Application



Integration of MGB2 *Modular* in Beckhoff TwinCAT 3

From V1.5.8

Contents

1.	Abou	It this document	4
	1.1.	Version	4
	1.2.	Scope	4
	1.3.	Target group	4
	1.4.	Supplementary documents	4
	1.5.	Notice	4
2.	Com	ponents/modules used	5
	2.1.	EUCHNER	5
		2.1.1. Items included in the MGB2 <i>Modular</i> set	5
	2.2.	Others	5
	2.3.	Software	6
3.	Fund	tional description	6
4.	Over	view of the communication data	7
	4.1.	Input	7
	4.2.	Output	7
5	Insta	lling the GSD file	8
о. с	0		
6.	Setti	ng the control system parameters	8
7.	Twin	SAFE and PROFIsafe hardware addressing	9
	7.1.	TwinSAFE	9
	7.2.	PROFIsafe	9
8.	Conf	iguration of the MBM and the I/O peripherals	10
	8.1.	Adding the I/O devices in the project	10
	8.2.	Setting the MGB2 <i>Modular</i> parameters	14
		8.2.1. PROFINET	
	83	Assigning PROFINET device name to the bus module MBM	16 17
•			10
9.		Structure of the connection for PROFINET I/O configuration	
	9.1.	Structure for readability of the inputs (autputs	
	9.2.	Structure for readability of the inputs/outputs	20 22
	9.3.	9.3.1 Conving the CPU input structure to the MGB2 Modular structure	22 22
		9.3.2. Copying the MGB2 Modular output structure to the CPU structure	23
	9.4.	PROFINET program	23
	9.5.	EtherCAT program	24
	9.6.	Main program MAIN	24
	9.7.	Linking the program variables	24

	9.8.	Transferring program to the PLC	27
	9.9.	Observing the non-safe variables	27
10.	Confi	guration of TwinSAFE – ProfiSAFE	28
11.	Creat	ting the safety program	35
	11.1.	Example of a safety program	35
	11.2.	Transferring safety program	37
12.	Impo	rtant note – please observe carefully!	38

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	www
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	www
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 <i>Modular</i> with modular bus module	156386 / MGB2-L1HB-PN-U-S4-D-R-156386
lock monitoring	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

			MO	BB2 Mo	odular	set	
Description	Order number/item number	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-MLHJ-Y0000-BJ-136776					-	-
	156392 / MGB2-L2-MLHJ-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 <u>www.euchner.com</u>. 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	СХ9020-0110-М930
BECKHOFF TwinSAFE Logic	EL6910
BECKHOFF 4-channel digital output terminal, TwinSAFE, 24 V DC	EL2904

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 O
1st byte	BM.E_G	-	BM.E_SYS	-	-	BM.E_ML2	BM.E_ML1	BM.D_RUN
2nd byte				Diagnostics b	yte (pluggable)			
PROFINET	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit O
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 b	yte (pluggable)			
PROFINET	Rit 7	Rit 6	Rit 5	Rit /	Rit 2	Rit 2	Rit 1	Rit O
1 et hyto	Dit 7	Ditto	SM F S1	Dit 4	Dit S	SMI S3	SML S2	SML S1
2nd hute	-	-	SWI.L_51	- Diagnostica h		3101.1_33	3111.1_32	311.1_31
2nd byte				Diagnostics b	yte (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		1	PROI	Flsafe intern genutz d within PROFIsafe	t (Steuerbyte, CRC, (control byte, CRC,	usw.) etc.)		
1.2. 0	utput							
PROFINET	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1st byte	BM.ACK_G	-	-	-	-	-	-	-
DROFINET	D:+ 7	Dia 6	Dia E		D:+ 0	Di+ 0	Dia 1	Dia O
			BILS	DIL 4	DIUS	DIL Z	DILI	
I St Dyte	LIVI.ACK_G	LIVI.ACK_ER	-	-	-	-	-	LIVI.U_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
	D1 7		D1 5	D': 4	D': 0	D'' 0	D'- 1	51.0
PROFIsate	Bit /	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit U
I st byte	-	-	-	-	-	-	-	LM.FO_CL
2nd byte	-	-	-	-	-	-	-	-
Bytes 3-6			PROF	Flsafe intern genutz	t (Statusbyte, CRC,	usw.)		

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*.

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 <u>www.euchner.com</u>. Always use the latest GSD file.

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

C:/TwinCAT/3.1/Config/lo/Profinet

Name	Änderungsdatum	Тур	Größe
GSDML-0135-0301-MBM.bmp	04.04.2018 17:10	Bitmap-Bild	4 KB
GSDML-V2.33-EUCHNER-MBM_2512512_T14-20190122.xml	22.01.2019 13:43	XML-Dokument	239 KB

Figure 1: Content of the ZIP file

Computer > Windows (C:) > TwinCAT > 3.1 > Config > Io > Profinet

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.



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	29	28	27	26	2 ⁵	24	2 ³	2 ²	21	20
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.

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

8.1. Adding the I/O devices in the project



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*.

	MAC address	IP address	Subnet	Rescan Devices
	00:1A:5D:03:ED:63	0, 0, 0, 0	0.0.0	Add Devices
∢ Stationname	ш		4	
				Set Stationname
IP configuration				Set Stationname Set IP configuration
IP configuration IP address	0.0.0.0	DHCP er	nable	Set Stationname
IP configuration IP address Subnet	0.0.0.0	DHCP er	nable	Set Stationname Set IP configuration Start Signal

Figure 6: Adding MBM

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



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

🔺 🔀 I/O	
⊿ * 🖥 D	evices
4	# Device 1 (Profinet Controller CCAT (RT))
	🛟 Image
Þ	Inputs
Þ	Outputs
4	euchner-mbm
	👂 🛄 Inputs
	Outputs
	A 📑 API
	Term 1 (DAP Module)
	Term 5 (PROFIsafe 2 Bytes)
	Term 3 (MBM DIAGNOSE EXTENDED)
	Term 4 (MGB2-L1-MLI-U-Y0000-BJ-136776 Diagnose Extended)
	Subterm 1 (MGB2-L1-MLI-U-Y0000-BJ-136776 Diagnose Extended)
	Subterm 2 (EmptySubslot)
	Subterm 3 (MSM-1-P-CA-BPP-A1-136687 Diagnose Basic)



(\mathbf{i})	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 <i>MBM DIAGNOSE EXTENDED</i> module in slot 2 is added automatically. It can be replaced with the <i>MBM DIAGNOSE BASIC</i> module.

ΕN

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

Solution Explorer	• 4 ×	AP000236_MG	B2modular_T	winCAT 👳 🗙	C	
© ⊃ ☆ `o - ฮ ≠ <mark></mark>		General De	vice Diagnos	sis <mark>Fe</mark> atures <i>A</i>	ADS	
Search Solution Explorer (Ctrl+ü)	ρ-	Adapter F	roperties			
 Solution 'AP000236_MGB2modular_TwinCAT' (1 project) AP000236_MGB2modular_TwinCAT SYSTEM 		Station	name ner-mbm			
MOTION		Vendo	rld l	DeviceId	HW Version	SW Version
PLC SAFETY		0x01	35	0x0301	0.00	T 0.00
₩ C++ Σ I/O		IP con	figuration			
Devices		IP ac	Idress 192	. 168 . 1	. 2	
Evice 1 (Profinet Controller CCAT (RT)) Image		Subr	et 255	. 255 . 255	. 0	
 Inputs Imputs 		Gate	way 192	. 168 . 1	. 1	Refresh GSDML
⊿ 🚟 euchner-mbm						
Inputs		Instance	Properties			
Dutputs		0.000		-rameid		
 Igit API Igit Term 1 (DAP Module) 		0000		UKOUUU		
Term 2 (EmptySlot)		MaxLe	ngthln N	MaxLengthOut	ActLengthIn	ActLengthOut
 Term 3 (MBM DIAGNOSE EXTENDE) Term 4 (MGB2-L1-MLI-U-Y0000-BJ- 	D) 136776 Di	1440	Byte	1440 Byte	13 Byte	10 Byte
Subterm 1 (MGB2-L1-MLI-U-Y00	JOO-RJ-130					
 Inputs Outputs 						

Figure 11: PROFINET parameters

olution Explorer 🔹 4 🗙 🖌	AP000236_MGB2modular_1w	INCAI -P	×
○ ○ ☆ [™] - 司 ⊁ -	General Device Diagnosis	Features	ADS
Search Solution Explorer (Ctrl+ü)	IO Custia Data		
 Solution 'AP000236_MGB2modular_TwinCAT' (1 project) AP000236_MGB2modular_TwinCAT SYSTEM MOTION PLC SAFETY C++ I/O Devices Devices Image Inputs Outputs 	Controller Cycle Time Device Cycle Time Min Device Interval Send Clock Factor Reduction Ratio Phase Watchdog Factor Watchdog Time	4 ms 16 ms 512 32 16 1 3 48 ms	Cycle time from master task DevCycleTime = SendClockFactor * 31,25us * RedRe Default = 3 Watchdog Time = Watchdog Factor * DevCycleTime
 euchner-mbm Inputs Outputs API Term 1 (DAP Module) Term 2 (EmptySlot) Term 3 (MBM DIAGNOSE EXTENDED) Term 4 (MGB2-L1-MLI-U-Y0000-BJ-136776 Di Subterm 1 (MGB2-L1-MLI-U-Y0000-BJ-136 Inputs Outputs 	Comment The timing parameters an	re OK!	

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)

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].

rameterize Module				
Isafe	Name	R/W	Offline Value	Online Value
lex 0x1	F_SIL	R	SIL3	
	F_CRC_Length	R	3-Byte-CRC	
	F_Block_ID	R	0	
	F_Par_Version	R	1	
	F_Source_Add	R/W	1	
	F_Dest_Add	R/W	12	
	F_WD_Time	R/W	600	
	F_Par_CRC	R	21146	
Read	RecordData to Index 0x	1, Transfer Sequ	ience is 0. If you click	'Set to Default' the whole index will be set to default!
vvnte				
et to Default				

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. \enskip \text{To assign the name to MGB2} \enskip \text{Modular via TwinCAT, right-click the PROFINET controller and then select $Scan$}.$

olution Explorer	• 4 ×	AP000236_	MGB2mo	odular_lwin(AI 🗝 X	
0 0 🔂 '0 - 🗗 🕨 🗕		General	Adapter	PROFINET	Sync Task	Setting
Search Solution Explorer (Ctrl+ü)	- م	IP com	figuration			
Solution 'AP000236_MGB2mo	dular_TwinCAT' (1 project) TwinCAT	IP ad	dress	192 . 168	. 1 . 1	
SYSTEM		Subn	et	255 . 255	. 255 . 0	
 PLC SAFETY 		Gate	way	192 . 168	. 1 . 1	
<u>₩</u> C++		Name	of Pnlo Co	ontroller Statio	n	
		ccat-	oncontrolle	er		
Device 1 (Press	Add New Item		I	ns		
Dinputs	Add Existing Item		9	Shift+Alt+A		
👂 🖷 Outputs 🗙	Remove		[Del		
Billion Billion	Change NetId				Port	
Dutp	Save Device 1 (Profinet Controller CC	AT (RT)) As				
API	Online Reset					
	Online Reload				signment	
	Online Delete					
🔺 🏭 Te 😒	Scan 🕞					
▲ ■	Change Id					
× ت	Сору		(Ctrl+C		
🖌 🗮 Device 2 (Etł 💥	Cut		(Ctrl+X		
Image	Paste		(Ctrl+V		
SyncUnit	Paste with Links					
👂 🛄 Inputs 🔓	Independent Project File					
 Outputs InfoData 	Disable					

Figure 14: Searching for devices online

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.

Stationname	MAC address	IP address	Subnet	Rescan Devices
	00:1A:5C:03:ED:63	0. 0. 0. 0	0. 0. 0. 0	Add Devices
tationname euchner-mbm	III		•	Set Stationname
Cationname euchner-mbm			•	Set Stationname
Stationname euchner-mbm IP configuration IP address	192 . 168 . 1 . 2	DHCP e	nable	Set Stationname Set IP configuration
Stationname euchner-mbm IP configuration IP address Subnet	192 . 168 . 1 . 2 255 . 255 . 255 . 0	DHCP e	nable	Set Stationname Set IP configuration Start Signal

Figure 15: Assigning device names

(1)	
(1)	

TIP!

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_MGB-2modular_PN_IO structure.

S	T_MGB2	2modular_PN_IO 🗢 🗙	
E	1	TYPE ST_MGB2modular_PN_IO :	
	2	//Struct for MGB2-Set Inputs and Ou	tputs
E	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 [01] OF BYTE;
	9	//Locking module Inputs	
	10	nMGB2modularI_LM	AT %I* : ARRAY [02] OF BYTE;
	11	//Submodule Inputs	
	12	nMGB2modularI_SM	AT %I* : ARRAY [01] OF BYTE;
	13		
	14	//MGB2 System Diagnostic Output	
	15	nMGB2modularQ_PnIoBoxCtrl	AT Q^* : UINT;
	16	//MBM Outputs	
	17	nMGB2modularQ_MBM	AT %Q* : ARRAY [00] OF BYTE;
	18	//Locking module Outputs	
	19	nMGB2modularQ_LM	AT %Q* : ARRAY [00] OF BYTE;
	20	//Submodule Outputs	
	21	nMGB2modularQ_SM	AT Q^* : ARKAI [00] OF BITE;
	22	END_STRUCT	
	23	END_TIPE	



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_N	/IGB2m	odular_l	PN + ×	
	1	TYF	E ST_MGB2modular_PN :	
	2	STR	NCT	
	3		//MGB2 System Diagnostic Inputs	
	4		nMGB2modularI PnIoBoxState	: VINT;
	5		nMGB2modularI PnIoBoxDiag	: VINT;
	6			
	7		//MBM Inputs	
	8		//nMGB2modularI MBM	AT %I* : ARRAY [01] OF BYTE;
	9		xBM D RUN	: BOOL;
	10		xBM E ML1	: BOOL;
	11		xBM E ML2	: BOOL;
	12	11	x103 BM	: BOOL;
	13	11	x104 BM	: BOOL;
	14		xBM E SYS	: BOOL;
	15	11	x105 BM	: BOOL;
_	16		×BM E G	: BOOL:
	17		BBM ExtendedDiagnostic	: BYTE:
	18			
	19		//Locking module Inputs	
	20		//Inputs nMGB2modularI LM	AT %I* : ARRAY [02] OF BYTE:
	21		xLM D RUN	: BOOL;
	22		xLM I SK	: BOOL;
	23		xLM I UK	: BOOL;
	24	11	x203 LM	: BOOL;
	25		xLM E SMO	: BOOL;
	26		xLM E SM1	: BOOL;
	27		xLM E ER	: BOOL;
	28		xLM E G	: BOOL;
	29		xLM I OD	: BOOL;
	30		xLM_I_OT	: BOOL;
	31		xLM_I_OL	: BOOL;
	32		BLM_ExtendedDiagnostic	: BYTE;
	33			
	34		//Submodul Inputs	
	35		//nMGB2modular_SM	AT %I* : ARRAY [01] OF BYTE;
	36		xSM_I_S1	: BOOL;
	37		xSM_I_S2	: BOOL;
	38		xSM_I_S3	: BOOL;
	39	11	x303_SM	: BOOL;
	40	11	x304_SM	: BOOL;
	41		xSM_E_S1	: BOOL;
	42	11	x306_SM	: BOOL;
	43	11	x307_SM	: BOOL;
	44		BSM_ExtendedDiagnostic	: BYTE;
	45			

Figure 17: Structure of inputs

ST_N	/IGB2mo	dular_F	PN ⇔ ×	
	48			
	49		//MBM Outputs	
	50		//nMGB2modularQ_MBM	AT %Q* : ARRAY [00] OF BYTE;
	51	11	x100_BM	: BOOL;
	52	11	x101_BM	: BOOL;
	53	11	x102_BM	: BOOL;
	54	11	x103_BM	: BOOL;
	55	11	x104_BM	: BOOL;
	56	11	x105_BM	: BOOL;
	57	11	x106_BM	: BOOL;
	58		xBM_ACK_G	: BOOL;
	59			
	60		//Locking module Outputs	
	61		//nMGB2modularQ_LM	AT %Q* : ARRAY [00] OF BYTE;
	62		xLM_O_CL	: BOOL;
	63	11	x201_LM	: BOOL;
	64	11	x202_LM	: BOOL;
	65	11	x203_LM	: BOOL;
	66	11	x204_LM	: BOOL;
	67	11	x205_LM	: BOOL;
	68		xLM_ACK_ER	: BOOL;
	69		xLM_ACK_G	: BOOL;
	70			
	71		//Submodule Outputs	
	72		//nMGB2modularQ_SM	AT %Q* : ARRAY [00] 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	11	x306_SM	: BOOL;
	80	11	x307_SM	: BOOL;
	81			
	82	END	STRUCT	
	83	END	TYPE	

Figure 18: Structure of outputs

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 + X						
	1	FUNCTION_BLOCK FB_EUCHNER_MGB2modular				
	2	VAR_IN_OUT				
	3	stMGB2modularPN :S	T_MGB2modular_PN;			
	4	END_VAR				
	5	VAR				
	6	stMGB2modularPNIO :S	T_MGB2modular_PN_IO;			
	7	END_VAR				
	1	//MGB2 System Diagnostic	Inputs			
	2	stMGB2modularPN.nMGB2modu	larI_PnIoBoxState :=	stMGB2modularPNIO.nMGB2modularI_PnIoBoxState;		
	3	stMGB2modularPN.nMGB2modu	larI PnIoBoxDiag :=	stMGB2modularPNIO.nMGB2modularI PnIoBoxDiag;		
	4					
	5	//MBM Inputs				
	6	stMGB2modularPN.xBM D RUN	:=	<pre>stMGB2modularPNIO.nMGB2modularI MBM[0].0;</pre>		
	7	stMGB2modularPN.xBM E ML1	:=	stMGB2modularPNIO.nMGB2modularI MBM[0].1;		
	8	stMGB2modularPN.xBM E ML2	:=	stMGB2modularPNIO.nMGB2modularI MBM[0].2;		
	9	stMGB2modularPN.xBM E SYS	:=	stMGB2modularPNIO.nMGB2modularI MBM[0].5;		
	10	stMGB2modularPN.xBM E G :	=	stMGB2modularPNIO.nMGB2modularI MBM[0].7;		
	11	stMGB2modularPN.BBM Extend	dedDiagnostic :=	stMGB2modularPNIO.nMGB2modularI MBM[1];		
	12	_	2			
	13	//Locking module Inputs				
	14	stMGB2modularPN.xLM D RUN	:=	stMGB2modularPNIO.nMGB2modularI LM[0].0:		
	15	stMGB2modularPN.xLM I SK	:=	stMGB2modularPNIO.nMGB2modularI LM[0].1;		
	16	stMGB2modularPN.xLM I UK	:=	stMGB2modularPNIO.nMGB2modularI LM[0].2;		
	17	stMGB2modularPN.xLM E SM0	:=	stMGB2modularPNIO.nMGB2modularI LM[0].4:		
	18	stMGB2modularPN.xLM E SM1	:=	stMGB2modularPNIO.nMGB2modularI LM[0].5;		
	19	stMGB2modularPN.xLM F FR	:=	stMGB2modularPNIO.nMGB2modularI LM[0].6;		
	20	stMGB2modularPN.xLM E G	=	stMGB2modularPNIO.nMGB2modularI LM[0].7:		
	21	stMGB2modularPN.xLM T OD	:=	stMGB2modularPNIO.nMGB2modularT_LM[1].0:		
	22	stMGB2modularPN.xLM T OT	:=	stMGB2modularPNIO.nMGB2modularI LM[1].1:		
	23	stMGB2modularPN.xLM T OT	:=	stMGB2modularPNIO.nMGB2modularI_LM[1].2:		
	24	stMGB2modularPN_BLM_Fyter	dedDiagnostic :=	stMGB2modularPNIO.nMGB2modularI_IM[2]:		
	25	- Chopemodululin, phi-Excell	acadiagnosoito .			
	26	//Submodule Inputs				
	27	etMGB2modularDN_vSM_T_S1	•=	stMGB2modularPNIO pMGB2modularI SM[0] 0.		
	20	atMCR2modularPN xSM_1_S1		atMCP2modularPNIO.nMCP2modularI_SM[0].0;		
	20	atMCR2modularPN.xSM_1_52	. –	atMCP2madularPNIO.nMCP2madularI_SM[0].1;		
	29	atMCR2module=DN +SM T C1	·	atMCR2modularPNIO nMCR2modularI_SM[0].2;		
	30	stMGB2modularPN.XSM_E_S1		stwepzmodularPNIO.nMGBZmodularI_SM[U].5;		
	31	SIMGBZMOQUIATPN.BSM_Extend	ueuviagnostic :=	sumedzmodularPNIO.nmeszmodulari_SM[1];		
	32					

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
-
    2
       VAR IN OUT
           stMGB2modularPN
    3
                                  :ST_MGB2modular_PN;
       END VAR
    4
       VAR
    5
6
           stMGB2modularPNI0
                                  :ST_MGB2modular_PN_IO;
       END VAR
   32
   33
        //MGB2 System Diagnostic Outputs
   34
        stMGB2modularPNIO.nMGB2modularQ PnIoBoxCtrl := stMGB2modularPN.nMGB2modularQ PnIoBoxCtrl;
   35
   36
        //MBM Outputs
   37
        stMGB2modularPNIO.nMGB2modularQ_MBM[0].7 :=
                                                           stMGB2modularPN.xBM_ACK_G;
   38
   39
        //Locking module Outputs
    40
         stMGB2modularPNIO.nMGB2modularQ LM[0].0 :=
                                                           stMGB2modularPN.xLM O CL;
   41
         stMGB2modularPNIO.nMGB2modularQ_LM[0].6 :=
                                                           stMGB2modularPN.xLM ACK ER;
   42
        stMGB2modularPNIO.nMGB2modularQ_LM[0].7 :=
                                                           stMGB2modularPN.xLM_ACK_G;
   43
   44
        //Submodule Outputs
    45
        stMGB2modularPNIO.nMGB2modularQ SM[0].0 :=
                                                         stMGB2modularPN.xSM O H1;
    46
        stMGB2modularPNIO.nMGB2modularQ_SM[0].1 :=
                                                         stMGB2modularPN.xSM O H2;
    47
        stMGB2modularPNIO.nMGB2modularQ_SM[0].2 :=
                                                         stMGB2modularPN.xSM O H3;
   48
        stMGB2modularPNIO.nMGB2modularQ_SM[0].3 :=
                                                         stMGB2modularPN.xSM_O_H1_B;
        stMGB2modularPNIO.nMGB2modularQ_SM[0].4 :=
    49
                                                           stMGB2modularPN.xSM_O_H2_B;
   50
        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_	Profi	VET 🕫 🗙
	1	PROGRAM PRG_ProfiNET
	2	VAR
	3	//ProfiNET diagnostics
	4	nProfiNet_DevState AT %I* :VINT;
	5	nProfiNet_PnIoError AT %I* :VINT;
	6	nProfiNet_PnIoDiag AT %I* :VINT;
	7	nProfiNet_DevCtrl AT %Q* :VINT;
	8	
	9	//MGB2modular
	10	stMGB2modular : ST_MGB2modular_PN;
	11	fbMGB2modular : FB_EUCHNER_MGB2modular;
	12	END_VAR
	13	
	1	//Connect ProfiNet I/O to EUCHNEP MGB2modular structure
	2	fbMGB2modular(
	2	atMCP2modularDN:= atMCP2modular);
	3	StMGDZmodularPN:- StMGDZmodular);



9.5. EtherCAT program

The EtherCAT diagnosis can be read with the PRG_EtherCAT program.



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	-Þ	Х	
	1		PROGRAM MAIN
	2		VAR
	3		
	4		END_VAR
	-		
	1		PRG_EtherCAT();
	2		<pre>PRG_ProfiNET();</pre>
	3		<pre>PRG_TwinSAFE();</pre>
	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.



Name:	RG_ProfiNET.fbMGB2modula	ar.stMGB2modula	rPNIO.nM	IGB2modularl_LM[0]		
Type:	BYTE					
Group:	PlcTask Inputs	Size:	1.0	.0		
Address:	512626 (0x7D272)	User ID:	0			
Linked to						
Comment:				^		
ADS Info:	Port: 350, IGrp: 0x8502000, 1	Offs: 0x8007D27	2, Len: 1	Y		
Symbol Info:	Port: 851, 'PRG ProfiNET.fbl	MGB2modular.stN	/GB2mod	ularPNIO.nMGB2mo		
-,	TIRCOMORO Madulas CMORO Madulas Instances CRIs Tasle Instance DRC					
Full Name:	TIPC^MGB2_Modular^MGB2	2_Modular Instan	ce [^] PlcTa	sk Inputs^PRG_Prof		
Full Name: Attach Variable PRG arch: Devices Devices Comparison Devices	TIPC^MGB2_Modular^MGB2 ProfiNET.fbMGB2modular.stMGB2modul 1 (Profinet Controller CCAT (RT)) hmer-mbm AP Term 5 (PROFIsate 2 Bytes) GMDF1ate_2B(2) > 18 4 G	2_Modular Instance 	e^PicTas	sk Inputs*PRG_Prof Input) Show Variables Unused Ugrused Exclude dirabled Exclude stabled Exclude stab		

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.



Figure 24: Variables to be linked

9.8. Transferring program to the PLC

Transfer the program to the control system by clicking Activate configuration **1**, 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* 2.

PRG_ProfiNET [Online] → ×						
AP000236_MGB2modular_TwinCAT.MGB2_Modular.PRG_ProfiNET						
Expression	Туре	Value	Prepared value	Address	*	jii)
🖃 🍫 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				
<pre>wbm_e_g</pre>	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				
<pre> xLM_I_OD </pre>	BOOL	TRUE				
<pre> xLM_I_OT </pre>	BOOL	TRUE				
<pre> xLM_I_OL </pre>	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				
<pre> xBM_ACK_G </pre>	BOOL	FALSE				
<pre> xLM_0_CL </pre>	BOOL	FALSE				
xLM_ACK_ER	BOOL	FALSE				
🛷 xLM_ACK_G	BOOL	FALSE				
<pre> xSM_0_H1 </pre>	BOOL	FALSE				
xSM_0_H2	BOOL	FALSE				
xSM_0_H3	BOOL	FALSE				
<pre> xSM_0_H1_B </pre>	BOOL	FALSE				
xSM_0_H2_B	BOOL	FALSE			-	
	1			►		
<pre>1 //Connect ProfiNet I/O to EUCHNER MGB2modular structure 2 fbMGB2modular(3 stMGB2modularPN:= stMGB2modular);RETURN</pre>						



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.



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.



Figure 27: Acknowledgment using the submodule's button.

4. Call PRG_TwinSAFE in the main program

MAIN	P × Contraction of the second s
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

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.

ErrorAcknowledgement.sds 👳 🗙	Run.sds 🕫 🗙			
Linking Process Image	Linking Process Image			
ErrorAcknowledgement.sds * × Linking Process Image Linking Mode: Manual Full Name: TID*Device 2 (EtherCAT)*Term 1 (EK1200)*Term 3 (EL6910)*St Linked to: Name: Standard In Var 1 Attach Variable Standard In Var 1 (Output) Search: MGB2_Modular MGB2_Modular Instance PRG_TwinSAFE_ack > QB 512641.0.B00L [1.0] Show Variable Topics Show Variable	Run.sds + × Linking Process Image Linking Mode: Manual Full Name: TIID*Device 2 (EtherCAT)*Term 1 (EK1200)*Term 3 (EL6910)*SI Linked to: Name: Standard In Var 2 Attach Variable Standard In Var 2 (Output) Search: MGB2_Modular MGB2_Modular Instance MGB3_GRUN Show Toolips Show Toolips Show Variable Groups Show Variable Groups			
Controlog Show Dialog Variable Name / Comment / Hand over / Take over Cancel OK	Show Dialog Variable Name / Comment / Hand over / Take over Cancel OK			

Figure 31:Alias ErrorAcknowledgement

Figure 32: Alias Run

8. Add the PROFIsafe connection:



Figure 33: Adding PROFIsafe connection

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

 $10. \ \mbox{Add}$ 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).

EL2904, 4 digital outputs_1.sds* 👳 🗙	
Linking Connection Safety Parameters Process Image	
FSoE Address: 2 🔅 External Safe Address:	
Linking Mode: Automatic 🔻	EL2904, 4 digital outputs_1.sds* 😔 🗙
Physical Device:	Linking Connection Safety Parameters Process Image
Dip Switch:	Connection Settings
Input: Full Name: TIID^Device 2 (EtherCAT)^Term 1 (EK1200)^Term 3 (EL6910)^C(Conn-No: 2 Conn-Id: 6
Linked to:	Mode: FSoE master Map Inputs
Output: Full Name: TIID^Device 2 (EtherCAT)^Term 1 (EK1200)^Term 3 (EL6910)^C(Watchdog (ms): 100
Linked to:	Module Fault (Fail Safe Data) is COM ERR
Name: Message_6	
Choose physical channel	EL2904, 4 digital outputs_1.sds* + × Linking Connection Safety Parameters Process Image Index Name Value Value Unit * 8000:0 FSOE Settings >5< 8000:01 Standard outputs active FALSE (0) 8000:02 Current measurement active TRUE (1) 8000:03 Testing of outputs active TRUE (1) 8000:04 Error acknowledge active FALSE (0)

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!

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

ΕN

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

TwinSafeGroup1.sal + ×		TwinSafeGroup1.sal* 🗢 🗙
Map to TwinSAFE-ProfiSAFE A G TwinSafeGroup1 A Alias Devices Channel 1 Channel 1 Clear Command :	Standard I/Os Very Local group Other groups Logic internal I/Os Very Local group Other groups Usage Usage Unused only Used and unused Direction In Out Ott Cancel	Map to
Variable Mapping Variables Group Ports Replacement Values Max Start D	eviation	Variable Mapping
Group Port Direction Alias Port		Variables Group Ports Replacement Values Max Start Deviation
Err Ack input		Group Port Direction Alias Port
Run/Stop input		Err Ack input ErrorAcknowledgement.In (TwinSafeGroup1)
Module Fault input		Run/Stop input
Com Err output		Module Fault input
		Com Err output

Figure 40: Mapping Err Ack

Figure 41: Mapping Run

11.2. Transferring safety program

Save the overall project with Save All \checkmark and transfer the configuration with Activate Configuration \checkmark . Subsequently check \checkmark \checkmark and transfer \checkmark 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.

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

Copyright: © EUCHNER GmbH + Co. KG, 02/2021

Subject to technical modifications; no responsibility is accepted for the accuracy of this information. $% \label{eq:sub_constraint}$