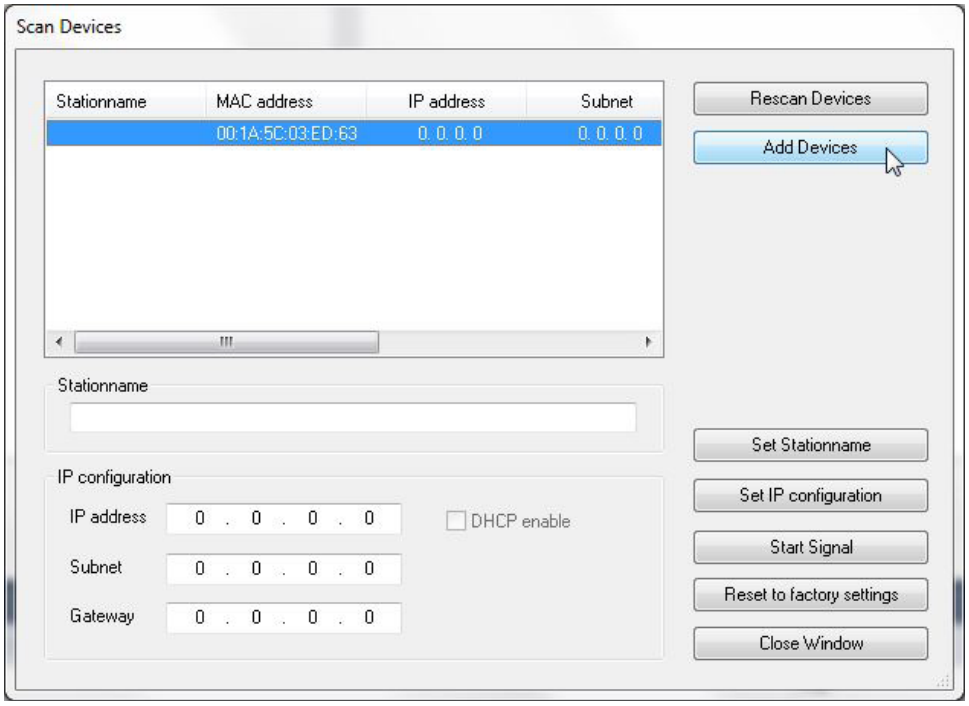



Figure 6: Adding MBM

EN

5. Then scan the real configuration. After completion of the scanning process, the hardware configuration appears as shown in Figure 7.

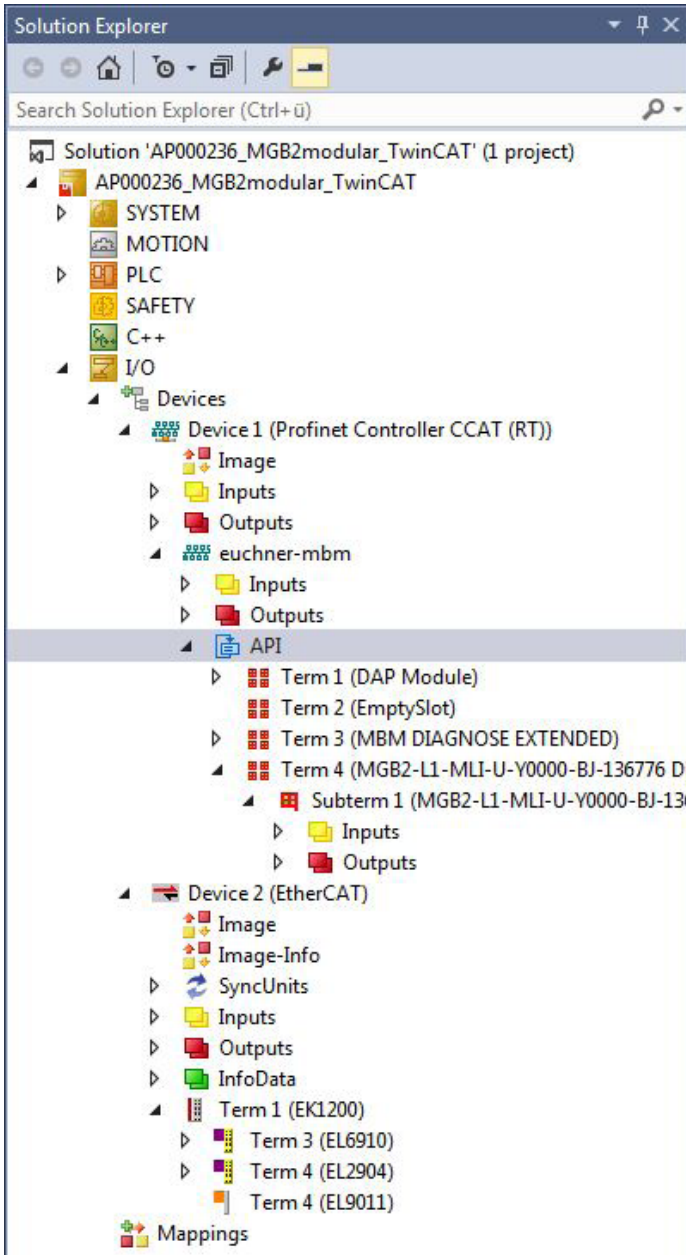



Figure 7: Structure of the modules used

6. Complete the configuration of the MGB2 *Modular* with the modules used in the example. Begin by right-clicking on *Term 2 (EmptySlot)*, and use Insert *New Item...* to insert the module *PROFIsafe 2 Bytes*, *PROFIsafe 4 Bytes* or *PROFIsafe 8 Bytes*.

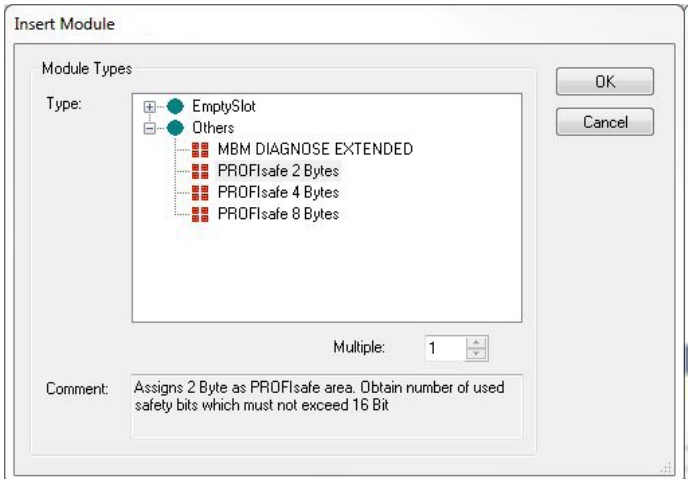



Figure 8: Adding PROFIsafe modules

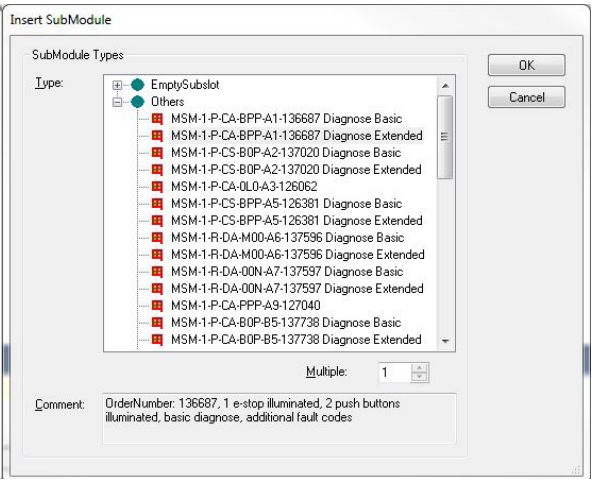



Figure 9: Adding additional submodules

7. Add modules and submodules to MGB2 *Modular* corresponding to your layout.

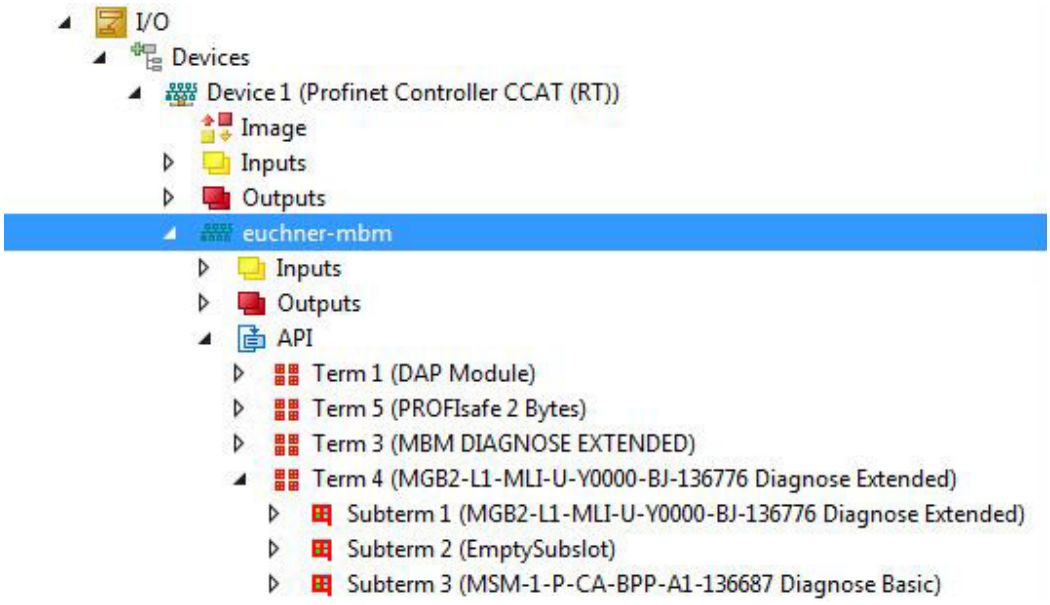



Figure 10: Hardware layout, MGB2 *Modular* set

**NOTE!**

- Only modules with extended diagnostics are configured in the application example. It is possible to configure the modules and submodules with basic diagnostics.
- You will find the list of parameters that can be set for modules and submodules in the operating instructions for the bus module.
- The *MBM DIAGNOSE EXTENDED* module in slot 2 is added automatically. It can be replaced with the *MBM DIAGNOSE BASIC* module.

EN

## 8.2. Setting the MGB2 *Modular* parameters

### 8.2.1. PROFINET

The following PROFINET parameters must be set:

- Device name/station name (factory setting from GSD file): [*euchner-mbm*].
- IP address: fixed

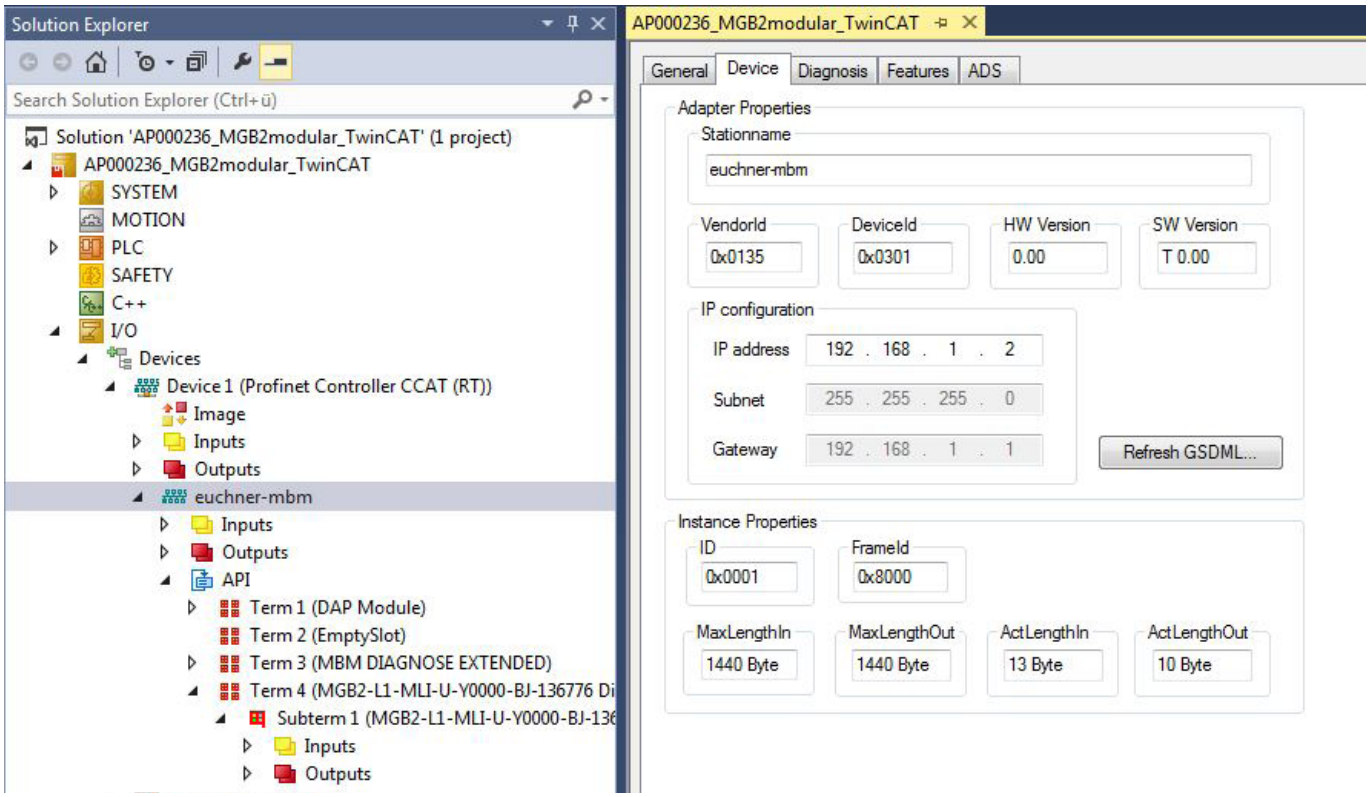



Figure 11: PROFINET parameters

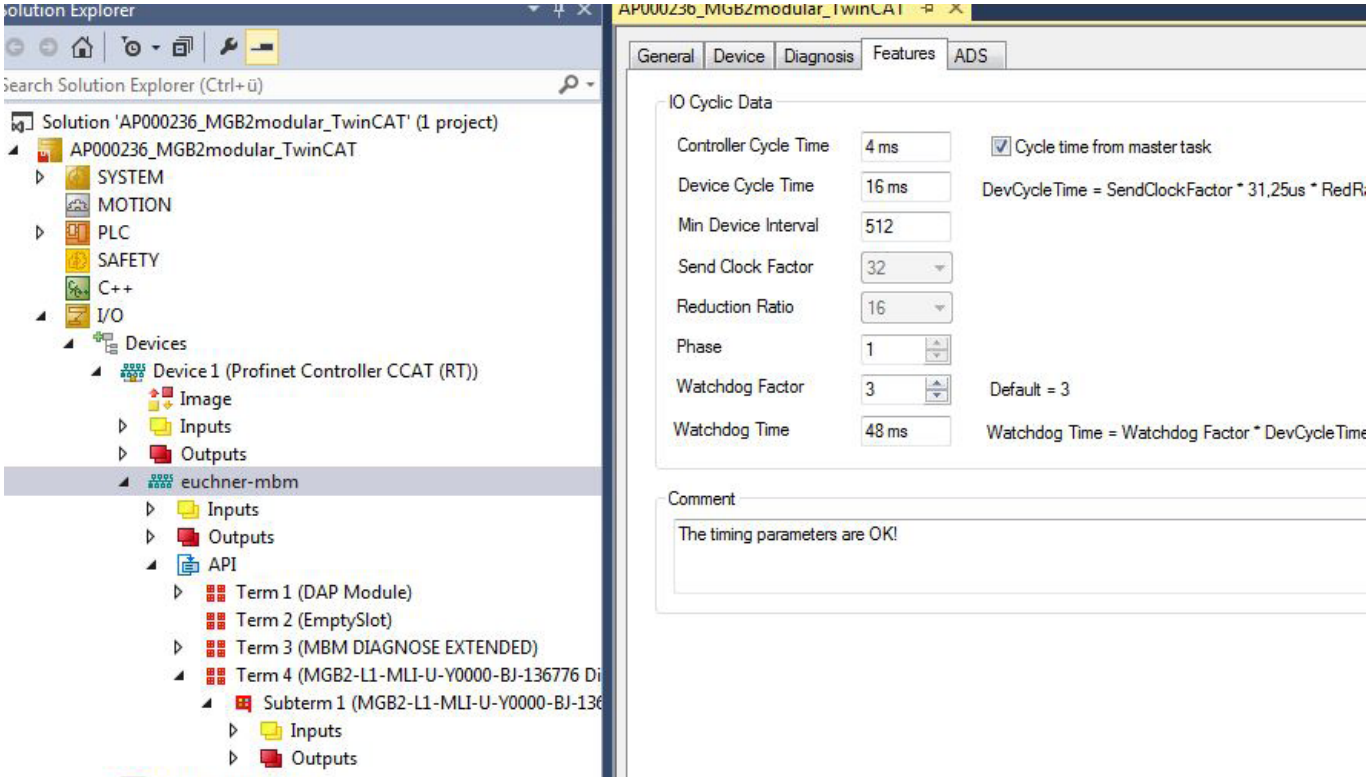



Figure 12: PROFINET real time settings

- IO cycle real time settings
  Update time: Calculate update time automatically (recommended)
  Watchdog time: accepted update cycles without IO data: 3 (recommended)

EN

### 8.2.2. PROFIsafe

The following PROFIsafe parameters must be set:

- F_Dest_Add (PROFIsafe address): 12 (TwinCAT 3 sets the default PROFIsafe address; addressing can be changed manually).
- F_WD_Time (time during which the control system expects a response from the PROFIsafe device): 600 ms. Factory setting from GSD file: [600 ms].

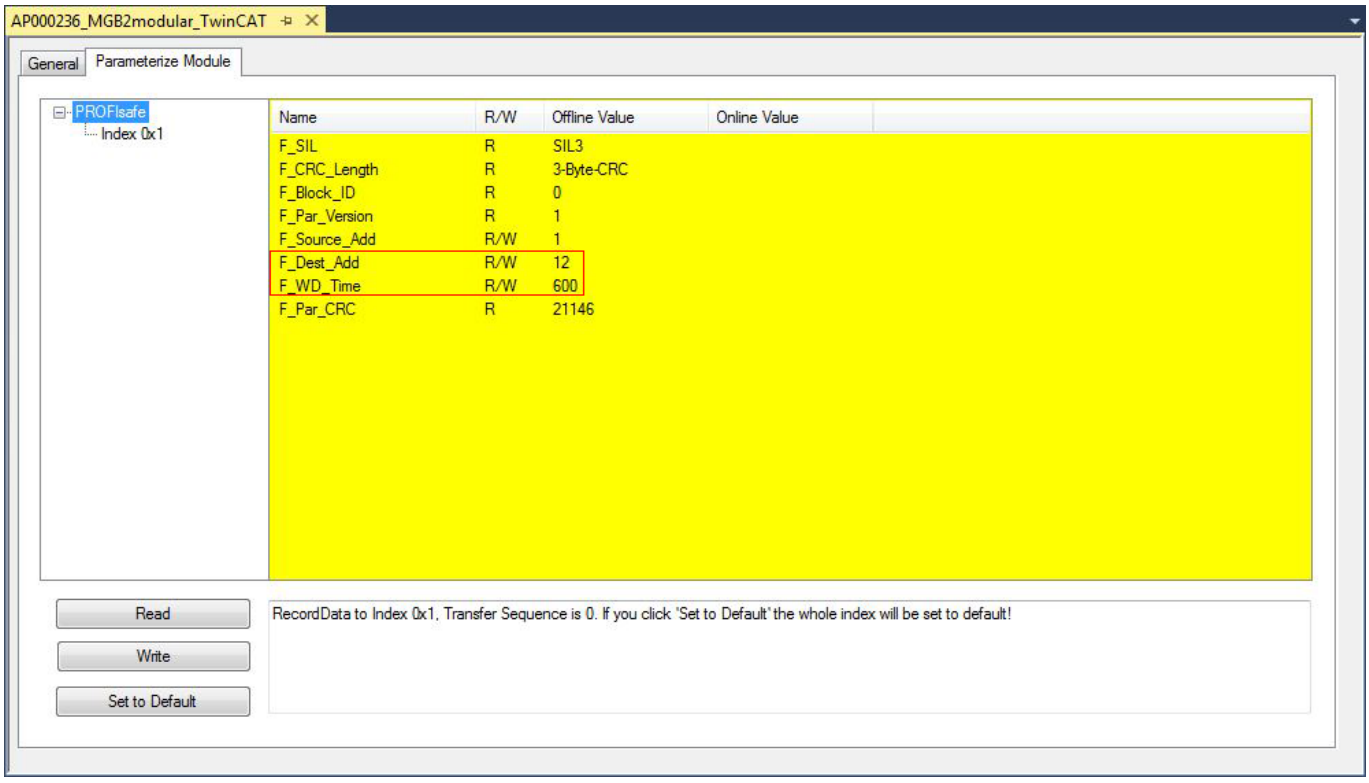



Figure 13: PROFIsafe parameters

**ATTENTION!**

- The set PROFIsafe address of the MBM and the address configured in TwinCat must match.
- The PROFIsafe address set using the DIP switches is applied only after the MBM is restarted.

## 8.3. Assigning PROFINET device name to the bus module MBM

1. To assign the name to MGB2 *Modular* via TwinCAT, right-click the PROFINET controller and then select *Scan*.

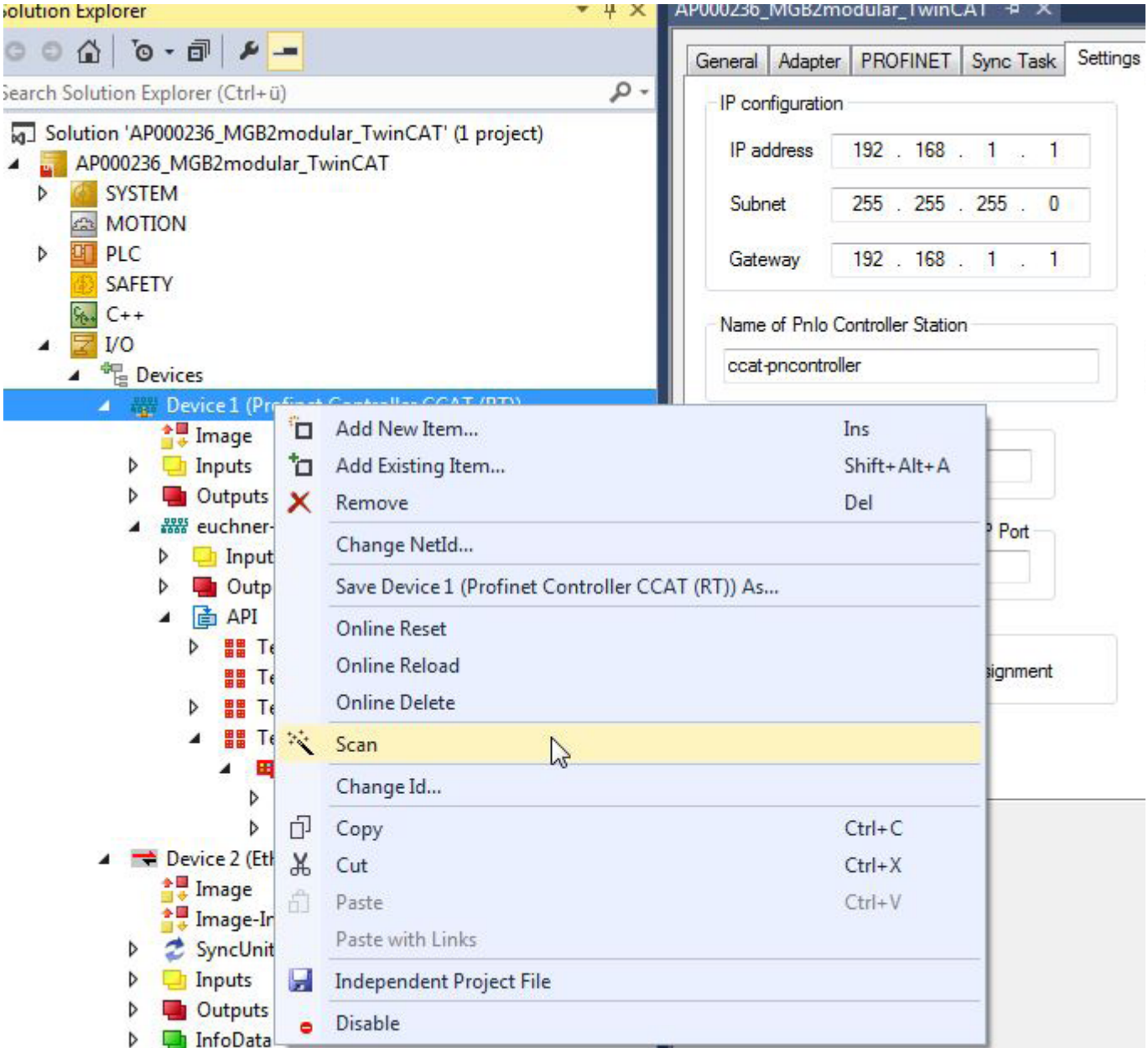



Figure 14: Searching for devices online

EN

2. Select the MBM from the list. Enter the station name and assign it using *Set Stationname*. Additionally assign the IP address with *Set IP configuration*.

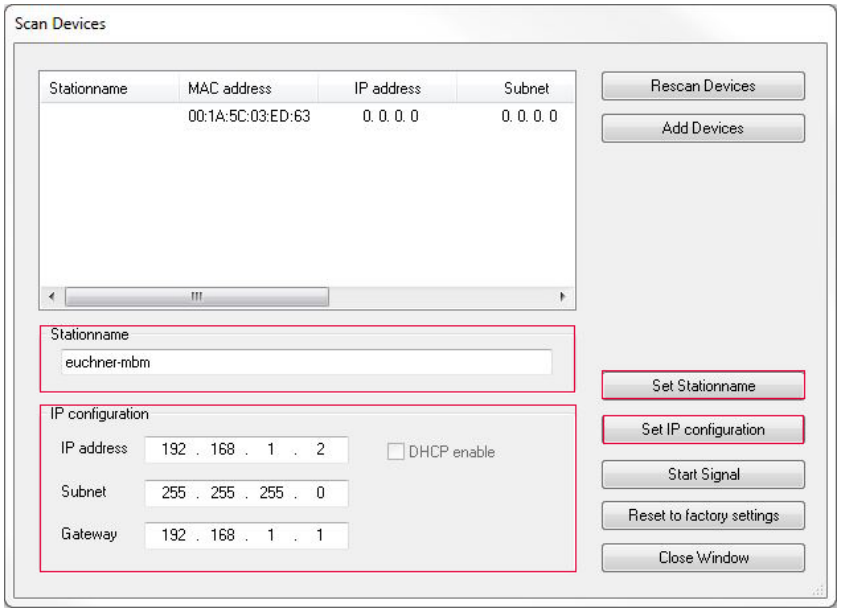
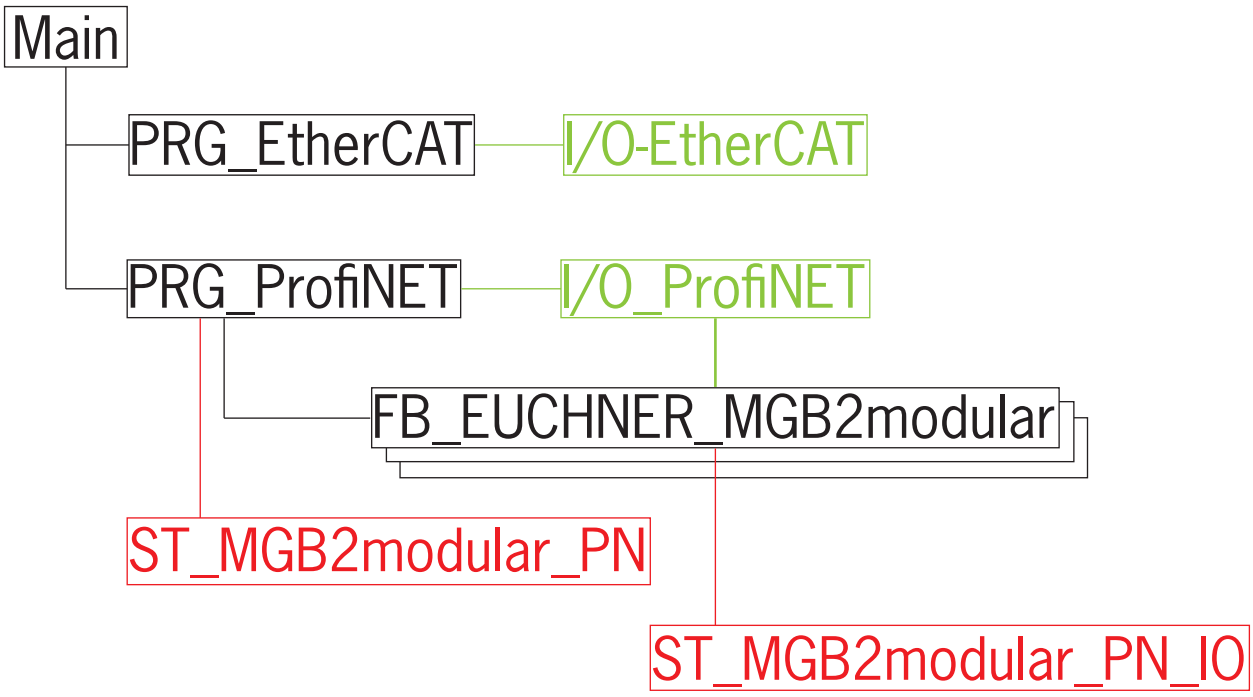



Figure 15: Assigning device names

| | |
|---|---|
| | **TIP!**<br><br>As an alternative to MAC address comparison, you can use *Start Signal* to see whether you have selected the correct device. The Link1 and Link2 LEDs on the MBM flash. |

# 9. PLC program creation

The following program structure is used for PROFINET communication (non-safe communication):




## 9.1. Structure of the connection for PROFINET I/O configuration

The input/output structure of the MGB2 *Modular* sets is mapped equivalently to the communication data in the *ST_MGB2modular_PN_IO* structure.

ST_MGB2modular_PN_IO

```
1  TYPE ST_MGB2modular_PN_IO :
2      //Struct for MGB2-Set Inputs and Outputs
3  STRUCT
4      //MGB2 System Diagnostic Inputs
5      nMGB2modularI_PnIoBoxState              AT %I* : UINT;
6      nMGB2modularI_PnIoBoxDiag               AT %I* : UINT;
7      //MBM Inputs
8      nMGB2modularI_MBM                       AT %I* : ARRAY [0..1] OF BYTE;
9      //Locking module Inputs
10     nMGB2modularI_LM                        AT %I* : ARRAY [0..2] OF BYTE;
11     //Submodule Inputs
12     nMGB2modularI_SM                        AT %I* : ARRAY [0..1] OF BYTE;
13
14     //MGB2 System Diagnostic Output
15     nMGB2modularQ_PnIoBoxCtrl               AT %Q* : UINT;
16     //MBM Outputs
17     nMGB2modularQ_MBM                       AT %Q* : ARRAY [0..0] OF BYTE;
18     //Locking module Outputs
19     nMGB2modularQ_LM                        AT %Q* : ARRAY [0..0] OF BYTE;
20     //Submodule Outputs
21     nMGB2modularQ_SM                        AT %Q* : ARRAY [0..0] OF BYTE;
22 END_STRUCT
23 END_TYPE
```

*Figure 16: ST_MGB2modular_PN_IO*

EN

## 9.2. Structure for readability of the inputs/outputs

The inputs and outputs of the MGB2 *Modular* are prepared for better readability in the *ST_MGB2modular_IO* structure. The data structure shown on the data sheet [chapter 4] is used as the template for this.

ST_MGB2modular_PN

```
TYPE ST_MGB2modular_PN :
STRUCT
    //MGB2 System Diagnostic Inputs
    nMGB2modularI_PnIoBoxState                  : UINT;
    nMGB2modularI_PnIoBoxDiag                   : UINT;

    //MBM Inputs
    //nMGB2modularI_MBM                         AT %I* : ARRAY [0..1] OF BYTE;
    xBM_D_RUN                                   : BOOL;
    xBM_E_ML1                                   : BOOL;
    xBM_E_ML2                                   : BOOL;
//  x103_BM                                     : BOOL;
//  x104_BM                                     : BOOL;
    xBM_E_SYS                                   : BOOL;
//  x105_BM                                     : BOOL;
    xBM_E_G                                     : BOOL;
    BBM_ExtendedDiagnostic                      : BYTE;

    //Locking module Inputs
    //Inputs nMGB2modularI_LM                   AT %I* : ARRAY [0..2] OF BYTE;
    xLM_D_RUN                                   : BOOL;
    xLM_I_SK                                    : BOOL;
    xLM_I_UK                                    : BOOL;
//  x203_LM                                     : BOOL;
    xLM_E_SM0                                   : BOOL;
    xLM_E_SM1                                   : BOOL;
    xLM_E_ER                                    : BOOL;
    xLM_E_G                                     : BOOL;
    xLM_I_OD                                    : BOOL;
    xLM_I_OT                                    : BOOL;
    xLM_I_OL                                    : BOOL;
    BLM_ExtendedDiagnostic                      : BYTE;

    //Submodul Inputs
    //nMGB2modular_SM                           AT %I* : ARRAY [0..1] OF BYTE;
    xSM_I_S1                                    : BOOL;
    xSM_I_S2                                    : BOOL;
    xSM_I_S3                                    : BOOL;
//  x303_SM                                     : BOOL;
//  x304_SM                                     : BOOL;
    xSM_E_S1                                    : BOOL;
//  x306_SM                                     : BOOL;
//  x307_SM                                     : BOOL;
    BSM_ExtendedDiagnostic                      : BYTE;

```

Figure 17: Structure of inputs

ST_MGB2modular_PN

```
48
49          //MBM Outputs
50          //nMGB2modularQ_MBM                    AT %Q* : ARRAY [0..0] OF BYTE;
51      //  x100_BM                                : BOOL;
52      //  x101_BM                                : BOOL;
53      //  x102_BM                                : BOOL;
54      //  x103_BM                                : BOOL;
55      //  x104_BM                                : BOOL;
56      //  x105_BM                                : BOOL;
57      //  x106_BM                                : BOOL;
58          xBM_ACK_G                              : BOOL;
59
60          //Locking module Outputs
61          //nMGB2modularQ_LM                     AT %Q* : ARRAY [0..0] OF BYTE;
62          xLM_O_CL                               : BOOL;
63      //  x201_LM                                : BOOL;
64      //  x202_LM                                : BOOL;
65      //  x203_LM                                : BOOL;
66      //  x204_LM                                : BOOL;
67      //  x205_LM                                : BOOL;
68          xLM_ACK_ER                             : BOOL;
69          xLM_ACK_G                              : BOOL;
70
71          //Submodule Outputs
72          //nMGB2modularQ_SM                     AT %Q* : ARRAY [0..0] OF BYTE;
73          xSM_O_H1                               : BOOL;
74          xSM_O_H2                               : BOOL;
75          xSM_O_H3                               : BOOL;
76          xSM_O_H1_B                             : BOOL;
77          xSM_O_H2_B                             : BOOL;
78          xSM_O_H3_B                             : BOOL;
79      //  x306_SM                                : BOOL;
80      //  x307_SM                                : BOOL;
81
82      END_STRUCT
83      END_TYPE
```

Figure 18: Structure of outputs

EN

## 9.3. Function block *FB_EUCHNER_MGB2modular*

The structure of the variables is copied to the structure of the inputs/outputs in the function block *FB_EUCHNER_MGB2modular*.

### 9.3.1. Copying the CPU input structure to the MGB2 *Modular* structure

FB_EUCHNER_MGB2modular

```
FUNCTION_BLOCK FB_EUCHNER_MGB2modular
VAR_IN_OUT
    stMGB2modularPN             :ST_MGB2modular_PN;
END_VAR
VAR
    stMGB2modularPNIO           :ST_MGB2modular_PN_IO;
END_VAR
```

```
//MGB2 System Diagnostic Inputs
stMGB2modularPN.nMGB2modularI_PnIoBoxState :=   stMGB2modularPNIO.nMGB2modularI_PnIoBoxState;
stMGB2modularPN.nMGB2modularI_PnIoBoxDiag :=    stMGB2modularPNIO.nMGB2modularI_PnIoBoxDiag;

//MBM Inputs
stMGB2modularPN.xBM_D_RUN :=                    stMGB2modularPNIO.nMGB2modularI_MBM[0].0;
stMGB2modularPN.xBM_E_ML1 :=                    stMGB2modularPNIO.nMGB2modularI_MBM[0].1;
stMGB2modularPN.xBM_E_ML2 :=                    stMGB2modularPNIO.nMGB2modularI_MBM[0].2;
stMGB2modularPN.xBM_E_SYS :=                    stMGB2modularPNIO.nMGB2modularI_MBM[0].5;
stMGB2modularPN.xBM_E_G :=                      stMGB2modularPNIO.nMGB2modularI_MBM[0].7;
stMGB2modularPN.BBM_ExtendedDiagnostic :=       stMGB2modularPNIO.nMGB2modularI_MBM[1];

//Locking module Inputs
stMGB2modularPN.xLM_D_RUN :=                    stMGB2modularPNIO.nMGB2modularI_LM[0].0;
stMGB2modularPN.xLM_I_SK :=                     stMGB2modularPNIO.nMGB2modularI_LM[0].1;
stMGB2modularPN.xLM_I_UK :=                     stMGB2modularPNIO.nMGB2modularI_LM[0].2;
stMGB2modularPN.xLM_E_SM0 :=                    stMGB2modularPNIO.nMGB2modularI_LM[0].4;
stMGB2modularPN.xLM_E_SM1 :=                    stMGB2modularPNIO.nMGB2modularI_LM[0].5;
stMGB2modularPN.xLM_E_ER :=                     stMGB2modularPNIO.nMGB2modularI_LM[0].6;
stMGB2modularPN.xLM_E_G :=                      stMGB2modularPNIO.nMGB2modularI_LM[0].7;
stMGB2modularPN.xLM_I_OD :=                     stMGB2modularPNIO.nMGB2modularI_LM[1].0;
stMGB2modularPN.xLM_I_OT :=                     stMGB2modularPNIO.nMGB2modularI_LM[1].1;
stMGB2modularPN.xLM_I_OL :=                     stMGB2modularPNIO.nMGB2modularI_LM[1].2;
stMGB2modularPN.BLM_ExtendedDiagnostic :=       stMGB2modularPNIO.nMGB2modularI_LM[2];

//Submodule Inputs
stMGB2modularPN.xSM_I_S1 :=                     stMGB2modularPNIO.nMGB2modularI_SM[0].0;
stMGB2modularPN.xSM_I_S2 :=                     stMGB2modularPNIO.nMGB2modularI_SM[0].1;
stMGB2modularPN.xSM_I_S3 :=                     stMGB2modularPNIO.nMGB2modularI_SM[0].2;
stMGB2modularPN.xSM_E_S1 :=                     stMGB2modularPNIO.nMGB2modularI_SM[0].5;
stMGB2modularPN.BSM_ExtendedDiagnostic :=       stMGB2modularPNIO.nMGB2modularI_SM[1];
```

Figure 19: Copying the CPU input structure

### 9.3.2. Copying the MGB2 *Modular* output structure to the CPU structure

FB_EUCHNER_MGB2modular

```
FUNCTION_BLOCK FB_EUCHNER_MGB2modular
VAR_IN_OUT
    stMGB2modularPN             :ST_MGB2modular_PN;
END_VAR
VAR
    stMGB2modularPNIO           :ST_MGB2modular_PN_IO;
END_VAR
```

```
//MGB2 System Diagnostic Outputs
stMGB2modularPNIO.nMGB2modularQ_PnIoBoxCtrl :=  stMGB2modularPN.nMGB2modularQ_PnIoBoxCtrl;

//MBM Outputs
stMGB2modularPNIO.nMGB2modularQ_MBM[0].7 :=     stMGB2modularPN.xBM_ACK_G;

//Locking module Outputs
stMGB2modularPNIO.nMGB2modularQ_LM[0].0 :=      stMGB2modularPN.xLM_O_CL;
stMGB2modularPNIO.nMGB2modularQ_LM[0].6 :=      stMGB2modularPN.xLM_ACK_ER;
stMGB2modularPNIO.nMGB2modularQ_LM[0].7 :=      stMGB2modularPN.xLM_ACK_G;

//Submodule Outputs
stMGB2modularPNIO.nMGB2modularQ_SM[0].0 :=      stMGB2modularPN.xSM_O_H1;
stMGB2modularPNIO.nMGB2modularQ_SM[0].1 :=      stMGB2modularPN.xSM_O_H2;
stMGB2modularPNIO.nMGB2modularQ_SM[0].2 :=      stMGB2modularPN.xSM_O_H3;
stMGB2modularPNIO.nMGB2modularQ_SM[0].3 :=      stMGB2modularPN.xSM_O_H1_B;
stMGB2modularPNIO.nMGB2modularQ_SM[0].4 :=      stMGB2modularPN.xSM_O_H2_B;
stMGB2modularPNIO.nMGB2modularQ_SM[0].5 :=      stMGB2modularPN.xSM_O_H3_B;
```

Figure 20: Copying the MGB2 *Modular* output structures

## 9.4. PROFINET program

The variable structure for PROFINET diagnostics is created in the *PRG_ProfiNET* program. Additionally, an instance of the function block *FB_EUCHNER_MGB2modular* is called.

PRG_ProfiNET

```
PROGRAM PRG_ProfiNET
VAR
    //ProfiNET diagnostics
    nProfiNet_DevState      AT %I*  :UINT;
    nProfiNet_PnIoError     AT %I*  :UINT;
    nProfiNet_PnIoDiag      AT %I*  :UINT;
    nProfiNet_DevCtrl       AT %Q*  :UINT;

    //MGB2modular
    stMGB2modular                   : ST_MGB2modular_PN;
    fbMGB2modular                   : FB_EUCHNER_MGB2modular;
END_VAR
```

```
//Connect ProfiNet I/O to EUCHNER MGB2modular structure
fbMGB2modular(
    stMGB2modularPN:= stMGB2modular);
```

*Figure 21: PRG_ProfiNET program*

EN

## 9.5. EtherCAT program

The EtherCAT diagnosis can be read with the *PRG_EtherCAT* program.

PRG_EtherCAT

```
1  PROGRAM PRG_EtherCAT
2  VAR
3      //EtherCAT diagnostics
4      nEtherCAT_DevState          AT %I*   :   UINT;
5  END_VAR
6
```

```
1
```

*Figure 22: PRG_EtherCAT*

## 9.6. Main program *MAIN*

The main program *MAIN* is used to call the subprograms *PRG_ProfiNET* and *PRG_EtherCAT*.

MAIN

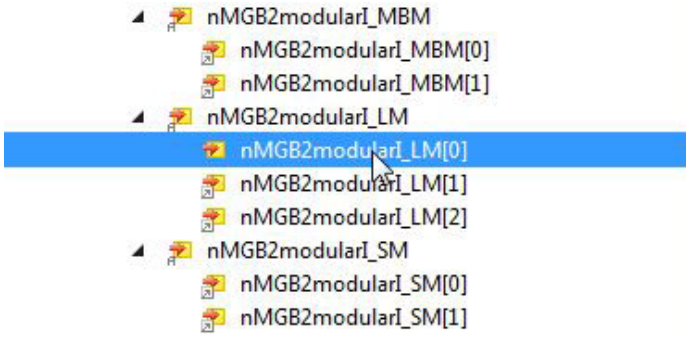
```
1  PROGRAM MAIN
2  VAR
3
4  END_VAR
```

```
1  PRG_EtherCAT();
2  PRG_ProfiNET();
3  PRG_TwinSAFE();
4
```

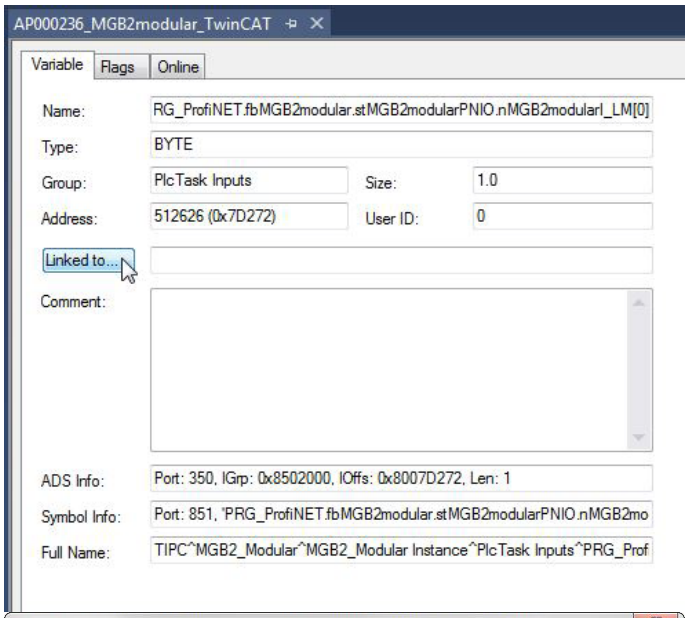
Figure 23: MAIN program

## 9.7. Linking the program variables

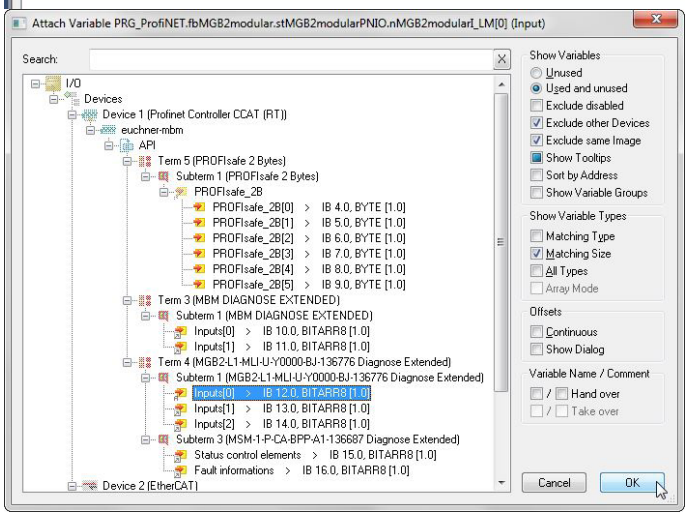
Linking establishes a connection between the MGB2 *Modular* input and output variables and the program structure. The CPU program must first be compiled for this purpose. The program can be compiled with *Build Solution (Ctrl+Shift+B)*. You can then find the variables to be linked under the created CPU instance.




1. Double-click the variable to be linked




2. Link the variables with *Linked to...*




3. Select the input area to be linked, and complete with OK

All created variables must be linked as described in this example.

EN

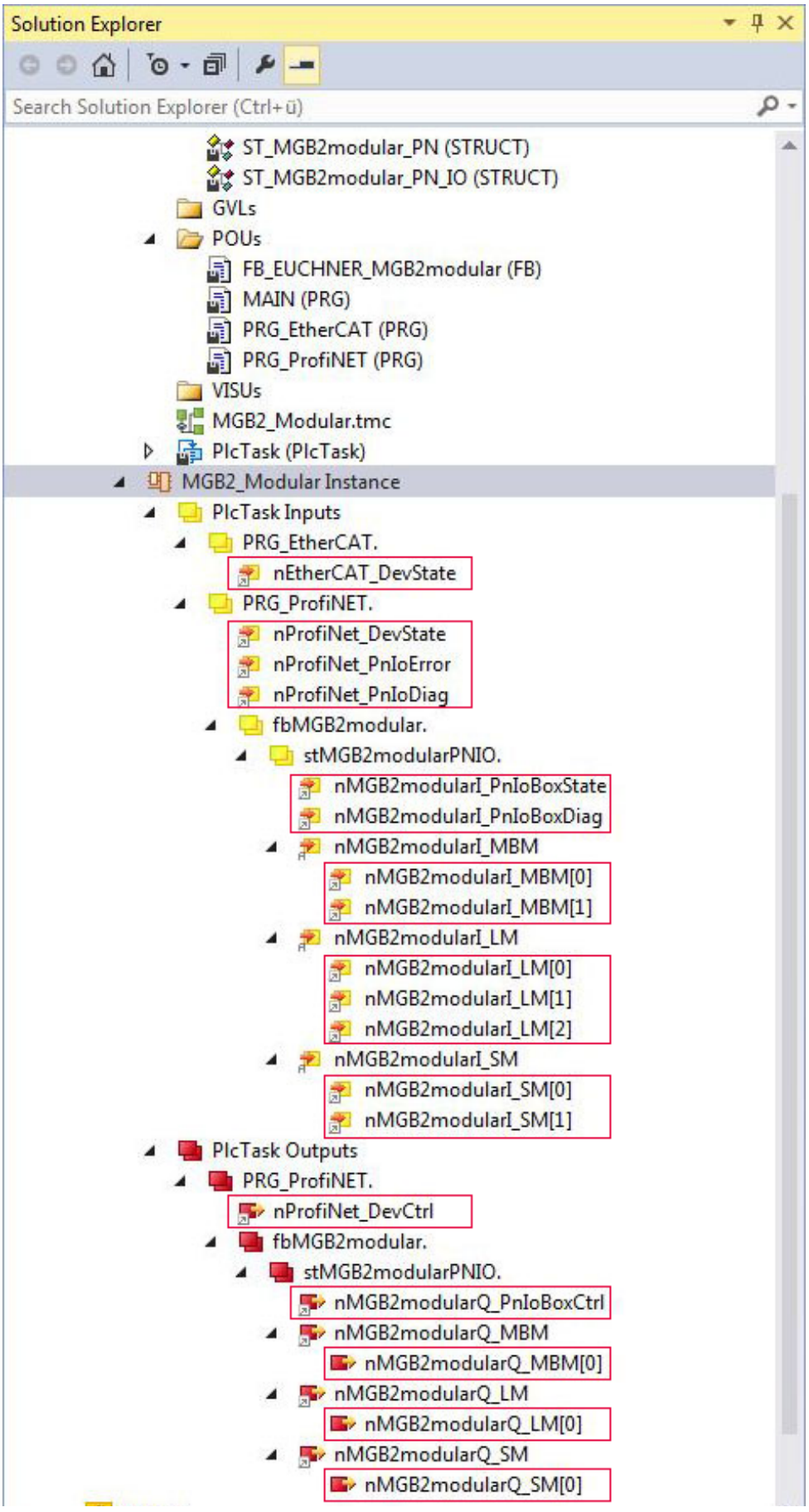



Figure 24: Variables to be linked

## 9.8. Transferring program to the PLC

Transfer the program to the control system by clicking *Activate configuration*  , and set the control system to *Run mode*.

## 9.9. Observing the non-safe variables

The inputs and outputs of the MGB2 *Modular* can be viewed using the block interface of the PRG_ProfiNET program. Go online by clicking *Login*  .

PRG_ProfiNET [Online]

AP000236_MGB2modular_TwinCAT.MGB2_Modular.PRG_ProfiNET

| Expression | Type | Value | Prepared value | Address |
|---|---|---|---|---|
| stMGB2modularPN | REFERENCE TO ST_... | | | |
| nMGB2modularI_PnIoBoxState | UINT | 0 | | |
| nMGB2modularI_PnIoBoxDiag | UINT | 2 | | |
| xBM_D_RUN | BOOL | TRUE | | |
| xBM_E_ML1 | BOOL | FALSE | | |
| xBM_E_ML2 | BOOL | FALSE | | |
| xBM_E_SYS | BOOL | FALSE | | |
| xBM_E_G | BOOL | FALSE | | |
| BBM_ExtendedDiagnostic | BYTE | 0 | | |
| xLM_D_RUN | BOOL | TRUE | | |
| xLM_I_SK | BOOL | TRUE | | |
| xLM_I_UK | BOOL | TRUE | | |
| xLM_E_SM0 | BOOL | FALSE | | |
| xLM_E_SM1 | BOOL | FALSE | | |
| xLM_E_ER | BOOL | FALSE | | |
| xLM_E_G | BOOL | FALSE | | |
| xLM_I_OD | BOOL | TRUE | | |
| xLM_I_OT | BOOL | TRUE | | |
| xLM_I_OL | BOOL | TRUE | | |
| BLM_ExtendedDiagnostic | BYTE | 0 | | |
| xSM_I_S1 | BOOL | TRUE | | |
| xSM_I_S2 | BOOL | FALSE | | |
| xSM_I_S3 | BOOL | FALSE | | |
| xSM_E_S1 | BOOL | FALSE | | |
| BSM_ExtendedDiagnostic | BYTE | 0 | | |
| nMGB2modularQ_PnIoBoxCtrl | UINT | 0 | | |
| xBM_ACK_G | BOOL | FALSE | | |
| xLM_O_CL | BOOL | FALSE | | |
| xLM_ACK_ER | BOOL | FALSE | | |
| xLM_ACK_G | BOOL | FALSE | | |
| xSM_O_H1 | BOOL | FALSE | | |
| xSM_O_H2 | BOOL | FALSE | | |
| xSM_O_H3 | BOOL | FALSE | | |
| xSM_O_H1_B | BOOL | FALSE | | |
| xSM_O_H2_B | BOOL | FALSE | | |

```
1  //Connect ProfiNet I/O to EUCHNER MGB2modular structure
2  fbMGB2modular(
3      stMGB2modularPN:= stMGB2modular);RETURN
```

Figure 25: Observing the MGB2 *Modular* variables

EN

# 10. Configuration of TwinSAFE – ProfiSAFE

The following chapter describes the configuration the TwinSAFE output and the ProfiSAFE connection of the MGB2 *Modular*.

1. Set the *PRG_TwinSAFE* program as the PLC program.
2. *PRG_TwinSAFE* variable declaration: The *xTwinSAFE_Run* and *xTwinSAFE_Ack* variables are required as transfer variables to the safe control system.

```
PRG_TwinSAFE
1  PROGRAM PRG_TwinSAFE
2  VAR
3      xTwinSAFE_RUN          AT %Q*      : BOOL := TRUE; // Enable for Safety Programm
4      xTwinSAFE_Ack          AT %Q*      : BOOL ;        // Diagnose: Acknowledge from Reset-button for Safety Program
5  END_VAR
6
```

*Figure 26: PRG_TwinSAFE* variable declaration

3. The submodule's S3 button is used to implement acknowledgment of the safe control system in the event of an error. For this purpose, assign the variables as shown in Figure 27.

```
PRG_TwinSAFE
1  PROGRAM PRG_TwinSAFE
2  VAR
3      xTwinSAFE_RUN          AT %Q*      : BOOL := TRUE; // Enable for Safety Programm
4      xTwinSAFE_Ack          AT %Q*      : BOOL ;        // Diagnose: Acknowledge from Reset-button for Safety Program
5  END_VAR
6

1  xTwinSAFE_Ack := PRG_ProfiNET.stMGB2modular.xSM_I_S3;
```

Figure 27: Acknowledgment using the submodule's button.

4. Call *PRG_TwinSAFE* in the main program

```
MAIN
1  PROGRAM MAIN
2  VAR
3
4  END_VAR

1  PRG_EtherCAT();
2  PRG_ProfiNET();
3  PRG_TwinSAFE();
4
```

*Figure 28: MAIN (PRG)*

5. Add the safety project

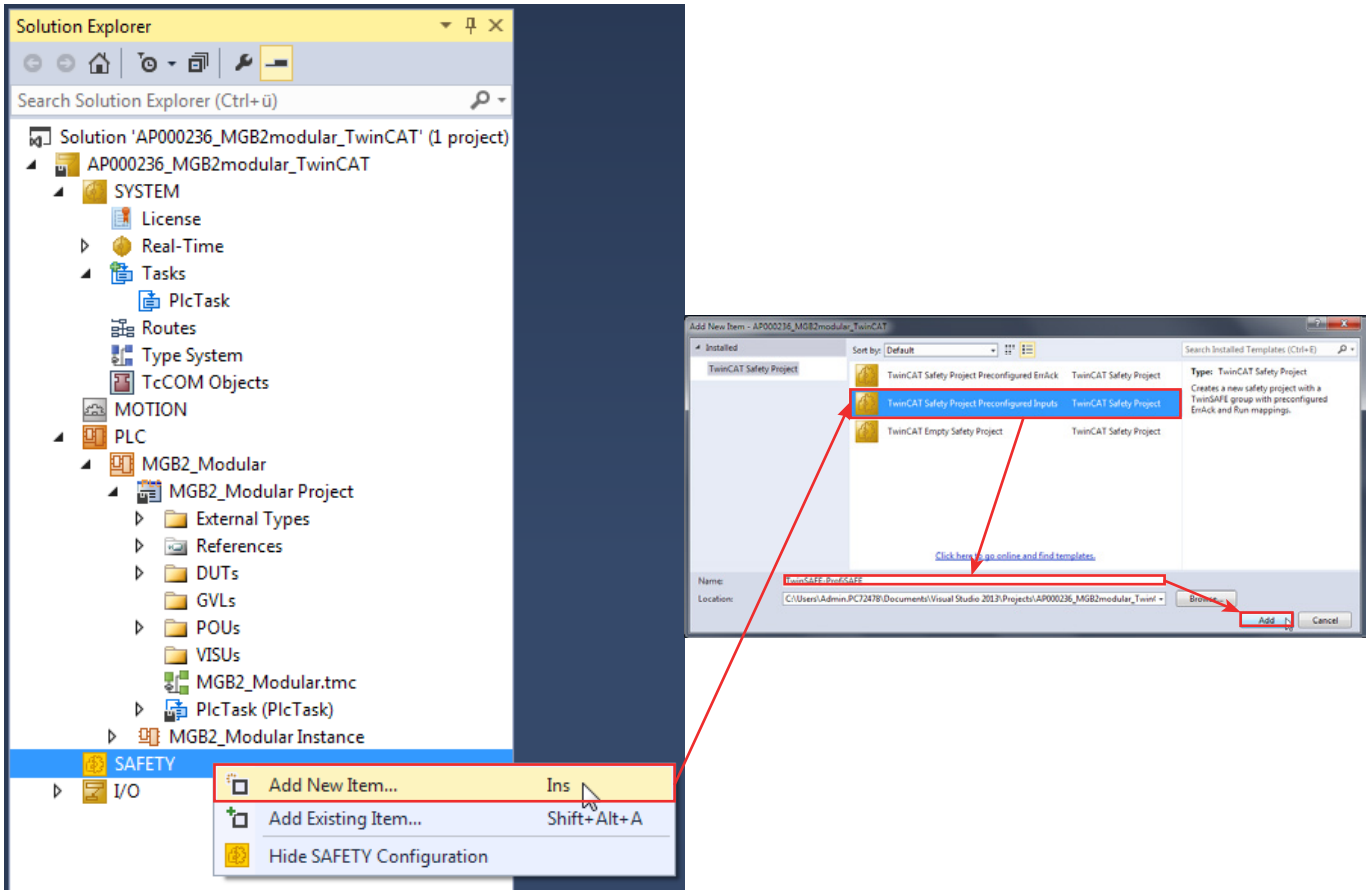



Figure 29: Adding safety project



EN

6. Create the target system: the Beckhoff terminal EL6910 must be selected as the target system. The terminal functions as the PROFIsafe controller as well. The safe address is also entered. *Map Serial Number* and *Map Project CRC* can be activated for expanded diagnostics.

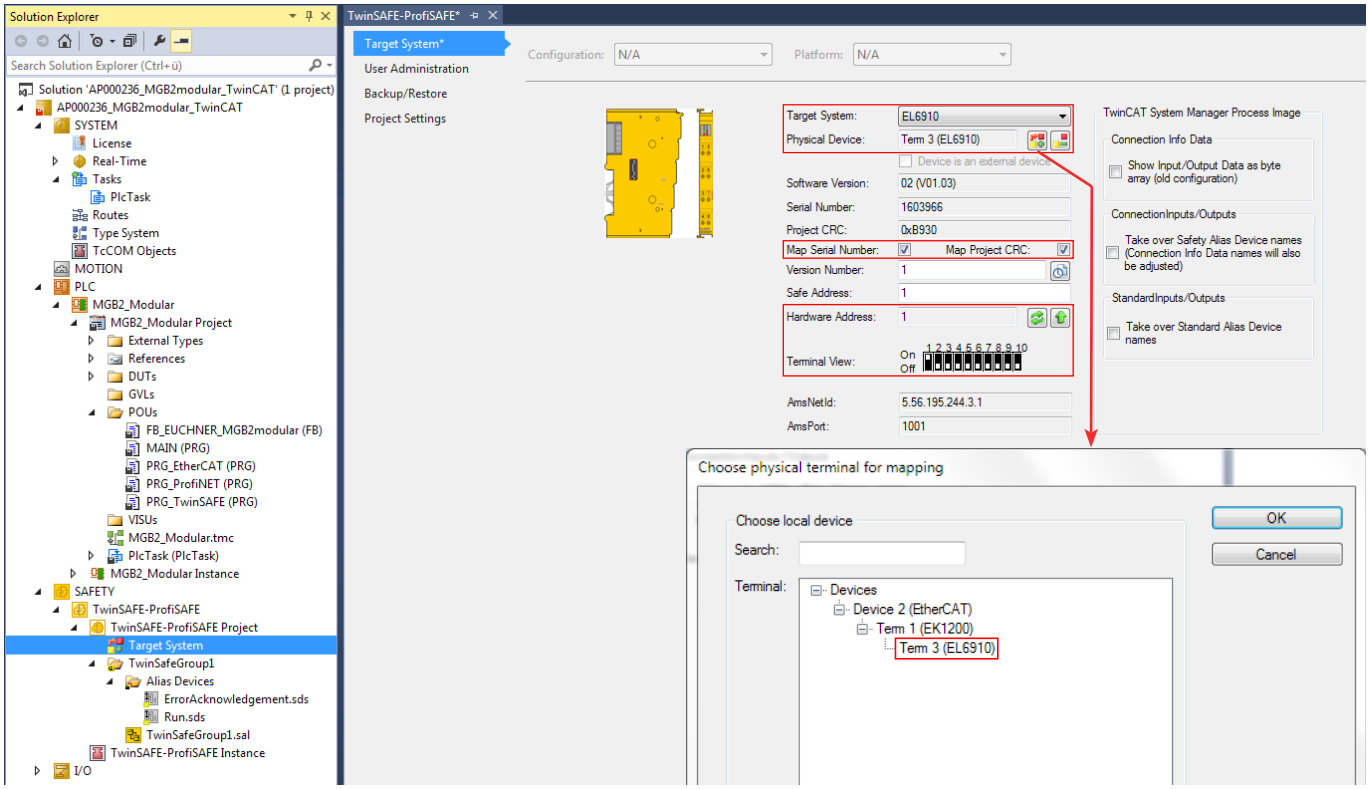



Figure 30: Target system

7. In the next step, the variables from PRG_TwinSAFE are assigned to the created *Alias Devices ErrorAcknowledgement* and *Run*. Open the properties by double-clicking the variable.

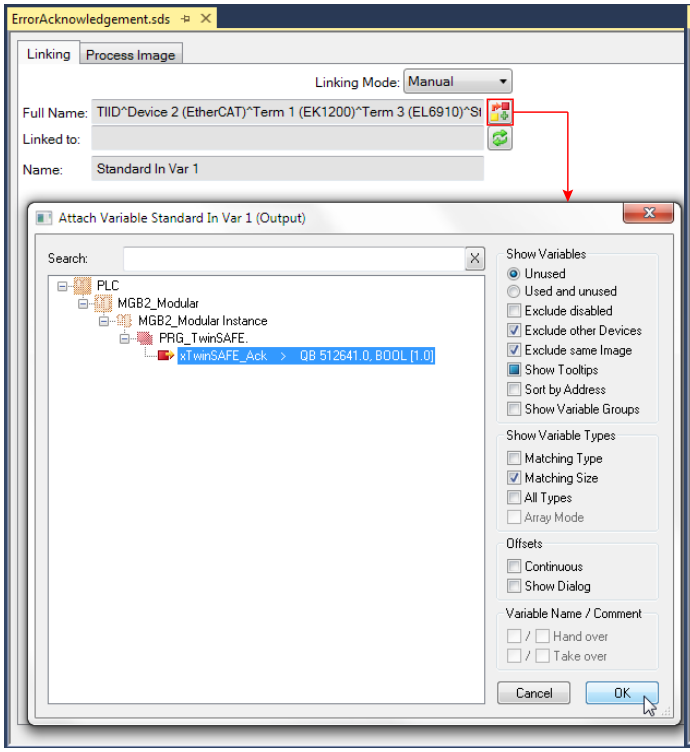



Figure 31: Alias *ErrorAcknowledgement*

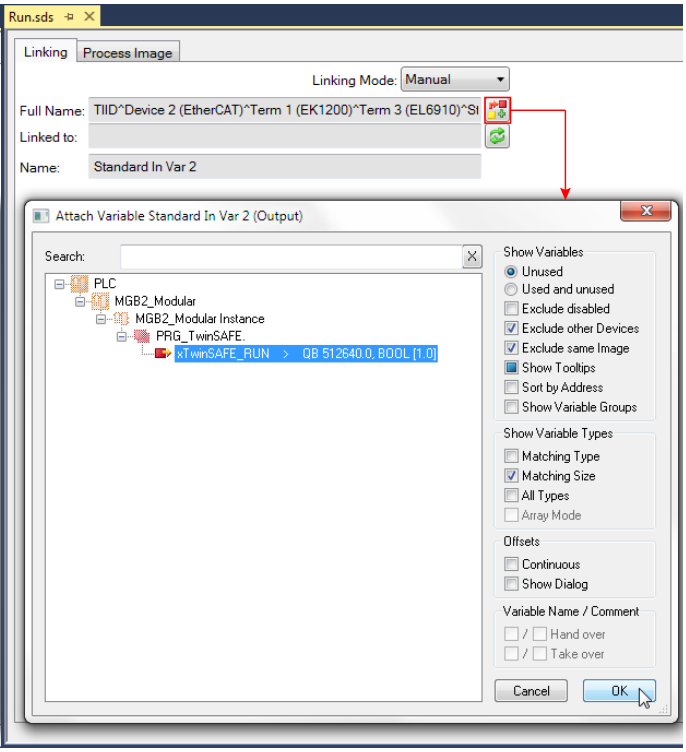



Figure 32: Alias *Run*

8. Add the PROFIsafe connection:

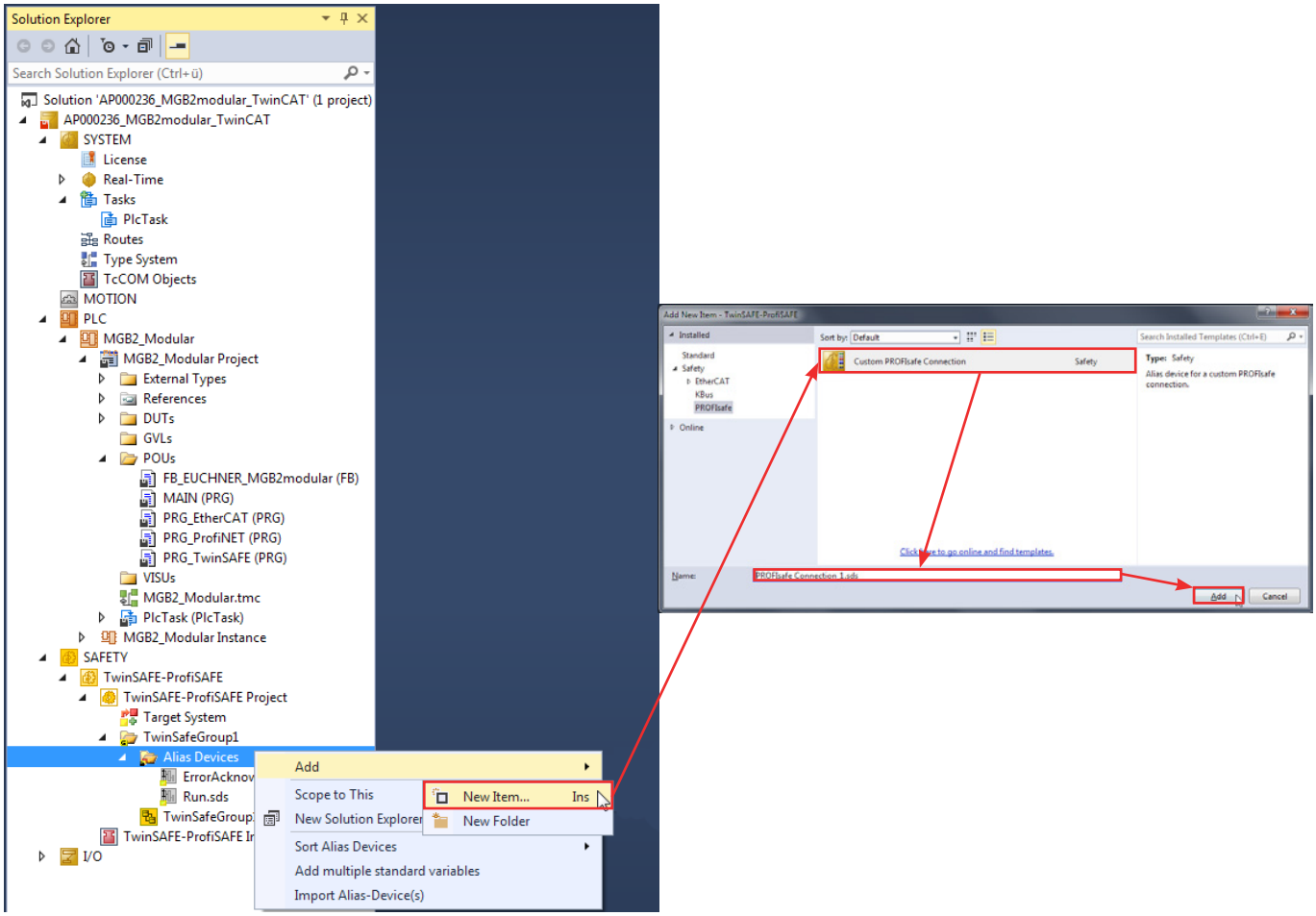



Figure 33: Adding PROFIsafe connection

EN

9. PROFIsafe settings of the MGB2 *Modular*: The assignment (mapping) to the slot *PROFIsafe 2 Bytes*, the safe address (physical DIP switch setting) and the F_WD_Time (600 ms from GSDML factory setting) must be set.

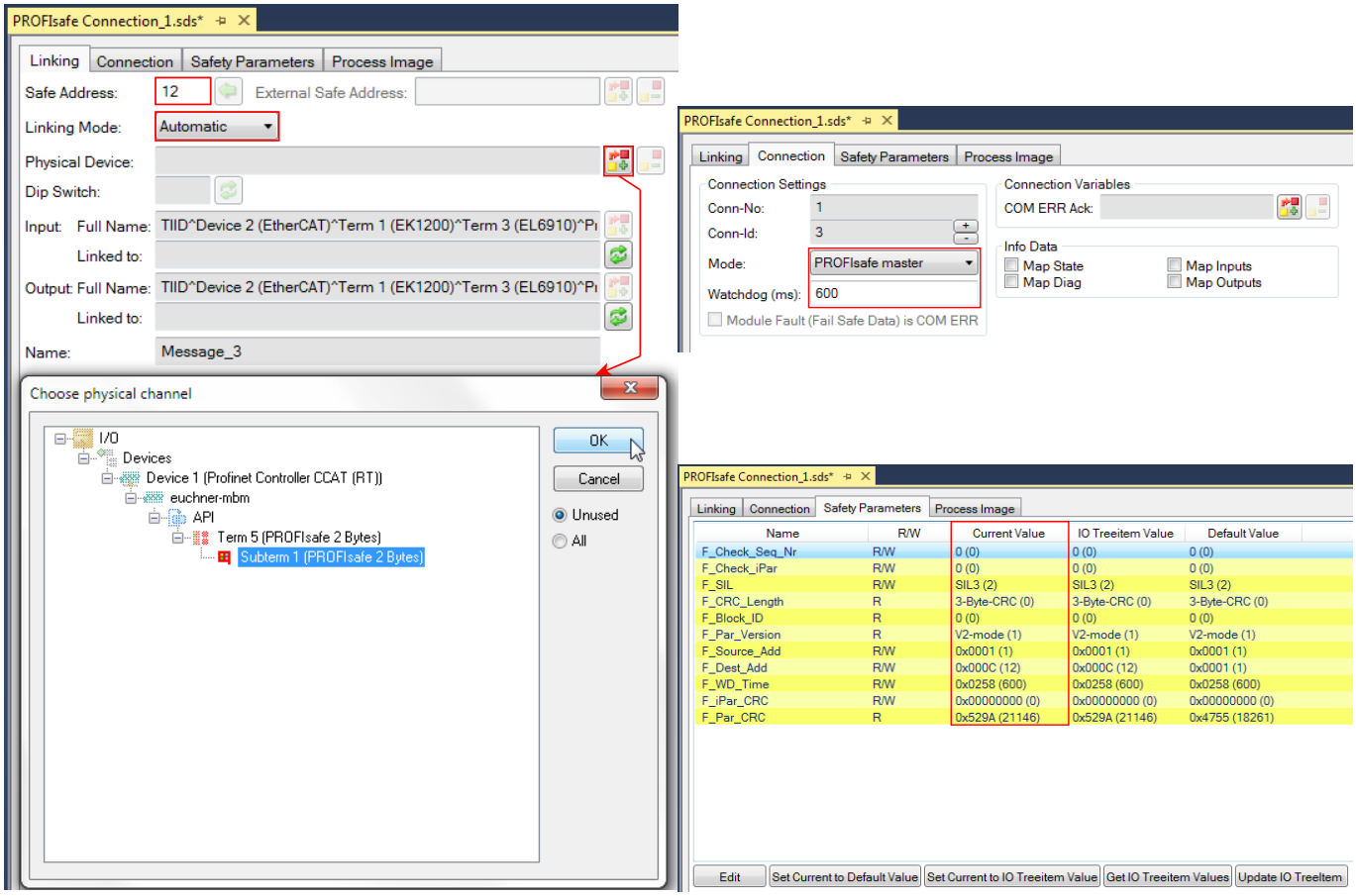



Figure 34: PROFIsafe settings

10. Add the TwinSAFE connection to terminal EL2904.

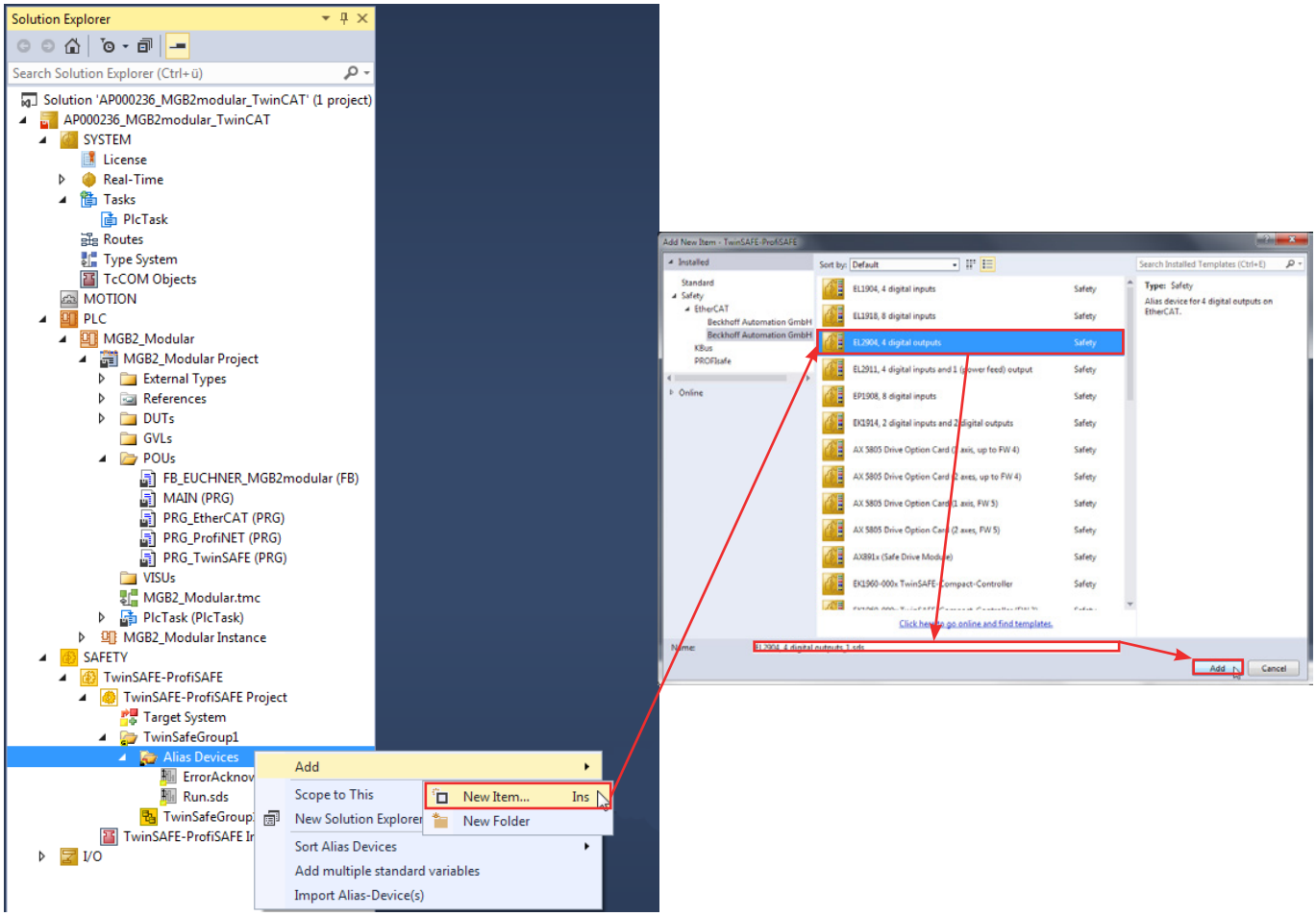



Figure 35: Adding TwinSAFE connection

11. Set terminal parameters: link to physical device and FSoE address (Fail Safe over EtherCAT; physical DIP switch setting).

EN

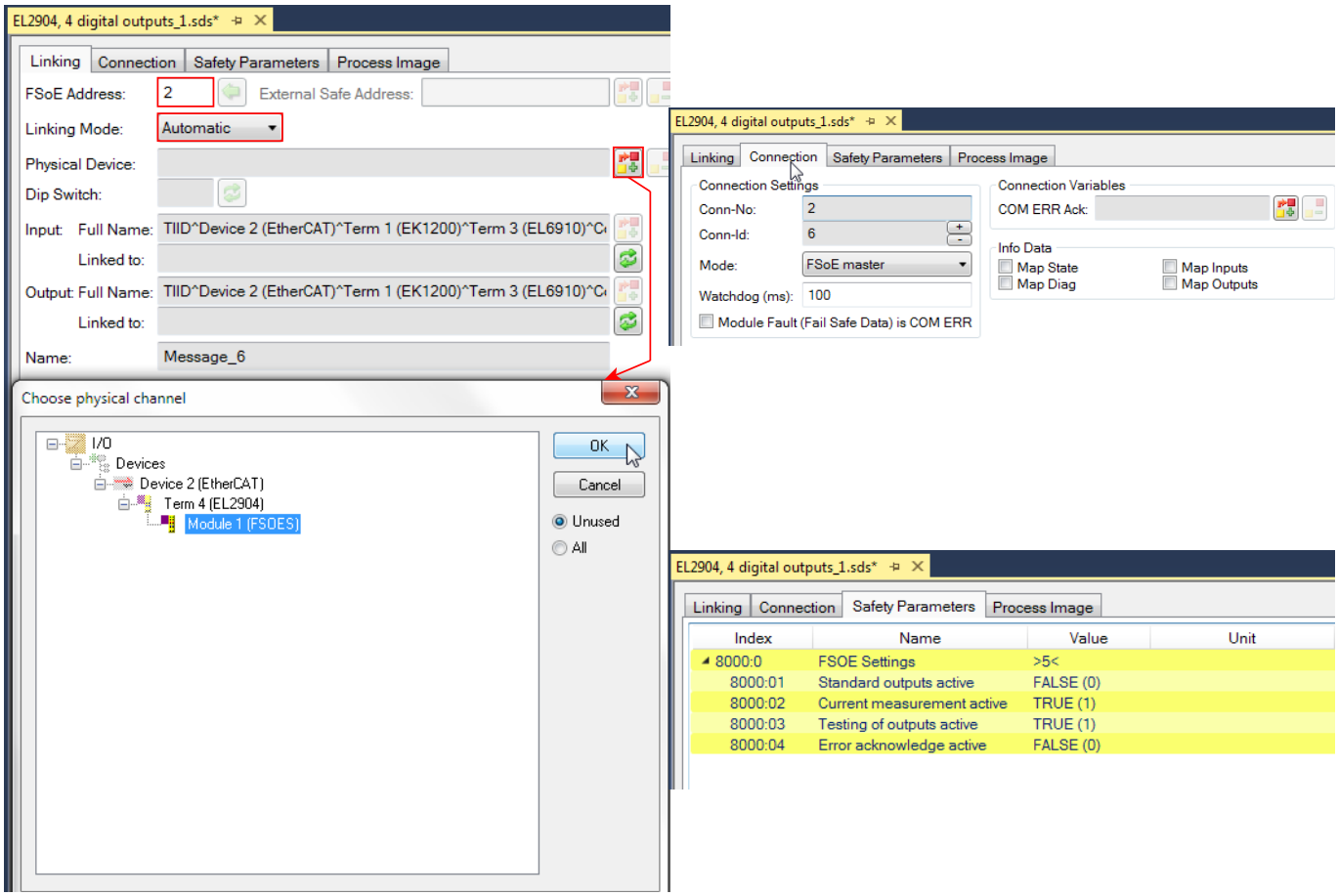



Figure 36: TwinSAFE settings EL2904

# 11. Creating the safety program

The safety engineering application is implemented in the sal (sal: Safety Application Language) worksheet belonging to the TwinSAFE group. It represents only an example of an application.

| | |
|---|---|
| ⓘ | **NOTE!**<br>There must be at least one call for an MBM safety bit in the safe part of the program to prevent the device from being passivated. |

## 11.1. Example of a safety program

In the following example, the safe output of terminal EL2904 (channel 1) is controlled by the bit LM_FI_UK. The conditions for the bit LM_FI_UK are met if the door is closed, the bolt tongue is in the locking module and guard locking is active.

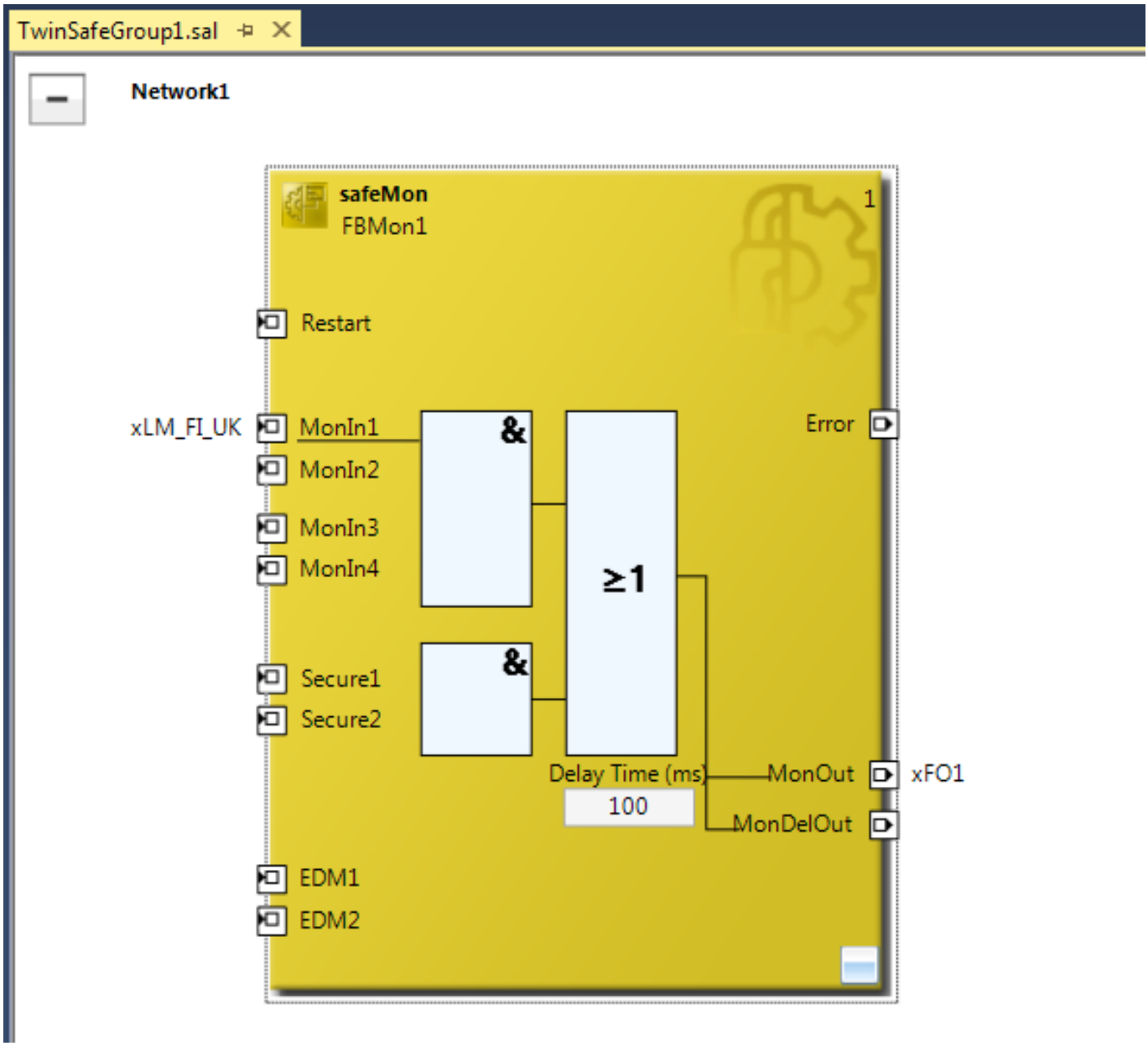



Figure 37: Example of a safety program

The variables are assigned for the TwinSAFE group (mapping) after addition of the blocks. Mapping must be performed for the variables *xLM_FI_UK*, *xFO_1*, *Err Ack* and *Run/Stop*.

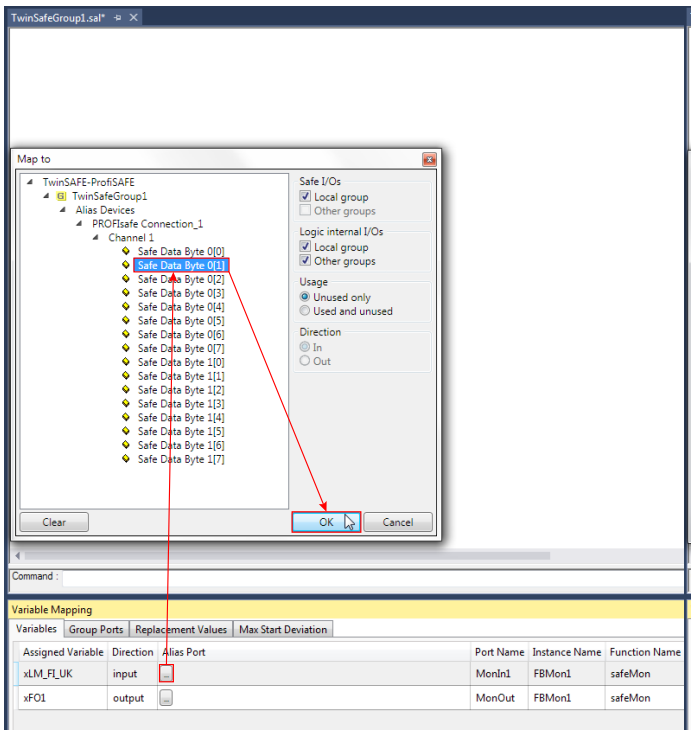



Figure 38: Mapping xLM_FI_UK

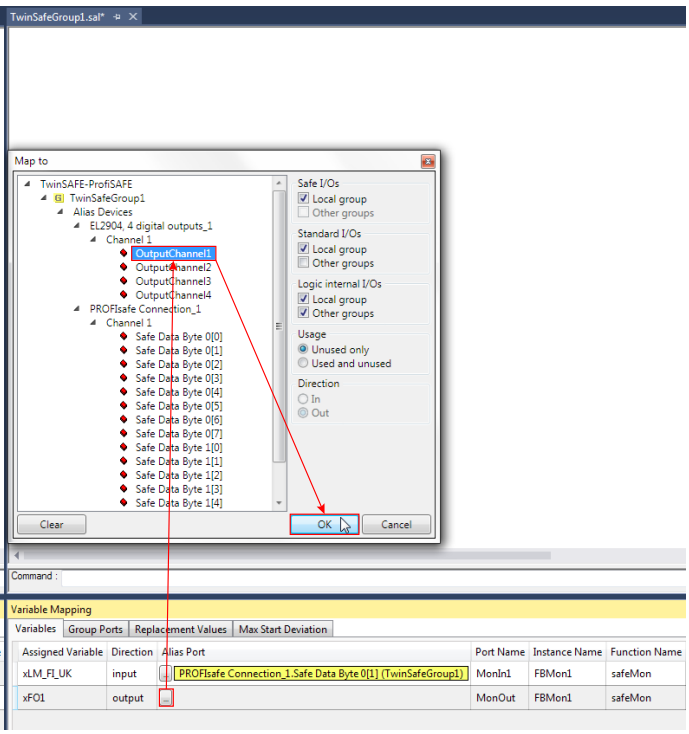



Figure 39: Mapping xFO_1

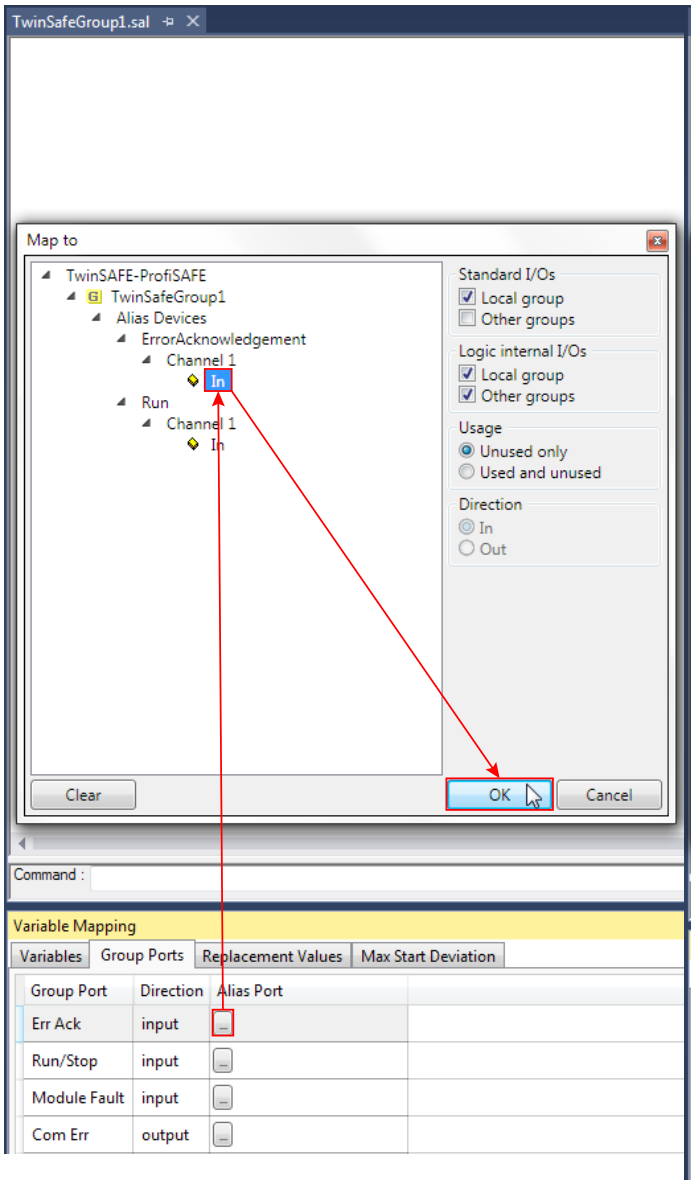



Figure 40: Mapping Err Ack

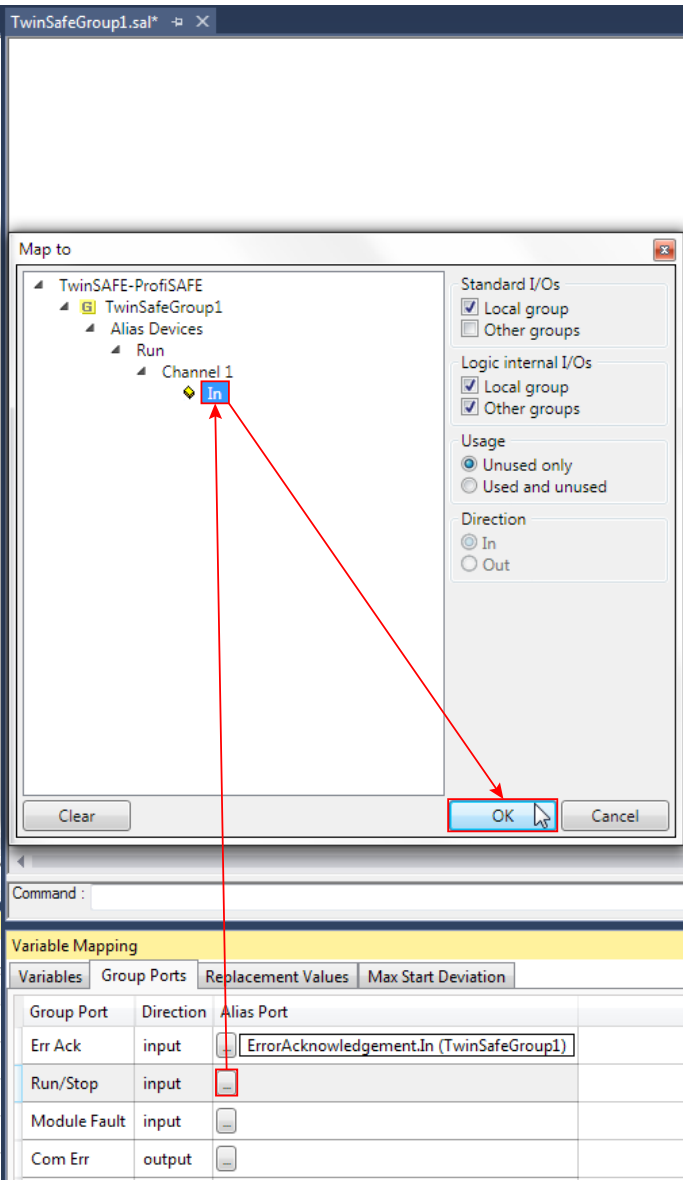



Figure 41: Mapping Run

## 11.2. Transferring safety program

Save the overall project with *Save All* and transfer the configuration with *Activate Configuration*. Subsequently check and transfer the TwinSAFE program to the control system. The user name [default: Administrator], the password [default: TwinSAFE] and the serial number of the target system will be required for transfer.

EN

# 12. Important note – please observe carefully!

This document is intended for a design engineer who possesses the requisite knowledge in safety engineering and knows the applicable standards, e.g. through training for qualification as a safety engineer. Only with the appropriate qualification is it possible to integrate the example provided into a complete safety chain.

The example represents only part of a complete safety chain and does not fulfill any safety function on its own. In order to fulfill a safety function, the energy switch-off function for the danger zone and the software must also be considered in the safety evaluation, for example.

The applications provided are only examples for solving certain safety tasks for protecting safety doors. The examples cannot be comprehensive due to the application-dependent and individual protection goals within a machine/installation.

**If questions concerning this example remain open, please contact us directly.**

According to the Machinery Directive 2006/42/EC, the design engineer of a machine or installation has the obligation to perform a risk assessment and take measures to reduce the risk. While doing this, the engineer must comply with the applicable national and international safety standards. Standards generally represent the current state-of-the-art. Therefore, the design engineer should continuously inform himself about changes in the standards and adapt his considerations to them. Relevant standards for functional safety include EN ISO 13849 and EN 62061. This application must be regarded only as assistance for the considerations about safety measures.

The design engineer of a machine/installation has the obligation to assess the safety technology himself. The examples must not be used for an assessment, because only a small excerpt of a complete safety function was considered in terms of safety engineering here.

In order to be able to use the safety switch applications correctly on safety doors, it is indispensable to observe the standards EN ISO 13849-1, EN ISO 14119 and all relevant C-standards for the respective machine type. Under no circumstances does this document replace the engineer's own risk assessment, and it cannot serve as the basis for a fault assessment.

In particular in relation to a fault exclusion, it must be noted that a fault can be excluded only by the machine's or installation's design engineer and this action requires justification. A general fault exclusion is not possible. More information about fault exclusion can be found in EN ISO 13849-2.

Changes to products or within assemblies from third-party suppliers used in this example can lead to the function no longer being ensured or the safety assessment having to be adapted. In any event, the information in the operating instructions on the part of EUCHNER, as well as on the part of third-party suppliers, must be used as the basis before this application is integrated into an overall safety function. If contradictions should arise between the operating instructions and this document, please contact us directly.

**Use of brand names and company names**

All brand names and company names stated are the property of the related manufacturer. They are used only for the clear identification of compatible peripheral devices and operating environments in relation to our products.

EUCHNER GmbH + Co. KG
Kohlhammerstraße 16
70771 Leinfelden-Echterdingen
Germany
info@euchner.de
www.euchner.com

Edition:
AP000236-01-02/21
Title:
Application MGB2
Integration of MGB2 Modular in Beckhoff TwinCAT 3