INTER CONTROL

digsy®fusion S

Manual

V 01.15.01

© INTER CONTROL

Hermann Köhler Elektrik GmbH & Co. KG
Schafhofstraße 30

90411 Nürnberg

Germany

Telefon:	++49 911 9522-5
Fax:	++49 911 9522-857
E-Mail:	info@intercontrol.de
Internet:	www.intercontrol.de

digsy®$_{\text{fusion}}$ S

Manual

Edition dated 05.12.2023

No. 04-68491

Great care has been taken in regard to the content of this document. The information presented is checked and updated at regular intervals and is subject to change and additions without prior notice at any time. Nevertheless, INTER CONTROL cannot assume liability for the correctness of the information in this document, since, despite considerable effort, it is not always possible to exclude errors. The specified technical data were determined by us in a test environment (we will provide information on this on request) and only represent the agreed quality on this basis. The customer/client is responsible for checking the suitability for the purpose intended by the customer/client or the use under the specific conditions of use; we do not assume any warranty for this. We reserve the right to make changes.

Any ideas or suggestions for changes, as well as the reporting of errors in this document, are welcome. Please contact INTER CONTROL.

Revision list:

Version	Date	Author	Changes
V1.0.0	Nov 17, 2015	LS	Review completed, released
V1.1.0	Nov 30, 2015	LS	Added Description for safe inputs and outputs. Added version V02.07.XX.
V1.2.0	Dec 11, 2015	LS	Added Certificate and Declaration of conformity
V1.2.1	Feb 11, 2016	LS	Added changes from review
V1.3.1	Mar 09, 2016	LS	Added product version V02.08.XX
V01.04.00	Jul 18, 2016	LS	Added product version V02.10.XX, added new Safetyrequirements, changed value of hysteresis and switching threshold of inputs Type D
V01.04.02	Mar 23, 2017	LS	Added Chapter Retain, CE document update, added the device description into the validity table
V01.06.02	Sep 11 2017	LS/HJE	Added version V02.15.XX and V03.25.XX, added calculation of MTTFd values for CAN, modified Errorcodes 10.13 and LED-Behaviour, added paek detection, modified IPON Hysteresys, modified 24h operation time, modified memory overview, modified description of system logbook, Rework §7372/§7421/§149, modified CODESYS-Configfile, import new declaration of conformity, Added Standard [N2] to [N9], Modify table 5-5 (IPON), Description of tasks, Modification of safety information of IO-Toggling
V01.06.03	Sep 12 2017	LS	Added new Certificate and Declaration of Conformity
V01.07.00	Oct 19 2017	LS	Added Versions V02.16.XX and V03.26.XX, versions V02.15XX and V03.25.XX are invalidated
V01.08.00	18.01.2018	LS	Approved FPU and mathematical functions (Chapter 7.4)
V01.08.01	19.03.2018	HJE	Reference to the FPU / mathematical functions safety manual adapted
V01.09.00	14.11.2018	LS	Device version 03.30.02 added, accuracy with respect to the type A PWM frequencies changed, information on peak current detection in type A PWM added. Added description for FTP. Paragraphs added. Invalid versions V02.15.XX and V03.25.XX removed.
V01.09.01	22.01.2019	LS / JR	GCM-P inserted Function CAN_Tra_Async added

Version	Date	Author	Changes
V01.10.00	14.06.2019	LS / JR	PLC digsy fusion S-P and module GCM-P added, version list extended, paragraphs amended or adapted, chapter 7.4 adapted, description RS232 added.
V01.10.01	17.06.2019	LS	Added variant 4888.03.32X, expand library chapter, updated version of Norm ISO25119
V01.10.02	03.07.2019	LS	Added variant 4888.03.32X in table, updated the certificate and declaration of conformity
V01.11.00	10.03.2020	MK	Added Libraryfunktion SetUserOperationTime Changed Chapter 9.3 library name Chapter 3.2.4 added table device variants Chapter 10.13 changed list of PLC versions Chapter 4.6.2.5, 4.6.3.4 revised Chapter 8.1.14.2 SysLogRead revised CODESYS V3.5 SIL2 versions revised §2027 revised Chapter 7.1.4.2 revised Chapter 7.3, 7.3.1, 7.3.2, 7.5.4,7.5.7.4 revised Chapter 1.6, 7.6.1.2,7.8.1 revised Chapter 7.3.7 added Chapter 7.8.3 added Chapter 7.4.11,§7421 revised Chapter 8.1.5.1 und 8.1.5.4 revised Chapter 4.4 revised §5378 revised Chapter 4.6.5 table changed Chapter 3.2.5.3, 3.2.5.4, 4.6.6.3 revised Chapter 7,4, 7.5.3, 7.5.7 revised Chapter 4.4.5 deleted §7414/§3549 revised §7011 deleted §5252 changed Chapter 1.4, 7.4.10, 7.7.1.2, 7.8.1 changed Chapter 5.5, 7.1.6.1 added MS2019-1 GIOM added Chapter 10.15 Errorcodes revised §6856, §149 changed §8094, §8095, §8096, §8097, §8098 added Added new version of Declaration of conformity Chapter 10.6 MTTFD revised Chapter 10.7 MuB added Problems with links and headers solved
V01.11.01	12.03.2020	MK	Chapter 1.6 errat sheet version changed changed Chapter 7.1.2, 6.1.6 and 6.1.7 added chapter 4.3.1 §153
V01.11.02	19.03.2020	MK	Chapter 1.6 library versions changed
V01.11.03	16.06.2020 21.12.2020	MK	Figure 4-11 changed fixed translation gaps (table captions)

Version	Date	Author	Changes
V01.12.00	11.01.2021 01.03.2021	MK	Chapter 1.6 Version Errata-Sheet changed, Item number 4888.04.XXX added Chapter 3.2, 3.2.2, 3.2.3, 3.2.4.1, 3.2.4.2, 3.2.4.3, 3.2.4.4, 3.2.5.6, 5.2, 7.1.2, 7.5.2, 7.5.6.2, 7.6.1.1 digsy fusion S-PL and GCM-PL added Chapter 3.2.3.4 GCM-PL added Chapter 3.2.4 Item 4888.03.3X5 and 4888.04.XXX added Chapter 8.3.2 SysGetGcmpState added Chapter 10.14 Item 4888.03.3X5 and 4888.04.XXX added chapter 1.6, 7.3.7, 10.1.14 PLC version 01.06.XX added chapter 4.3.1 item number 4888.04.XXX added chapter 10.13.3 power consumption for digsy fusion S-PL added chapter 4.8.9: §5492 edited chapter 7.1.4.2 corrected .icd file prefix chapter 10.13.3 added missing data
V01.12.01	23.04.2021	MK	Chapter 10.14 changed
V01.12.02	20.07.2021 04.05.2022	MK BS	Chapter 4.6.5.4 Added Note Version list updated
V01.13.0	13.07.2022 21.07.2022	BS HJE	Legal text updated, Version list updated, Chapter 4.1.3 added: 24V/12V system voltage, The former chapter 4.1.3 is now 4.1.4, §7124 and §7125 added, Chapter 7.6.1.1 modified
V01.13.01	15.09.2022	BS	Version list updated (Modbus variants)
V01.14.00	08.02.2023	BS	Version list updated (V03.37.03 resp. V01.10.XX)
V01.14.01	25.04.2023	LS	Safety release of inputs changed; chapter 4.6.4 added; chapter 4.6.5, 4.6.6 and 4.6.7 reworked; §1691, §3765 and §2024 reworked; §8192 and §8103 added;
V01.15.00	15.09.2023 26.09.2023 26.09.2023	BS WP LS	Chapter 8.1.12 updated; several formattings adjusted; version list updated (V04.40.XX resp. V06.40.XX); Chapter 3.2.6.6: Formatting of table 3.7 reworked Chapter 10.13.5: short circuit current A1 for new HW version added Chapter 10.12: data concerning protection class and mechanical protection added
V01.15.01	28.11.2023 05.12.2023	HJE BS	Device data and safety instructions type A1 adjusted Chapter 7.6.1.1: reference to chapter 4.1.3 added

Table of Contents

1 Introduction

1.1 Contents of this document

digsy®$_{fusion}$ S is a modular, freely programmable electronic control system for outdoor machines operated under harsh environmental conditions. It fulfills all functional safety requirements set out in the European Machinery Directive 2006/42/EC.

This document includes the following:

- Basic general safety notes
- Generic system description
- Description of the system components
- Description of the functional components
- Description of the programming
- Detailed description of all accordingly available software libraries
- Description of the operation of the control system
- Description of the safety functions
- Appendix (Technical Data / Environmental Impact)

1.2 Target group

This document is intended for:

- Personnel responsible for machine project engineering
- Personnel responsible for machine development
- Personnel responsible for machine manufacture
- Personnel responsible for machine servicing
- Personnel responsible for machine documentation
- All persons who must be familiar with the technical characteristics, especially the safety features, of the product described.

1.3 Validity

This document applies to products labeled ***digsy***®$_{fusion}$ S.

digsy®$_{fusion}$ S is available in different variants. These differ e.g. with regard to the memory extension and/or the number of inputs/outputs.

The following ***digsy***®$_{fusion}$ S variants (see chapter 10.14) are available at the date of issue of this document.

1.4 Reference documents

In addition to this document, other documents are necessary for understanding the use of the product as well as its programming and handling.

- H2__CODESYS_Safety_SIL2_-_IEC_Programming_Guidelines.pdf (3S Smart Software Solutions) Version V5.0
- 04-68509-010400_CODESYS_Runtime_Math-Lib_Safety_Manual.pdf

Safety Manual for the usage of the FPU and the mathematical functions in the safe context from the company INTER CONTROL.

- [MAN-GCM-P] 04-68518-010000_digsy-fusion-S-P_GCM-P_Manual_E.pdf Manual GCM-P (V01.00.00 or newer)
- [MAN-GCM-PL] 04-68539-010000_digsy-fusion_GCM-PL_Manual_E.pdf (V01.00.00 or newer)

Please contact INTER CONTROL for any of the above documents.

1.5 Norms and standards

This document refers to the norms and standards mentioned below.

Reference	Norm / Standard	Comment
[N1]	EN ISO 13849-1:2015 Safety of Machinery – Safety-Related Parts of Control Systems – Part 1: General Principles for Design	See 4.3.2
[N2]	ISO 25119-1:2018 Tractors and machinery for agriculture and forestry — Safety-related parts of control systems — Part 1: General principles for design and development	See 4.3.2
[N3]	ISO 25119-2:2018 Tractors and machinery for agriculture and forestry — Safety-related parts of control systems — Part 2: Concept phase	See 4.3.2
[N4]	ISO 25119-3:2018 Checklist for Tractors and machinery for agriculture and for-estry — Safety-related parts of control systems — Part 3: Series development, hardware and software	See 4.3.2
[N5]	ISO 25119-4:2018 Tractors and machinery for agriculture and forestry — Safety-related parts of control systems — Part 4: Production, operation, modification and supporting processes	See 4.3.2
[N6]	EN 16590-1:2014 Tractors and machinery for agriculture and forestry – Safetyrelated parts of control systems – Part 1: General principles for design and development (ISO 251191:2010 modified)	See 4.3.2
[N7]	EN 16590-2:2014 Tractors and machinery for agriculture and forestry – Safety-related parts of control systems – Part 2: Concept phase (ISO 25119-2:2010 modified)	See 4.3.2
[N8]	EN 16590-3:2014 Tractors and machinery for agriculture and forestry – Safety-related parts of control systems – Part 3: Series development, hardware and software (ISO 25119-3:2010 modified)	See 4.3.2

Reference	Norm / Standard	Comment
[N9]	EN 16590-4:2014 Tractors and machinery for agriculture and forestry – Safety-related parts of control systems – Part 4: Production, operation, modification and supporting processes (ISO 25119-4:2010 modified)	See 4.3.2

1.6 Safety information and safety requirements

Safety information and safety requirements

To use the ***digsy***®fusion *S* the Erratasheet must be observed:

Name: 04-68493-011400_digsy-fusion-S_Erratasheet_E.pdf

Version: V01.14.00

The versions of the device description (devdesc), the system libraries and the CODESYS development environment must be used for the respective PLC version.

(see chapter 7.3.7 for versions of the standard application)

4888.02.XXX

PLC version V02.07.XX:

CODESYS Version:	3.5 SP5 Patch 0 (3.5.5.0)
digsy_fusion_S-Safety.devdesc.xml	Version: 3.5.5.2
digsy_fusion_S-Safety.devdesc_T2.xml	Version: 3.5.5.2
SCM-Ethernet.devdesc.xml	Version: 1.0.0.0
SIOM-Basic.devdesc.xml	Version: 1.0.0.0
DFS_Safe.compiled-library	Version: 1.4.0.0

PLC version V02.08.XX:

CODESYS Version:	3.5 SP5 Patch 0 (3.5.5.0)
digsy_fusion_S-Safety.devdesc.xml	Version: 3.5.5.2
digsy_fusion_S-Safety.devdesc_T2.xml	Version: 3.5.5.2
SCM-Ethernet.devdesc.xml	Version: 1.0.0.0
SIOM-Basic.devdesc.xml	Version: 1.0.0.0
DFS_Safe.compiled-library	Version: 1.4.0.0

PLC version V02.10.XX:

CODESYS Version:	3.5 SP5 Patch 0 (3.5.5.0)
digsy_fusion_S-Safety.devdesc.xml	Version: 3.5.5.3
digsy_fusion_S-Safety.devdesc_T2.xml	Version: 3.5.5.3
digsy_fusion_S-Safety.devdesc_T3.xml	Version: 3.5.5.3
SCM-Ethernet.devdesc.xml	Version: 1.0.0.0
SIOM-Basic.devdesc.xml	Version: 1.0.0.0

Safety information and safety requirements

DFS_Safe.compiled-library	Version: 1.4.0.0

PLC version V02.16.XX:

CODESYS Version:	3.5 SP5 Patch 0 (3.5.5.0)
digsy_fusion_S-Safety.devdesc.xml	Version: 3.5.5.4
digsy_fusion_S-Safety.devdesc_T2.xml	Version: 3.5.5.4
digsy_fusion_S-Safety.devdesc_T3.xml	Version: 3.5.5.4
SCM-Ethernet.devdesc.xml	Version: 1.1.0.0
SIOM-Basic.devdesc.xml	Version: 1.1.0.0
DFS_Safe.compiled-library	Version: 1.6.0.0

PLC version V03.26.XX:

CODESYS Version:	3.5 SP5 Patch 0 (3.5.5.0)
digsy_fusion_S-Safety.devdesc.xml	Version: 3.5.5.4
digsy_fusion_S-Safety.devdesc_T2.xml	Version: 3.5.5.4
digsy_fusion_S-Safety.devdesc_T3.xml	Version: 3.5.5.4
SCM-Ethernet.devdesc.xml	Version: 1.1.0.0
SIOM-Basic.devdesc.xml	Version: 1.1.0.0
DFS_Safe.compiled-library	Version: 1.6.0.0
Fusionx01.compiled-library	Version: 1.0.0.0
lib_digsyfusion_Util_Safety.compiled-library	Version: 1.4.0.0
Lib_Easy280_Safety.compiled-library	Version: 1.0.1.1

PLC version V03.30.XX:

CODESYS Version:	3.5 SP11 Patch 0 (3.5.11.0)
digsy_fusion_S-Safety.devdesc.xml	Version: 3.5.11.0
digsy_fusion_S-Safety.devdesc_T2.xml	Version: 3.5.11.0
digsy_fusion_S-Safety.devdesc_T3.xml	Version: 3.5.11.0
SCM-Ethernet.devdesc.xml	Version: 1.1.0.0
SIOM-Basic.devdesc.xml	Version: 1.1.0.0
DFS_Safe.compiled-library	Version: 1.8.0.0
Fusionx01.compiled-library	Version: 1.0.0.0
lib_digsyfusion_Util_Safety.compiled-library	Version: 1.4.0.0
Lib_Easy280_Safety.compiled-library	Version: 1.0.1.1

PLC version V03.35.XX:

CODESYS Version:	3.5 SP11 Patch 0 (3.5.11.0)
digsy_fusion_S-Safety.devdesc.xml	Version: 3.5.11.0
digsy_fusion_S-Safety.devdesc_T2.xml	Version: 3.5.11.0
digsy_fusion_S-Safety.devdesc_T3.xml	Version: 3.5.11.0
SCM-Ethernet.devdesc.xml	Version: 1.1.0.0

Safety information and safety requirements	
SIOM-Basic.devdesc.xml	Version: 1.1.0.0
GIOM-Basic.devdesc.xml	Version: 1.2.0.0
DFS_Safe.compiled-library	Version: 1.9.0.0
Fusionx01.compiled-library	Version: 1.0.0.0
lib_digsyfusion_Util_Safety.compiled-library	Version: 1.4.0.0
Lib_Easy280_Safety.compiled-library	Version: 1.0.1.1
PLC version V03.36.XX:	
CODESYS Version:	3.5 SP11 Patch 0 (3.5.11.0)
digsy_fusion_S-Safety.devdesc.xml	Version: 3.5.11.0
digsy_fusion_S-Safety.devdesc_T2.xml	Version: 3.5.11.0
digsy_fusion_S-Safety.devdesc_T3.xml	Version: 3.5.11.0
SCM-Ethernet.devdesc.xml	Version: 1.1.0.0
SIOM-Basic.devdesc.xml	Version: 1.1.0.0
GIOM-Basic.devdesc.xml	Version: 1.2.0.0
DFS_Safe.compiled-library	Version: 1.9.0.0
Fusionx01.compiled-library	Version: 1.0.0.0
lib_digsyfusion_Util_Safety.compiled-library	Version: 1.4.0.0
Lib_Easy280_Safety.compiled-library	Version: 1.0.1.1
PLC version V03.37.XX:	
CODESYS Version:	3.5 SP11 Patch 0 (3.5.11.0)
digsy_fusion_S-Safety.devdesc.xml	Version: 3.5.11.0
digsy_fusion_S-Safety.devdesc_T2.xml	Version: 3.5.11.0
digsy_fusion_S-Safety.devdesc_T3.xml	Version: 3.5.11.0
SCM-Ethernet.devdesc.xml	Version: 1.1.0.0
SIOM-Basic.devdesc.xml	Version: 1.1.0.0
GIOM-Basic.devdesc.xml	Version: 1.2.0.0
DFS_Safe.compiled-library	Version: 1.9.0.0
Fusionx01.compiled-library	Version: 1.0.0.0
lib_digsyfusion_Util_Safety.compiled-library	Version: 1.4.0.0
Lib_Easy280_Safety.compiled-library	Version: 1.0.1.1
4888.03.XXX	
PLC version V01.00.XX:	
CODESYS Version:	3.5 SP11 Patch 0 (3.5.11.0)
digsy_fusion_S-P-Safety.devdesc.xml	Version: 3.5.11.1
SCM-Ethernet_SL.devdesc.xml	Version: 1.1.0.0
SIOM-Basic.devdesc.xml	Version: 1.1.0.0
DFS-P_Safe.compiled-library	Version: 1.1.0.0
Fusionx01.compiled-library	Version: 1.0.0.0

Safety information and safety requirements	
lib_digsyfusion_Util_Safety.compiled-library	Version: 1.4.0.0
Lib_Easy280_Safety.compiled-library	Version: 1.0.1.1
PLC version V01.05.XX:	
CODESYS Version:	3.5 SP11 Patch 0 (3.5.11.0)
digsy_fusion_S-P-Safety.devdesc.xml	Version: 3.5.11.1
SCM_Eth_SL.devdesc.xml	Version: 1.1.0.0
SIOM-Basic.devdesc.xml	Version: 1.1.0.0
GIOM-Basic.devdesc.xml	Version: 1.2.0.0
DFS-P_Safe.compiled-library	Version: 1.2.0.0
Fusionx01.compiled-library	Version: 1.0.0.0
lib_digsyfusion_Util_Safety.compiled-library	Version: 1.4.0.0
Lib_Easy280_Safety.compiled-library	Version: 1.0.1.1
PLC version V01.06.XX:	
CODESYS Version:	3.5 SP11 Patch 0 (3.5.11.0)
digsy_fusion_S-P-Safety.devdesc.xml	Version: 3.5.11.1
SCM_Eth_SL.devdesc.xml	Version: 1.1.0.0
SIOM-Basic.devdesc.xml	Version: 1.1.0.0
GIOM-Basic.devdesc.xml	Version: 1.2.0.0
DFS-P_Safe.compiled-library	Version: 1.2.0.0
Fusionx01.compiled-library	Version: 1.0.0.0
lib_digsyfusion_Util_Safety.compiled-library	Version: 1.4.0.0
Lib_Easy280_Safety.compiled-library	Version: 1.0.1.1
PLC version V01.09.XX:	
CODESYS Version:	3.5 SP11 Patch 0 (3.5.11.0)
digsy_fusion_S-P-Safety.devdesc.xml	Version: 3.5.11.1
SCM_Eth_SL.devdesc.xml	Version: 1.1.0.0
SIOM-Basic.devdesc.xml	Version: 1.1.0.0
GIOM-Basic.devdesc.xml	Version: 1.2.0.0
DFS-P_Safe.compiled-library	Version: 1.2.0.0
Fusionx01.compiled-library	Version: 1.0.0.0
lib_digsyfusion_Util_Safety.compiled-library	Version: 1.4.0.0
Lib_Easy280_Safety.compiled-library	Version: 1.0.1.1
PLC version V01.10.XX:	
CODESYS Version:	3.5 SP11 Patch 0 (3.5.11.0)
digsy_fusion_S-P-Safety.devdesc.xml	Version: 3.5.11.1
SCM_Eth_SL.devdesc.xml	Version: 1.1.0.0
SIOM-Basic.devdesc.xml	Version: 1.1.0.0

Safety information and safety requirements	
GIOM-Basic.devdesc.xml	Version: 1.2.0.0
DFS-P_Safe.compiled-library	Version: 1.2.0.0
Fusionx01.compiled-library	Version: 1.0.0.0
lib_digsyfusion_Util_Safety.compiled-library	Version: 1.4.0.0
Lib_Easy280_Safety.compiled-library	Version: 1.0.1.1
4888.04.XXX	
PLC version V01.08.XX:	
CODESYS Version:	3.5 SP11 Patch 0 (3.5.11.0)
digsy_fusion_S-P-Safety.devdesc.xml	Version: 3.5.11.1
SCM_Eth_SL.devdesc.xml	Version: 1.1.0.0
SIOM-Basic.devdesc.xml	Version: 1.1.0.0
DFS-P_Safe.compiled-library	Version: 1.3.0.0
Fusionx01.compiled-library	Version: 1.0.0.0
lib_digsyfusion_Util_Safety.compiled-library	Version: 1.4.0.0
Lib_Easy280_Safety.compiled-library	Version: 1.0.1.1
PLC version V01.09.XX:	
CODESYS Version:	3.5 SP11 Patch 0 (3.5.11.0)
digsy_fusion_S-P-Safety.devdesc.xml	Version: 3.5.11.1
SCM_Eth_SL.devdesc.xml	Version: 1.1.0.0
SIOM-Basic.devdesc.xml	Version: 1.1.0.0
DFS-P_Safe.compiled-library	Version: 1.3.0.0
Fusionx01.compiled-library	Version: 1.0.0.0
lib_digsyfusion_Util_Safety.compiled-library	Version: 1.4.0.0
Lib_Easy280_Safety.compiled-library	Version: 1.0.1.1
PLC version V01.10.XX:	
CODESYS Version:	3.5 SP11 Patch 0 (3.5.11.0)
digsy_fusion_S-P-Safety.devdesc.xml	Version: 3.5.11.1
SCM_Eth_SL.devdesc.xml	Version: 1.1.0.0
SIOM-Basic.devdesc.xml	Version: 1.1.0.0
DFS-P_Safe.compiled-library	Version: 1.3.0.0
Fusionx01.compiled-library	Version: 1.0.0.0
lib_digsyfusion_Util_Safety.compiled-library	Version: 1.4.0.0
Lib_Easy280_Safety.compiled-library	Version: 1.0.1.1

2 Basic Safety Notes

2.1 General notes – warranty – liability

This manual contains the information required in order to comply with the intended use (as defined in Chapter 2.6) of the product described. It addresses technically qualified personnel with special training or having relevant knowledge in the field of instrumentation and control engineering, hereinafter referred to as automation engineering, and safety requirements.

The knowledge and technically correct implementation of the safety notes and warnings contained in this manual are a prerequisite for safe installation and commissioning, as well as for the safe operation and maintenance of the product described. Only qualified personnel as defined in Chapter 2.2 possess the expertise required to correctly interpret and implement the general safety notes and warnings given in this document under specific conditions.

For reasons of clarity, the manual does not include all details for all the designs of the product described and cannot consider every conceivable arrangement, operation or maintenance case. Should you require further information or if special problems are not sufficiently covered in this document, please request further information from INTER CONTROL (see address on Page 1) or its representatives.

Please note that the contents of this product documentation are not part of an earlier or existing agreement, commitment, or of a legal relationship or an amendment to this. All obligations of INTER CONTROL are laid down in the respective purchase contract, which also includes the complete and exclusively valid warranty regulations. These contractual warranty regulations are neither extended nor limited by the text of this document.

Note

INTER CONTROL is not liable for damage resulting from the improper use of the components supplied or from failure to observe the instructions given in this manual or in other documents relating to this product.

2.2 Qualified personnel

Unqualified interventions in the device/machine or non-compliance with the warnings specified in this manual or attached to the device/machine can result in severe physical injury or damage to property. Therefore, only properly **qualified personnel** may intervene in a machine equipped with this device. Any intervention is not permitted. Only INTER CONTROL staff is allowed to perform interventions.

Qualified personnel in the sense of the safety-related notes in this manual or on the product itself are:

- Persons who, as either project engineering or development personnel, are familiar with the safety concepts of automation engineering in general and particularly with the safety requirements regarding the use of this product, or
- Persons who, as operating personnel, are trained in the handling of automation systems and are familiar with the operation-related contents of this manual, or
- Persons who, as commissioning and service personnel, have special training enabling them to repair such automation systems or have the authorization to put such systems into operation and to ground and label circuits and devices/machines in compliance with applicable safety standards.

	HAZARD
	Abuse by unauthorized persons / non-qualified personnel Compromising the safety function 1. The operator must ensure that all ***digsy***®fusion *S* interfaces are properly protected against access (abuse) by unauthorized persons / non-qualified personnel.

2.3 Hazard notes

The following notes are, on the one hand, intended for your personal safety and, on the other hand, to prevent damage to the product or connected devices described.

Safety notes and warnings pertaining to the life and limb of the users or the maintenance personnel or for the prevention of material damage are emphasized in this manual by the signal terms defined below. In regard to this manual and the notes on the products themselves, the various terms have the following meaning:

	HAZARD
	Type and source of the hazard 'Consequences' 1st 'Remedy' 2nd 'Remedy'

HAZARD means that failure to observe the relevant precautions and corrective actions **will** result in death, severe injury or considerable material damage.

	WARNING
	Type and source of the hazard 'Consequences' 1st 'Remedy' 2nd 'Remedy'

WARNING means that failure to observe the relevant precautions and corrective actions can result in death, severe injury or considerable material damage.

	CAUTION
	Type and source of the hazard 'Consequences' 1st 'Remedy' 2nd 'Remedy'

CAUTION means that failure to observe the relevant precautions and corrective actions can result in slight injuries or material damage.

	Note
	'Title' ‚Standard text' 1. 'Enumeration' 2. 'Remedy'

NOTE indicates important information about the product, handling of the product or the relevant section of the manual requiring special attention.

2.4 Safety instructions / Safety information

Safety information and safety requirements	
Text	Paragraph

Safety information includes information to be observed by the user in order to ensure the safety of control systems in accordance with EN ISO 13849-1:2015 ([N1]) PLd (Cat 2 or 3). This paragraph is unambiguous and can be found in chapter 10.14.

2.5 Safe State

	WARNING
	To reduce the risk of a fault, the PLC control changes in safe state at an active or passive error. The safe state is achieved when all outputs are switched deenergized. To occupy the safe state, the plc control changes the operating state to **OPMODE_FAILSAFE_IO** or **OPMODE_FAILSAFE_STOP** (see chapter 6.1).

2.6 Intended use

- The device/system or the respective system component may only be used in applications in which the technical characteristics are not overextended and only in conjunction with devices or components (from external suppliers) recommended or approved by INTER CONTROL.
- The product described was developed, manufactured, tested, and documented in compliance with the relevant safety standards (EN ISO 13849-1:2015 ([N1])). Therefore, the operation of the product does not normally lead to physical or material hazards when the handling instructions and safety notes for project planning, assembly/installation, normal use and maintenance/repair are observed.

	WARNING
	Opening/removing the housing of all devices described For all devices described, opening/removing the housing enables access to the device electronics. The electronics may be damaged by the effect of ESD, for example, in such a way that failures or hazards to safety may occur. 1. The user may not open / remove the housing of any of the devices described! 2. Only INTER CONTROL staff is allowed to perform interventions in these devices!

WARNING

Improper use

Improper use of all devices described in this document can lead to outage and safety risks.

1. Only qualified personnel may work on the devices described, within the scope of machine development, machine production, test mode, service and all other phases of handling
2. Qualified personnel must be familiar with the general and specific sources of hazards and all development rules and maintenance activities, particularly those mentioned in this document!
3. Prerequisites for the fault-free and safe operation of all devices described are proper transportation, proper storage, and proper setup and installation, as well as careful operation and maintenance.

2.7 General notes regarding the project engineering and installation of the devices

2.7.1 Project engineering of automation equipment

Since the product is generally used as a part of larger systems or equipment, these notes are intended to provide a guideline for the safe integration of the product in its environment.

In this context, special attention has to be paid to the following facts:

WARNING

Project engineering of automation equipment

Although the project engineering of automation equipment achieves a maximum level of design safety (e.g. by multi-channel design), improper handling or non-compliance with the specifications may, under certain circumstances, interfere with the precautions taken to avoid dangerous faults or create additional hazard sources!

1. All the instructions included in this document, and in particular the safety notes, must be strictly followed!

For the installation and commissioning of the devices described, the general notes and warnings below must be followed, depending on the individual situation:

WARNING

Safety and accident prevention regulations

Failure to observe general and/or specific safety and accident prevention regulations may lead to dangerous failures and safety risks.

1. All application-related general and specific safety and accident prevention regulations must be followed!

2.7.2 Stationary operation from AC networks

The devices described here require a DC source for their operation. For stationary operation from an AC network, only power supply units having certain properties may be used:

CAUTION

Stationary operation from AC networks

The devices described are specified for extra-low voltage (DC) operation. They provide a conductive connection (functional grounding) between their housings and internal ground. If stationary operation from an AC network utilizes a power supply unit (e.g. with 230 V~) to generate the extra-low voltage required, make sure to consider the following:

1. Only PELV-type power supply units may be used (see EN 50178)!
2. For 12 V or 24V power supplies, care must be taken to ensure the safe electrical isolation of the extra-low voltage. Only use power supply units manufactured in accordance with IEC 364-4-41, or HD 384.04.41 (VDE 0100 Part 410)!
3. The control system must be supplied via a mains disconnector or a fuse!
4. For the connection to the protective conductor the relevant regulations must be considered, and in particular:
   - DIN VDE 0100-410:2007-06, section 414.4.1, General installation in buildings, and
   - EN 60204-1:2006 (VDE 0113-1), section 6.4.1, Electrical equipment of machines!
5. For devices/machines with a permanently attached, non-detachable connection cable and without an all-pole mains disconnector, the socket outlet with ground contact must be mounted near the machine and must be easily accessible!
6. Prior to commissioning check if the rated voltage range set corresponds to the local power system voltage!
7. Fluctuations or deviations from the rated value of the mains voltage may not exceed the tolerance limits specified in the Technical Data of the PELV power supply units; otherwise malfunctions and hazardous conditions in the electric and electronic sub-assemblies/ devices cannot be excluded!
8. Precautions must be taken so that an interrupted program can be continued properly after a system voltage dip or a power failure. Not even short-term dangerous operating conditions may occur. In case of doubt, an emergency stop must be performed!
9. Emergency stop devices in accordance with EN 60204/IEC 204 (VDE 0113) must remain effective in all modes of operation of the automation system. Unlocking the emergency stop devices may not lead to an uncontrolled or undefined restart.

Notice

Installation instructions

The following must be considered for installation in systems:

1. Connecting cables and signal lines must be installed such that inductive and capacitive interference does not adversely affect the automation functions!
2. Automation system equipment and its control elements must be mounted so as to be sufficiently protected against unintentional operation or actuation.
3. Interface cables may be plugged / unplugged without power supply disconnection only.
4. Interface cables can only be plugged / unplugged while the power supply is switched on and if the following requirements are fulfilled:
   a. The interface type must be designed for this application (e.g. USB)!
   b. The cable must be shielded and the shielding must be connected to a conductive metal cover!
   c. Make sure that equipotential bonding is provided by connecting the PE carrying connector components before the cables are plugged!
5. To avoid undefined conditions in the automation system caused by a cable or wire break in the signal lines, appropriate safety measures have to be taken on the hardware and software side of the I/O coupling!

Note

Declaration of conformity

Prior to use in systems and prior to their official commissioning, particularly for systems used in Europe, conformity with the applicable legal regulations must be reviewed and confirmed, if necessary!

2.8 Active and passive errors of an automation system

- Depending on the individual task of an electronic automation facility, both active and passive errors can be dangerous. Active errors occurring in a drive controller, for example, are usually dangerous because they can lead to an unauthorized switch-on of the drive. In the case of a message function, however, a passive error can prevent the indication of a dangerous operating condition.
- This distinction of potential errors and their job-related assignment to dangerous and harmless errors is essential for all safety considerations with the product delivered.

WARNING

Dangerous faults

Dangerous faults occurring in the automation system can cause material damage and even personal injury.

1. Additional precautions have to be taken or equipment has to be provided that ensures or enforces a safe operating state even in the event of a dangerous fault!

2.9 How to proceed in the event of maintenance or repair

If measurements or testing are necessary on the active device, the requirements instructions for implementation set out in the Accident Prevention Regulations BGV-A3 must be observed. Suitable electrical tools must be used.

	WARNING
	How to proceed in the event of maintenance and repair In the event of maintenance and repair make sure that the safety properties of the machine are not disabled and that the personnel is not exposed to risk while executing maintenance and repair work. This can be ensured with the following precautions: 1. Only qualified personnel as defined in section 2.2 may carry out maintenance and repair work! 2. Unauthorized opening of devices or improperly performed repair work can lead to serious safety hazards! 3. Only use OEM parts or parts included in the spare parts list to replace parts or components! 4. Always open the mains disconnector switch or unplug the power plug in the case of line-powered devices! 5. Input/output signal cables may be plugged or unplugged under de-energized conditions only! 6. Interface cables can only be plugged / unplugged while the power supply is switched on if the following requirements are fulfilled: a. The interface type must be designed for this application (e.g. USB)! b. The cable must be shielded and the shielding must be connected to a conductive metal cover! c. Make sure that equipotential bonding is provided by connecting the PE carrying connector components before the cables are plugged! 7. Fuses must be replaced only by fuse types which comply with the specifications of the Technical Data or the maintenance manual of the device / machine!

2.10 How to proceed when carrying out welding work

The following regulations have to be observed for welding work on machinery and installations:

- Accident prevention regulations BGV-A3 (electrical systems and equipment)
- Accident prevention regulations VBG 15 (welding and similar procedures), in particular §43 (Connecting and disconnecting welding current circuits).

	Notice
	Welding work Only qualified personnel may carry out welding jobs and, in particular, electrical welding on machinery and systems. To protect electronic equipment, proceed as follows with electrical welding: 1. Switch off the power supply (in vehicles: switch off the ignition)! 2. Disconnect the machine from the power line, or disconnect the starter battery in vehicles! First disconnect the negative pole and then the positive pole. An equivalent method for safe disconnection of the electric circuits can also be used. 3. Connecting the ground terminal on electrical welding machines: The ground terminal must be directly connected to the part to be welded. The ground terminal must not be connected above rotating assembly components. Avoid stray welding currents! 4. Welding cables may not be laid parallel to electric cables! 5. Do not touch electronic components, and in particular any housings of electronic control units and their components, as well as power cables, with the welding electrode! If one of the specified measures cannot be met, unplug the connectors of all the electronic devices **prior to starting any welding job**! In the case of **plasma welding** also make sure that the connectors are unplugged from the electronic devices **prior to starting any welding job**! Contact between the welding electrode of an electric welder and a non-insulated cable (e.g. where the insulation has been destroyed) will destroy the electronics immediately!

2.11 How to proceed in the case of quick battery charging

	CAUTION
	Quick battery charging Quick charging a vehicle battery can cause damage to the electronics if the battery is not disconnected. 1. Quick charging of batteries which e.g. supply the control systems and control components may only be carried out with the battery disconnected from the terminals!

2.12 How to jump start an internal combustion engine

	CAUTION
	Jump starts for internal combustion engines Jump starts for internal combustion engines can cause damage to the electronics if the battery is disconnected. 1. The starter battery must be connected to jump start an internal combustion engine! 2. The jump start may under no circumstances be made with the quick charging device!

2.13 Possible consequential damage when cutting through cable harnesses

Welding flames, welding arcs, cutting tools or other mechanical severing can cut through cable harnesses. Both cable breaks and short circuits can occur as a result.

The sections below consider only damage in the cable harness that is either caused by a short circuit between supply cables carrying the operating voltage and the various interface cables or by a short-circuit between the supply cables carrying the operation voltage or interface cables and GND.

2.13.1 Short circuit between power supply cables

Fused/protected power supply units (by fuse, automatic circuit-breakers, etc.) and cables are protected against short circuits.

	⚠	CAUTION
⚠		**Unprotected power supply cables** Power supply units or supply cables with improper protection or without any protection can be destroyed if a short circuit occurs! 1. Make sure that the power supply units and power supply cables are fused!

2.13.2 Short circuits on analog inputs

Analog voltage inputs have a functional range from 0 to 10V or from 0 to 32V. Analog current inputs have a functional range from 0 to 20mA.

Normally a short circuit between GND and the analog inputs has a negative impact on the proper functioning of the inputs and the entire system. However, the analog inputs are not destroyed.

	⚠	CAUTION
⚠		**Short circuit between operating voltage and analog inputs** A short circuit between supply cables carrying the operating voltage and analog inputs, in particular current inputs, may destroy the electronics! 1. Make sure that supply cables carrying the operating voltage and wires connected to analog inputs are laid separately!

2.13.3 Short circuits on communication interfaces

Normally a short circuit between GND and the inputs/outputs of communication interfaces has a negative impact on the proper functioning of the inputs and the entire system. However, the communication interfaces are not destroyed.

CAUTION

Short circuit between operating voltage and wires connected to communication interfaces

A short circuit between supply cables carrying the operating voltage and connections to

RS232 interfaces,
CAN interfaces,
USB interfaces

can lead to the destruction of the electronics!

1. Make sure that supply cables carrying the operating voltage and wires connected to communication interfaces are laid separately!

2.13.4 Short circuit on shaft encoder inputs

Shaft encoder inputs often function at 5V or 12V levels. A short circuit between GND and the shaft encoder inputs has a negative impact on the proper functioning of the inputs and the entire system. However, the shaft encoder inputs are not destroyed.

CAUTION

Short circuit between operating voltage and shaft encoder inputs

A short circuit between supply cables carrying the operating voltage and the shaft encoder inputs may destroy the electronics!

1. Make sure that supply cables carrying the operating voltage and wires connected to shaft encoder inputs are laid separately!

2.13.5 Short circuit on counter inputs

Counter inputs often operate at 5V, 12V or 24V levels. A short circuit between GND and counter inputs has a negative impact on the proper functioning of the inputs and the entire system. However, the counter inputs are not destroyed.

Normally, a short circuit between supply cables carrying the operating voltage and counter inputs with levels designed on the basis of the operating voltage has a negative impact on the proper functioning of the inputs and the entire system. However, the counter inputs are not destroyed.

CAUTION

Short circuit between operating voltage and wires connected to counter inputs

A short circuit between supply cables carrying the operating voltage and counter inputs with a voltage level designed for less than the operating voltage (e.g. 12V level at 24V operating voltage) may destroy the electronics!

1. Make sure that supply cables carrying the operating voltage and wires connected to counter inputs are laid separately!

2.13.6 Short circuit on digital inputs

A short circuit between GND and digital inputs has a negative impact on the proper functioning of the inputs and the entire system. However, the digital inputs are not destroyed. A short circuit between supply cables carrying the operating voltage and digital inputs designed for operating voltage has a negative impact on the proper functioning of the digital inputs. However, the digital inputs are not destroyed.

CAUTION

Short circuit between operating voltage and wires connected to digital inputs

A short circuit between supply cables carrying the operating voltage and digital inputs with a voltage level designed for less than the operating voltage (e.g. 12V level at 24V operating voltage) may destroy the electronics!

1. Make sure that supply cables carrying the operating voltage and wires connected to digital inputs are laid separately!

2.13.7 Short circuit on digital outputs

Digital outputs are protected against short circuits to ground. A short circuit between GND wires and digital outputs has a negative impact on the proper functioning of the outputs. However, the digital outputs are not destroyed.

CAUTION

Short circuit between operating voltage and wires connected to digital outputs

A short circuit between supply cables carrying the operating voltage and digital outputs can
- destroy the electronics in exceptional cases
- in principle be a dangerous event!

1. Make sure that supply cables carrying the operating voltage and wires connected to digital outputs are laid separately!

CAUTION

Short circuit between GND and wires connected to digital outputs

A sustained short circuit between GND and digital outputs may destroy the electronics!

1. Digital outputs must be continuously checked for short circuits using the methods provided by the corresponding controller type. After a short circuit has been identified the outputs affected must be switched off immediately. In exceptional cases a sustained short circuit can destroy electronic components and lead to hazardous situations!

2.13.8 Short circuit on reference voltage outputs

Reference voltage outputs are normally protected against short circuit to ground by control circuit components. A short circuit between GND wires and reference voltage outputs has a negative impact on the proper functioning of outputs. However, the digital outputs are not destroyed.

CAUTION

Short circuit between operating voltage and wires connected to reference voltage outputs

A short circuit between supply cables carrying the operating voltage and reference voltage outputs may destroy the electronics in exceptional cases!

1. Make sure that supply cables carrying the operating voltage and wires connected to reference voltage outputs are laid separately!

	⚠ CAUTION
⚠	**Short circuit between GND and wires connected to reference voltage outputs** A sustained short circuit between GND and the reference voltage outputs may destroy the electronics! 1. Reference voltage outputs must be continuously checked for short circuits using the methods provided by the corresponding controller type. After a short circuit has been identified the outputs affected must be switched off immediately. In exceptional cases a sustained short circuit can destroy electronic components and lead to hazardous situations!

2.13.9 Short circuit on PWM outputs

PWM outputs are normally protected against short circuits to ground via control circuit components. A short circuit between GND wires and PWM outputs has a negative impact on the proper functioning of the outputs. However, the digital outputs are not destroyed.

	⚠ CAUTION
⚠	**Short circuit between operating voltage and wires connected to PWM outputs** A short circuit between supply cables carrying the operating voltage and PWM outputs can - destroy the electronics in exceptional cases - in principle be a dangerous event! 1. Make sure that supply cables carrying the operating voltage and wires connected to PWM outputs are laid separately!

	⚠ CAUTION
⚠	**Short circuit between GND and lines to PWM outputs** A sustained short circuit between GND and PWM outputs may destroy the electronics! 1. PWM outputs must be continuously checked for short circuit using the methods provided by the corresponding controller type. After a short circuit has been identified the outputs affected must be switched off immediately. In exceptional cases a sustained short circuit can destroy electronic components and lead to hazardous situations!

2.14 System diagnosis

The ***digsy***® family provides a wide range of diagnostic options which can support you identifying any malfunctions of the control system and its components as well as problems in the related environment in time and bring the machine to a safe condition. The products of the ***digsy***® family provide various tools to display status and error conditions which can be evaluated directly in the process.

A detailed description of these tools can be found in the section dedicated to the respective component.

	Note
	System diagnosis and error handling System diagnosis and error handling are necessary to ensure the safety of people and machines. 1. The component-related sections provide a description of the methods and tools available for system diagnosis and error handling! 2. The methods and tools available for system diagnostics and error handling must be used!

3 System Description

3.1 Introduction to the product

The ***digsy***®fusion S controller was developed for the following applications and satisfies the environmental conditions set out below:

- Mobile work machines (e.g. cranes, road finishers, elevated work platforms, etc.)
- Agricultural machines
- Other outdoor applications.

digsy®fusion S complies with the following requirements:

- 32-bit technology
- Modular design
- Designed as a complex central control system or as local function node
- Free programming according to IEC 61131-3
- Functional safety: PL d Category 3 / Category 2 (see 4.3.3) in accordance with EN ISO 13849-1:2015 ([N1])
- Protection class IP6K7/ IP6K9K in accordance with DIN 40050-9,
- Operating temperature range (-40°C to 80 C in accordance with DIN EN 60068-2-2), satisfying the requirements for mobile work machines.

digsy®fusion S is designed as a modular system to which various expansion modules can be connected. The combination of these modules in the correct quality and number allows customized control systems with a large range of product variants.

digsy®fusion S is freely programmable in accordance with IEC 61131-3 using the CODESYS development system (see chapter 1.6 for version to be used)from 3S-Smart Software Solutions. This feature and this programming tool allow the machine manufacturer to directly implement customized applications.

digsy®fusion S complies with the requirements in respect of protection class and operating temperature range for outdoor applications, in particular for mobile work machines.

digsy®fusion S meets the requirements set out in PL d Category 3 / Category 2 (see 4.3.3) (EN ISO 13849-1:2015 ([N1])) – and therefore the requirements of the European Machinery Directive 2006/42/EC – for the functional safety of mobile machinery.

	Process Safety Time
	For more detailed information on the Process Safety Time refer to Chapter 4.4.

3.2 System architecture

There are 5 variants of ***digsy***®fusion S:

- ***digsy***®fusion S (T1)
- ***digsy***®fusion S T2
- ***digsy***®fusion S T3
- ***digsy***®fusion S-P
- ***digsy***®fusion S-PL

3.2.1 Difference between *digsy*®fusion S T1, *digsy*®fusion S T2 and *digsy*®fusion S T3

digsy®fusion S is available in three variants. These three variants differ with regard to their memory partitioning. Variant T2 has a higher performance on the standard side than variant T1. When compared to variant T2, variant T1 offers a larger storage capacity for the application programs (see Table 3-1). In variant T3 doesn't exist a standard application. Therefore, variant T3 has more memory for the safe application avaible than variant T1 and variant T2.

	digsy®fusion **S T1**	***digsy***®fusion **S T2**	***digsy***®fusion **S T3**
Standard Application	2048 kB	896 kB	-
Safety Application	896 kB	384 kB	1536 kB

Table 3-1: Memory partitioning

Safety information and safety requirements	
In variant T3 the memory for the safe application is 1.5 MB	§8003

3.2.2 *digsy*®fusion S-P

The ***digsy***®fusion S-P is a variant of the ***digsy***®fusion S. In terms of storage, the SCM-SL module of the DFS-P is equivalent to variant T3.

The safety application is executed on the SCM SL module.

The standard application is executed on the GCM-P module.

Difference *digsy*®fusion S-P

Difference between the ***digsy***®fusion S-P and the variants T1/T2/T3

If the following chapters talk about ***digsy***®fusion S, the ***digsy***®fusion S-P is included. Should there be a difference to the ***digsy***®fusion S (T1 / T2 / T3) with the ***digsy***®fusion S-P, this is indicated separately.

3.2.3 *digsy*®fusion S-PL

The ***digsy***®fusion S-PL is a variant of the ***digsy***®fusion S. In terms of storage, the SCM-SL module of the DFS-PL is equivalent to variant T3.

The safety application is executed on the SCM SL module. The standard application is executed on the GCM-PL module.

The difference between the ***digsy***®$_{\text{fusion}}$ S-P and the ***digsy***®$_{\text{fusion}}$ S-PL (where the "L" stands for Linux) is that there is no CODESYS runtime system on the GCM-P, but the Linux operating system. The standard application(s) are executed as userspace programs on Linux.

	Difference digsy®fusion S-PL
	Difference between the digsyfusionS-PL and the variants T1/T2/T3 If the following chapters talk about ***digsy***®$_{\text{fusion}}$ S, the ***digsy***®$_{\text{fusion}}$ S-PL is included. Should there be a difference to the ***digsy***®$_{\text{fusion}}$ S (T1 / T2 / T3) with the ***digsy***®$_{\text{fusion}}$ S-PL, this is indicated separately.

3.2.4 *digsy*®$_{\text{fusion}}$ components

The following modules are available:

- SCM Ethernet   Safety Controller Module (only variants T1/T2/T3)
- SCM Eth SL   Safety Controller Module (only variant ***digsy***®$_{\text{fusion}}$ S-P(L)*)*
- SIOM   Safety I/O Module

The SCM Ethernet, SCM Eth SL and SIOM modules can be used for safety applications according to PL d Cat 3 or Cat 2 (see 4.3.3).

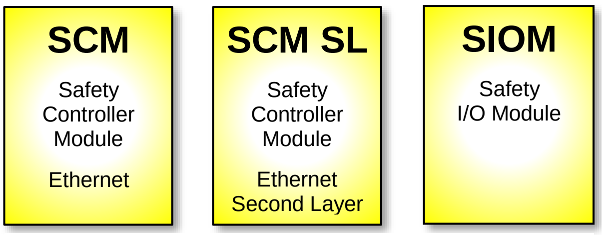


Figure 3-1: Safe modules

- GCM-P   General Controller Module Performance (only variant ***digsy***®$_{\text{fusion}}$ S-P*)*
- GCM-PL   General Controller Module Performance (only variant ***digsy***®$_{\text{fusion}}$ S-PL*)*
- GIOM   General I/O Module

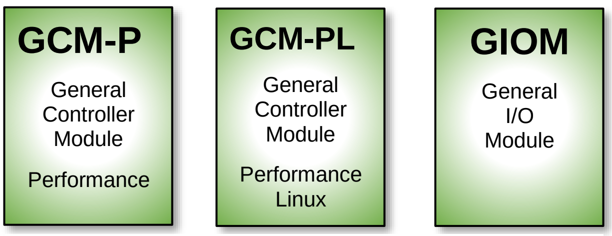


Figure 3-2: Standard modules

3.2.4.1 Safety Controller Module – SCM (SL)

The *Safety Controller Module SCM* is at the heart of the modular system.

	Note
	Functional safety Provided that the defined safety conditions are met, a Safety Controller Module (SCM) complies with the following safety requirements: 1. Performance Level PL d, Category 3 / Category 2 (see 4.3.3) in accordance with EN ISO 13849-1:2015 ([N1]).

The *SCM* is freely programmable via CODESYS 3.5 (exact version see chapter 1.6).

The *SCM* includes a 32-bit microcontroller with zero-voltage-protected memory and random access memory, thus ensuring free programming of conventional and even demanding customer applications.

The dual core type microcontroller is designed particularly for safety-critical applications (Dual Core Controller operated in the Lockstep Mode).

The *SCM* includes e.g. the following functional components:

- 32-bit Safety Controller
- 2-channel µC in the Lockstep Mode
- Floating Point Unit (FPU)
- On-board LED state indicator
- Internal bus connection for system communication
- PL d, category 2, or category 3 in accordance with EN ISO 13849-1:2015 ([N1])
- Application programming via CODESYS V3.5 SIL 2 from the company 3S
- Functionally safe runtime system CODESYS V3.5 SIL 2 (SRTS) for Safe Application program
- Runtime system CODESYS V3.5 SIL 2 (SRTS) for the Standard Application program (independent of SRTS)
- Self-diagnosis in accordance with safety requirements
- 2x RS232 interface (RS232-1 = Diagnosis interface, RS232-2 = freely programmable)
- 4x CAN interface
- 1x USB interface
- 1x Ethernet interface (option, diagnosis, and service interface)
- Various types of inputs/outputs (see Table 5-3)
- On-board power supply

Memory	Size	Usage
On-Chip Flash-Memory	3MB	FW (safe, standard) safe application (size see 3.2.1) standard application (variant T2, size see 3.2.1)
On-Chip RAM- Memory	256kB	safe RAM
On-Board Flash- Memory (parallel)	32MB	internal filesystem 16MB standard application (variant T1, size see 3.2.1) Bootloader
On-Board SDRAM	16MB	shared memory 96kB for user 2 x 8kB avaiable (see chapter 8.1.13 / 8.2.7) Standard RAM
On-Board FRAM- Memory	32kB	retain data 16kB (2 x 8kB) (see chapter 7.5.7.5) systemlogbook 14kB
On-Board Flash- Memory (serial)	64MB	userlogbook 64MB

Table 3-2: SCM (SL) memory

For a detailed description of the *SCM* properties refer to Chapter 5.1.

3.2.4.2 Safety I/O Module – SIOM

The *Safety I/O Module (SIOM)* is a component for the extension of the number of functionally safe inputs/outputs within a modular system. In order to be able to use the *SIOM*, a *SCM (SL)* module must exist which controls the inputs/outputs available on the *SIOM*.

Note

Functional safety

Provided that the defined safety conditions are met, a Safety I/O-Module SIOM complies with the following safety requirements:

1. Performance Level PL d, Category 3 / Category 2 (see 4.3.3) in accordance with EN ISO 13849-1:2015 ([N1]).

The *SIOM* is not freely programmable.

The *SIOM* includes e.g. the following functional components:

- 32-bit µC, 2-channel, in the Lockstep Mode,
- 3MB internal flash memory,
- 256kB internal RAM memory,
- Internal system buses,
- Self-diagnosis options,
- Up to 48 digital / analog inputs and outputs
- Reference voltage and reference current outputs.

For a detailed description of the *SIOM* properties refer to chapter 5.3.

3.2.4.3 GCM-P

The General Controller Module Performance GCM-P is the module on which the Digsy Fusion S-P runs the standard application. It requires an existing SCM SL.

Note

Functional safety

A General Controller Module Performance GCM-P does not meet any safety requirements.

The GCM-P is freely programmable.

The GCM-P includes i.a. the following functional components:

- 32-Bit-Prozessor (400 MHz)
- Memory - Byte Order: Big-Endian
- Floating Point Unit FPU
- 4x CAN interfaces
- 1x RS232 interface
- 1x Ethernet interface (10/100 MBit/s) to the processor
- 1x Ethernet switch (2 ports outwards)
- RTC realtime clock (approx. 1h power reserve, external buffering possible)
- Temperature sensor

- Voltage monitoring
- Watchdog

Memory	Size	Usage
SDRAM	32MB	Firmware Application
Flash	32MB	Internal filesystem 26MB Bootloader Firmware
FRAM	32kB	Retain-Data User-Data
Compact Flash Card	1GB / 4GB	filesystem

Table 3-3: GCM-P memory

A detailed description of the properties of the GCM-P can be found in the manual [MAN-GCM-P]

3.2.4.4 GCM-P(L)

The General Controller Module Performance Linux GCM-P(L) is the module on which the digsy fusion S-P(L) runs the standard application on a Linux operating system. It requires an existing SCM S(L).

	Note
	Functional safety A General Controller Module Performance GCM-PL does not meet any safety requirements.

The GCM-P(L) is freely programmable.

The GCM-P(L) includes u. a. the following functional components:

- 32-Bit-Prozessor (400 MHz)
- Memory - Byte Order: Big-Endian
- Floating Point Unit FPU
- 4x CAN interfaces
- 1x RS232 interface
- 1x Ethernet interface (10/100 MBit/s) to the processor
- 1x Ethernet switch (2 ports outwards)
- RTC realtime clock (approx. 1h power reserve, external buffering possible)
- Temperature sensor
- Voltage monitoring
- Watchdog

Memory	Size	Usage
SDRAM	256MB	Firmware Application
Flash	32MB	Internal filesystem 26MB Bootloader Firmware

Memory	Size	Usage
FRAM	32kB	User-Data
Compact Flash Card	1GB	filesystem

Table 3-4: GCM-P(L) memory

A detailed description of the properties of the GCM-P can be found in the manual [MAN-GCM-PL].

3.2.4.5 General I/O Module – GIOM

The *General I/O Module (GIOM)* is a component for the extension of the number of standard inputs/outputs within a modular system. In order to be able to use the *GIOM*, a *SCM* (SL) module must exist which controls the inputs/outputs available on the G*IOM*.

The *GIOM* is not freely programmable.

The *GIOM* includes e.g. the following functional components:

- 32-bit µC, 2-channel, in the Lockstep Mode
- 3MB internal flash memory
- 256kB internal RAM memory
- Internal system buses
- Self-diagnosis options
- Up to 48 digital / analog inputs and outputs
- Reference voltage and reference current outputs

For a detailed description of the *GIOM* properties refer to chapter 5.5.

3.2.5 ***digsy***®$_{\textbf{fusion}}$ **device variants**

The modular structure of the *digsy*®$_{\text{fusion}}$ system enables different device variants.

Item number	Description		Housing variants
4888.02.001	*digsy*®$_{\text{fusion}}$ S T1	SCM	SMALL
4888.02.002		SCM + 1 x SIOM-Basic	SMALL
4888.02.004		SCM + 2 x SIOM-Basic	MEDIUM
4888.02.005		SCM + 3 x SIOM-Basic	MEDIUM
4888.02.007		SCM + 3 x GIOM-Basic	MEDIUM
4888.02.101	*digsy*®$_{\text{fusion}}$ S T2	SCM	SMALL
4888.02.102		SCM + 1 x SIOM-Basic	SMALL
4888.02.104		SCM + 2 x SIOM-Basic	MEDIUM
4888.02.105		SCM + 3 x SIOM-Basic	MEDIUM
4888.02.107		SCM + 3 x GIOM-Basic	MEDIUM
4888.02.201	*digsy*®$_{\text{fusion}}$ S T3	SCM	SMALL
4888.02.202		SCM + 1 x SIOM-Basic	SMALL
4888.02.204		SCM + 2 x SIOM-Basic	MEDIUM
4888.02.205		SCM + 3 x SIOM-Basic	MEDIUM
4888.02.207		SCM + 3 x GIOM-Basic	MEDIUM

Item number	Description		Housing variants
4888.03.301	*digsy*®$_{fusion}$ S-P 1GB CF Card	GCM-P + SCM SL	SMALL
4888.03.302		GCM-P + SCM SL + 1 x SIOM-Basic	MEDIUM
4888.03.304		GCM-P + SCM SL + 2 x SIOM-Basic	MEDIUM
4888.03.305		GCM-P + SCM SL + 3 x SIOM-Basic	LARGE
4888.03.321	*digsy*®$_{fusion}$ S-P 4GB CF Card	GCM-P + SCM SL	SMALL
4888.03.322		GCM-P + SCM SL + 1 x SIOM-Basic	MEDIUM
4888.03.324		GCM-P + SCM SL + 2 x SIOM-Basic	MEDIUM
4888.03.325		GCM-P + SCM SL + 3 x SIOM-Basic	LARGE
4888.04.301	*digsy*®$_{fusion}$ S-PL 1GB CF Card	GCM-PL + SCM SL	SMALL
4888.04.302		GCM-PL + SCM SL + 1 x SIOM-Basic	MEDIUM
4888.04.304		GCM-PL + SCM SL + 2 x SIOM-Basic	MEDIUM

Table 3-5: device variants

3.2.5.1 Controller

This variant consists of a *SCM* with integrated inputs/outputs. This variant constitutes a fully-fledged, freely programmable controller which can be used for both Standard Applications and Safe Applications.

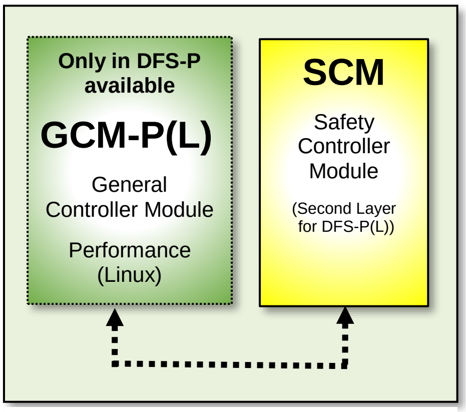


Figure 3-3: Functionally safe controller

For variant ***digsy***®$_{fusion}$ S-P(L), a GCM-P(L) module is additionally installed and a SCM-SL module is used in place of a SCM module.

3.2.5.2 Controller with one input/output extension

This variant consists of a *SCM* with integrated inputs/outputs, including an extension of the functionally safe inputs/outputs by an *SIOM*.

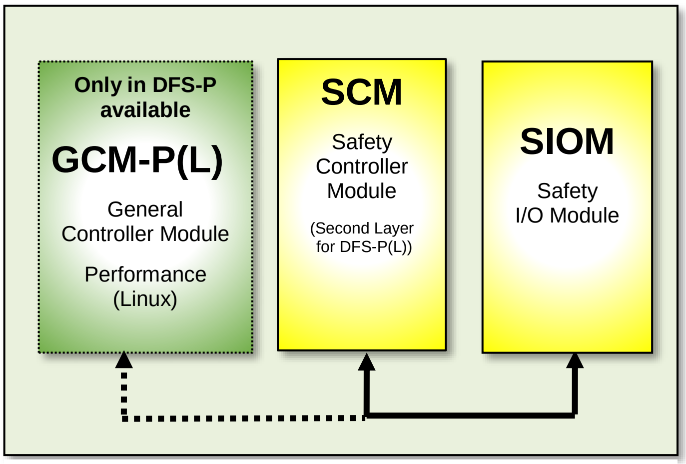


Figure 3-4: Functionally safe controller with I/O extension

For variant ***digsy***®$_{\text{fusion}}$ S-P(L), a GCM-P(L) module is additionally installed and a SCM-S(L) module is used in place of a SCM module.

3.2.5.3 Controller with two input/output extension

This variant consists of a *SCM* with integrated inputs/outputs, including an extension of the functionally safe inputs/outputs by two *SIOM*s.

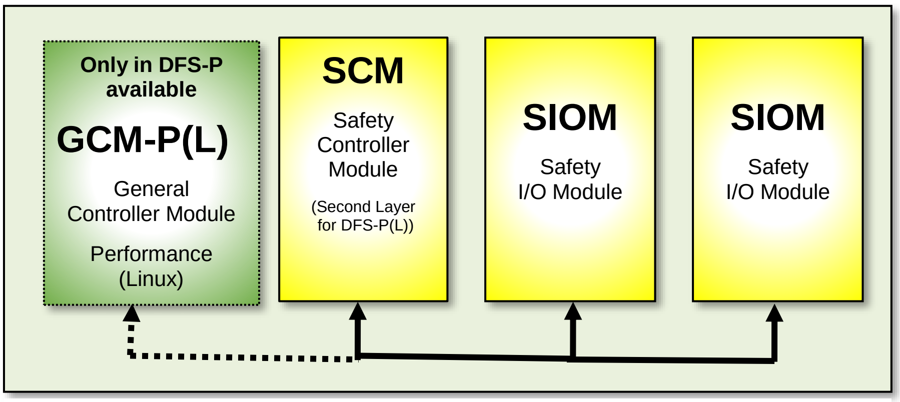


Figure 3-5: Functionally safe controller with additional I/O extensions

For variant ***digsy***®$_{\text{fusion}}$ S-P(L), a GCM-P(L) module is additionally installed and a SCM-S(L) module is used in place of a SCM module.

3.2.5.4 Controller with three input/output extension

This variant consists of a *SCM* with integrated inputs/outputs, including an extension of the functionally safe inputs/outputs by three *SIOM*s.

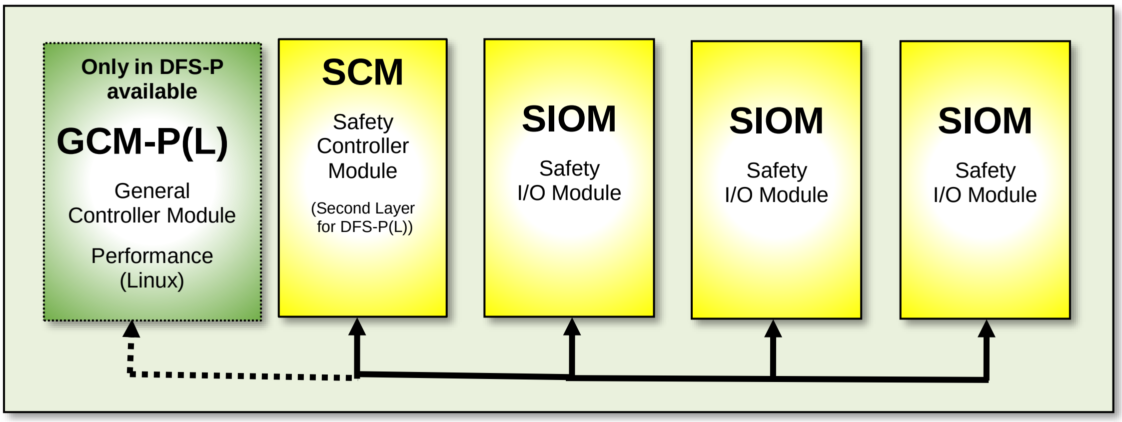


Figure 3-6: Functionally safe controller with additional safety I/O extensions

For variant ***digsy***®$_{\text{fusion}}$ S-P(L), a GCM-P(L) module is additionally installed and a SCM-S(L) module is used in place of a SCM module.

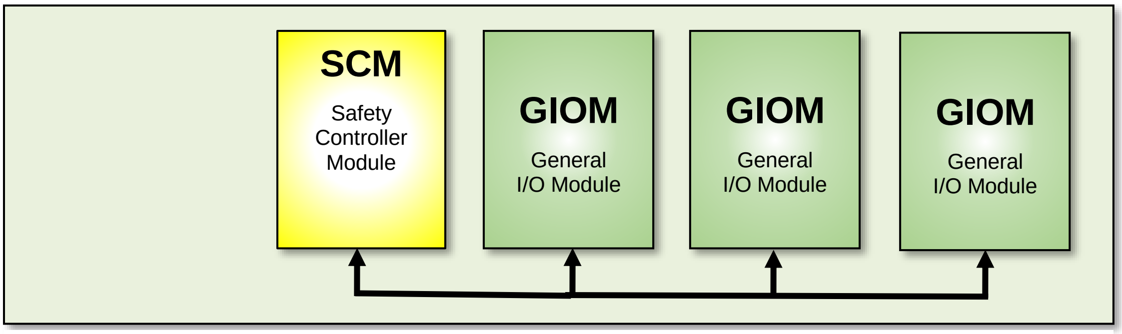


Figure 3-7: Functionally safe controller with additional standard I/O extensions

This variant consists of a SCM with integrated inputs/outputs, including an extension of the standard inputs/outputs by three GIOMs.

3.2.6 ***digsy***®$_{\text{fusion}}$ housing

3.2.6.1 Housing elements

The ***digsy***®$_{\text{fusion}}$ housing is made of die-cast aluminum. The housing provides excellent mechanical protection for the embedded electronic elements and superior EMC properties, and thus makes ***digsy***®$_{\text{fusion}}$ suitable for applications in mobile work machines and outdoor applications.

The housing consists of two basic elements which form the basis of all ***digsy***®$_{\text{fusion}}$ device variants:

- A basic housing element
- An intermediate housing element

Both the central plug and the M12 plug provide the contact for the housing elements. The central plugs guarantee high contact density with low plug volume. M12 plug connectors allow the direct wiring of sensors or actuators with customary, directly pluggable cables.

3.2.6.2 Housing variants

The following housing configurations are possible:

- SMALL: 2 basic housing elements (see Figure 3-8)
- MEDIUM: 2 basic housing elements + 1 intermediate housing element (see Figure 3-9)
- LARGE: 2 basic housing elements + 2 intermediate housing elements (see Figure 3-10)

The modular structure is a feature of ***digsy***®fusion. The housing configurations possible allow various implementation variants of ***digsy***®fusion. These controller versions correspond to serial devices of the manufacturer. The customer can order the required housing configurations from the manufacturer. The customer may not assemble or configure the housings on its own.

	CAUTION
⚠	**Electronic devices are sensitive to static discharge!** Electronic devices may be damaged when the housing is opened! 1. The user / operator may not open the housing of *digsy*®fusion! The housings may only be opened by INTER CONTROL staff at the factory.

***digsy*®fusion SMALL with central plug**

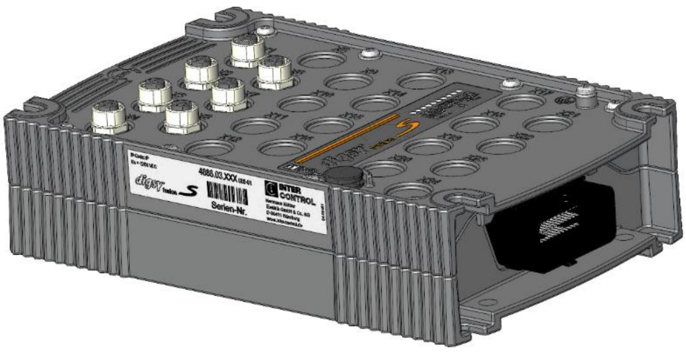


Figure 3-8: ***digsy***®fusion SMALL with central plug

Figure 3-8 shows a ***digsy***®fusion-controller in the SMALL configuration, consisting of two basic housing elements with central plug. A printed circuit board includes all the necessary control components. The communication interfaces are connected to external components via M12 plug connectors. The central plug ensures the compact connection of the input/output signals.

Figure 3-8 corresponds e.g. to the solution illustrated in Figure 3-3.

digsy®$_{fusion}$ **MEDIUM with central plug and input/output extensions**

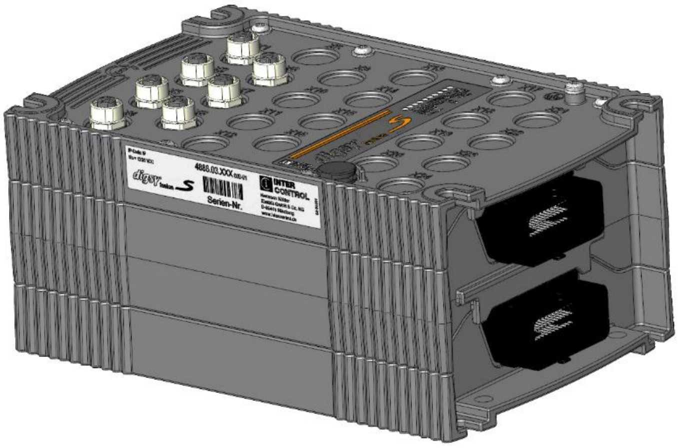


Figure 3-9: ***digsy***®$_{fusion}$ MEDIUM with central plugs

Figure 3-9 shows a ***digsy***®$_{fusion}$-controller in the MEDIUM configuration, consisting of two basic housing elements, one intermediate housing element, and several central plugs. M12 plug connectors connect the communication interfaces to external components. The central plugs ensure the compact connection of the input/output signals.

The configuration in Figure 3-9, for example, represents the solution described in Figure 3-4

The top module is a SCM as control module. The two underlying modules can, for example, be SIOMs and extend the number of inputs/outputs available.

digsy®$_{fusion}$ **LARGE with central plug and input/output extensions**

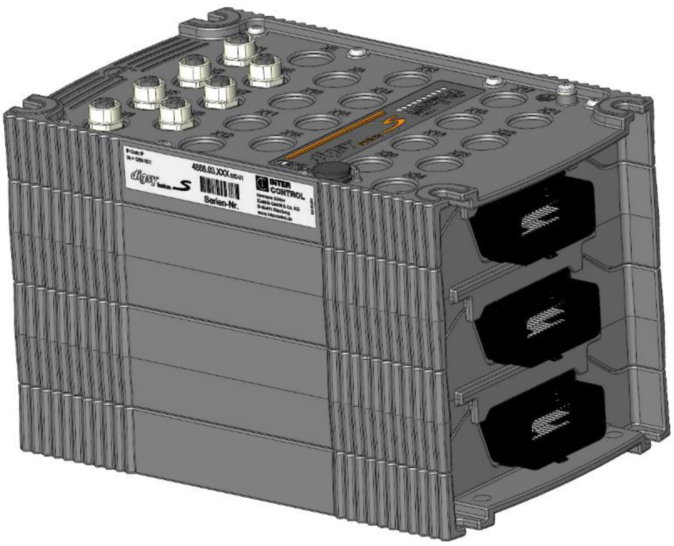


Figure 3-10: digsy®fusion LARGE with central plugs

Figure 3-10 shows a *digsy*®$_{fusion}$controller in the LARGE configuration, consisting of 1 SCM and 4 SIOMs in its maximum extension.

The configuration in Figure 3-10, for example, represents a solution which corresponds to the device variant described in Figure 3-4.

3.2.6.3 M12 plug connector

In order to be able to use the interfaces, M12 plug connectors are used with the SCM. The use of M12 plug connectors / M12 sockets, and the encoding of the M12 plug connectors and sockets, depend on their function on the respective board.

In principle, all M12-compatible mating connectors can be used. The M12 plug connectors of the ***digsy***®fusion support the Phoenix SPEEDCON quick locking system.

	CAUTION
	M12 plug connectors not properly inserted If M12 plug connectors not properly inserted, water may destroy the contacts. Water on the contacts can lead to corrosion on the contacts and, consequently, to malfunction! 1. Always make sure that the M12 plug connectors are properly inserted. This ensures adequate sealing and prevents contact problems! 2. We recommend using the Phoenix mating plug with the SPEEDCON quick locking system! 3. Unused M12 sockets must be fitted with sealing caps! 4. Before plugging in an M12 connector, make sure that there is no contamination.

3.2.6.4 Central plug connector

The central plug connector is from the CMC series, **64** pins, hybrid (**56**/8), 90°angular, made by FCI. For more detailed information on the wiring and the connections of the central plug refer to Chapter 5.1.2.2.

	CAUTION
	Handling of the central plug connector 1. The maximum number of mating cycles allowed for the central plug connector is 20! 2. Before plugging in the central connector, make sure that there is no contamination (e.g. sand, metal chips, etc.), especially on/in the sockets of the mating connector.

3.2.6.5 Connection cable

M12 cable

All standard, customary cables can be used as a connection cable for the M12 plug connectors. However, we recommend using cables equipped with the Phoenix SPEEDCON quick locking system.

INTER CONTROL offers a cable kit which includes all the necessary cables and plug codes (item number 4306.51.010).

Connection cable for the central plug

We recommend using standard cables suitable for mobile machines or vehicles as a connection cable for the central plug. The required socket is from the CMC series, 64 pins, hybrid 56/8 (item number FCI PPI0001500) made by FCI.

Wires with a maximum cross section of 0.5mm², or wires with a maximum cross section of 0.75mm² having reduced insulation thickness, can be used on the "small" contacts (GTS 0.64, FCI item number PPI0000489) of the central plug.

Wires with a maximum cross section of 2.0mm² can be used on the "large" contacts (GTS 1.50, FCI item number PPI0001484) of the central plug.

We recommend the use of a Leoni FLRY cable.

INTER CONTROL offers cables pre-assembled on one side with a 64-pole socket (56 x 0.5mm², 8 x 2.0mm², 3.0m or 5.0m cable length), item numbers 4306.20.001 and 4306.20.002.

3.2.6.6 LED indicators

digsy®fusion includes ten LEDs, some of which are multi-colored, for the display of system states and system processes (see Figure 3-11). In cases where there is a difference between the digsyfusionS (T1 / T2 / T3) and the digsyfusionS-P(L), a comment "DFS Tx only" or "DFS-P(L) only" is inserted.

Figure 3-11: ***digsy***®fusion S (T1/T2/T3) central label with LED indicators

Figure 3-12: ***digsy***®fusion S-P(L) central label with LED indicators

These are explained in further detail in Table 3-6:

LED name	Color 1	Color 2	Description
RUN	red OFF	green OFF	VIM switched off
	red ON	green OFF	VIM switched on, system in RESET mode
	red OFF	green ON	VIM switched on, system running
START	red OFF	green OFF	VIM switched off
	red ON	green OFF	System start program (Fork):
	red flashing (1Hz)	green OFF	Boot Loader: - Software update running
	red flashing (100ms pulse / 2s)	green OFF	Boot Loader: - Software update properly completed (the device must be switched off and on!)
	red flashing (2Hz)	green OFF	Boot Loader: - Software update completed with error (the device must be switched off and on!)
	red flashing (5Hz)	green OFF	Boot Loader: - The operating software cannot be started (the device must be switched off and on)
	red OFF	green flashing (1Hz)	Operating software: - Initialization of the operating software (normally takes a few seconds) - Alternative: boot loader update (can take a few minutes)
	red flashing (1x100ms pulse / 2s)	green flashing (1Hz)	Operating software: - Boot Loader update properly completed (the device must be switched off and on!)

LED name	Color 1	Color 2	Description
	red flashing (2Hz alternating)	green flashing (2Hz alternating)	Operating software: - Boot Loader update completed with error
	red flashing (5Hz alternating)	green flashing (5Hz alternating)	Operating software: - Operating software cannot be started after self-test
	red OFF	green ON	Operating software: - System initialization and self-tests successfully completed!
S-APP	red OFF	green OFF	Safe Applications not available or cannot be started!
	red OFF	green flashing (1Hz)	The Safe Application has started successfully in the safe mode and is running properly!
	red ON	green OFF	Safe Application not in the safe diagnosis mode; program stopped.
	red flashing (1Hz)	green OFF	Safe Application not in the safe diagnosis mode; program running.
APP	red OFF	green OFF	Standard Application not available or cannot be started!
	red OFF	green flashing (1Hz)	The Standard Application has been started successfully and is running properly!
	red ON	green OFF	Standard Application diagnosis mode; program is stopped.
	red flashing (1Hz)	green OFF	Standard Application in diagnosis mode; program is running.
ERROR	red OFF	green OFF	OPMODE_NORMAL, no error
	red OFF	yellow flashing (1Hz)	OPMODE_FAILURE
	red flashing (1Hz)	yellow OFF	OPMODE_FAILSAFE_IO
	red ON	yellow OFF	OPMODE_FAILSAFE_STOP or Start Up If the RUN and START indicators light up green, this indicates that OPMODE_NORMAL, but no APP, was detected.
ETH	---	green OFF	No Ethernet device connected
(only	---	green ON	Ethernet device connected; no data traffic
DFS Tx)	---	green flashing (1Hz)	Ethernet device connected; data traffic active
ETH 1	---	green OFF	No Ethernet device connected
(only	---	green ON	Ethernet device connected; no data traffic to SCM
DFS-P(L))	---	green flashing (1Hz)	Ethernet device connected; data traffic active to SCM
ETH 2	---	green OFF	No Ethernet device connected
(only	---	green ON	Ethernet device connected; no data traffic to GCM-P
DFS-P(L))	---	green flashing (1Hz)	Ethernet device connected; data traffic active to GCM-P
USB	---	green OFF	No USB device connected
	---	green ON	USB device connected; no data traffic
	---	green flashing (1Hz)	USB device connected; data traffic active

LED name	Color 1	Color 2	Description
USER1 (only DFS Tx)	red ON / OFF / flashing (1Hz)	green ON / OFF / flashing (1Hz)	Controllable via the Safe APP
USER2 (only DFS Tx)	red ON / OFF / flashing (1Hz)	green ON / OFF / flashing (1Hz)	Controllable via the Safe APP
USER3 (only DFS Tx)	red ON / OFF / flashing (1Hz)	green ON / OFF / flashing (1Hz)	Controllable via the Safe APP
USER S (only DFS-P(L))	red ON / OFF / flashing (1Hz)	green ON / OFF / flashing (1Hz)	Controllable via the Safe APP (SCM)
USER G (only DFS-P(L))	red ON / OFF / flashing (1Hz)	green ON / OFF / flashing (1Hz)	Controllable via the Standard APP (GCM-P)

Table 3-6: LED function display

Note

Simple rules for LED indicators (see Table 3-7):

The operating modes are described in Chapter 6.1

1. OPMODE_NORMAL operating mode:
   1. the LEDs *RUN* and *START* show a steady green light
   2. the LEDs *S-APP* and/or *APP* are flashing green at 1 Hz,
   3. the LED *ERROR* is off
2. OPMODE_FAILURE operating mode:
   - the LEDs *RUN* and *START* show a steady green light
   - the LEDs *S-APP* and/or *APP* are flashing green at 1 Hz,
   - the LED *ERROR* is flashing yellow at 1 Hz.
3. OPMODE_FAILSAFE_IO operating mode:
   - the LEDs *RUN* and *START* show a steady green light
   - the LEDs *S-APP* and/or *APP* are flashing green at 1 Hz,
   - the LED *ERROR* is flashing red at 1 Hz.
4. OPMODE_FAILSAFE_STOP operating mode
   - the LED *ERROR* shows a steady red light,
   - all the other LEDs are not relevant!

Operating mode	LED RUN	LED START	LED S_APP *)	LED APP *)	LED ERROR
OPMODE_NORMAL	green ON	green ON	green flashing 1Hz	green flashing 1Hz	OFF
OPMODE_NORMAL (no APP)	green ON	green ON	OFF	OFF	red ON
OPMODE_FAILURE	green ON	green ON	green flashing 1Hz	green flashing 1Hz	yellow flashing 1 Hz
OPMODE_FAILSAFE_IO	green ON	Not defined	Not defined	Not defined	red flashing 1 Hz
OPMODE_FAILSAFE_STOP	Not defined	Not defined	Not defined	Not defined	red ON

*) Flashing LEDs for LED S_APP and LED APP optional, depending on the APPs loaded

Table 3-7: LED indicators

3.3 Delivery condition

If the ***digsy***®$_{\text{fusion}}$ S is switched on for the first time by the user, the PLC is in the OPMODE_NORMAL state (LED RUN and LED START are green) after the boot process has finished.

In addition, there is no application on the control (LED ERROR is permanently red) and the retaindata are cleared (0xFF).

4 System Features of *digsy*®$_{\text{fusion}}$

4.1 Supply voltage

4.1.1 Functional description

digsy®$_{\text{fusion}}$ is supplied with electric power in two different ways:

- The logic, the functional electronics and the sensor supply outputs are supplied via the *VIM.*
- The power electronics are supplied via the *VIQ.*

Electric power from the *VIM* is fed in via a central plug connector.

The *VIM* of an SCM supplies the logic and functional electronics of the entire device as well as the local sensor supply outputs.

The *VIM* of an SIOM supplies only the local sensor supply outputs.

The *P_VIM* voltage is generated in the control system via the *VIM* on the SCM according to the individual control circuit components, such as overvoltage, undervoltage and polarity reversal protection.

Electric power from the *VIQ* is fed in via one or several central plug connectors. The *VIQ* supplies the power outputs. Different supply connectors identified as *VIQ_1 – VIQ_n* are available for the output groups.

Note

VIM, VIQn (n = 1 – 4) and GND: reference potential

1. The connections between the *VIM* and *VIQ_n* in the control system are **nonconductive** (insulation resistance ≥ 100kΩ).
2. The *VIQ_n* and *VIQ_m* (n ≠ m) connectors in the control system are **not** connected to each other (insulation resistance ≥ 40kΩ).
3. *GND_x* is the reference potential for *VIM* and *VIQ_n*. The connection between all the *GND_x* connectors in the control system is low resistive.

CAUTION

Fuse protection of the *VIM* and *VIQ_n* connectors

Currents supplied into the *VIM* and *VIQ_n* connectors which exceed the rated current can cause damage inside the control system and lead to the loss of its safety properties.

1. The supply cable routed to the *VIM* connector must be equipped with a fusible cutout to ensure that the maximum rated current specified in the technical data is not exceeded!
2. All supply cables routed to the *VIQ_n* connectors must be equipped with a fusible cutout to ensure that the maximum rated current specified in the Technical Data is not exceeded!
   This fusible cutout also trips in the case of polarity reversal on *VIQ_n*.

	WARNING
	Connection of the functional grounding Risk of control system malfunction 1. The user must connect the functional grounding connector (see Figure 10-1) of the control system to the ground (GND).

In vehicles the *VIM* must be connected to terminal 30 (permanent positive).

The *IPON* (to be connected to the ignition switch, terminal 15) energizes the DC/DC converters internally and supplies them via the *VIM*. The control system starts up when the supply voltage is present on the *VIM* and a rising edge is present on the *IPON*. The system is de-energized in the event of a falling edge on the *IPON*.

The *Power Hold* signal can be activated via the application program. As a result, the application program takes control over the shutdown of the control system. Consequence: A falling edge on the *IPON* does not directly cause the control system to shut down. Instead, an indication is transmitted to the application program, and the application program initiates a program-controlled shutdown of the control system by resetting the *Power Hold* signal.

4.1.2 VIM – IPON block diagram

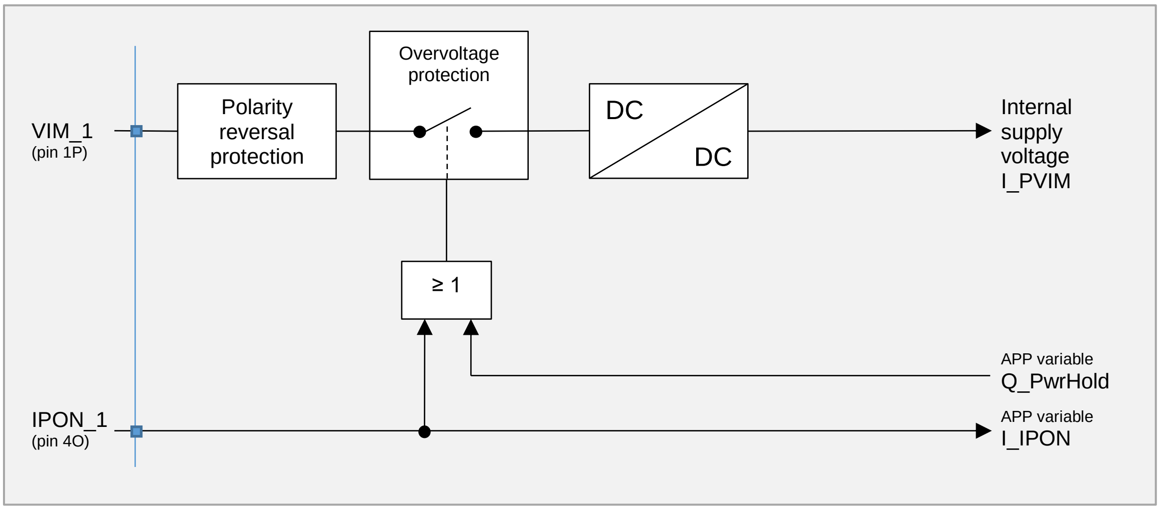


Figure 4-1: I_PVIM – I_IPON block diagram

4.1.3 24V/12V System voltage

The ***digsy***®fusion can be operated with both 24V and 12V system voltage from PLC version V03.36.XX (***digsy***®fusion S) and from V01.09.XX (***digsy***®fusion S-P and ***digsy***®fusion S-PL) see also 4.1.4.

For devices with smaller PLC versions than those mentioned above, only a system voltage of 24V is permitted.

4.1.4 Safety information and safety requirements – Supply voltage

Safety information and safety requirements – Supply voltage	
All individual power supply inputs of the control system must be supplied from a single voltage source. 12V/24V mixed operation of individual control system components is not permitted. Furthermore, the supply of individual control system components via two or more batteries separated from each other is not permitted.	§7122
If the user replaces a TX PLC variant with a TY PLC variant and vice versa, the user must repeat all of its tests to ensure that the safe user program still responds within the defined response time. X and Y = 1, 2, 3, 4, X <> Y	§5723
If the control system is designed for a 24V system voltage, the control system must be supplied via a 24V battery for mobile operation.	§7120
A 24V system voltage corresponds to a maximum operating voltage of 32V and a minimum operating voltage of 16V.	§7126
If the control system is designed for a 12V system voltage, the control system must be supplied via a 12V battery for mobile operation. Operation with a 12V system voltage is only permitted from PLC version V03.36.XX (***digsy***®$_{fusion}$ S) or from V01.09.XX (***digsy***®$_{fusion}$ S-P and ***digsy***®$_{fusion}$ S-PL).	§7124
A 12V system voltage corresponds to a maximum operating voltage of 16V and a minimum operating voltage of 8V.	§7125
If the external *VIM* supply voltage exceeds 33V +0.0V/-0.8V, all ***digsy***®$_{fusion}$ S (SCM, SIOM) safety components switch to the OPMODE_FAILSAFE_STOP operating mode.	§6888
If the internal protected *P_VIM* supply voltage on an SCM exceeds 32.5V, the SCM logs an error message in its internal system logbook and switches to the OPMODE_FAILSAFESTOP operating mode.	§6871
If the internal protected *P_VIM* supply voltage on an SIOM exceeds 32.5V, the system automatically carries out the following steps: - All sensor supplies of the SIOM (Q_SENS, UREFx, IREF) are switched off. - The POWER_HOLD signal of the SIOM is deactivated. - An error message is saved to the internal error memory of the SCM.	§6886

Safety information and safety requirements – Supply voltage	
All supply connections of the *VIQn*, n ∈ [1, 2, 3, 4] power outputs must be protected with an upstream fusible cutout (standard automobile fuse). - VIQ1: **10A** - VIQ2: **10A** - VIQ3: **10A** - VIQ4: **20A** - VIM: **10A**	§4259 §7025 §7026 §7272
The incoming cable impedances of the VIQ1, VIQ2, VIQ3 and VIQ4 inputs must be selected so that a short circuit current trips the upstream fuse within no more than 500ms in the event of polarity reversal. Note: Furthermore, the user must ensure that the onboard power supply or the power supply system provides the required short circuit current.	§7249
When totaling the output currents of the individual power drivers of a VIQ1, VIQ2 or VIQ3 power supply, a summation current of up to 10A per power supply is possible. The user must monitor the summation current of all outputs assigned to the VIQn power supply with n=1..3. The maximum summation current of the respective power supply (VIQ1, VIQ2 or VIQ3) must not exceed 10A. If this maximum summation current is exceeded, the user must take appropriate countermeasures. The maximum summation current of the VIQ4 power supply must not exceed 20A.	§7027 §7028 §7029
The user must monitor the summation current of all outputs assigned to the VIQ4 power supply. The maximum summation current of the VIQ4 power supply must not exceed 20A. The user must monitor the summation current and take appropriate countermeasures if the maximum summation current is exceeded.	§7030
Stationary operation supplied from the alternating current network: The devices described here are specified for extra-low (DC) voltage operation. The connection (functional grounding) between the housings and the internal grounding connector is conductive. Note the following if the necessary extra-low voltage is generated via a power supply unit (e.g. 230V~) in the case of stationary operation with supply via the AC power network: **Only PELV-type power supply units may be used (see EN 50178)!**	§6972
Ground connections: Make sure that all GND connections on the central plug are separately grounded. This ensures that the ground connection of the entire system remains intact if an individual ground connection is interrupted.	§6970

Safety information and safety requirements – Supply voltage	
For stationary operation the user must connect the "functional earth" (FE) housing connection to the grounding potential of the power supply. For mobile operation the user must connect the "functional earth" (FE) housing connection to the vehicle chassis.	§7210 §7211

Table 4-1: Safety information and safety requirements – Supply voltage

4.2 Comparison of the modules

The modules of the ***digsy***®fusion family support the following functions:

Function	SCM	SIOM	GIOM
VIM supply	■	■	■
VIQ supply	■	■	■
Undervoltage detection (VIM and VIQ)	■	■	■
Overvoltage detection (VIM and VIQ)	■	■	■
Power fail signal generation	■		
Overcurrent detection (VIM and VIQ)	■	■	■
Overcurrent shutdown (VIM and VIQ)	■	■	■
Polarity reversal protection	■	■	■
Load dump protection	■	■	■
IPON connection (terminal 15)	■		
DC/DC converter for the generation of the Vcc central voltage	■		
Power Hold circuit	■		
Watchdog with system reset generation	■		
Watchdog without system reset generation		■	■
System logbook, error memory	■		
Production data memory (electronic rating plate)	■	■	■
LED status indicator	■		

Table 4-2: General properties of the modules

4.3 Functional safety

4.3.1 Safety information and safety requirements – General

Safety information and safety requirements – General	
digsy®fusion S has a service life (proof test interval) of 20 years from the date of manufacture, irrespective of its operating time. The maximum service life starts with the date of manufacture (see Chapter 10.2). The control system does not support autonomous service life monitoring. The machine manufacturer / operator is responsible for monitoring and observing the maximum service life.	§6810 §6890 §7072
4888.02.XXX product version older than V02.15.XX The maximum uninterrupted operation time of the PLC is 24 hours. The user is responsible for a restart of the PLC after this time. The user must monitor the operating time and switch the control system to the OPMODE_FAILSAFE_STOP mode after more than 24 hours of uninterrupted operation. **4888.02.XXX from product version 02.15.XX to V03.26.XX** **4888.03.XXX V01.00.XX** The maximum uninterrupted operation time of the PLC is 24 hours. The user is responsible for a restart of the PLC after this time. After an uninterrupted operation time of 24 hours the PLC switches the operation mode automaticly to OPMODE_FAILURE and writes a warning into the system logbook. If the PLC enters OPMODE_FAILURE mode after 24 hours and if it does not leave this status for another hour, an error is entered in the system logbook. **4888.02.XXX from product version V03.35.XX** **4888.03.XXX from V01.05.XX** **4888.04.XXX from V01.08.XX** The maximum uninterrupted operation time of the PLC can be set between 1 to 90 days. The user is responsible for a restart of the PLC after this time. After an uninterrupted operation time of the set maximum uninterrupted operation time the PLC switches the operation mode automaticly to OPMODE_FAILURE and writes a warning into the system logbook. To prevent the control system from changing to OPMODE_FAILURE mode after the maximum uninterrupted operating time has reached, the user can perform a planned reset of the control system before the maximum uninterrupted operating time has reached. To do this, the system function "SysRestart" can be used.	§7421 §149 §8017

Safety information and safety requirements – General	
4888.02.XXX from product version V03.35.XX **4888.03.XXX from V01.05.XX** **4888.04.XXX from V01.08.XX** The maximum uninterrupted permissible operating time can be parameterized between 1 and a maximum of 90 days. The maximum uninterrupted permissible operating time is preset with the value 1 day. (default setting)	§8094 §8095 §8096
The internal system time is recorded in milliseconds within a 32 bit information. Due to the maximum permissible operating time of the system, a timer overflow can occur if a maximum uninterrupted operating time of > 49.71 days is reached. An entry is made in the system logbook approximately 10s before the timer overflow occurs.	§8097
The supply of the power outputs via the VIQx is not ensured when starting the control system. It can also be interrupted for technical reasons during operation. For this reason the voltages on the VIQx are continuously monitored. The voltages on the VIQx can therefore be externally connected / disconnected at any time without causing a malfunction. The system automatically detects voltage connections on the VIQx. A state machine is now executed, during which the second shutdown path is tested. After the execution of the state machine – and following successful testing – the VIQx_VALID flag is set. The system automatically detects voltage disconnections on the VIQx. All relevant outputs and the second shutdown path are then de-energized and the VIQ_VALID byte is reset. It is therefore absolutely necessary to evaluate the VIQ_VALID byte in the application program. Outputs can only be switched if the VIQ_VALID byte has been set.	§5272
Note the following when connecting a PC or a notebook computer to the control system for the purpose of programming, program upload/download, program diagnosis, etc.: - Ensure that the PC or the notebook computer is not connected directly to the AC voltage network via a power cable. - Alternatively, it is necessary to connect the PC or the notebook computer to the AC voltage network via an electrically isolated connection (e.g. an isolating transformer).	§6973
With the exception of the USB plug, no plugs may be plugged into or unplugged from the ***digsy***®$_{fusion}$ S control system during operation.	§6983

Safety information and safety requirements – General	
The SCM (Safety Controller Module) is the central element of the ***digsy***®fusion S control system for functional safety applications. The SCM is the master board and has control over the entire control system. The extension of inputs/outputs in the context of functional safety can be performed via an SIOM (Safety I/O Module).	§3986 §3987
The application program provides a function for reading and evaluating all messages saved to the internal error memory of ***digsy***®fusion S.	§3454
Error flags are not reset, even if the error causing the error flag (e.g. a short circuit on an output) has been remedied. To reset the error flag, first eliminate the error and then switch the control system off and back on.	§1920
A warning, irrespective of its source, saved to the error memory does not cause ***digsy***®fusion S to change its operating mode. ***digsy***®fusion S remains in its current operating mode.	§3472
If a diagnostic error is saved to the error memory while the system is in the OPMODE_NORMAL operating mode, the system always switches to OPMODE_FAILURE, regardless of the error source.	§3471
The input information may only be considered as valid in the application programs (Safe Application and Standard Application) if the *IO_Valid* flags have the value "TRUE". Otherwise, the data from the inputs of the boards may not be used in the application programs. The IO_Valid flag must be evaluated in both the Safe Application and the Standard Application programs. As long as this flag marks the input data as invalid, no input data may be used in the application programs. Furthermore, an expiration time stamp may only be transmitted if IO_Valid = 170.	§7289 §6854
The execution of the free-running task in the Safe Application is not guaranteed. If the free-running task is required for the safety function, the user must take appropriate measures in order to ensure that this task is correctly executed, e.g. by monitoring the free-running task via one of the cyclical safety tasks.	§5678 §7004
A reset of the POWER_HOLD flag after a falling edge on the IPON input in the application program causes the control system to shut down. If a new rising edge appears on the IPON in the meantime (for example, after a quick ignition off/ ignition on sequence), the control system must be restarted via the application program. The function library provides a special function for this purpose.	§5303

Safety information and safety requirements – General	
Firmware updates for the ***digsy***®fusion S control system may only be carried out by competent and instructed personnel, regardless of whether the update is performed via the CODESYS software tool or via a USB stick. When updating the firmware make sure that the manufacturer has approved the firmware and software versions used – either individually or together with various components – for the available hardware version. Following every update, competent and trained personnel must set the machine to the service mode again and check its proper functioning.	§6853 §6981
It is absolutely necessary to take account of the abnormal machine condition in the Safe Application. The error flags and system logbook entries of the system components used within the Safe Application must therefore be evaluated in order to ensure that the program initiates reactions that restore the machine to a safe condition.	§6975
The Safe Application must be programmed in such a way that any communication via the CANopen SAFETY protocol is stopped once the control system has switched to the OPMODE_FAILSAFE_IO operating mode. This applies to both the evaluation of Receive Objects and the creation of Transmit Objects.	§7008
In the OPMODE_FAILURE operating state, the functionality of the user programs remains active. Chapter 6.3 describes the system behavior in the OPMODE_FAILURE operating state. The user should decide how to react to a detected error. He should decide when and whether to change to the OPMODE_FAILSAFE_IO or OPMODE_FAILSAFE_STOP operating state.	§153

Table 4-3: Safety information and safety requirements – General

4.3.2 Standards

digsy®fusion S devices meet the requirements of Perfomance Level PL d, Category 3 / Category 2 (see 4.3.3) (EN ISO 13849-1:2015 ([N1])).

digsy®fusion S devices meet the requirements of ISO 25119 [N2] to [N5] and EN 16590 [N6] to [N9]. Agrar Performance Level (AgPL) and category are equivalent to EN ISO 13849-1:2015 ([N1]) as PL d, Cat 3 and Cat 2 (see 4.3.3). The safety parameters in chapter 10.5 also apply analogously to these standards.

The following modules fulfill the functional safety requirements described above:

- Safety Controller Module SCM (SL)
- Safety I/O Module SIOM

Almost all modules can be mixed freely within a device. Only the manufacturer may assemble ***digsy***®fusion control systems. Customers may not open ***digsy***®fusion controllers, add on additional components, or replace components!

	CAUTION
	Electronic devices are sensitive to static discharge! Electronic devices can be damaged if the housing is opened! 1. The user / operator may not open the housing of *digsy*®fusion. Only trained INTER CONTROL staff at the factory may open the housing.

4.3.3 Safety information and safety requirements – Standards

Safety information and safety requirements – Standards	
digsy®fusion S fulfills the requirements set out in EN ISO 13849-1:2015 ([N1]), Category 3, PL d, provided that input channels in two-channel topology are used.	§5078
All output channels of ***digsy***®fusion S fulfill the requirements set out in EN ISO 13849-1:2015 ([N1]), Category 3, PL d.	§4000
The input channels of ***digsy***®fusion S fulfill the requirements set out in: - EN ISO 13849-1:2015 ([N1]), Category 2, PL d – single-channel topology – and - EN ISO 13849-1:2015 ([N1]), Category 3, PL d – two-channel topology.	§4001 §4002

Table 4-4: Safety information and safety requirements – Standards

4.3.4 Safety information and safety requirements – Ambient conditions

The temperature inside the housing is measured by two sensors. Internal temperature limits apply for ***digsy***®fusion S.

Safety information and safety requirements – Ambient conditions	
If a temperature T in the -40°C ≤ T ≤ -35°C range is detected on at least one of the two sensors of a module and if the control system is in the OPMODE_NORMAL or OPMODE_FAILURE operating mode, the system saves a "diagnostic error" to the internal error memory. If this error message occurs, the user must ensure that the control system switches to the OPMODE_FAILSSAFE_STOP operating mode.	§4321 §7446
If a temperature T in the +100°C ≤ T ≤ +105°C range is detected on at least one of the two sensors of a module and if the control system is in the OPMODE_NORMAL or OPMODE_FAILURE operating mode, the system saves a "diagnostic error" to the internal error memory. If this error message occurs, the user must ensure that the control system switches to the OPMODE_FAILSSAFE_STOP operating mode.	§4327 §7446

Safety information and safety requirements – Ambient conditions	
If a temperature T less than -40°C is detected on at least one of the two sensors of a module, a fatal error message is generated and the control system switches to the OPMODE_FAILSAFE_STOP operating mode.	§4328
If a temperature T greater than +105 C is detected on at least one of the two sensors of a module, a fatal error message is generated and the control system switches to the OPMODE_FAILSAFE_STOP mode.	§4329

Table 4-5: Safety information and safety requirements – Ambient conditions

4.4 System response time and error response time

4.4.1 Controller response time – general information

The controller response time is not calculated under worst case conditions and indicated in the safety manual.

Instead, the user must define the maximum permissible controller response time for his specific application.

A method is implemented which ensures compliance with the maximum permissible tPLC_RT (controller response time) even in the worst case. This method is based on the monitoring of the actual tPLC_RT and a safe shutdown if the maximum permissible tPLC_RT was exceeded.

HAZARD

Defining the maximum controller response time

Reaching the safe mode within the specified time

1. To ensure safety, the user must determine the maximum permissible controller response time for his individual safety function in accordance with EN ISO 13849-1:2015 ([N1]).

4.4.2 Input data monitoring

During the acquisition of input data, the acquisition time is logged as the time stamp of the input data. The time stamp considers all filters and signal propagation times in the hardware and software under worst case conditions. This time stamp is transmitted together with the input data and made available to the APP for processing each module (SCM/SIOM). The time stamp is a value in the input mapping (I_InputTimestamp). It is given in microseconds.

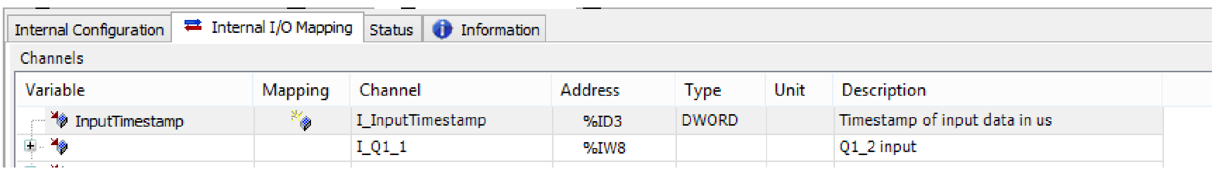


Figure 4-2: Input time stamp

Using the current system time and the time stamp of the inputs, the user must now monitor whether the input data is still valid or whether the maximum permissible system response time (*dwMaxControlResponseTime)* has already been exceeded. This monitoring step is necessary if the output is connected via an external interface (e.g. CAN) and not available on a module (SCM/SIOM). In this context, note the formulas given in Table 4-6.

Example:

```
dwCurrentTime_in_us := SysGetTimestampUs();
IF ((dwCurrentTime_in_us - InputTimestamp) < dwMaxControlResponseTime) THEN
        dataValid := TRUE;
ELSE
        dataValid := FALSE;
END_IF
```

In order to also ensure correct calculation in the case of a one-time overflow (32 bit => occurs after approx. 71 minutes), it is necessary to consider the time differences for this calculation. A one-time overflow can be remedied via the instruction "*((dwCurrentTime_in_us – InputTimestamp) < dwMaxControllerResponseTime)" instead of "((InputTimestamp + dwMaxControllerResponseTime) < dwCurrentTime_in_us)"*.

4.4.3 Output data monitoring

The user must provide an expiration time stamp for the output data generated by the APP. This expiration time stamp corresponds to a value in the output mapping (Q_OutputExpirationTime) and available to every module (SCM/SIOM).

Variable	Mapping	Kanal	Adresse	Typ	Einheit	Beschreibung
dw_Q_OutputExpirationTime_SCM		Q_OutputExpirationTime	%QD1	DWORD		Time of expiration of the output data in us
		Q_Q1_1	%QW4			Q1_1 output

Figure 4-3: Expiration time stamp

The user must calculate the expiration time stamp as follows:

Expiration time stamp = Input time stamp + maximum permissible controller response time

dw_Q_OutputExpirationTime_SCM := dw_I_InputTimestamp_SCM +
dwMaxControlResponseTime*;*

The expiration time stamp is then monitored by the lowest driver layer of the controller.

If the maximum permissible system response time cannot be met, i.e. if the expiration time stamp has expired, the controller switches to the OPMODE_FAILSAFE_IO operating mode. In addition, all output error flags are set.

4.4.3.1 Example: Input and output on the same module

If the input and the output of the safety function are on the same module (SCM/SIOM), the output time stamp for the safety-relevant data can be determined as follows:

```
VAR CONSTANT
        dwExpirationTime   :DWORD := DWORD#200_000;        // expiration time [us]
END_VAR
```

```
IF (I_System_SCM.I_IO_Valid = BYTE#170
        AND I_System_SIOM1.I_IO_Valid      = BYTE#170
        AND I_System_SIOM2.I_IO_Valid      = BYTE#170
        ) THEN
        dw_Q_OutputExpirationTime_SCM    := dw_I_InputTimestamp_SCM + dwExpirationTime;
        dw_Q_OutputExpirationTime_SIOM1 := dw_I_InputTimestamp_SIOM1 + dwExpirationTime;
        dw_Q_OutputExpirationTime_SIOM2 := dw_I_InputTimestamp_SIOM2 + dwExpirationTime;
END_IF
```

4.4.3.2 Example: Input and output on different modules

If the input and the output of the safety function are on different modules (SCM/SIOM), the output time stamp for the safety-relevant data can be determined as follows:

Example: Input (safety function 1) on SIOM1 and output (safety function 1) on SCM. Input (safety function 2) on SCM and output (safety function 2) on SIOM2.

```
VAR CONSTANT
        dwExpirationTime   :DWORD := DWORD#200_000;            // expiration time [us]
END_VAR
```

```
IF (I_System_SCM.I_IO_Valid           = BYTE#170
        AND I_System_SIOM1.I_IO_Valid      = BYTE#170
        AND I_System_SIOM2.I_IO_Valid      = BYTE#170
        ) THEN
        dw_Q_OutputExpirationTime_SCM    := dw_I_InputTimestamp_SIOM1 + dwExpirationTime;
        dw_Q_OutputExpirationTime_SIOM1 := dw_I_InputTimestamp_SIOM1 + dwExpirationTime;
        dw_Q_OutputExpirationTime_SIOM2 := dw_I_InputTimestamp_SCM + dwExpirationTime;
END_IF
```

4.4.3.3 Example: Recommended use without CAN

If the inputs and outputs of the safety functions are on different modules (SCM/SIOM), the user can determine a minimum controller response time for each task. This time is independent of the modules used and always refers to the oldest input stamp.

Determining a half maximum value of DWORD

```
MAX_DWORD_HALF := 16#7FFFFFFF;
```

Determination of the oldest input time stamp (SCM/SIOM1/SIOM2 in a task).

```
IF ((dw_I_InputTimestamp_SCM - dw_I_InputTimestamp_SIOM1) < MAX_DWORD_HALF) THEN
        dwInputTimestampMin := dw_I_InputTimestamp_SIOM1;
ELSE
        dwInputTimestampMin := dw_I_InputTimestamp_SCM;
END_IF
IF ((dwInputTimestampMin - dw_I_InputTimestamp_SIOM2) < MAX_DWORD_HALF) THEN
        dwInputTimestampMin:= dw_I_InputTimestamp_SIOM2;
```

```
END_IF
```

Setting the expiration time stamps (SCM/SIOM1/SIOM2 in a task)

```
VAR CONSTANT
        dwExpirationTime   :DWORD    := DWORD#200_000;        // expiration time [us]
END_VAR
```

```
dw_Q_OutputExpirationTime_SCM := dwInputTimestampMin + dwExpirationTime;
dw_Q_OutputExpirationTime_SIOM1 := dwInputTimestampMin + dwExpirationTime;
dw_Q_OutputExpirationTime_SIOM2 := dwInputTimestampMin + dwExpirationTime;
```

4.4.4 Using several tasks

The user defines a specific, maximum permissible controller response time for every safety task. All the I/O modules (SIOM, GIOM), including the SCM (SL) module, are assigned to one task each. For every I/O module assigned, the APP of every task calculates in every cycle an output data expiration time depending on the controller response time. This expiration time is monitored on every I/O module as described. The described actions are executed if no new output data is available before the expiration time has expired.

If a task includes more than one module, the actions performed by the modules can be different.

4.4.5 Determining the maximum permissible controller response time without errors

The user must define the maximum permissible tPLC_RT. The methods for calculating the expiration time described in the following chapters must be performed on the basis of the maximum permissible tPLC_RT. The control system even guarantees in the worst case that the maximum permissible tPLC_RT is not exceeded.

Table 4.6 indicates controller response times which can be reliably reached without triggering the OPMODE_FAILSAFE_IO mode. If the selected maximum permissible controller response time is less than these values, it must be assumed that the availability is strongly reduced and that the control system frequently switches to the OPMODE_FAILSAFE_IO mode. If the maximum permissible tPLC_RT is defined too low, this does not lead to a safety gap, but causes an availability problem.

When using the CANopen safety protocol, the SCT (Safeguard Cycle Time in the CANopen safety protocol - maximum distance between 2 telegrams to an SRDO) must also be observed.

Local input	CAN input	Local output	CAN output	Reachable controller response time
X		X		Z1 = 10ms (input) + 3 x task cycle + 10ms (output)
	X	X		Z2 = 2 x SCT (input) + 3 x task cycle + 15ms (output)
	X		X	Z3 = 2 x SCT (input) + 2 x task cycle + 2 x SCT (output)
X	X	X		Max(Z1, Z2)
X	X	X	X	Max(Z1, Z2, Z3)

Table 4-6: Controller response time

These formulas only apply if the input data is processed within a task cycle and the outputs are set. If the processing does not take place within one cycle, the additional task cycle times must be added to the response time determined:

$tPLC\_RT_{Total}$ = tPLC_RT + (N-1) * cycle time (N: Number of task cycles run)

4.4.6 Determining the maximum permissible controller response time with errors

The guaranteed maximum controller response time after errors have occurred in the control system depends on the error type.

All possible errors can be classified as follows depending on their effects on the controller response time:

Error class	Response time
Fatal errors in the controller logic	100ms
Single board failure	tPLC_RT + 100ms
Output failure	tPLC_RT + 100ms
Input failure	tPLC_RT + 100ms

Table 4-7: Controller response time with errors

4.4.7 Safety information

⚠ Safety information and safety requirements	
For each I/O module (board) of the PLC the user must define a controller response time which is monitored during the expiration time.	§7148
No controller response time can be defined for a free-running, safety-relevant task. When using a safe, free-running task, the user must select an expiration time which is sufficiently long in order to prevent availability problems.	§7149
The firmware generates an input data time stamp for each I/O module. The time stamps are made available to the AWP together with the input data. The acquisition time stamp is set so that it is ensured that all signals pending by this time are recorded in the data. The solution considers all filters and signal propagation times with regard to both the hardware and software. The following applies: Time stamp < Time of the earliest signal which has not been captured	§7425
In the case of fatal errors and I/O errors the controller response time is extended by up to 100ms (tFDR_fatal).	§7201 §7326
If I/O errors have occurred on the inputs/outputs, the controller response time is extended by up to 100ms	§7175 §7177
By modifying the functions described in Chapter 4.4.5, the user can calculate the maximum cycle time from a pre-defined controller response time	§7153

Safety information and safety requirements	
If the safety function inputs and outputs of a control system are distributed over several boards, all outputs of this safety function within the "controller respond time tPLC_BrdLoss related to the board loss" change to the safe mode in case of a fatal error of a board assigned to the safety function concerned. The controller response time related to the board loss is calculated as follows: tPLC_BrdLoss = tPLC_RT + tFDR_fatal = tPLC_RT + 100ms.	§7202
If an individual module on a safety-relevant board (SCM (SL) or SIOM) detects that the safety response time specified for the outputs concerned cannot be observed, these outputs change to the safe mode before the controller response time has expired.	§7152
The controller diagnostic time is 50ms	§7271
When using a single-channel input, the diagnostic time/request time ratio must be greater than or equal to 100 in order to be able to fulfill the requirements set out in EN ISO 13849-1:2015 ([N1]) PLd Category 2. This means: If the diagnostic time is 50 ms, the request time must be greater than or equal to 5 s.	§7268
In order to consider signal propagation times under safety-relevant aspects, the low level drivers must calculate a modified expiration date t_Expiration_LLD from the expiration date OutputExpirationTime of the output state predefined by the application: t_Expiration_LLD = t_Expiration_LLD – tFilter – tDelay_Out or t_Expiration_LLD = t_Expiration_LLD – tHW_SW_Delay t_Expiration_LLD is the expiration date which is monitored by the low level driver layer.	§7377
The end user must perform a risk analysis in order to calculate the process safety time on the system level.	§7378
The end user must determine the required controller response time tPLC_RT based on the process safety time.	§7382
If several controller response times are required, the end user must assign I/O modules to the various controller response times.	§7383
For every board in every task cycle, the application must determine an expiration date of the output states of this board from the input time stamp tDataIn: OutputExpirationTime = InputTimestamp + tPLC_RTx	§7384
Based on the controller response time tPLC_RT, the user can determine a target value for the cycle time tCycle of the task: tCycle < (tPLC_RT – tIO_Delay) / 2	§7385

Safety information and safety requirements	
The time tIO_Delay is 18.1ms (15ms for the inputs (filters) and 3.1ms for the outputs).	§7386

Table 4-8: Safety information concerning the error response time

4.5 Internal communication

The communication between the ***digsy***®fusion modules (SCM (SL) / SIOM / GIOM / GCM-P) in a control system is managed automatically on the system level and cannot be accessed by the user.

4.6 Inputs and outputs

4.6.1 Typification of inputs and outputs

Various input and output functionalities are typified for the *digsy*®fusion modules. This means that every input/output type has defined properties assigned.

Some of these types are assigned different classes. This means that different parameters may apply with the same functional properties.

The different types are marked with alphanumeric characters (e.g. type A).

The various type classes are complemented by figures (e.g. type A2).

WARNING

Use of sensors / actuators in two-channel systems

1. It must be ensured that the sensors/actuators used in a two-channel system actually support two channels.

Safety information and safety requirements – Inputs/outputs	
When defining a configuration with a rated output current of 0.5A, the controller switches to the safe mode.	§7256
If an input/output group of a board is configured as 'not safe' (standard) and if at least one input/output is configured as an input, all I/O terminals of this group (connected to ground) have an input resistance of 3.9kΩ.	§7252
If an input/output group of a board (safe or standard) is only configured with outputs, the 3.9kΩ input resistances are between the I/O terminals of this group and the ground connection.	§7253
If the current measuring resistance of a current input has been switched off by the hardware fuse and if the input has an odd number (e.g. I2.1, I2.3, I2.5, ...), the error flag for the next input is also set (e.g. overload I2.5, error flag set for I2.6). "High CrossTalk" is saved to the error memory as the error cause.	§7236

Safety information and safety requirements – Inputs/outputs	
If the current measuring resistance of a current input has been switched off by the hardware fuse and if the input has an even number (e.g. I2.2, I2.4, I2.6, ...), the error flag for the previous input must also be set (e.g. overload I2.6, setting the error flag for I2.5). "High CrossTalk" must be logged as the error cause in the error memory.	§7237
If the error flag has been set for a 10V input due to an overvoltage (> 10.75V) and if the input has an odd number (e.g. I2.1, I2.3, I2.5, ...), the error flag is also set for the next input (e.g. overload I2.5, setting the error flag for I2.6). "High CrossTalk" is saved to the error memory as the error cause.	§7228
If the error flag has been set for a 10V input due to an overvoltage (> 10.75V) and if the input has an even number (e.g. I2.2, I2.4, I2.6, ...), the error flag is also set for the previous input (e.g. overload I2.6, setting the error flag for I2.5). "High CrossTalk" is saved to the error memory as the error cause.	§7229
If the error flag has been set for a 32V input due to an overvoltage (> 32.5V) and if the input has an odd number (e.g. I2.1, I2.3, I2.5, ...), the error flag is also set for the next input (e.g. overload I2.5, setting the error flag for I2.6). "High CrossTalk" is saved to the error memory as the error cause.	§7230
If the error flag has been set for a 32V input due to an overvoltage (> 32.5V) and if the input has an even number (e.g. I2.2, I2.4, I2.6, ...), the error flag is also set for the previous input (e.g. overload I2.6, setting the error flag for I2.5). "High CrossTalk" is saved to the error memory as the error cause.	§7231
If the error flag has been set for a current input due to an overcurrent (> 21.75mA) and if the input has an odd number (e.g. I2.1, I2.3, I2.5, ...), the error flag is also set for the next input (e.g. overload I2.5, setting the error flag for I2.6). "High CrossTalk" is saved to the error memory as the error cause.	§7232
If the error flag has been set for a current input due to an overcurrent (> 21.75mA) and if the input has an even number (e.g. I2.2, I2.4, I2.6, ...), the error flag is also set for the previous input (e.g. overload I2.6, setting the error flag for I2.5). "High CrossTalk" is saved to the error memory as the error cause.	§7233
If the error flag has been set for an analog (current or voltage) input which is configured as safe and if "Shunt Protection Fuse Activated", "Input Overload" or "High Cross Talk" has been logged as the error cause, a diagnostic error is saved to the error memory together with the cause of the error.	§7234
If the error flag has been set for an analog (current or voltage, not safety-relevant) input which is configured as 'standard' and if "Shunt Protection Fuse Activated", "Input Overload" or "High Cross Talk" has been logged as the error cause, a warning is saved to the error memory together with cause of the error.	§7235

Table 4-9: Safety information and safety requirements – Inputs/outputs

4.6.2 Type A outputs

Type A outputs have the following properties and can be configured as follows:

Function	Output	Input	Current measurement	Functionally safe (Cat2)	Functionally safe [Cat3]
Digital output, pulled down	■	Read back	Diagnosis	*	Yes
PWM output, pulled down	■		Diagnosis	*	Yes
PWM output, pulled down with current control	■		Closed loop control	*	Yes

Table 4-10: Properties of type A outputs

* Not configurable for Cat2 but usable for Cat2-applications.

4.6.2.1 Class A1, A2, and A3 types

Three classes, A1, A2, and A3 – with different output currents and different current measuring ranges – are defined for type A outputs.

Type	Rated current	Switchable amplification	Measuring range
A1	2.5A	No	0 – 2.5A
A2	4.0A	No	0 – 4.0A
A3	2.5A	No	0 – 2.5A

Table 4-11: Comparison of the type A outputs

4.6.2.2 Wiring of the central plug

PIN	PIN NAME	Input Digital +	Input Digital -	Input Count	Input Analog	Output Digital +	Output Digital -	Output PWM	Output PWM I_{REG}	VIX / GND	I/O Type	Note
3B	Q1.1					■		■	■		A1	Safe Out, group 1, 2.5A
3A	Q1.2					■		■	■		A1	Safe Out, group 1, 2.5 A
4A	Q1.3					■		■	■		A1	Safe Out, group 1, 2.5 A
4B	Q1.4					■		■	■		A1	Safe Out, group 1, 2.5 A
2A	Q2.1					■		■	■		A3	Safe Out, group 2, 2.5A
2B	Q2.2					■		■	■		A3	Safe Out, group 2, 2.5A
1B	Q2.3					■		■	■		A2	Safe Out, group 2, 4A
1A	Q2.4					■		■	■		A2	Safe Out, group 2, 4A
2O	Q3.1					■		■	■		A3	Safe Out, group 3, 2.5A
2N	Q3.2					■		■	■		A3	Safe Out, group 3, 2.5A
1N	Q3.3					■		■	■		A2	Safe Out, group 3, 4A
1O	Q3.4					■		■	■		A2	Safe Out, group 3, 4A

Table 4-12: Pin wiring of type A outputs

4.6.2.3 Technical data

See Chapter 10.13.5.

4.6.2.4 Type A block diagram

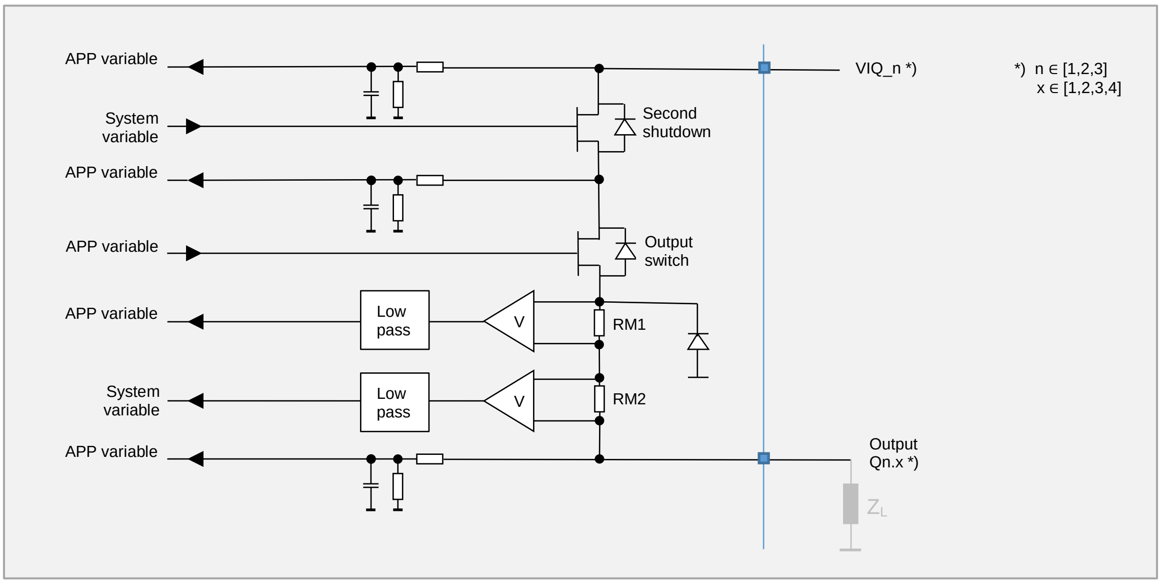


Figure 4-4: Block diagram of type A outputs

4.6.2.5 Functional description

The output function provides power outputs which can be controlled both digitally and analogously. The analog control is implemented via the PWM outputs of a microcontroller.

The output current is measured via two independent analog inputs. A second line for diagnostic purposes (functional safety) is thus implemented in addition to a complete current control loop.

The outputs are protected by freewheeling diodes.

The corresponding static output state can be read back via a digital read back input.

A second shutdown path of the *SCM* and the *SIOM* ensures single fault safety. This second shutdown path is implemented as a multi-switch for several (preferably 4) outputs. All *VIQn* supply connections have a second shutdown path.

Type A outputs must be configured in groups for both the Safe Application and the Standard Application. The following settings apply:

Safe Application	Standard Application
Safe / Standard / Disabled	
→	Standard / Disabled

This means that the Safe, Standard, and Disabled settings are possible for the Safe Application. The output can only be used in the Standard Application and can be set as Standard or Disabled in this application, if Disabled is selected in the Safe Application.

4.6.2.6 Safe State

⚠	WARNING
	Safe State If the plc control is in safe state, the output type A are deenergized.

4.6.2.7 Safety information and safety requirements – Type A outputs

Safety information and safety requirements – Type A outputs	
All type A outputs, including all A1, A2, and A3 classes, fulfill the requirements set out in EN ISO 13849-1:2015 ([N1]), Category 3, Performance Level d.	§1753
An unexpected output level can be read back in the case of high-resistance output loads. To avoid this, the user must connect a minimum load of 500Ω.	§7408
If the power supply (VIQ1, VIQ2 or VIQ3) of an energized output fails and is reconnected later, the output is de-energized once this power supply has recovered (restart protection). This applies to the following outputs: • Type A digital outputs • Type A PWM outputs • Current-controlled type A outputs	§7105 §7111 §7114
If the restart protection blocks the energizing of an output after the VIQ1, VIQ2 or VIQ3 power supply has failed and recovered, the output can be re-energized via the application as follows: 1) De-energize the output via Q_StartStop_State = 0 (this is a prerequisite for starting of the VIQ switch-on diagnosis) 2) Wait until the corresponding VIQ-Valid is set to TRUE by the VIQ switch-on diagnostic function 3) Energize the input n via Q_Start_Stop_State = TRUE.	§7106 §7112 §7115
In the de-energized state, the output voltage of all type A outputs is less than 2.2V with a tolerance of +0.1V, or -0.1V.	§5282
After being de-energized, all type A outputs have a leakage current of up to 1.1mA with a tolerance of +0.1mA, or -0.1mA within a maximum time of 35ms.	§5283
The parallel connection of several type A outputs in order to obtain an output with an increased output current is <u>prohibited</u>!	§4900
All type A outputs have a readback channel which is used to read back the signal on the output terminal (sampling once per millisecond). This information is made available to the user programs via the variable I_Q1_X.I_State (X: 1..4) in the input/output mapping. **PLEASE NOTE: The readback information concerning the output state must <u>not</u> be used in the safe context of the Safe Application!**	§5252
For type A outputs of the classes A2 and A3, the output current in the PWM operating mode is limited to up to 15mA during the interpulse periods (for HW version < VXX.40.XX).	§1778

<table>
<tr><th colspan="2">⚠ Safety information and safety requirements – Type A outputs</th><th></th></tr>
<tr><td colspan="2">On all outputs type A, for classes A1, A2 and A3, the maximum output current in PWM mode during impulse pause is limited to maximal 15mA (for HW version ≥ VXX.40.XX).</td><td></td></tr>
<tr><td colspan="2">For type A outputs the transmission function of the integrated current measurement low pass has a cutoff frequency of 3.4Hz ± 10%</td><td>§3559</td></tr>
<tr><td colspan="2">If a mixed configuration is defined within a type A output group, i.e. if the outputs in this group are configured
- for both the Safe Application and the Standard Application
- or for both the Safe Application and as "not used" or "disabled",
the system automatically switches to the OPMODE_FAILSAFE_IO operating mode upon the start.
NOTE: The CODESYS programming system allows this kind of mixed configurations. However, type A output groups must be completely configured for the Safe Application in order to be able to use them within the Safe Application!</td><td>§3576
§6767</td></tr>
<tr><td colspan="2">For every type A output the output voltage is captured on the plug-in contact. The following levels apply:

Low Level / High Level
24V system: ≤ 4.0V (+0.0V / -0.1V) / ≥ 5.0V (+0.0V / – 0.1V)</td><td>§3587
§3588</td></tr>
<tr><td colspan="2">The system measures the output current for all type A outputs. The output current is captured with a precision of ±1.5% with reference to the rated current of the respective output. It the rated current of an output is configurable, the configured rated current applies.

Rated currents:
Q1.1 – Q1.4, 2.5A
Q2.1 – Q2.2, Q3.1 – Q3.2 2.5A
Q2.3 – Q2.4, Q3.3 – Q3.4 4.0A</td><td>§1821
§1822
§1823</td></tr>
<tr><td colspan="2">If a type A output is configured as a digital output – irrespective of whether it is used in the Safe Application or the Standard Application – the following applies:
- The type A digital output cannot be energized by the application program if – irrespective of the Q_StartStop_State parameter – VIQ_VALID has not been set for its output group (see Chapter 7.6.2.4).
- However, a digital type A output can only be energized via the application program if VIQ_VALID has been set for its output group (see Chapter 7.6.2.4).
- The digital type A output is automatically de-energized by the system, i.e. without a command from the application program, if VIQ_VALID (see Chapter 7.6.2.4) is reset.
NOTE: This occurs automatically, for example if the VIQx is disconnected from the supply voltage outside the control system.</td><td>§6660
§1846
§1890</td></tr>
</table>

Safety information and safety requirements – Type A outputs	
If a type A output is configured as a digital output for the Safe Application or the Standard Application, the following applies: - The digital type A output must keep up the new level for at least 50 ms upon every level change. The programmer must define this setting in the application program. This is necessary to guarantee defined conditions for the automatic self-diagnosis. The maximum signal output frequency is 10 Hz (= 1 / (2 x 50 ms)). - The digital type A output <u>cannot</u> be energized via the application program if the error flag has been set. If the digital output is configured for the safety application, an error (ERR_DIAG_FAST_TOGGLING_TYPE_A) is entered in the system logbook in the event of an error. If the digital output is configured for the standard application, a warning (WARN_DIAG_FAST_TOGGLING_TYPE_A) is entered in the system logbook in the event of an error. In both cases, the output is switched off in the event of a fault.	§5298 §3619 §6905
If a type A output is configured as a digital output for the Standard Application, the following applies: - The digital type A output can be energized via the application program even if the error flag has been set.	§6763
Current readback precision: 2.5 A configuration (Q1.1 – Q1.4): +/-2% 4 A configuration (Q2.3, Q2.4, Q3.3, Q3.4): +/-2% 2.5 A configuration (Q2.1, Q2.2, Q3.1, Q3.2): +/-2% A 5.5% safety precision factor with reference to the nominal scale end value applies to the safe current measurement of type A outputs. If this safety precision can no longer be ensured due to an error, an output configured as safe is de-energized within 100ms after the error has occurred.	§7410 §7411 §7412 §6899
The control system detects an overcurrent if the real, averaged current on a type A output exceeds the rated current by a certain value for at least 0.10s. If the overcurrent persists, the system saves a warning to the error memory (after 0.25s at the latest). Rated current / Overcurrent detection Type A1: 2.5A / 3.0A Type A3: 2.5A / 3.0A Type A2: 4.0A / 4.8A PWM configured outputs are: Rated current / no peak detection / peak detection Type A1: 2,5A / 3,5A +/- 0,1A / 5,0A +/- 0,1A	§1815 §1816 §1809 §4927 §4928 §4929 §4930 §4934

Safety information and safety requirements – Type A outputs	
Type A3 2,5A 3,5A +/- 0,1A 5,0A +/- 0,1A Type A2 4,0A 5,5A +/- 0,1A 8,0A +/- 0,1A If the application program does not de-energize the output affected by overcurrent within 0.9s (max. 1.1s), the system automatically de-energizes the output and sets the error flag.	
The system performs a two-channel current measurement on a type A output. If the values measured for the two channels of an output differ by more than 3% – with reference to the rated current -, the system sets the error flag for the output.	§3634
If a type A output is configured as a PWM output or as a current-controlled PWM output – irrespective of whether it is used in the Safe Application or the Standard Application – the following applies: - The PWM frequency can be individually adjusted for each of Group 1 PWM outputs (Q1.1 – Q1.4). - For Group 2 PWM outputs (Q2.1 – Q2.4), only one PWM frequency can be set which applies to the entire group. - For Group 3 PWM outputs (Q3.1 – Q3.4), only one PWM frequency can be set which applies to the entire group.	§1862 §1863 §1864
If a type A output is configured as a PWM output/current-controlled PWM output and energized via the flag Q_StartStop_State = TRUE, the selection of the PWM frequency is monitored. The following generally applies: The system permanently de-energizes a PWM output and sets the error flag if: - a PWM frequency of 0 Hz is selected, - a PWM frequency ≥ 1001 Hz is selected, - a PWM frequency of 1 Hz ≤ f_{PWM} ≤ 49 Hz is selected, provided that the output has been configured for the Safe Application. If a PWM output is switched inactive, the PWM frequency is not monitored. "Freq. out of range" is logged as the error cause in the error memory.	§3610 §3611 §6952 §4823 §4824 §4825
If a type A output is configured as a PWM output or as a current-controlled PWM output, the following applies to the precision of the frequencies selected: Configuration for Standard Application only: 1 – 49 Hz ± 0.5 Hz Configuration for Standard and Safe Application: 50 – 200 Hz 0Hz / -2.5 Hz 201 – 300 Hz 0Hz / -5 Hz 301 – 500 Hz 0Hz / -10 Hz	§1854 §1853 §4905 §1855 §1856 §7013

Safety information and safety requirements – Type A outputs	
501 – 700 Hz 0Hz / -20 Hz 701 – 1000 Hz 0Hz / -37.5 Hz	
If a type A output is configured as a PWM output or as a current-controlled PWM output – irrespective of whether it is used in the Safe Application or the Standard Application – the following applies: The type A PWM output cannot be energized via the application program if the flag in the VIQ_Valid byte (see Chapter 7.6.2.4) is not TRUE.	§6661 §6662 §1861
If the PWM setpoint value of a type A PWM output is preset to Q_PWM = 0, the output is de-energized irrespective of the Q_StartStop_State parameter setting. If a current setpoint value of 0mA is preset for a current-controlled type A PWM output, the output is de-energized irrespective of the Q_StartStop_State parameter setting.	§1859 §1888
- If an output current of nominal 2.5A is configured for a current-controlled type A1 PWM output and a setpoint value > 2,500mA has been preset, the output is permanently de-energized and the error flag is set. - If a setpoint value > 2,500mA has been preset for a current-controlled type A3 PWM output (rated current > 2.5A), the output is permanently de-energized and the error flag is set. - If a setpoint value > 4,000mA has been preset for a current-controlled type A2 PWM output (rated current > 4.0A), the output is permanently de-energized and the error flag is set.	§3614 §3615
If a type A PWM output or a current-controlled PWM output has been configured for the Safe Application, the following applies: The type A PWM output cannot be energized via the application program if the error flag has been set.	§3620 §3621
For a type A output configured as a PWM output or as a current-controlled PWM output for the Safe Application, the following applies: The system only monitors the safe OFF state. The system does not perform a safety monitoring of the pulse-interpulse ratio. For this reason the system behavior and the current values must be subjected to a plausibility check in the Safe Application. The system does not monitor the current control safety functions. For this reason the transient response must be monitored via the Safe Application. Using the safe current measurement functions provided by the system, it must also be monitored whether the current setpoint is actually reached.	§6978 §6833 §6834
If a type A PWM output or a current-controlled PWM output is configured for the Standard Application, the following applies:	§6764 §6765

Safety information and safety requirements – Type A outputs	
- The type A PWM output can be energized via the application program even if the error flag has been set.	
If a type A output is configured for the Safe Application and if the error flag has been set (i.e. the output is de-energized), a diagnostic error including the error cause is saved to the internal error memory.	§3630
If a current flow is detected on a de-energized type A output – irrespective of whether it is configured for the Safe Application or the Standard Application – the control system does not take any measures.	§4910
If the diagnostic function detects an error on a type A output which is configured for the Safe Application, the second shutdown path towards the VIQ1, VIQ2 or VIQ3 power supply opens and the outputs assigned to this group are de-energized. Furthermore, error flags are set for all outputs assigned to the group affected by the error.	§3737 §3738
If an error flag has been set for a type A output configured for the Safe Application due to an overload, a diagnostic error is also saved to the internal error memory.	§1871
Type A class A1, A2, and A3 outputs can switch resistive, capacitive and inductive loads. To avoid diagnostic errors, it must be ensured that capacitive-resistive loads have an electrical discharge time constant of τ < 1.5ms.	§7001 §7002

Table 4-13: Safety information and safety requirements – Type A inputs/outputs

4.6.3 Type B inputs/outputs

Type B inputs/outputs have the following properties and can be configured as follows:

Function	Output	Input	Current measurement	Functionally safe (Cat2)	Functionally safe [Cat3]
Digital output	■	Read back	---	*	Yes
Digital input, pulled down		■	---	No	No

Table 4-14: Properties of type B inputs/outputs

* Not configurable for Cat2 but usable for Cat2-applications.

The output function provides power outputs which can be digitally controlled.

Type B designates digital inputs/outputs with pull down resistor.

The static output state of a type B output can be read back via a digital input. Alternatively, this readback input can be used as a digital input via the application, if it has been configured as such.

A second shutdown path ensures single fault safety of type B outputs. The second shutdown path is implemented as a multi-switch for several outputs. The outputs of a multi-switch represent an output group. A second shutdown path is provided for every *VIQ4* supply connection.

If **one** type B digital input is configured within an output group, all other outputs of this group must be configured as standard outputs (i.e. not as safe outputs!) or as digital inputs in order to avoid negative effects on the outputs. This means that no output of the entire group is available to perform functional safety tasks.

	WARNING
	Use of type B digital inputs Type B digital inputs do **not** fulfill the requirements set out in PL d, Category 2 and 3! If at least **one** of digital input of an input/output group is configured as a type B input, all other outputs of this group must not be used as functionally safe outputs, i.e. in the safe context!

4.6.3.1 Wiring of the central plug

PIN	PIN NAME	Input Digital +	Input Digital -	Input Count	Input Analog	Output Digital +	Output Digital -	Output PWM	Output PWM IREG	VIX / GND	I/O type	Note
1J	Q4.1	■				■					B	Safe Out, Std. In, group 4, 4.0A
1K	Q4.2	■				■					B	Safe Out, Std. In, group 4, 4,0A
1L	Q4.3	■				■					B	Safe Out, Std. In, group 4, 4,0A
1M	Q4.4	■				■					B	Safe Out, Std. In, group 4, 4,0A
2L	Q4.5	■				■					B	Safe Out, Std. In, group 4, 4,0A
2M	Q4.6	■				■					B	Safe Out, Std. In, group 4, 4,0A

Table 4-15: Pin assignment of type B inputs/outputs

4.6.3.2 Technical data

See Chapter 10.13.6.

4.6.3.3 Type B block diagram

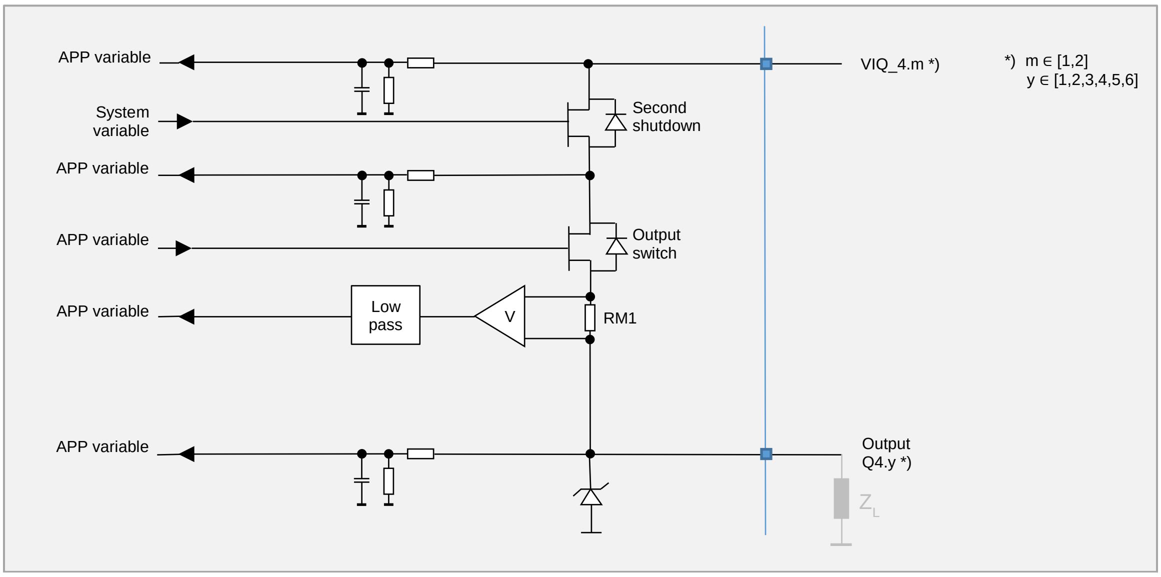


Figure 4-5: Block diagram of type B outputs

4.6.3.4 Functional description

Type B outputs can be configured in groups for both the Safe Application and the Standard Application. The following settings apply:

Safe Application	Standard Application
Safe / Standard / Disabled	
→	Standard / Disabled

This means that the Safe, Standard, and Disabled settings are possible for the Safe Application. The output can only be used in the Standard Application and can be set as Standard or Disabled, if Disabled is selected in the Safe Application.

The outputs are protected by freewheeling diodes.

4.6.3.5 Safe State

⚠ **WARNING**

Safe State

If the plc control is in safe state the output type B are deenergized.

4.6.3.6 Safety information and safety requirements for type B inputs/outputs

⚠ **Safety information and safety requirements – Type B inputs/outputs**	
Type B outputs are "protected outputs" in accordance with EN 61131-2 (Chapter 5.2.2.2.2).	§1634

Safety information and safety requirements – Type B inputs/outputs	
Digital type B outputs fulfill the requirements set out in EN ISO 13849-1:2015 ([N1]), Category 3, Performance Level d.	§1660
Signals connected to type B inputs must be fuse-protected against overload (e.g. in the case of polarity reversal) on the system side. The maximum rated current of the fuse can be up to 3A.	§4976
If the VIQ4 power supply fails in the case of an energized type B digital output and is re-connected afterwards, the output is de-energized once the VIQ4 power supply has recovered (restart protection).	§7081
If the restart protection function blocks the re-energizing of an output after the VIQ4 power supply has failed and recovered, restart is enabled via the Sequence • De-energizing the output via Q_State_Q4 = 0 • Energizing the output n via Q_State_Q4n = TRUE controlled by the application. Prerequisite: VIQ4_Valid = TRUE.	§7082
If a mixed configuration is performed in a type B input/output group, i.e. if the inputs/outputs within the group are configured - for both the Safe Application and the Standard Application, - or for both the Safe Application and as "not used" or "disabled", the system saves a serious diagnostic error to the internal error memory, and automatically switches to the OPMODE_FAILSAFE_IO operating mode upon the start. NOTE: The CODESYS programming system allows this kind of mixed configurations. For use in the Safe Application, however, the type B input/output groups must be completely configured for the Safe Application and as pure outputs!	§6768
All type B outputs have a readback channel which is used to read back the signal on the output terminal. This information is made available to the application programs via the variable I_Q4.I_Current_Q4X (X: 1..4) in the input/output mapping. **PLEASE NOTE: The readback information concerning the output state must <u>not</u> be used in the safe context of the Safe Application!**	§6752
For every type B output the output voltage is captured on the plug-in contact. The following levels apply: Low Level / High Level 24V system: ≤ 4.0V (+0.0V / -0.1V) / ≥ 5.0V (+0.0V / – 0.1V)	§3672 §3673

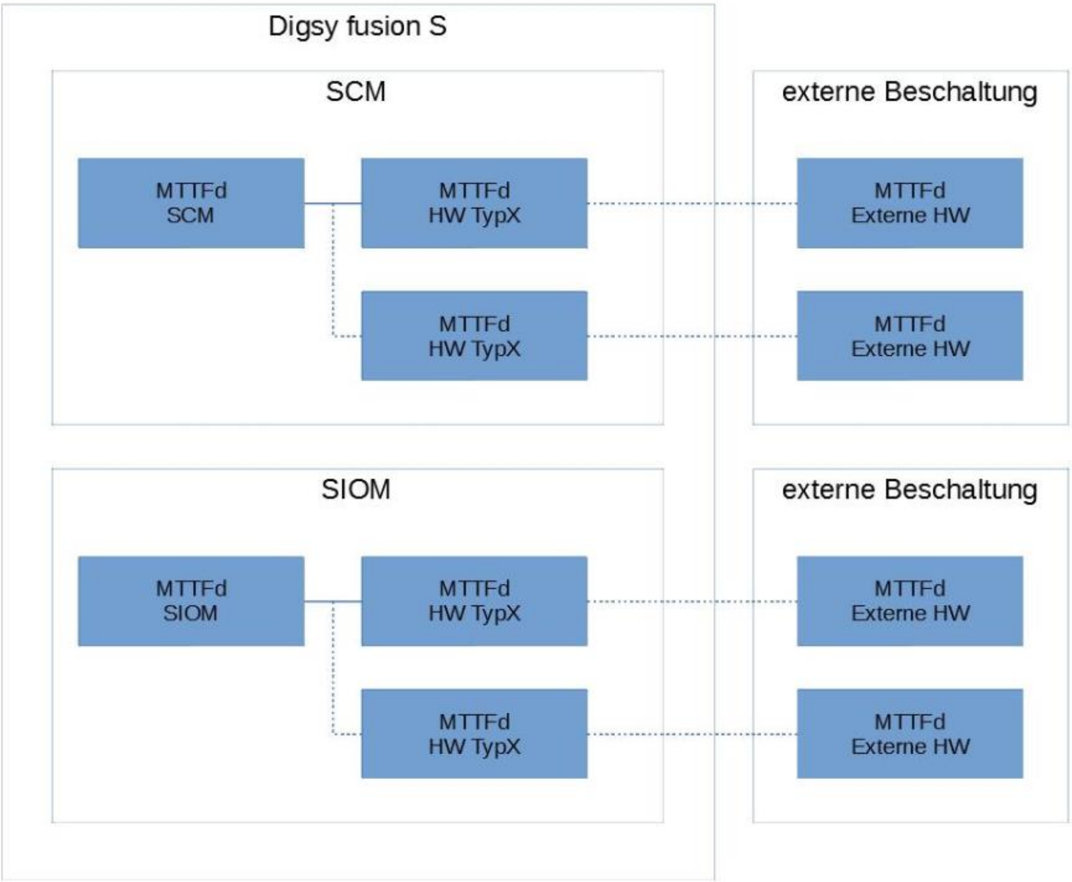
Safety information and safety requirements – Type B inputs/outputs	
The parallel connection of several type B outputs with the purpose of obtaining an output with an increased output current is permissible within a Standard Application. The parallel connection of two type B outputs provides a rated current of 1.7 * 4.0A = 6.8A. The parallel connection of three type B outputs provides a rated current of 2.2 * 4.0A = 8.8A. **PLEASE NOTE: Type B outputs configured for the Safe Application must <u>not</u> be used for parallel connection!**	§3877 §3878
If a type B output is configured as a digital output – irrespective of whether it is used in the Safe Application or the Standard Application – the following applies: - The digital type B output cannot be energized via the application program if – irrespective of the Q_State_Q4i (1 ≤ i ≤ 6) parameter setting – the flag (VIQ4) for its output group has not been set in the VIQ_VALID byte. The digital type B output is automatically de-energized by the system, i.e. without a command issued by the application program, if the VIQ4 flag has been reset in the VIQ_VALID byte. NOTE: This occurs automatically if, for example, VIQ4 is disconnected from the supply voltage outside the control system.	§6747 §3689 §3690
If type B digital inputs are used, the VIQ4 power supply is applied to the corresponding controller terminals on the system side.	§7206
If a type B input/output is configured as a digital output for the Safe Application or the Standard Application, the following applies: - The digital type B output must keep up the new level for at least 50ms upon every level change. The programmer must define this setting in the application program. This is necessary in order to guarantee defined conditions for automatic self-diagnosis. The maximum signal output frequency is thus 10 Hz (= 1 / (2 x 50ms)). - The digital type B output <u>cannot</u> be energized via the application program if its error flag has been set. If the digital output is configured for the safety application, an error (ERR_DIAG_FAST_TOGGLING_TYPE_B) is entered in the system logbook in the event of an error. If the digital output is configured for the standard application, a warning (WARN_DIAG_FAST_TOGGLING_TYPE_B) is entered in the system logbook in the event of an error. In both cases, the output is switched off in the event of a fault.	§3688 §6918
If the diagnostic function detects an error on a type B output which is configured for the Safe Application, the second shutdown path towards the VIQ4 power supply opens and the outputs Q4.1 – Q4.6 assigned to this group are de-energized. Furthermore, the error flag is set for all outputs.	§1987 §1988

Safety information and safety requirements – Type B inputs/outputs	
If the second shutdown path has been activated for the VIQ4 power supply, error flags are set for all Q4.1 to Q4.6 outputs.	
If an error flag has been set for a type B output which is configured for the Safe Application, a diagnostic error is also saved to the internal error memory. For more detailed information on further steps to be performed refer to the "Error Handling" chapter.	§3718
The control system detects an overcurrent if the real current on a type B output exceeds the rated current by a certain value for at least 0.10 s. If the overcurrent persists, the system saves a warning to the error memory after 0.25 s at the latest. Rated current / Overcurrent detection Type B: 4.0A / 5.0A If the application program does not de-energize the output affected by the overcurrent within 0.9s (max. 1.1s), the system automatically de-energizes the output and sets its error flag.	§3716 §3717
Type B outputs can switch resistive, capacitive, and inductive loads. To avoid diagnostic errors, it must be ensured that capacitive-resistive loads have an electrical discharge time constant of τ < 1.5ms.	§3650 §4973

Table 4-16: Safety information and safety requirements – Type B inputs/outputs

4.6.4 MTTF$_d$ values for inputs

The user has to calculate MTTFd values of used inputs.



For one channel the user can calculate the total MTTF$_d$ value by given MTTF$_d$ values of the modules.

The formula is as followed:

$$MTTFd\ (Channel) = \frac{1}{\frac{1}{MTTFd\ (SCM)} + \frac{x}{MTTFd\ (SIOM)} + \frac{a}{MTTFd\ (TypC)} + \frac{b}{MTTFd\ (TypD)} + \frac{c}{MTTFd\ (TypE)}}$$

with:

x: amount of SIOMs

a: amount of used type C inputs

b: amount of used type D inputs

c: amount of used type E inputs

MTTF$_d$ value safety channel

	MTTF$_d$ (per channel) at 85°C	MTTF$_d$ (per channel) at 55°C	MTTF$_d$ (per channel) at 25°C
SCM	94 years	296 years	869 years
SIOM	116,4 years	369,4 years	1069,4 years

	$MTTF_d$ (per channel) at 85°C	$MTTF_d$ (per channel) at 55°C	$MTTF_d$ (per channel) at 25°C
Type C input	4327 years	11613 years	28108 years
Type D input	3693,6 years	10152 years	26549 years
Type E input	1416 years	4457 years	12734 years

Tabelle 4-17: $MTTF_d$ for safety channel

$MTTF_d$ value diagnostic channel

	$MTTF_d$ (per channel) at 85°C	$MTTF_d$ (per channel) at 55°C	$MTTF_d$ (per channel) at 25°C
SCM	141 years	444 years	1436 years
SIOM	259,5 years	802,8 years	2589,27 years
Type C input	1664,8 years	5688 years	19093 years
Type D input	3658,8 years	11427 years	36538 years
Type E input	1115,9 years	3507 years	11157 years

Tabelle 4-18: $MTTF_d$ for test channel

Example: Use of type C inputs for safety function at 85°C

$$MTTFd\ (Safety\ channel) = \frac{1}{\frac{1}{94} + \frac{1}{4327}} = 92{,}34\ years$$

$$MTTFd\ (Test\ channel) = \frac{1}{\frac{1}{141} + \frac{1}{1664{,}8}} = 130\ years$$

4.6.5 Type C inputs

Type C inputs have the following properties can be configured as follows:

Function	Output	Input	Current measurement	Functionally safe (Cat2)	Functionally safe [Cat3]
Digital input, pulled down		■	---	Yes	Yes
Digital input, pulled up		■	---	No	No

Table 4-19: Properties of type C inputs/outputs

4.6.5.1 Performance Level using single channel structure

With following values a classification in a performance level (PL) can be done:

- $MTTF_d$ value (see chapter 4.6.4)
- DC type C: 90% (medium)
- CCF: 65 points

4.6.5.2 Wiring of the central plug

	PIN	PIN NAME	Input Digital +	Input Digital -	Input Count	Input Analog	Output Digital +	Output Digital -	Output PWM	Output PWM I_{REG}	VIX / GND	I/O Type	Note
	4C	I3.1	■	■								C	Safe In, Standard In Low, group 3
	3C	I3.2	■	■								C	Safe In, Standard In Low, group 3
	4D	I3.3	■	■								C	Safe In, Standard In Low, group 3
	3D	I3.4	■	■								C	Safe In, Standard In Low, group 3
	4E	I3.5	■	■								C	Safe In, Standard In Low, group 3
	3E	I3.6	■	■								C	Safe In, Standard In Low, group 3

Table 4-20: Pin assignment of type C inputs

4.6.5.3 Technical data

See Chapter 10.13.7.

4.6.5.4 Type C block diagram

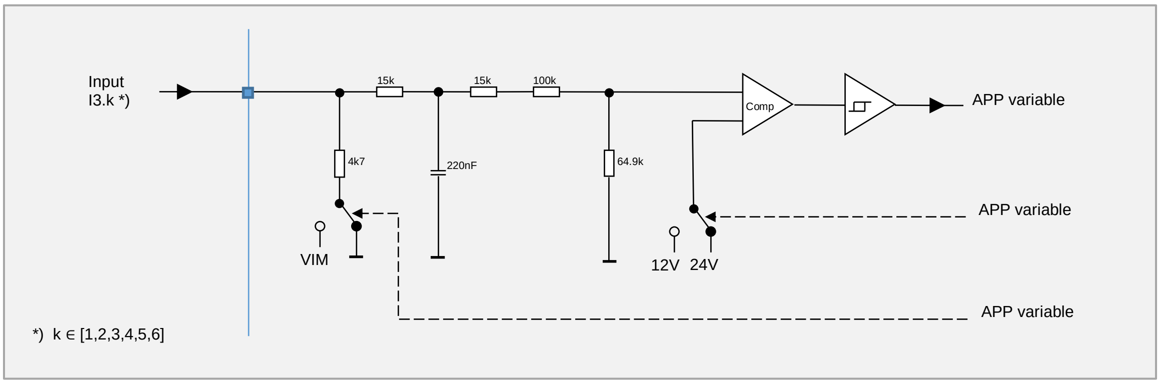


Figure 4-6: Block diagram of type C inputs

4.6.5.5 Functional description

Type C inputs include inputs for the connection of external sensors.

Type C enables the connection of digital sensors with pull up or pull down resistor. The changeover between pull down and pull up is carried out group-wise (Group A / Group B).

To use type C inputs in two-channel mode, they must be connected as described in Figure 4-7.

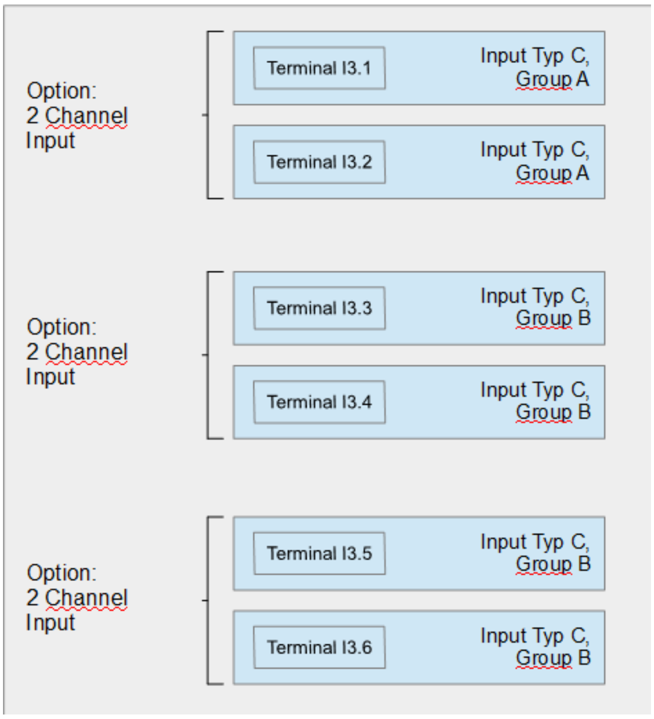


Figure 4-7: Two-channel type C inputs

4.6.5.6 Safe State

	WARNING
⚠	**Safe State** All type C inputs can safely detect the logical 0 (low). In the safe state, all type C inputs deliver the logical 0 (low) to the application (CODESYS). It is not allowed to implement safety functions with a logical 1 (high) as safe state for input type C.

4.6.5.7 Safety information and safety requirements – Type C inputs

Safety information and safety requirements – Type C inputs	
A two-channel structure (e.g. PL d, cat. 3) is possible if the following inputs are coupled: I3.1 and I3.2 I3.3 and I3.4 I3.5 and I3.6. Other pairings of type C inputs must not be used. The CODESYS application program must be used to evaluate the two channels. The low level drivers cannot determine whether a single- or a two-channel structure has been implemented.	§1676 §6855

Safety information and safety requirements – Type C inputs	
Type C inputs in two-channel configuration fulfill the requirements set out in EN ISO 13849-1:2015 ([N1]), Category 3, Performance Level d.	§1690
Digital plus switching inputs of type C correspond in single channel configuration to requirements according to EN ISO 13849-1:2015 ([N1]), category 2	§8102
Digital ground switching inputs of type C in single channel configuration are not safe. Digital plus switching inputs of type C in single channel configuration are safe. Following points has to be considered: 1. Requirement rate must be <= 1/100 times test rate 2. $MTTF_d$ test channel >= 0,5 * $MTTF_d$ functional channel	§1691
Type C inputs are subject to cyclical diagnosis in order to verify whether the inputs can always detect the 0-level. If the 1-level is detected twice during this diagnosis, the error flag of the input is set.	§3758
If a type C input is configured for the Safe Application and if its error flag has been set, a diagnostic error message is saved to the error memory.	§3453
If a type C input is configured for the Standard Application and if the error flag has been set, a warning is saved to the error memory.	§3456

Table 4-21: Safety information and safety requirements – Type C inputs

4.6.6 Type D inputs

Type D inputs have the following properties and can be configured as follows:

Function	Output	Input	Current measurement	Functionally safe (Cat 2)	Functionally safe (Cat 3)
Digital input, pulled down		■	---	Yes	Yes
Counter input		■	---	*	Yes
AB counter input		■	---	No	Yes
Pulse length measurement input		■	---	No	Yes
Phase measurement input (only I1.3, I1.4, I1.7, and I1.8)		■	---	No	No

Table 4-22: Properties of the D input type
*see safety instruction §6856

4.6.6.1 Performance Level using single channel structure

With following values a classification in a performance level (PL) can be done:

- MTTFd values (see chapter 4.6.4)
- DC type D:
  - digital inputs: 90% (medium)
  - counter inputs: 60% (low)
- CCF: 65 points

4.6.6.2 Wiring of the central plug

PIN	PIN NAME	Input Digital +	Input Digital -	Input Count	Input Analog	Output Digital +	Output Digital -	Output PWM	Output PWM I...	VIX / GND	I/O Type	Note
1C	I1.1	■		■							D	Safe In, group 1, AB counter 1A
1D	I1.2	■		■							D	Safe In, group 1, AB counter 1B
2C	I1.3	■		■							D	Safe In, group 1, AB counter 2A
2D	I1.4	■		■							D	Safe In, group 1, AB counter 2B
2E	I1.5	■		■							D	Safe In, group 1, AB counter 3A
2F	I1.6	■		■							D	Safe In, group 1, AB counter 3B
2G	I1.7	■		■							D	Safe In, group 1, AB counter 4A
2H	I1.8	■		■							D	Safe In, group 1, AB counter 4B

Table 4-23: Pin assignment of type D inputs

4.6.6.3 Technical data

See Chapter 10.13.8.

4.6.6.4 Type D block diagram

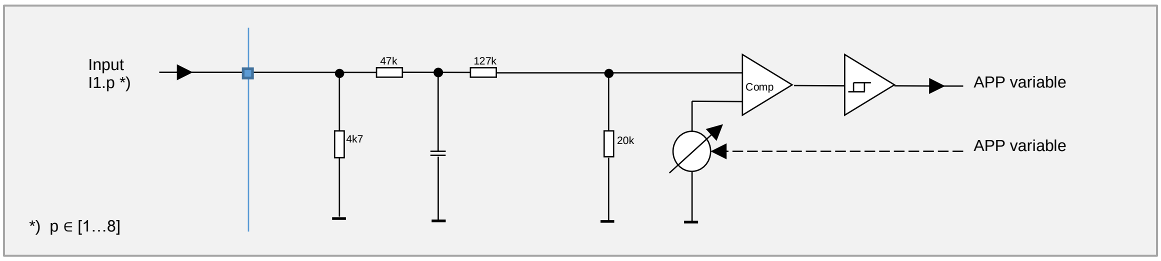


Figure 4-8: Block diagram of type D inputs

4.6.6.5 Functional description

Type D inputs are inputs for the connection of external sensors. The inputs can be operated as both digital and counter / pulse inputs.

If the inputs are combined in pairs, AB signals can be captured, and the speed and the direction of rotation as well as the phase assignment / phase shift of two signals can be recorded.

The switching threshold can be set group-wise in order to be able to adapt it to the clock signals of pulse generators (see Figure 7-48). The threshold setting signal is measured back for diagnostic purposes.

	Note
	Using the type D inputs on the SIOM When using the type D inputs on the SIOM, the VIM must be connected to the power supply on the SIOM. Otherwise startup tests fail, which set the error flags of the inputs and enter a diagnostic error "ERR_TYPE_STARTUP_DIAG_WRONG_CNT_TYPE_D" in the system log.

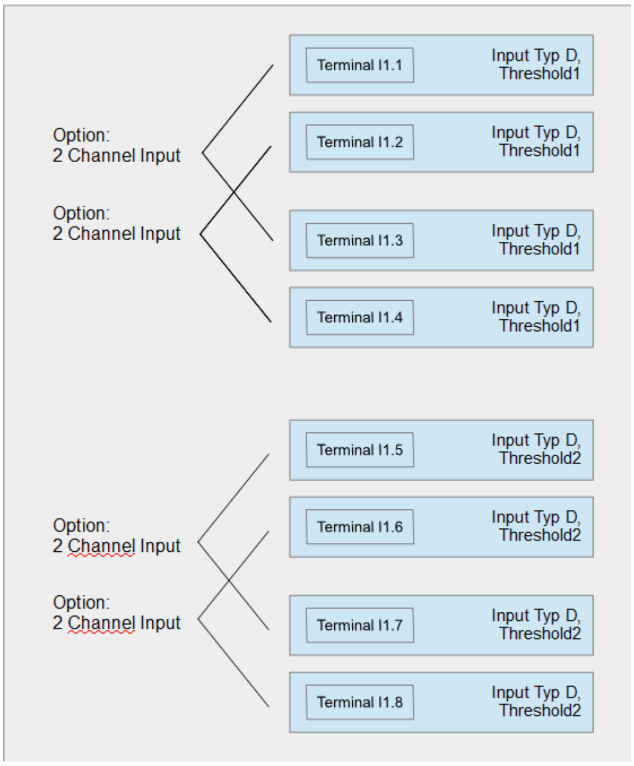


Figure 4-9: Type D topology "single-terminal input"

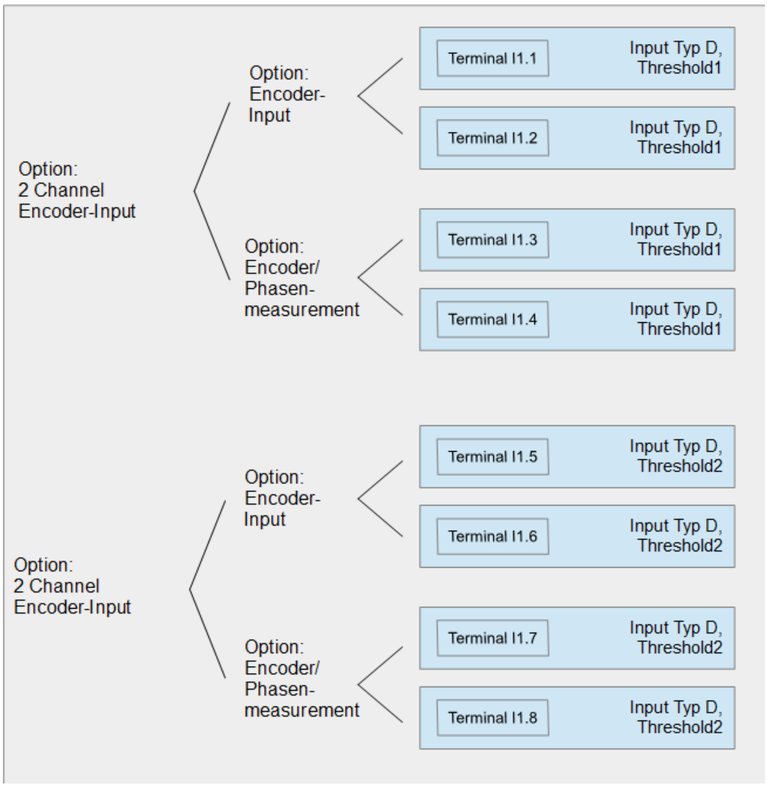


Figure 4-10: Type D topology "two-terminal input"

4.6.6.6 Safe State

WARNING

Safe State

All type D inputs could safely detect the logical 0 (low). In the safe state, all type D inputs deliver the logical 0(low) to the application (CODESYS).
It is not allowed to implement safety functions with a logical 1 (high) as safe state for input type D.

4.6.6.7 Safety information and safety requirements – Type D inputs

Safety information and safety requirements – Type D inputs	
Type D inputs in two-channel configuration fulfill the requirements set out in EN ISO 13849-1:2015 ([N1]), Category 3, Performance Level d.	§3764
Type D digital inputs in the single-channel configuration fulfill the requirements set out in EN ISO 13849-1:2015 ([N1]), Category 2. Following point have to be considered: 1. Requirement rate must be <= 1/100 times test rate 2. $MTTF_d$ test channel >= 0,5 * $MTTF_d$ functional channel	§3765 §8103

Safety information and safety requirements – Type D inputs	
In order to ensure that type D inputs comply with the input levels set out in EN 61131-2 Type 3, the switching threshold (in the variables Q_Threshold1 and Q_Threshold2) must be set to 6.400V.	§3819
For type D inputs in the Pulse Length Measurement mode, the maximum input signal frequency depends on the number of inputs configured for this purpose. - The maximum input signal frequency must not exceed 1kHz if up to two measuring inputs from the I1.1, I1.2, I1.5, I1.6 group or from the I1.3, I1.4, I1.7, I1.8 terminal group are used at the same time. - The maximum input signal frequency must not exceed 500Hz if three or four measuring inputs of the I1.1, I1.2, I1.5, I1.6 group or the I1.3, I1.4, I1.7, I1.8 group are used at the same time.	§4694 §4722
For a type D input in the Period Measuring mode, a period of 65535µs is interpreted as exceeding the measuring limit.	§3844
Irrespective of whether - a static high signal or - a static low signal is present on type D inputs upon the start of pulse length measurement, the value ZERO is first transmitted to the application program as a measured value for the pulse duration (variable I_Counter_Pulse) and the period duration (variable I_Freq_Period). NOTE: When measuring the positive pulse duration, it does not need to be considered that the measuring limit is exceeded, because the period duration is a 16-bit value and the shortest pulse duration is a 32-bit value.	§3866, §3869
The phase measurement between two inputs is one of the possible operating modes for type D inputs. PLEASE NOTE: Phase measurement is not a safe application!	§5375
For type D inputs in the Phase Measurement mode, the maximum input signal frequency depends on the number of inputs configured for this purpose, and on their use for pulse length measurement. - The maximum input signal frequency must not exceed 1kHz if only one channel is used for phase measurement on the module concerned and no pulse length measurement is performed. - The maximum input signal frequency must not exceed 500Hz if both channels on the board are used for phase measurement. - The maximum input signal frequency must not exceed 500Hz if a phase measurement and up to two pulse length measurements are active on the board.	§4690 §5354 §4691

Safety information and safety requirements – Type D inputs	
For type D inputs in the Phase Measurement mode, the phase shift is the signal lagging on channel B in relation to the reference signal on channel A. Only the positive values in the 0° - 359.9° range are considered as phase shift.	§5380 §5381
If one of the type D inputs I1.3, I1.4, I1.7 or I1.8 is configured as "Phase Measurement Standard" and if the input assigned to the phase measurement is not configured as "Phase Measurement Standard",the controller switches to the OPMODE_FAILSAFE_IO state and saves a message for a serious error in the internal error memory. Combinations for phase measurement: - I1.3 + I1.4 - I1.7 + I1.8	§5378
If the error flag has been set for a type D input suitable for phase measurement, which is configured as a non-safety-relevant phase measurement input ("Phase Measurement standard"), the phase measurement is nevertheless evaluated as normal. The result values are mapped in the I_Counter_Pulse (Channels A and B) and I_Freq_Period (only Channel A) variables.	§3926
Type D inputs are subject to cyclical diagnosis in order to verify whether the inputs can always detect the 0-level. If the 1-level is detected twice during this diagnosis, the error flag of the input is set.	§3919
Type D inputs are subject to cyclical diagnosis in order to verify whether the voltage set for the switching threshold corresponds to the default setpoint value. If a deviation greater than 250 mV has been detected twice, the error flag is set for all entries belonging to this group.	§3925
If a type D input is configured for the Safe Application and if its error flag has been set, a diagnostic error message is saved to the error memory. For more detailed information on further steps to be performed refer to the "Error Handling" chapter.	§3920
If a type D input is configured for the Standard Application and if its error flag has been set, a warning is saved to the error memory.	§3921

Safety information and safety requirements – Type D inputs	
If a type D input has been configured for the Safe Application and if a single-channel topology is required, e.g. in order to fulfill the requirements set out in EN ISO 13849-1:2015 ([N1]), cat. 2, it must be verified for all possible configurations of the respective application – apart from the configuration as a digital input – whether a test rate of the set maximum uninterrupted operation time (1-90 days) is sufficient in order to detect potential errors to the required extent. NOTE: Alternatively, a plausibility check can be performed for single-channel systems, e.g. between the setpoint definition for a motor control and the back-measured counting rate.	§6856
On request of the application program, an encoder component transmits the direction of rotation to the I_StateDir variable. If the last time window was incremented, I_StateDir has the value 1 or TRUE. If the last time window was decremented, I_StateDir has the value 0 or FALSE.	§3927

Table 4-24: Safety information and safety requirements – Type D inputs

4.6.7 Type E inputs

Type E inputs have the following properties and can be configured as follows:

Function	Output	Input	Current measurement	Functionally safe (Cat 2)	Functionally safe (Cat 3)
Digital input, pulled down		■	---	*	Yes
Analog input		■	---	*	Yes

Table 4-25: Properties of the E input type

* see safety requirement §7372

4.6.7.1 Performance Level using single channel structure

With following values a classification in a performance level (PL) can be done:

- $MTTF_d$ value (see chapter 4.6.4)
- DC type E: 90% (medium) according to §7372, otherwise 87% (low)
- CCF: 65 points

4.6.7.2 Wiring of the central plug

PIN	PIN NAME	Input Digital +	Input Digital -	Input Count	Input Analog	Output Digital +	Output Digital -	Output PWM	Output PWM I_{REG}	VIX / GND	I/O Type	Note
4F	I2.1	■			■						E	Safe In, group 2
3F	I2.2	■			■						E	Safe In, group 2
4G	I2.3	■			■						E	Safe In, group 2
3G	I2.4	■			■						E	Safe In, group 2
4H	I2.5	■			■						E	Safe In, group 2
3H	I2.6	■			■						E	Safe In, group 2
4J	I2.7	■			■						E	Safe In, group 2
3J	I2.8	■			■						E	Safe In, group 2
4K	I2.9	■			■						E	Safe In, group 2
3K	I2.10	■			■						E	Safe In, group 2
4L	I2.11	■			■						E	Safe In, group 2
3L	I2.12	■			■						E	Safe In, group 2
4M	I2.13	■			■						E	Safe In, group 2
3M	I2.14	■			■						E	Safe In, group 2
4N	I2.15	■			■						E	Safe In, group 2
3N	I2.16	■			■						E	Safe In, group 2

Table 4-26: Pin assignment of type E inputs

4.6.7.3 Technical data

See chapter 10.13.9.

4.6.7.4 Type E block diagram

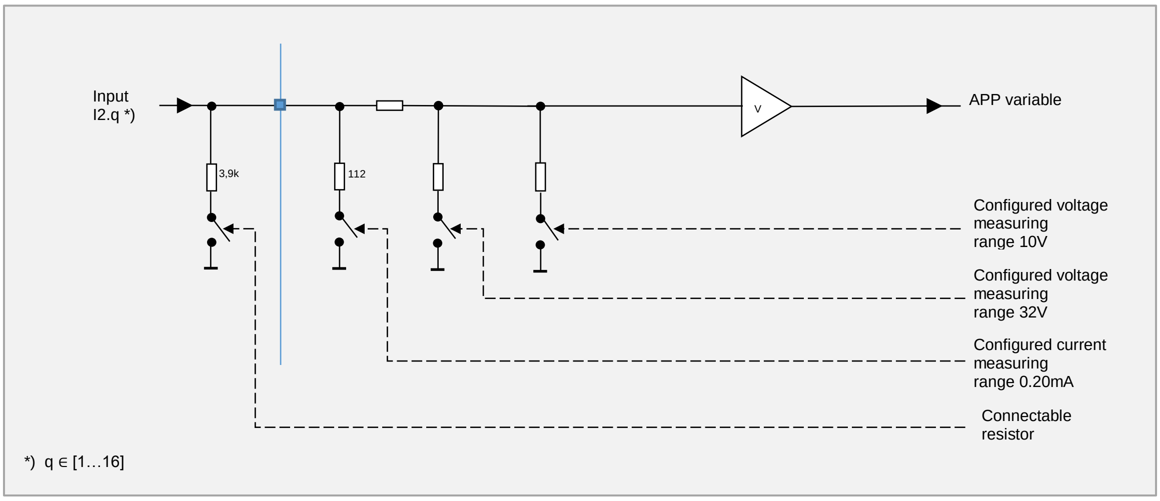


Figure 4-11: Block diagram of type E inputs

4.6.7.5 Functional description

Type E inputs/outputs include dedicated inputs for the connection of external sensors. The inputs can be operated as both digital and analog inputs.

Both digital sensors with pull down resistors and analog sensors for the 0 –10V, 0 – 32V, and 0 – 20mA ranges can be used. The actual measuring ranges and the measuring ranges which are required to ensure functional safety exceed the limit values mentioned above (e.g. 0…22mA or 0…11V).

4.6.7.6 Safe State

	WARNING
⚠	**Safe State** Digital type E inputs could safely detect the logical 0 (low). In the safe state, digital type E inputs deliver the logical 0 (low) to the application (CODESYS). It is not allowed to implement safety functions with a logical 1 (high) as safe state for digital inputs type E. Safety functions with analog inputs (type E) are allowed within the safety measurement range (see chapter 10.13.9)

4.6.7.7 Safety information and safety requirements – Type E inputs

⚠ **Safety information and safety requirements – Type E inputs**	
Inputs of type E are safe in single-channel topology. 1. Requirement rate must be <= 1/100 times test rate 2. $MTTF_d$ test channel >= 0,5 * $MTTF_d$ functional channel	§2023 §2024

Safety information and safety requirements – Type E inputs	
Inputs of type E inputs in two-channel configuration fulfill the requirements set out in EN ISO 13849-1:2015 ([N1]), Category 3, Performance Level d.	
dual-channel input type E For the case of a two-channel input (Cat. 3) for type E, the application shall compare the two measured values and initiate safety measures in the event of a deviation (to be defined by the user). See chapter 5.1.2 or table Table 5-4: Combination of *SCM* inputs. single-channel input type E For the case of a single-channel input (Cat. 2) for type E, a reference voltage (eg UREF1 or UREF2 to the affected safety group I2.1 ... I2.8 or I2.9 ... I2.16 and in the application) is to be applied to plausibility The application should then measure this reference voltage and compare the measured values with the expected values. In the event of a deviation (to be defined by the user) the user must initiate safety measures. Example for a single-channel input: A single-channel sensor (Cat. 2) is connected to I2.1. Then UREF1 may be connected to one of the free inputs of the same safety group (I2.2 ... I2.8) e.g. I2.2. In the application, the connected reference voltage UREF1 must be switched on and checked for the expected value of 10V. The reference voltage UREF1 (10V) is generated with a tolerance of ± 1% and read back with a tolerance of ± 1% at the analog input I2.2 (32V), thus resulting in a deviation of. ± 1% (UREF1) = 100mV ± 1% (I2.2) = 320mV = ± 420mV when reading back the reference voltage at analog input I2.2. This could result in the application checking the reference voltage for validity with "UREF1 (I2.2) <9.58V = error" or "UREF1 (I2.2)> 10.42V = error", but the user can also use a defined higher deviation, depending on its application. In case of a fault, the user must initiate appropriate measures.	§7372
The input resistances R_{IN} of the type E inputs (digital input and analog input 32V) can be activated/deactivated.	§3503
The inputs I2.1 to I2.8 belong to safety group 1. The inputs I2.9 to I2.16 belong to safety group 2.	§2027
The input resistance on type E current inputs (0 – 20mA) is 112Ω – 114Ω.	§2036
The input resistance is > 24kΩ on type E voltage inputs (0 – 10V, 0 – 32V).	§2037
On digital type E inputs the system detects a signal in the: - 0 – 5V +0.1V/-0.0V voltage range as LOW. - > 6.8 V +0.0V/-0.1V voltage range as HIGH.	§2068 §2069
Analog type E inputs measure voltages in the 0 – 10V or 0 – 32V range over the entire temperature range with a precision of ±1% in relation to the full scale values indicated above.	§2105 §3548

⚠ Safety information and safety requirements – Type E inputs	
The safety precision of the analog voltage inputs is ±4.5% in relation to the full scale values indicated above.	
If a voltage measurement for the Safe Application is configured for an analog type E input, the full measuring range must not be used. The following applies: - Configured measuring range 0 – 10V → Safe measuring range 0.5 – 10.0V Unusable measuring range: 0.0 – 0.5V - Configured measuring range 0 – 32V → Safe measuring range 1.5 – 32.0V Unusable measuring range 0.0 – 1.5V. A voltage value within the unusable measuring range must be evaluated as an error by the application program.	§3516 §3517
Analog type E inputs measure currents in the 0 – 20mA range over the entire temperature range (in relation to §7423) with a precision of ±1,5% in relation to the full scale values indicated above. The safety precision of the analog current inputs is ±6% in relation to the full scale values indicated above.	§7414 §3549
The current values are invalid at temperatures of more than 80°C (internal temperature sensor). The user must monitor the temperature. If the temperature limit is exceeded, the safety function for safe current measurement must be enabled.	§7423
If a current measurement for the Safe Application is configured for an analog type E input, the full measuring range must not be used. The following applies: - Configured measuring range 0 – 20mA → Safe measuring range 1.0 – 20.0mA Unusable measuring range 0.0 – 1.0mA. A current value in the unusable measuring range must be evaluated as an error by the application program.	§3520
The type E current inputs are protected against overload, for example in the case of incorrect wiring or short circuits. The measuring shunt of the current inputs is disconnected (the input becomes a high resistance input). The following applies: - From an input current > 30.0mA - From an input voltage ≥ +6.0V - From an input voltage ≤ -2.5V ±0.2V If one of the cases above occurs, the system sets the error flag of the input concerned and saves a diagnostic error to the internal error memory (Shunt Protection Fuse Activated).	§3531 §2124 §3403 §3404

⚠ Safety information and safety requirements – Type E inputs	
If a type E input – for the Safe Application or the Standard Application – is configured as a 10V voltage input, the following applies: - If a voltage > 10.75V is measured for a period of 50ms, the error flag of the input concerned is set and a diagnostic error is saved to the internal error memory (input overload).	§3521
If a type E input – for the Safe Application or the Standard Application – is configured as a 32V voltage input, the following applies: - If a voltage > 32.5V is measured for a period of 50ms, the error flag of the input concerned is set and a diagnostic error is saved to the internal error memory (input overload).	§3522
If a type E input – for the Safe Application or the Standard Application – is configured as a 20mA current input, the following applies: - If a current ≥ 21.75mA is measured for a period of 50ms, the error flag of the input concerned is set and a diagnostic error is saved to the internal error memory.	§3523
Type E inputs are subject to continuous diagnosis. If this diagnosis detects a measuring deviation > 3% of the full scale value (10V, 32V, or 20mA) twice on an input configured for the Safe Application, the error flag is set for this input and a diagnostic error is saved to the error memory.	§3536
If a type E input is configured as an analog input for the Safe Application and if its error flag has been set, the system transmits the input value ZERO for this input to the application program. If a type E input is configured as a digital input for the Safe Application and if its error flag has been set, the system transmits the input value LOW for this input to the application program.	§3538 §3540
If a type E input is configured as an analog input for the Standard Application and if its error flag has been set, the system transmits the input value measured for this input to the application program. If a type E input is configured as a digital input for the Standard Application and if its error flag has been set, the system transmits the input value measured for this input to the application program.	§6924 §6925
If a type E input is configured for the Safe Application and if its error flag has been set, the system does not reset the error flag even if the cause of the error no longer exists. To reset the error flag, the control system must be shut down and started again.	§3539
If a type E input has been configured for the Safe Application and if a single-channel topology is required, e.g. in order to fulfill the requirements set out in EN ISO 13849-1:2015 ([N1]), cat. 2, it must be verified for all possible configurations of	§8098

Safety information and safety requirements – Type E inputs	
the respective application – apart from the configuration as a digital input – whether a test rate of the set maximum uninterrupted operation time (1-90 days) is sufficient in order to detect potential errors to the required extent. NOTE: Alternatively, a plausibility check can be performed for single-channel systems.	

Table 4-27: Safety information and safety requirements – Type E inputs

4.7 Connecting sensors and actuators

Sensors equipped with standard interfaces are connected to digital or analog device inputs. In addition, sensors can be connected via an RS232 interface. Bus-capable sensors must be connected via the CAN bus.

In order to be able to connect sensors for functional safety applications, special safety sensors, or alternatively two single sensors, which fulfill the requirements of the application must be used.

The input information must be compared within the scope of the application program.

	Note
	Connection of functional safety sensors The sensors used for functional safety applications must be chosen depending on the individual safety requirements. To fulfill the requirements set out in PL d Category 2, single-channel sensors can be used and connected to inputs intended for this purpose. To fulfill the requirements set out in PL d Category 3, two sensors **must** be used. Two safety groups are available for connecting the sensors, whereby one of the sensors must be assigned to each safety group!

Actuators can be connected to digital or PWM-device outputs. In addition, actuators can be connected via an RS232 interface. Bus-capable actuators must be connected via the CAN bus.

	Note
	Connection of functional safety actuators Actuators used for functional safety applications must be connected to outputs providing a second shutdown path!

4.8 Sensor supply

4.8.1 Q_SENS sensor supply output

The Q_SENS sensor supply output is used to power sensors, switches, etc. The supply voltage is a "protected" VIM voltage. The output can be switched and read back (analog).

	Note

	Functional safety The Q_SENS sensor supply output fulfills the requirements set out in PL d, Category 3.

4.8.2 Q_SENS block diagram

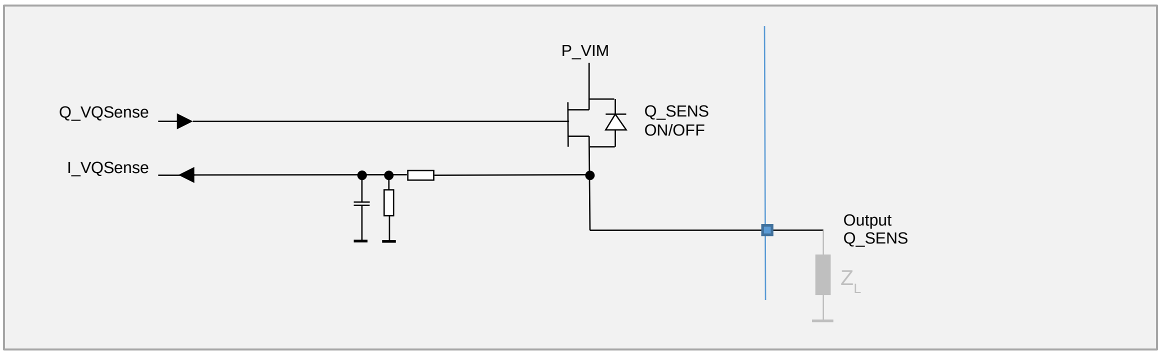


Figure 4-12: Q_SENS block diagram

For the technical data refer to chapter 10.13.10.

4.8.3 Safety information and safety requirements – Q_SENS sensor supply output

Safety information and safety requirements – Q_SENS sensor supply output	
The Q_SENS output for the supply of external sensors fulfills the requirements set out in EN ISO 13849-1:2015 ([N1]), Category 3, Performance Level d.	§4381
The Q_SENS output is exclusively used for the supply of external sensors. This output must not be used for controlling safety-relevant actuators or safety-relevant inputs!	§6848
If continuous safety monitoring is required for Q_SENS, this must be performed in the application program. The I_VQSense variables (readback of the voltage present on Q_SENS) and P_VIM (readback of the voltage supplying Q_SENS) are available for this purpose.	§6849
The Q_SENS output for the supply of external sensors is powered with the P_VIM voltage, i.e. an overvoltage- and overcurrent-protected voltage from the VIM. The Q_SENS output can be energized and de-energized via a dedicated switch. For diagnostic purposes the output voltage on Q_SENS is cyclically compared with P_VIM. The measuring result on Q_SENS is mapped to the application program via the I_VQSense variable. If the condition (P_VIM – 1,3V) ≤ Q_SENSE ≤ (P_VIM + 0.5V) has been violated, the system transmits the value ZERO with the I_VQSense variable and saves a warning to the internal error memory. From this time onwards, I_VQSense uses once again the value measured on Q_SENS, provided that the value of Q_SENS is within the permitted voltage window.	§4397

Table 4-28: Safety information and safety requirements – Q_SENS sensor supply output

4.8.4 IREF reference current source

The IREF reference current source is used to supply sensors, etc. with a defined current. This output is switchable.

4.8.5 IREF block diagram

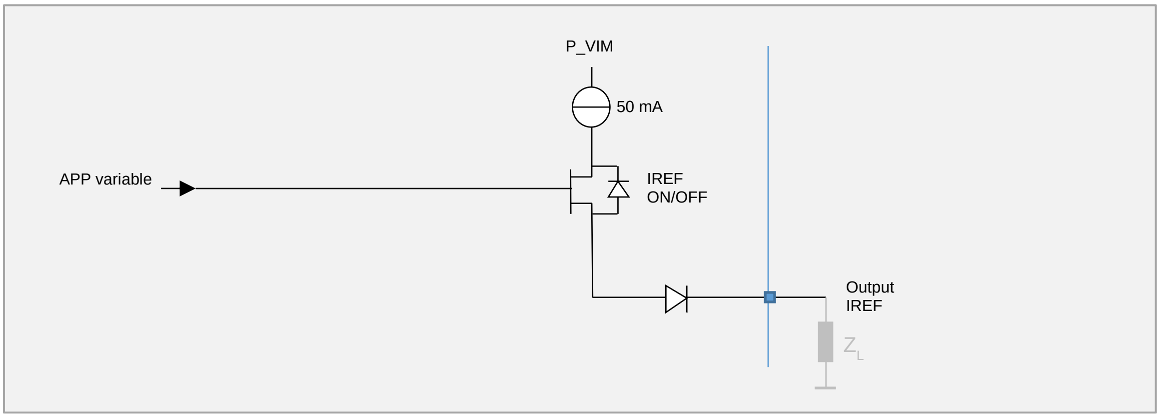


Figure 4-13: IREF block diagram

For the technical data refer to chapter 10.13.12.

4.8.6 Safety information and safety requirements – IREF reference current source

WARNING

Current source of the IREF sensor supply

The current source of the IREF sensor supply does <u>not</u> fulfill the requirements set out in EN ISO 13849-1:2015 ([N1]) standard. Further measures must be initiated in order to be able to use it for a safety function.

4.8.7 UREF_x reference voltage sources

The UREF_1 and UREF_2 reference voltage sources may be used to supply sensors, switches, etc. with a defined voltage. The output has switching and readback capability (analog).

UREF_1 and UREF_2 are supplied via the VIM. The two reference voltage sources only operate properly if the VIM supply voltage is ≥ 11V.

Note

Functional safety

The UREF_x reference voltage sources fulfill the requirements set out in PL d, Category 3.

4.8.8 UREF_x block diagram

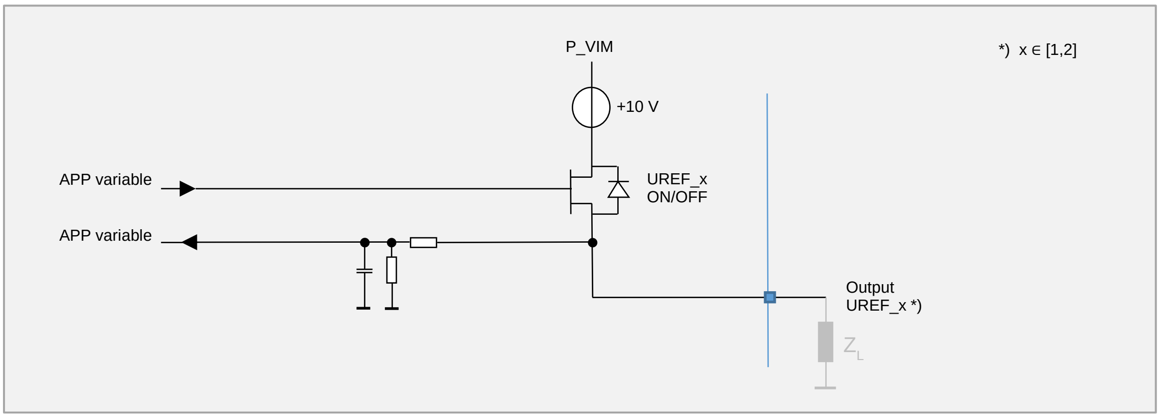


Figure 4-14: UREF_x block diagram

For the technical data refer to chapter 10.13.11.

4.8.9 Safety information and safety requirements – UREF_x reference voltage sources

Safety information and safety requirements – UREF_x reference voltage sources	
The UREF_1 and UREF_2 reference voltage outputs fulfill the requirements set out in EN ISO 13849-1:2015 ([N1]), Category 3, Performance Level d. The safety precision of the output voltage is ±1,5%.	§4594
The UREF_1 and UREF_2 outputs are exclusively used to supply external sensors. These outputs must not be used for controlling safety-relevant actuators or safety-relevant inputs!	§6869
If continuous safety monitoring is required for UREF_1 and UREF_2, this must be performed via the application program. The I_URef1 and I_URef2 variables (reference voltage readback) are available for this purpose.	§6876
4888.02.XXX product version before V03.36.XX **4888.03.XXX product version before V01.06.XX** The UREF_x (x = 1.2) reference voltage outputs are subject to system monitoring and are survived by the read back inputs UREF_FB_x. If in the switched on state for the read back voltage UREF_FB_x the condition 9.9V ≤ UREF_FB_x ≤ 10.1V is violated for UREF_FB_x, the following applies: - The system saves a warning to its internal error memory. - The control system retains its current operating mode. - The current measured values of the I_URefx reference voltages are mapped in the application program.	§5492

Safety information and safety requirements – UREF_x reference voltage sources	
<u>NOTE:</u> Due to the accuracy of the UREF read back (±1%), a voltage between 9.8V and 10.2V can be present at the output terminal in the event of a fault without a warning message being entered in the internal error memory (see also §6876). **4888.02.XXX from product version V03.36.XX** **4888.03.XXX from product version V01.06.XX** **4888.04.XXX** The UREF_x (x = 1, 2) reference voltage outputs are subject to system monitoring and are survived by the read back inputs UREF_FB_x. If in the switched on state for the read back voltage UREF_FB_x the condition 9.85V ≤ UREF_FB_x ≤ 10.15V is violated for UREF_FB_x, the following applies: - The system saves a warning to its internal error memory. - The control system retains its current operating mode. - The current measured values of the I_URefx reference voltages are mapped in the application program. <u>NOTE:</u> Due to the accuracy of the UREF read back (±1%), a voltage between 9.75V and 10.25V can be present at the output terminal in the event of a fault without a warning message being entered in the internal error memory (see also §6876).	

Table 4-29: Safety information and safety requirements – UREF_x Reference voltage sources

5 Modules

5.1 Safety Controller Module (SCM)

5.1.1 General

The *Safety Controller Module (SCM)* is a discrete control system with the following properties:

- 32-bit Safety Controller
- 2-channel µC in the Lockstep Mode
- Floating Point Unit (FPU)
- On-board LED state indicator
- Internal bus connection for system communication
- PL d, category 2 or category 3 in accordance with EN ISO 13849-1:2015 ([N1])
- Application programming via CODESYS V3.5 SIL 2 from the company 3S
- Functionally safe runtime system CODESYS V3.5 SIL 2 (SRTS) for Safe Application program
- Runtime system CODESYS V3.5 SIL 2 (SRTS) for the Standard Application program (independent of SRTS)
- FAT32 file system
- Self-diagnosis in accordance with safety requirements
- SCM: 2 x RS232 interface (RS232-1 = Diagnosis interface, RS232-2 = freely programmable)
- 4x CAN interface
- 1x USB interface
- 1x Ethernet interface (option, diagnosis, and service interface)
- Various types of inputs/outputs (see Table 5-3)
- On-board power supply

Memory	Size	Usage
On-Chip Flash-Memory	3MB	FW (safe, standard) safe application (size see 3.2.1) standard application (variant T2, size see 3.2.1)
On-Chip RAM- Memory	256kB	safe RAM
On-Board Flash- Memory (parallel)	32MB	internal filesystem 16MB standard application (variant T1, size see 3.2.1) Bootloader
On-Board SDRAM	16MB	shared memory 96kB for user 2 x 8kB avaiable (see chapter 8.1.13 / 8.2.7) Standard RAM
On-Board FRAM- Memory	32kB	retain data 16kB (2x 8kB) (see Chapter 7.5.7.5) systemlogbook 14kB
On-Board Flash- Memory (serial)	64MB	userlogbook 64MB

Table 5-1: SCM memory

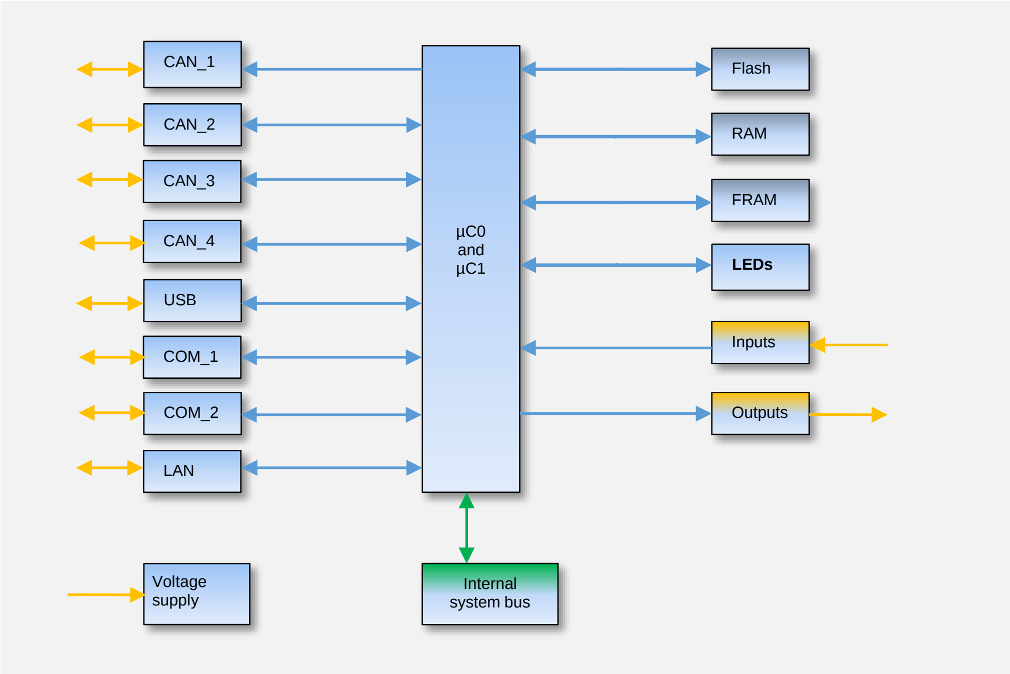


Figure 5-1: SCM block diagram

5.1.2 Pin assignment

5.1.2.1 M12 plug connectors

All the communication interfaces are connected via M12 plug connectors.

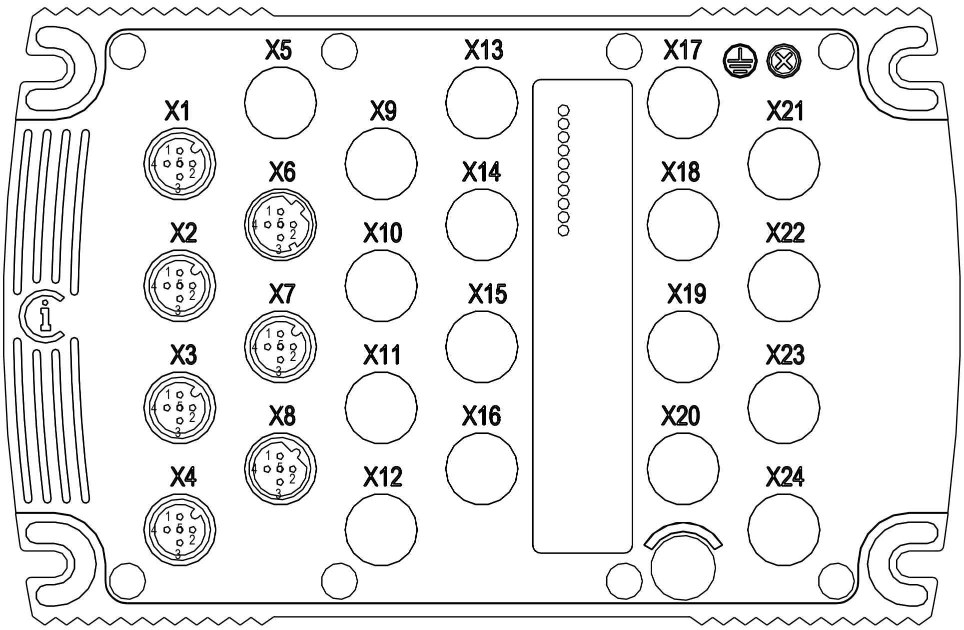


Figure 5-2: M12 plug connectors for SCM

Interface	Signal	Plug	Pin No.	Note
CAN_1	CAN_1_SHLD	X1	1	M12 socket, A-encoded
	---	X1	2	(NC)
	CAN_1_GND	X1	1 + 3	CAN_1_SHLD = CAN_1_GND
	CAN_1_H	X1	4	
	CAN_1_L	X1	5	
CAN_2	CAN_2_SHLD	X2	1	M12 socket, A-encoded
	---	X2	2	(NC)
	CAN_2_GND	X2	1 + 3	CAN_2_SHLD = CAN_2_GND
	CAN_2_H	X2	4	
	CAN_2_L	X2	5	
CAN_3	CAN_3_SHLD	X3	1	M12 socket, A-encoded
	---	X3	2	(NC)
	CAN_3_GND	X3	1 + 3	CAN_3_SHLD = CAN_3_GND
	CAN_3_H	X3	4	
	CAN_3_L	X3	5	
CAN_4	CAN_4_SHLD	X4	1	M12 socket, A-encoded
	---	X4	2	(NC)
	CAN_4_GND	X4	1 + 3	CAN_4_SHLD = CAN_4_GND
	CAN_4_H	X4	4	
	CAN_4_L	X4	5	
ETHERNET	Eth_TX+	X6	1	M12 socket, D-encoded
	Eth_RX+	X6	2	
	Eth_TX-	X6	3	
	Eth_RX-	X6	4	
	---	X6	5	(NC)
RS232	COM2_TX	X7	1	M12 socket, A-encoded
	COM1_TX	X7	2	
	GND	X7	3	
	COM1_RX	X7	4	
	COM2_RX	X7	5	
USB	VUSB	X8	1	M12 socket, B-encoded
	USB_D-	X8	2	
	USB_D+	X8	3	
	GND	X8	4	
	USB_Mode	X8	5	Special Inter Control Mode

Table 5-2: Pin assignment of the M12 plug connectors for *SCM*

5.1.2.2 Central plug connector

A 64-pin central plug connector supplies the electronics (i.e. the logic) and the circuit breakers, and provides for the connection of all the inputs/outputs.

For the pin assignment refer to Table 5-3.

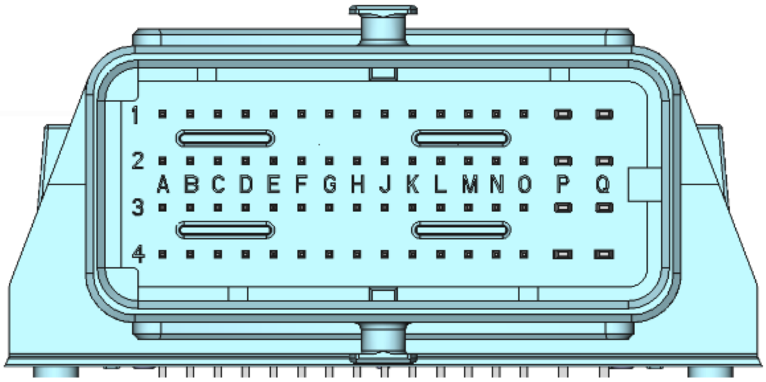


Figure 5-3: Central plug connector

	PIN	PIN NAME	Input Digital +	Input Digital -	Input Count	Input Analog	Output Digital +	Output Digital -	Output Ratiometer	Output PWM	Output PWM IREG	VIX / GND	I/O Type	Note
	3P	VIQ_1										■		Power supply group 1
	3B	Q1.1					■			■	■		A1	Safe Out, group 1, 2.5A
	3A	Q1.2					■			■	■		A1	Safe Out, group 1, 2.5A
	4A	Q1.3					■			■	■		A1	Safe Out, group 1, 2.5A
	4B	Q1.4					■			■	■		A1	Safe Out, group 1, 2.5A
	2P	VIQ_2										■		Power supply group 2
	2A	Q2.1					■			■	■		A3	Safe Out, group 2, 2.5A
	2B	Q2.2					■			■	■		A3	Safe Out, group 2, 2.5A
	1B	Q2.3					■			■	■		A2	Safe Out, group 2, 4A
	1A	Q2.4					■			■	■		A2	Safe Out, group 2, 4A
	2Q	VIQ_3										■		Power supply group 3
	2O	Q3.1					■			■	■		A3	Safe Out, group 3, 2.5A
	2N	Q3.2					■			■	■		A3	Safe Out, group 3, 2.5A
	1N	Q3.3					■			■	■		A2	Safe Out, group 3, 4A
	1O	Q3.4					■			■	■		A2	Safe Out, group 3, 4A
	1Q	VIQ_4.1										■		Power supply group 4
	3Q	VIQ_4.2										■		Power supply group 4
	1J	Q4.1	■				■						B	Safe Out, Std. In, group 4, 4,0A
	1K	Q4.2	■				■						B	Safe Out, Std. In, group 4, 4,0A
	1L	Q4.3	■				■						B	Safe Out, Std. In, group 4, 4,0A
	1M	Q4.4	■				■						B	Safe Out, Std. In, group 4, 4,0A
	2L	Q4.5	■				■						B	Safe Out, Std. In, group 4, 4,0A
	2M	Q4.6	■				■						B	Safe Out, Std. In, group 4, 4,0A
	1C	I1.1	■		■								D	Safe In, group 1, AB counter 1A
	1D	I1.2	■		■								D	Safe In, group 1, AB counter 1B
	2C	I1.3	■		■								D	Safe In, group 1, AB counter 2A
	2D	I1.4	■		■								D	Safe In, group 1, AB counter 2B
	2E	I1.5	■		■								D	Safe In, group 1, AB counter 3A
	2F	I1.6	■		■								D	Safe In, group 1, AB counter 3B

PIN	PIN NAME	Input Digital +	Input Digital -	Input Count	Input Analog	Output Digital +	Output Digital -	Output Ratiometer	Output PWM	Output PWM I_{REG}	VIX / GND	I/O Type	Note
2G	I1.7	■		■								D	Safe In, group 1, AB counter 4A
2H	I1.8	■		■								D	Safe In, group 1, AB counter 4B
4F	I2.1	■			■							E	Safe In, group 2
3F	I2.2	■			■							E	Safe In, group 2
4G	I2.3	■			■							E	Safe In, group 2
3G	I2.4	■			■							E	Safe In, group 2
4H	I2.5	■			■							E	Safe In, group 2
3H	I2.6	■			■							E	Safe In, group 2
4J	I2.7	■			■							E	Safe In, group 2
3J	I2.8	■			■							E	Safe In, group 2
4K	I2.9	■			■							E	Safe In, group 2
3K	I2.10	■			■							E	Safe In, group 2
4L	I2.11	■			■							E	Safe In, group 2
3L	I2.12	■			■							E	Safe In, group 2
4M	I2.13	■			■							E	Safe In, group 2
3M	I2.14	■			■							E	Safe In, group 2
4N	I2.15	■			■							E	Safe In, group 2
3N	I2.16	■			■							E	Safe In, group 2
4C	I3.1	■	■									C	Safe In, Standard In Low, group 3
3C	I3.2	■	■									C	Safe In, Standard In Low, group 3
4D	I3.3	■	■									C	Safe In, Standard In Low, group 3
3D	I3.4	■	■									C	Safe In, Standard In Low, group 3
4E	I3.5	■	■									C	Safe In, Standard In Low, group 3
3E	I3.6	■	■									C	Safe In, Standard In Low, group 3
1P	VIM_1										■		Supply for logic + inputs
4O	IPON_1										■		Connection of terminal 15
3O	GND_1										■		Ground 1 (logic + inputs)
1G	Q_SENS										■		Sensor supply output
1H	IREF										■		Reference current
1E	UREF_1										■		Reference voltage 1
1F	UREF_2										■		Reference voltage 2
2J	GND_A1										■		Ground – analog 1
2K	GND_A2										■		Ground – analog 2
4P	GND_2										■		Ground 2 (leading contact!)
4Q	GND_1										■		Ground 3 (leading contact!)
SUM	Total: Safe:	36 30	6 0	8 8	16 16	18 18	0 0	0 0	12 12	12 12	16	48	

Table 5-3: Pin assignment of the *SCM central plug connector*

Note

Explanations concerning the pin assignment of the central plug connector

1. The Q1.x – Q4.x. output groups have a different color coding on the left to assign them to the related voltage supply path (VIQ_x).
2. The inputs/outputs usable for safety applications and their functions are highlighted in yellow.

	CAUTION
	Overload of the VIQ_n [n = 1…4] supply connections The control system can be damaged if an overload occurs on the VIQ_n supply connections. 1. The supply connections VIQ_1 for output group Q1.x VIQ_2 for output group Q2.x VIQ_3 for output group Q3.x can be loaded up to the predefined rated current over the permissible temperature range. 2. SPECIAL FEATURE: Due to the high rated total current, two supply connections, i.e. VIQ_4.1 and VIQ_4.2, are provided for the Q4.x output groups. The two supply connections, i.e. VIQ_4.1 and VIQ_4.2, must be connected to the supply voltage in the same manner in order to ensure that the total input current of the Q4.x group is equally distributed over both connections. The supply connections VIQ_4.1 and VIQ_4.2 for the Q4.x output group may be loaded up to the predefined rated current over the permissible temperature range.

For applications in accordance with EN ISO 13849-1:2015 ([N1]), PL d, category 3 – i.e. two-channel information acquisition – the sensors must be connected in pairs to the inputs of an SCM. The special regulations which apply to the use of these inputs must be observed. To ensure this, the inputs are assigned to the 1 and 2 safety groups. For the two-channel acquisition of information for category 3 applications, one of the two sensors must be connected to an input of safety group 1, and the other sensor must be connected to an input of safety group 2. The inputs used must be configured for an operating mode which is suitable for functional safety applications. Functional safety in accordance with PL d, category 3, is not ensured if this specification is not observed!

Sensors with AB clock pulse outputs and sensor pairs for phase angle measurement can also be used. Both the processing of AB signals by the control systems and phase angle measurement are standard functions. Even though they do not fulfill the functional safety requirements, they can be used on the application level in the safe context e.g. after a plausibility check. Make sure that special purpose inputs are used for this type of application!

For more detailed information refer to Table 5-4.

	Safety group 1	Safety group 2	AB counter combinations	Phase measurement combinations	Note
Type C inputs	I3.2	I3.1			Pull up/pull down switchover applies to I3.1 and I3.2
	I3.4	I3.3			Pull up/down switchover applies to I3.3 and I3.4, I3.5, and I3.6
	I3.6	I3.5			
Type D inputs	I1.1	I1.3	I1.1 + I1.2		The switching thresholds can be adjusted separately for the groups I1.1 – 1.4 and I1.5 – 1.8
	I1.2	I1.4	I1.3 + I1.4	I1.3 + I1.4	
	I1.5	I1.7	I1.5 + I1.6		
	I1.6	I1.8	I1.7 + I1.8	I1.7 + I1.8	
Type E inputs	I2.1	I2.9			PLEASE NOTE: The operating mode (i.e. digital / 0 – 10V / 0 – 32V / 0 – 20mA) is set with the input pairs listed here in one of these table rows (i.e. I2.1 + I2.9, I2.2 + I2.10, etc.).
	I2.2	I2.10			
	I2.3	I2.11			
	I2.4	I2.12			
	I2.5	I2.13			
	I2.6	I2.14			
	I2.7	I2.15			
	I2.8	I2.16			

Table 5-4: Combination of *SCM* inputs

5.1.3 External interfaces

An *SCM* provides the following communication interfaces:

Interface	Physical	Transmission rate	Protocol 1	Protocol 2	Note
CAN_1	ISO11898 11 + 29-bit ID	Like CANopen-Standard DS 301	Proprietary (layer 2)	CANopen, CANopen Safety*)	J1939 library on the basis of layer 2
CAN_2	ISO11898 11 + 29-bit ID	Like CANopen-Standard DS 301	Proprietary (layer 2)	CANopen, CANopen Safety*)	J1939 library on the basis of layer 2
CAN_3	ISO11898 11 + 29-bit ID	Like CANopen-Standard DS 301	Proprietary (layer 2)	CANopen, CANopen Safety*)	J1939 library on the basis of layer 2
CAN_4	ISO11898 11 + 29-bit ID	Like CANopen-Standard DS 301	Proprietary (layer 2)	CANopen, CANopen Safety*)	J1939 library on the basis of layer 2
COM_1	RS232	Standard 300 - 115,200 baud	CODESYS proprietary	---	Programming and diagnosis interface
COM_2	RS232	Standard 300 - 115,200 baud	Freely programmable	---	User interface
USB	USB	USB V2.0 (Full Speed)	Host: Mass Storage Profile	Device: CODESYS diagnosis	USB stick: 1. System update 2. File system
Ethernet	PHY	10/100 Mbit/s	TCP/IP, UDP	---	

Table 5-5: *SCM* external interfaces

*) CANopen Safety Stack is based on the CODESYS CANopen Stack from 3S.

Note

The CAN1 to CAN4 interfaces are not provided with a sensor supply and have no internal resitance.

5.1.4 Firmware

The SCM is an independent, freely programmable control system. It is programmed via CODESYS V3.5 SIL 2 (exact version see chapter 1.6) from 3S. This programming tool meets the requirements set out in IEC 61131-3 and is also certified for safety applications in accordance with SIL 2 (IEC 61508).

The SCM provides a platform for two application programs:

- Functional safety application program (Safe Application).
- Application program for tasks without safety requirements (Standard Application).

The two programs can be developed independently of each other via CODESYS V3.5 SIL 2.

The SCM firmware includes the following components:

- Boot loader
- Start-up self-diagnosis
- Runtime self-diagnosis
- CODESYS runtime system for the Safe Application
- CODESYS runtime system for the Standard Application
- CAN stack for free programming on layer 2 level for 4 CAN interfaces
- CANopen and CANopen Safety via CODESYS stack
- RS232 driver for 2 COM interfaces
- FW modules to support all the existing input/output components
- File system for on-board flash memory and USB
- USB host for Mass Storage Profile
- TCP/IP / UDP stack

The CODESYS V3.5 SIL 2 runtime system implemented in accordance with the specifications of the 3S company is set up on a real time operating system from our own development.

5.1.5 Functional safety

The SCM fulfills all the functional safety requirements in accordance with PL d, category 3 (EN ISO 13849-1:2015 ([N1])). Exception: certain input/output combinations. For more detailed information refer to chapter 7.7 and 4.6.

An *SCM* is able to control and monitor all the processes within a modular *digsy*®$_{fusion}$control system (i.e. including those of additional components such as the *SIOM*), and to transfer the entire system into a failsafe state.

WARNING

In consequence of a serious or fatal error, the SCM shifts to the safe state (OPMODE_FAILSAFE_STOP or OPMODE_FAILSAFE_IO).

The safe state (OPMODE_FAILSAFE_STOP or OPMODE_FAILSAFE_IO) is the de-energized state.

5.2 Safety Controller Module Second Layer SCM SL

The ***digsy***®$_{fusion}$ S-P(L) uses a SCM SL instead of the SCM as a controller module. The SCM-SL is, with some differences, an SCM. These differences are described in the following chapter.

5.2.1 General

Unlike the SCM, the SCM SL has only one RS232 interface. The second RS232 interface is available for the installed GCM-P.

The RS232 interface of the SCM SL can be configured as a diagnostics or freely programmable interface (see chapter 7.1.5).

5.2.2 Pin assignment

5.2.2.1 M12 plug connectors

The M12 pin assignment differs between SCM and SCM SL, because the interfaces of the GCM-P were outwarded.

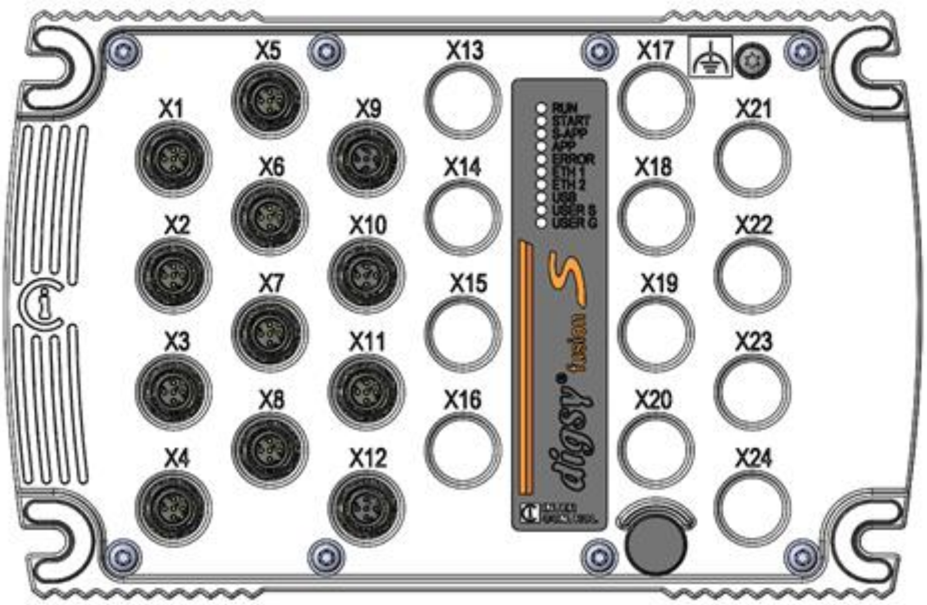


Figure 5-4: M12 plug connectors for SCM SL

Interface	Signal	Plug	Pin No.	Note
CAN_1	CAN_1_SHLD	X1	1	M12 socket, A-encoded
(SCM)	---	X1	2	(NC)
	CAN_1_GND	X1	1 + 3	CAN_1_SHLD = CAN_1_GND
	CAN_1_H	X1	4	
	CAN_1_L	X1	5	
CAN_2	CAN_2_SHLD	X2	1	M12 socket, A-encoded
(SCM)	---	X2	2	(NC)
	CAN_2_GND	X2	1 + 3	CAN_2_SHLD = CAN_2_GND
	CAN_2_H	X2	4	
	CAN_2_L	X2	5	
CAN_3	CAN_3_SHLD	X3	1	M12 socket, A-encoded
(SCM)	---	X3	2	(NC)
	CAN_3_GND	X3	1 + 3	CAN_3_SHLD = CAN_3_GND
	CAN_3_H	X3	4	

Interface	Signal	Plug	Pin No.	Note
	CAN_3_L	X3	5	
CAN_4	CAN_4_SHLD	X4	1	M12 socket, A-encoded
(SCM)	---	X4	2	(NC)
	CAN_4_GND	X4	1 + 3	CAN_4_SHLD = CAN_4_GND
	CAN_4_H	X4	4	
	CAN_4_L	X4	5	
CAN_1 (GCM-P)	CAN_1_SHLD	X5	1	M12 socket, A-encoded
	V_BUFF_RTC	X5	2	buffer voltage RTC[1]
	CAN_1_GND	X5	1 + 3	CAN_1_SHLD = CAN_1_GND
	CAN_1_H	X5	4	
	CAN_1_L	X5	5	
CAN_2 (GCM-P)	CAN_2_SHLD	X6	1	M12 socket, A-encoded
	---	X6	2	buffer voltage RTC[1]
	CAN_2_GND	X6	1 + 3	CAN_2_SHLD = CAN_2_GND
	CAN_2_H	X6	4	
	CAN_2_L	X6	5	
CAN_3 (GCM-P)	CAN_3_SHLD	X7	1	M12 socket, A-encoded
	---	X7	2	buffer voltage RTC[1]
	CAN_3_GND	X7	1 + 3	CAN_3_SHLD = CAN_3_GND
	CAN_3_H	X7	4	
	CAN_3_L	X7	5	
CAN_4 (GCM-P)	CAN_4_SHLD	X8	1	M12 socket, A-encoded
	---	X8	2	buffer voltage RTC[1]
	CAN_4_GND	X8	1 + 3	CAN_4_SHLD = CAN_4_GND
	CAN_4_H	X8	4	
	CAN_4_L	X8	5	
ETHERNET1	Eth_TX+	X9	1	M12 socket, D-encoded
	Eth_RX+	X9	2	
	Eth_TX-	X9	3	
	Eth_RX-	X9	4	
	---	X9	5	(NC)
ETHERNET2	Eth_TX+	X10	1	M12 socket, D-encoded
	Eth_RX+	X10	2	
	Eth_TX-	X10	3	
	Eth_RX-	X10	4	
	---	X10	5	(NC)
RS232	COM2_TX	X11	1	M12 socket, A-encoded
COM1=SCM	COM1_TX	X11	2	
COM2=GCM-P	GND	X11	3	
	COM1_RX	X11	4	
	COM2_RX	X11	5	
USB	VUSB	X12	1	M12 socket, B-encoded

[1] Pin for feeding an external buffer voltage for the internal RTC

Interface	Signal	Plug	Pin No.	Note
	USB_D-	X12	2	
	USB_D+	X12	3	
	GND	X12	4	
	USB_Mode	X12	5	Special mode Inter Control

Table 5-6: Pin assignment M12 plug connectors *SCM SL*

5.2.3 External Interfaces

An SCM SL is equipped with the following interfaces for communication:

Interface	Physics	Transmission rate	Protocol 1	Protocol 2	Note
CAN_1 SCM SL	ISO11898 11- u. 29-Bit-ID	Like CANopen-Standard DS 301	proprietary (Layer 2)	CANopen, CANopen Safety*)	J1939-Bibliothek Based on Layer 2
CAN_2 SCM SL	ISO11898 11- u. 29-Bit-ID	Like CANopen-Standard DS 301	proprietary (Layer 2)	CANopen, CANopen Safety*)	J1939-Bibliothek Based on Layer 2
CAN_3 SCM SL	ISO11898 11- u. 29-Bit-ID	Like CANopen-Standard DS 301	proprietary (Layer 2)	CANopen, CANopen Safety*)	J1939-Bibliothek Based on Layer 2
CAN_4 SCM SL	ISO11898 11- u. 29-Bit-ID	Like CANopen-Standard DS 301	proprietary (Layer 2)	CANopen, CANopen Safety*)	J1939-Bibliothek Based on Layer 2
COM_1	RS232	Standard 300… 115.200 Baud	CODESYS proprietary	---	Programming- and diagnostic interface / user interface
USB	USB	USB V2.0 (Full Speed)	Host: Mass storage profile	Device: Diagnose CODESYS	USB-Device: 1. System-Update 2. File-System
Ethernet	PHY	10/100 MBit/s	TCP/IP, UDP	---	

Table 5-7: *SCM SL External Interfaces*

*) CANopen Safety Stack is based on the CODESYS CANopen Stack from 3S.

	Note
i	The interfaces CAN1 to CAN4 of the SCM SL have no sensor supply and no internal termination resistance.

All other M12 connections are available for the GCM-P and are described in [MAN-GCM-P].

5.3 SIOM safety I/O modules

5.3.1 General

The *SIOM* module is intended to extend the inputs/outputs of a ***digsy***®fusion S. The *SIOM* provides the same number and quality of inputs/outputs as an *SCM*. These inputs/outputs can be used for functional safety applications.

Properties of the *SIOM*:

- 32-bit Safety Controller
- 2-channel µC in the Lockstep Mode
- On-chip flash memory 3MB incl. ECC
- On-chip RAM 256 kB incl. ECC
- Internal bus connection for system communication

- PL d, category 2, or category 3 in accordance with EN ISO 13849-1:2015 ([N1])
- Self-diagnosis in accordance with safety requirements
- Various types of inputs/outputs (see Table 5-8)
- On-board power supply

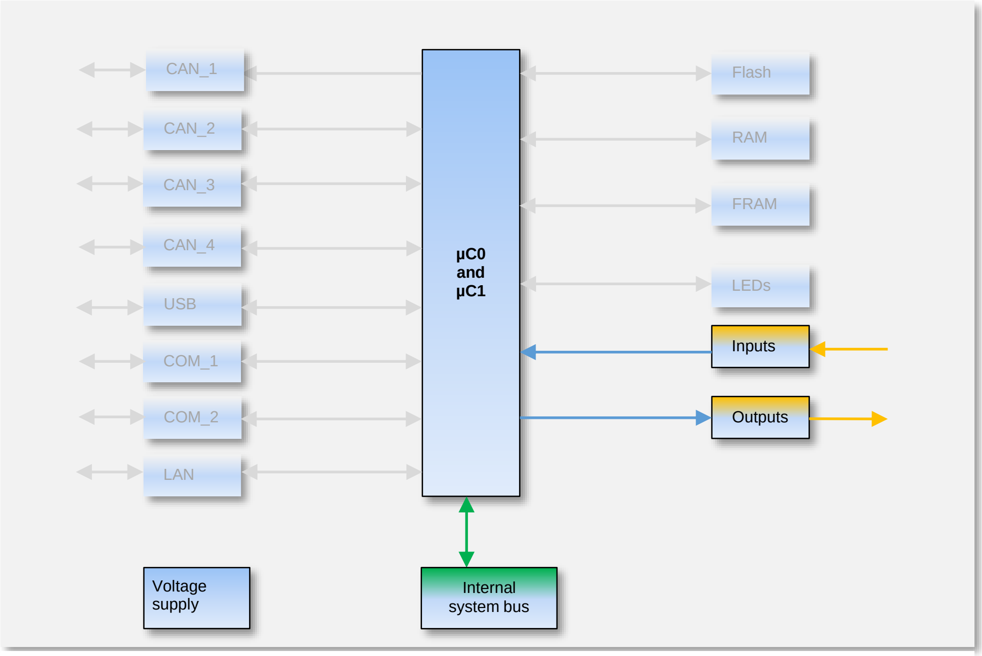


Figure 5-5: SIOM block diagram

5.3.2 Pin assignment of the central plug connector

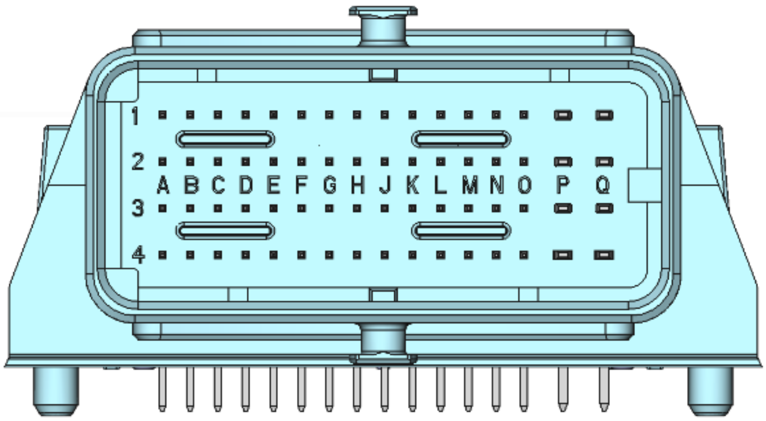


Figure 5-6: Central plug connector

	PIN	PIN NAME	Input Digital +	Input Digital -	Input Count	Input Analog	Output Digital +	Output Digital -	Output Ratiometer	Output PWM	Output PWM IREG	VIX / GND	I/O Type	Note
	3P	VIQ_1										■		Power supply group 1
	3B	Q1.1					■			■	■		A1	Safe Out, group 1, 2.5A

PIN	PIN NAME	Input Digital +	Input Digital -	Input Count	Input Analog	Output Digital +	Output Digital -	Output Ratiometer	Output PWM	Output PWM I_{REG}	VIX / GND	I/O Type	Note
3A	Q1.2					■			■	■		A1	Safe Out, group 1, 2.5A
4A	Q1.3					■			■	■		A1	Safe Out, group 1, 2.5A
4B	Q1.4					■			■	■		A1	Safe Out, group 1, 2.5A
2P	VIQ_2										■		Power supply group 2
2A	Q2.1					■			■	■		A3	Safe Out, group 2, 2.5A
2B	Q2.2					■			■	■		A3	Safe Out, group 2, 2.5A
1B	Q2.3					■			■	■		A2	Safe Out, group 2, 4A
1A	Q2.4					■			■	■		A2	Safe Out, group 2, 4A
2Q	VIQ_3										■		Power supply group 3
2O	Q3.1					■			■	■		A3	Safe Out, group 3, 2.5A
2N	Q3.2					■			■	■		A3	Safe Out, group 3, 2.5A
1N	Q3.3					■			■	■		A2	Safe Out, group 3, 4A
1O	Q3.4					■			■	■		A2	Safe Out, group 3, 4A
1Q	VIQ_4.1										■		Power supply group 4
3Q	VIQ_4.2										■		Power supply group 4
1J	Q4.1	■				■						B	Safe Out, Std. In, group 4, 4,0A
1K	Q4.2	■				■						B	Safe Out, Std. In, group 4, 4,0A
1L	Q4.3	■				■						B	Safe Out, Std. In, group 4, 4,0A
1M	Q4.4	■				■						B	Safe Out, Std. In, group 4, 4,0A
2L	Q4.5	■				■						B	Safe Out, Std. In, group 4, 4,0A
2M	Q4.6	■				■						B	Safe Out, Std. In, group 4, 4,0A
1C	I1.1	■		■								D	Safe In, group 1, AB counter 1A
1D	I1.2	■		■								D	Safe In, group 1, AB counter 1B
2C	I1.3	■		■								D	Safe In, group 1, AB counter 2A
2D	I1.4	■		■								D	Safe In, group 1, AB counter 2B
2E	I1.5	■		■								D	Safe In, group 1, AB counter 3A
2F	I1.6	■		■								D	Safe In, group 1, AB counter 3B
2G	I1.7	■		■								D	Safe In, group 1, AB counter 4A
2H	I1.8	■		■								D	Safe In, group 1, AB counter 4B
4F	I2.1	■			■							E	Safe In, group 2
3F	I2.2	■			■							E	Safe In, group 2
4G	I2.3	■			■							E	Safe In, group 2
3G	I2.4	■			■							E	Safe In, group 2
4H	I2.5	■			■							E	Safe In, group 2
3H	I2.6	■			■							E	Safe In, group 2
4J	I2.7	■			■							E	Safe In, group 2
3J	I2.8	■			■							E	Safe In, group 2
4K	I2.9	■			■							E	Safe In, group 2
3K	I2.10	■			■							E	Safe In, group 2
4L	I2.11	■			■							E	Safe In, group 2
3L	I2.12	■			■							E	Safe In, group 2
4M	I2.13	■			■							E	Safe In, group 2
3M	I2.14	■			■							E	Safe In, group 2
4N	I2.15	■			■							E	Safe In, group 2
3N	I2.16	■			■							E	Safe In, group 2
4C	I3.1	■	■									C	Safe In, Standard In Low, group 3
3C	I3.2	■	■									C	Safe In, Standard In Low, group 3
4D	I3.3	■	■									C	Safe In, Standard In Low, group 3
3D	I3.4	■	■									C	Safe In, Standard In Low, group 3
4E	I3.5	■	■									C	Safe In, Standard In Low, group 3
3E	I3.6	■	■									C	Safe In, Standard In Low, group 3
1P	VIM_1										■		Supply for logic + inputs

	PIN	PIN NAME	Input Digital +	Input Digital -	Input Count	Input Analog	Output Digital +	Output Digital -	Output Ratiometer	Output PWM	Output PWM I_{REG}	VIX / GND	I/O Type	Note
	4O	IPON_1										■		Connection of terminal 15
	3O	GND_1										■		Ground 1 (logic + inputs)
	1G	Q_SENS										■		Sensor supply output
	1H	IREF										■		Reference current
	1E	UREF_1										■		Reference voltage 1
	1F	UREF_2										■		Reference voltage 2
	2J	GND_A1										■		Ground – analog 1
	2K	GND_A2										■		Ground – analog 2
	4P	GND_2										■		Ground 2 (leading contact!)
	4Q	GND_1										■		Ground 3 (leading contact!)
	SUM	Total: Safe:	36 30	6 0	8 8	16 16	18 18	0 0	0 0	12 12	12 12	16	48	

Table 5-8: Pin assignment of the central plug connector of an *SIOM*

Note

Explanations concerning the pin assignment of the central plug connector

1. The Q1.x – Q4.x. output groups have a different color coding on the left to assign them to the corresponding voltage supply path (VIQ_x).
2. The inputs/outputs usable for safety applications and their functions are highlighted in yellow.

Depending on the individual application, all the inputs suitable for safety applications can be used for applications in accordance with EN ISO 13849-1:2015 ([N1]), PL d, category 2.

For applications in accordance with EN ISO 13849-1:2015 ([N1]), PL d, category 3 – i.e. two-channel information acquisition – the sensors must be connected in pairs to the inputs of an SCM. The special regulations which apply to the use of these inputs must be observed. To ensure this, the inputs are assigned to the 1 and 2 safety groups. For the two-channel acquisition of information for category 3 applications, one of the two sensors must be connected to a safety group 1 input, and the other sensor must be connected to a safety group 2 input. The inputs used must be configured for an operating mode which is suitable for functional safety applications (see Table 5-9). Functional safety in accordance with PL d, category 3, is not ensured if this specification is not observed!

Sensors with AB clock pulse outputs and sensor pairs for phase angle measurement can also be used. The AB signal processing by the control system and the phase angle measurement are standard functions. Although they do not comply with the functional safety requirements, they may be used in the safe context, e.g. after a plausibility check, on the application level. Make sure that special purpose inputs are used for this type of application!

For more detailed information refer to Table 5-9.

	Safety group 1	Safety group 2	AB counter combinations	Phase measurement combinations	Note
Type C inputs	I3.2	I3.1			Pull up/down switchover applies to I3.1 and I3.2
	I3.4	I3.3			Pull up/down switchover applies to I3.3, I3.4, I3.5, and I3.6
	I3.6	I3.5			
	I1.1	I1.3	I1.1 + I1.2		

	Safety group 1	Safety group 2	AB counter combinations	Phase measurement combinations	Note
Type D inputs	I1.2	I1.4	I1.3 + I1.4	I1.3 + I1.4	The switching thresholds can be adjusted separately for the groups I1.1 – 1.4 and I1.5 – 1.8
	I1.5	I1.7	I1.5 + I1.6		
	I1.6	I1.8	I1.7 + I1.8	I1.7 + I1.8	
Type E inputs	I2.1	I2.9			PLEASE NOTE: The operating mode (i.e. digital / 0 – 10V / 0 – 32V / 0 – 20mA) is set with the input pairs listed here in one of these table rows (i.e. I2.1 + I2.9, I2.2 + I2.10, etc.).
	I2.2	I2.10			
	I2.3	I2.11			
	I2.4	I2.12			
	I2.5	I2.13			
	I2.6	I2.14			
	I2.7	I2.15			
	I2.8	I2.16			

Table 5-9: Combination of *SIOM* inputs

5.3.3 External interfaces

The *SIOM* does **not** provide communication interfaces (RS232/ CAN/ Ethernet).

5.3.4 Firmware

The *SIOM* is a non-autonomous, fixedly programmed module which requires an *SCM* (SL) for normal operation.

The *SIOM* firmware includes the following components:

- Boot loader
- Start-up self-diagnosis
- Runtime self-diagnosis
- Function-dependent, fixed system program
- FW modules to support all the existing input/output components

5.3.5 Functional safety

The *SIOM* fulfills the functional safety requirements set out in PL d (EN ISO 13849-1:2015 ([N1])). Exception: certain input/output combinations. For more detailed information refer to chapters 7.7 and 4.6.

WARNING

In consequence of a serious or fatal error, the SIOM shifts to the safe state (OPMODE_FAILSAFE_STOP or OPMODE_FAILSAFE_IO).

The safe state (OPMODE_FAILSAFE_STOP or OPMODE_FAILSAFE_IO) is the de-energized state.

5.4 General Controller Module Performance GCM-P

For the Module GCM-P there is a separate manual [MAN-GCM-P] in which the characteristics of this module are described.**Shared Memory**

	WARNING
	Shared Memory -IOs If an IO module SCM (SL), SIOM or GIOM is to be used completely by the standard application program via the shared memory, it must be monitored in the secure application program to see whether the shared memory communication is transmitting current values. If no current values are received from the GCM-P in the safe application, the default values of the outputs must be set. The shared memory update of the GCM-P is not executed, in case of an exception, upload or stop of the standard application.

5.4.2 Diag level- Serial-Port

	WARNING
	Diag Level - Serial-Port If the Diag_level has been changed, it must be reset to 0 for safe operation of the controller. The environment variable Diag_level of the GCM-P module must not be used with the value 5.

5.5 General I/O-Module GIOM

5.5.1 General

The *GIOM* module is intended to extend the inputs/outputs of a ***digsy***®$_{fusion}$ S. The *GIOM* provides the same number and quality of inputs/outputs as an *SCM*.

Properties of the *GIOM*:

- 32-bit Safety Controller
- 2-channel µC in the Lockstep Mode
- On-chip flash memory 3MB incl. ECC
- On-chip RAM 256 kB incl. ECC
- Internal bus connection for system communication
- Various types of inputs/outputs (see Table 5-10)
- On-board power supply

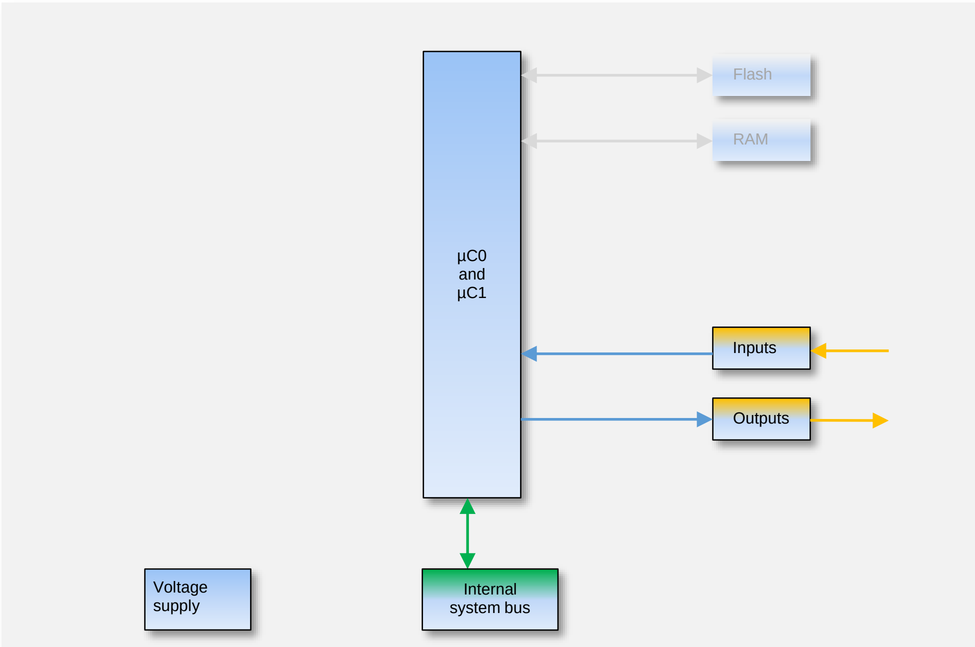


Figure 5-7: GIOM block diagram

5.5.2 Pin assignment of the central plug connector

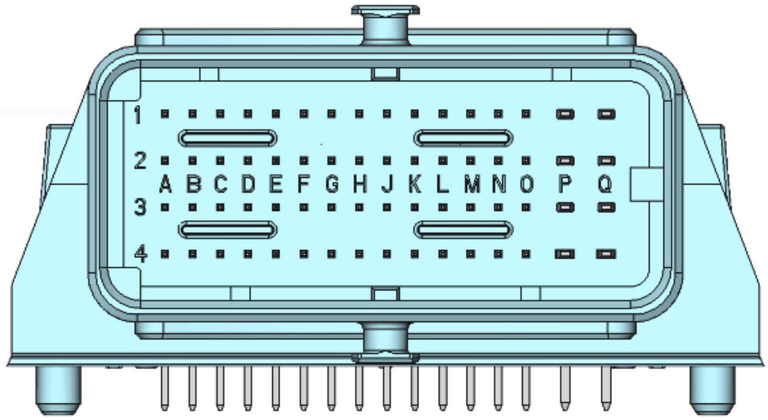


Figure 5-8: Central plug connector

	PIN	PIN-NAME	Input Digital +	Input Digital -	Input Count	Input Analog	Output Digital +	Output Digital -	Output Ratiometr.	Output PWM	Output PWM IREG	VIX / GND	I/O-Type	Note
	3P	VIQ_1										■		Versorgung Leistung Gruppe 1
	3B	Q1.1					■			■	■		A1	Out, Gruppe 1, 2,5A
	3A	Q1.2					■			■	■		A1	Out, Gruppe 1, 2,5A
	4A	Q1.3					■			■	■		A1	Out, Gruppe 1, 2,5A
	4B	Q1.4					■			■	■		A1	Out, Gruppe 1, 2,5A
	2P	VIQ_2										■		Versorgung Leistung Gruppe 2
	2A	Q2.1					■			■	■		A3	Out, Gruppe 2, 2,5A
	2B	Q2.2					■			■	■		A3	Out, Gruppe 2, 2,5A
	1B	Q2.3					■			■	■		A2	Out, Gruppe 2, 4A
	1A	Q2.4					■			■	■		A2	Out, Gruppe 2, 4A
	2Q	VIQ_3										■		Versorgung Leistung Gruppe 3
	2O	Q3.1					■			■	■		A3	Out, Gruppe 3, 2,5A
	2N	Q3.2					■			■	■		A3	Out, Gruppe 3, 2,5A
	1N	Q3.3					■			■	■		A2	Out, Gruppe 3, 4A
	1O	Q3.4					■			■	■		A2	Out, Gruppe 3, 4A
	1Q	VIQ_4.1										■		Versorgung Leistung Gruppe 4
	3Q	VIQ_4.2										■		Versorgung Leistung Gruppe 4
	1J	Q4.1	■				■						B	Out, In, Gruppe 4, 4,0A
	1K	Q4.2	■				■						B	Out, In, Gruppe 4, 4,0A
	1L	Q4.3	■				■						B	Out, In, Gruppe 4, 4,0A
	1M	Q4.4	■				■						B	Out, In, Gruppe 4, 4,0A
	2L	Q4.5	■				■						B	Out, In, Gruppe 4, 4,0A
	2M	Q4.6	■				■						B	Out, In, Gruppe 4, 4,0A
	1C	I1.1	■		■								D	In, Gruppe 1, AB-Zähler 1A
	1D	I1.2	■		■								D	In, Gruppe 1, AB-Zähler 1B
	2C	I1.3	■		■								D	In, Gruppe 1, AB-Zähler 2A
	2D	I1.4	■		■								D	In, Gruppe 1, AB-Zähler 2B
	2E	I1.5	■		■								D	In, Gruppe 1, AB-Zähler 3A
	2F	I1.6	■		■								D	In, Gruppe 1, AB-Zähler 3B
	2G	I1.7	■		■								D	In, Gruppe 1, AB-Zähler 4A
	2H	I1.8	■		■								D	In, Gruppe 1, AB-Zähler 4B
	4F	I2.1	■			■							E	In, Gruppe 2

	PIN	PIN-NAME	Input Digital +	Input Digital -	Input Count	Input Analog	Output Digital +	Output Digital -	Output Ratiometr.	Output PWM	Output PWM I_{REG}	VIX / GND	I/O-Type	Note
	3F	I2.2	■			■							E	In, Gruppe 2
	4G	I2.3	■			■							E	In, Gruppe 2
	3G	I2.4	■			■							E	In, Gruppe 2
	4H	I2.5	■			■							E	In, Gruppe 2
	3H	I2.6	■			■							E	In, Gruppe 2
	4J	I2.7	■			■							E	In, Gruppe 2
	3J	I2.8	■			■							E	In, Gruppe 2
	4K	I2.9	■			■							E	In, Gruppe 2
	3K	I2.10	■			■							E	In, Gruppe 2
	4L	I2.11	■			■							E	In, Gruppe 2
	3L	I2.12	■			■							E	In, Gruppe 2
	4M	I2.13	■			■							E	In, Gruppe 2
	3M	I2.14	■			■							E	In, Gruppe 2
	4N	I2.15	■			■							E	In, Gruppe 2
	3N	I2.16	■			■							E	In, Gruppe 2
	4C	I3.1	■	■									C	In, In Low, Gruppe 3
	3C	I3.2	■	■									C	In, In Low, Gruppe 3
	4D	I3.3	■	■									C	In, In Low, Gruppe 3
	3D	I3.4	■	■									C	In, In Low, Gruppe 3
	4E	I3.5	■	■									C	In, In Low, Gruppe 3
	3E	I3.6	■	■									C	In, In Low, Gruppe 3
	1P	VIM_1										■		Versorgung Logik + Eingänge
	4O													
	3O	GND_1										■		Masse 1 (Logik + Eingänge)
	1G	Q_SENS										■		Ausgang Sensorversorgung
	1H	IREF										■		Referenzstrom
	1E	UREF_1										■		Referenzspannung 1
	1F	UREF_2										■		Referenzspannung 2
	2J	GND_A1										■		Masse – Analog 1
	2K	GND_A2										■		Masse – Analog 2
	4P	GND_2										■		Masse 2 (voreilender Kontakt!)
	4Q	GND_1										■		Masse 3 (voreilender Kontakt!)
	SUMME	Total:	36	6	8	16	18	0	0	12	12	16	48	

Table 5-10: Pin assignment of the central plug connector of an *GIOM*

Note

Notes on the pin assignment of the central plug connector

1. For the output groups Q1.x to Q4.x the assignment to the respective voltage supply path (VIQ_x) is marked in color on the left.

5.5.3 External interfaces

The *GIOM* does **not** provide communication interfaces (RS232/ CAN/ Ethernet).

5.5.4 Firmware

The *GIOM* is a non-autonomous, fixedly programmed module which requires an *SCM* (SL) for normal operation.

The *GIOM* firmware includes the following components:

- Boot loader
- Start-up self-diagnosis
- Runtime self-diagnosis
- Function-dependent, fixed system program
- FW modules to support all the existing input/output components

6 Dynamic Properties

6.1 Operating modes

6.1.1 Overview

digsy®fusion can operate in various modes:

Operating mode	***digsy***®**fusionS**	***digsy***®**fusionS *SAFE***	***digsy***®**fusionS *STANDARD***
BOOT	●		
SELFTEST		●	
INITIALIZATION		●	
OPMODE_NORMAL		●	
OPMODE_FAILURE		●	
OPMODE_FAILSAFE_IO		●	
OPMODE_FAILSAFE_STOP		●	

Table 6-1: Operating modes

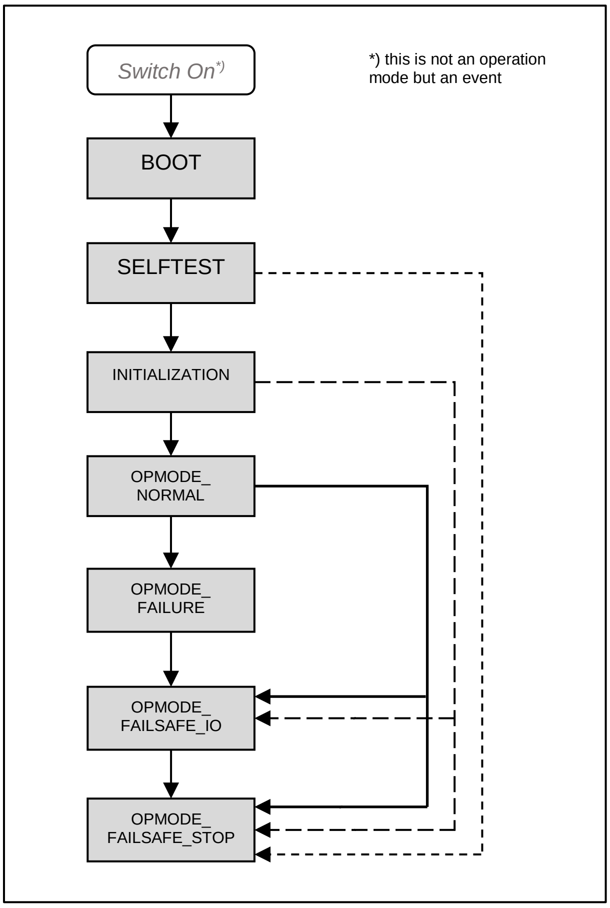


Figure 6-1: Switchover of the digsy®fusion operating mode

6.1.2 BOOT operating mode

Immediately after starting control system, the BOOT operating mode checks whether files are available for updating the firmware, the application program, or other functions. If this is the case, the update of these files is started. Once the update has been completed the normal operation continues.

During normal operation, i.e. if no files are available for updating, a check is performed in order to determine whether the operating system is available and ready to run. If the operating system is missing or detected as faulty, the BOOT operating mode remains in an infinite loop. If this is the case, *digsy*®$_{\text{fusion}}$S must be restarted by switching it off and on again!

If the operating system is available and fault-free, the system switches from the BOOT operating mode to the SELFTEST mode (see Figure 6-2).

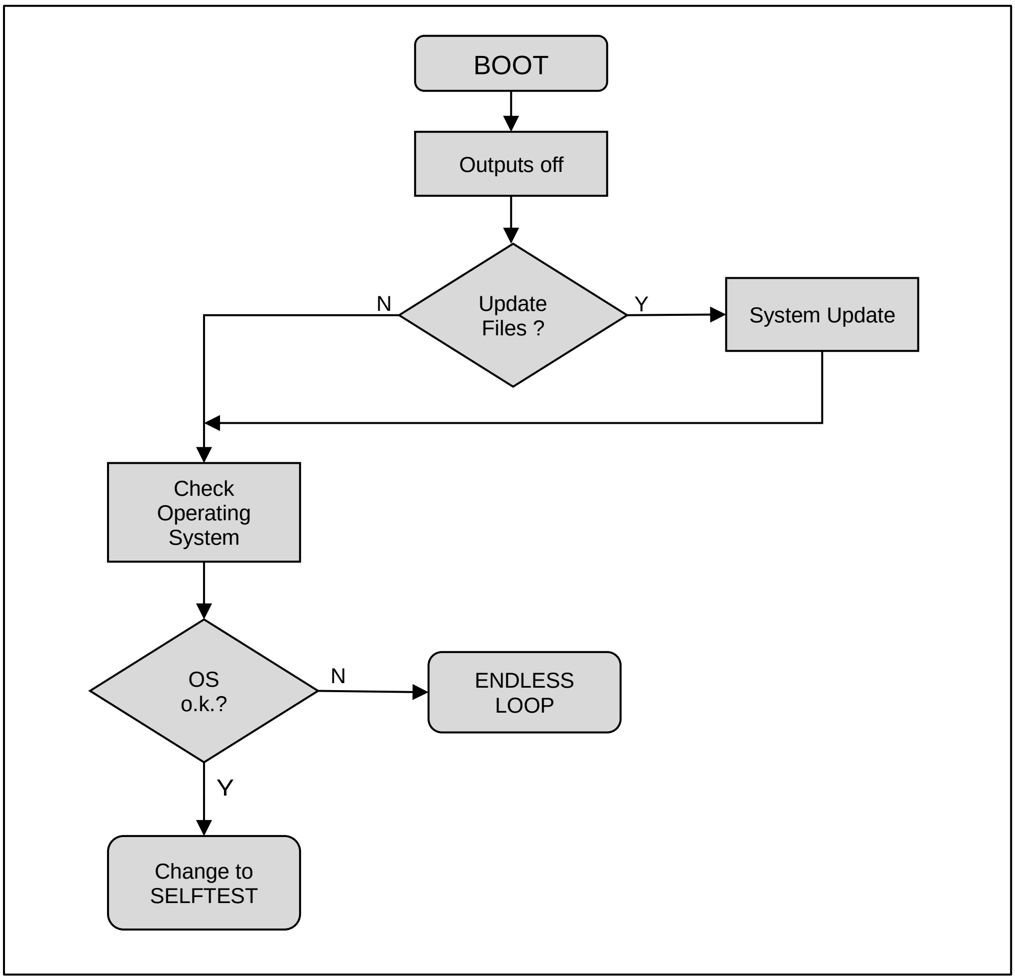


Figure 6-2: BOOT operating mode

	Note
	Outputs: All the outputs are de-energized in the BOOT operating mode!

6.1.3 SELFTEST operating mode

When entering the SELFTEST operating mode, the control system performs a detailed check as to whether the operating system and its components operate properly. Afterwards, the hardware of the control system is subjected to comprehensive tests using special diagnostic functions.

If a functional error is detected, the system quits the SELFTEST operating mode with an error message and switches to the OPMODE_FAILSAFE_STOP mode.

If no error is detected, the system switches from the SELFTEST operating mode to the INITIALIZATION operating mode.

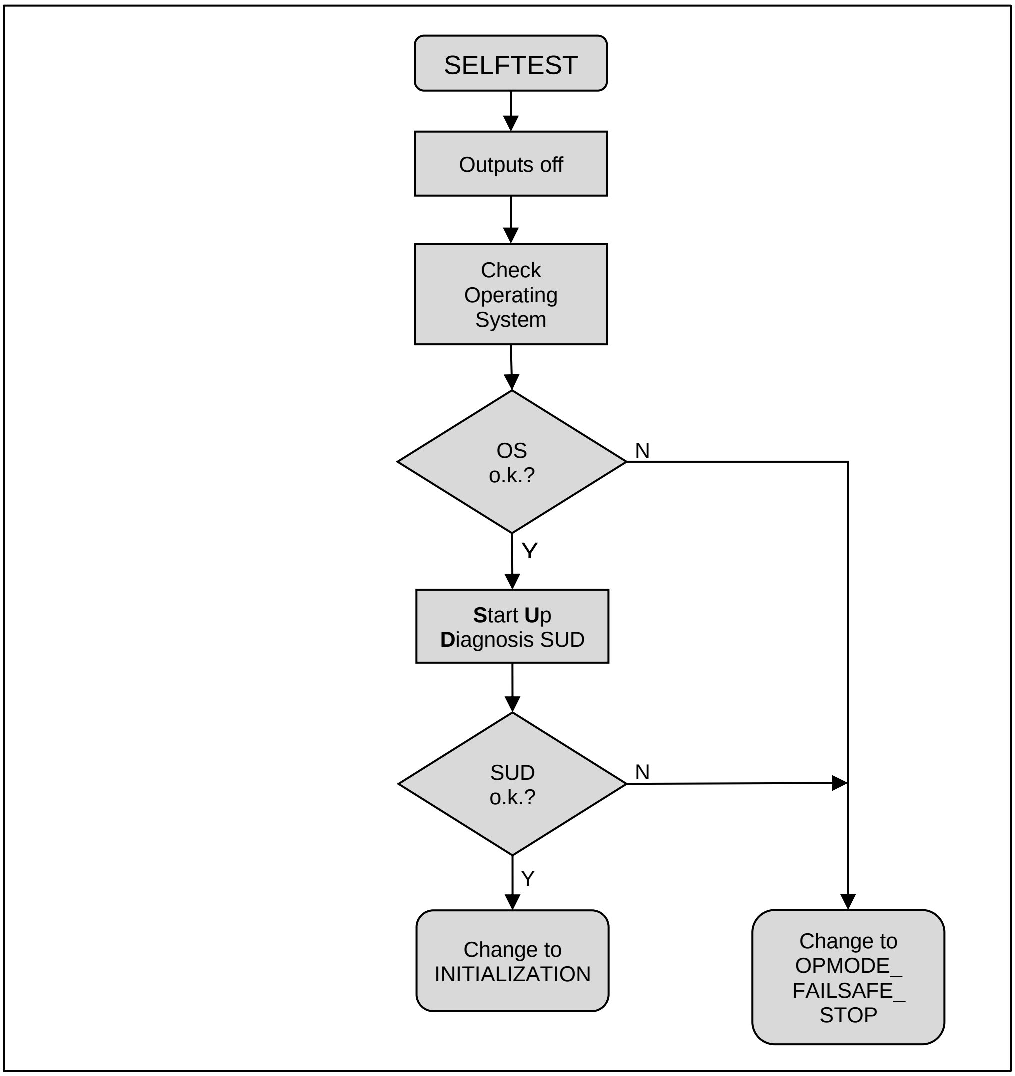


Figure 6-3: SELFTEST operating mode

	Note
	Outputs: All the outputs are de-energized in the SELFTEST operating mode!

6.1.4 INITIALIZATION operating mode

In the INITIALIZATION operating mode the following activities are performed in the order indicated below:

- Start of the runtime diagnosis
- Set-up of the CODESYS Safety runtime system
- Set-up of all communication interfaces
- Set-up / initialization of all the inputs and outputs
- Start of the Safe Application via CODESYS Safety
- Set-up of the CODESYS Standard runtime system
- Start of the Standard Application via CODESYS Standard

In the INITIALIZATION operating mode the runtime diagnosis of ***digsy***$^{®}_{\text{fusion}}$ S is started first. The runtime diagnosis operates cyclically. It runs a complete diagnostic cycle within 50ms.

Once the CODESYS Safety runtime system, the communication interfaces and all the inputs and outputs have been set up according to the predefined configuration, the Safe Application is started via CODESYS Safety.

Once the CODESYS Standard runtime system has been set up, the Standard Application is started.

Note

To set up the inputs and outputs:

Both the inputs/outputs of the Safe Application and the inputs/outputs of the Standard Application are set up. A (hidden) system data structure is generated for the Standard Application which is read by the CODESYS Safety runtime system for the configuration of the inputs and outputs.

PLEASE NOTE: Any change in the input/output configuration data of the Standard Application only becomes effective after a restart of the Safe Application. The Safe Application must be restarted in the Debug mode or the control system must be switched off and on again!

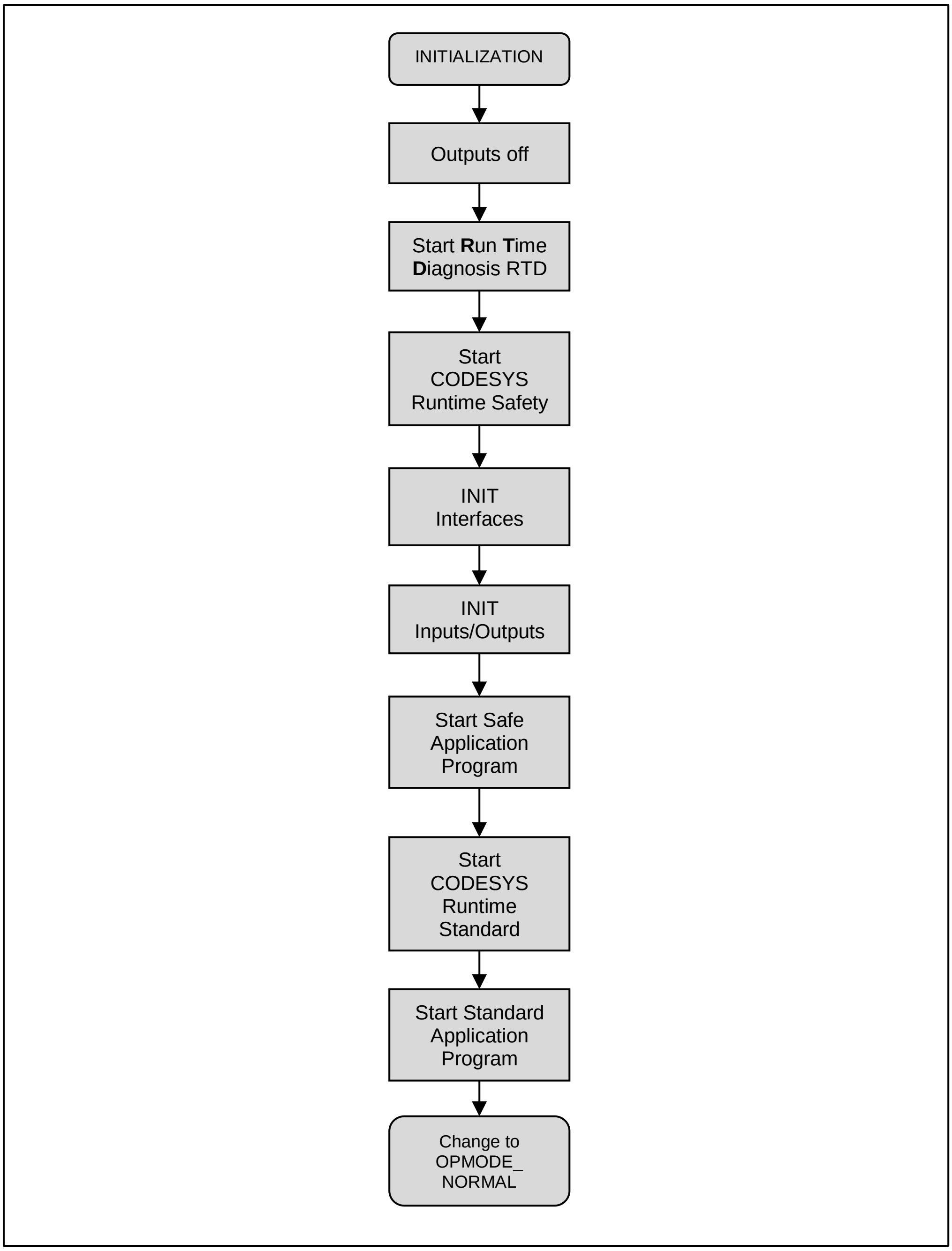


Figure 6-4: INITIALIZATION operating mode

	Note
	Outputs: All the outputs are de-energized in the INITIALIZATION operating mode!

6.1.5 OPMODE_NORMAL operating mode

After all the necessary activities have been completed in the operating modes described above, the OPMODE_NORMAL is the operating mode where the application runs without any limitation.

The following activities are performed in the OPMODE_NORMAL operating mode:

- Cyclical input data import
- Cyclical import/export of the communication interface data
- Cyclical execution of the application program(s)
- Cyclical runtime diagnosis
- Cyclical export of the output data

Note

OPMODE_NORMAL operating mode

When the system is in the OPMODE_NORMAL operating mode, application programs run (in addition to the OPMODE_FAILURE operating mode)! These application programs have not yet been started in all the preceding operating modes!

In the OPMODE_NORMAL operating mode the application programs switch the outputs on and off!

If an error is detected during the cyclical runtime diagnosis, the system quits the OPMODE_NORMAL operating mode and switches to one of the following modes:

- OPMODE_FAILURE
- OPMODE_FAILSAFE_IO
- OPMODE_FAILSAFE_STOP

The system switches to one of these operating modes depending on the quality and severity of the error detected.

Note

Switchover of the operating mode:

At runtime the system only changes the operating mode in one direction, i.e. from:

OPMODE_NORMAL →
OPMODE_FAILURE →
OPMODE_FAILSAFE_IO →
OPMODE_FAILSAFE_STOP!

The switchover of the operating mode cannot be undone!

For example, the system cannot be transferred back from the OPMODE_FAILURE operating mode to the OPMODE_NORMAL operating mode.

Provided that the errors detected can be remedied, switching the control system off and back on is the only way to return to the OPMODE_NORMAL operating mode.

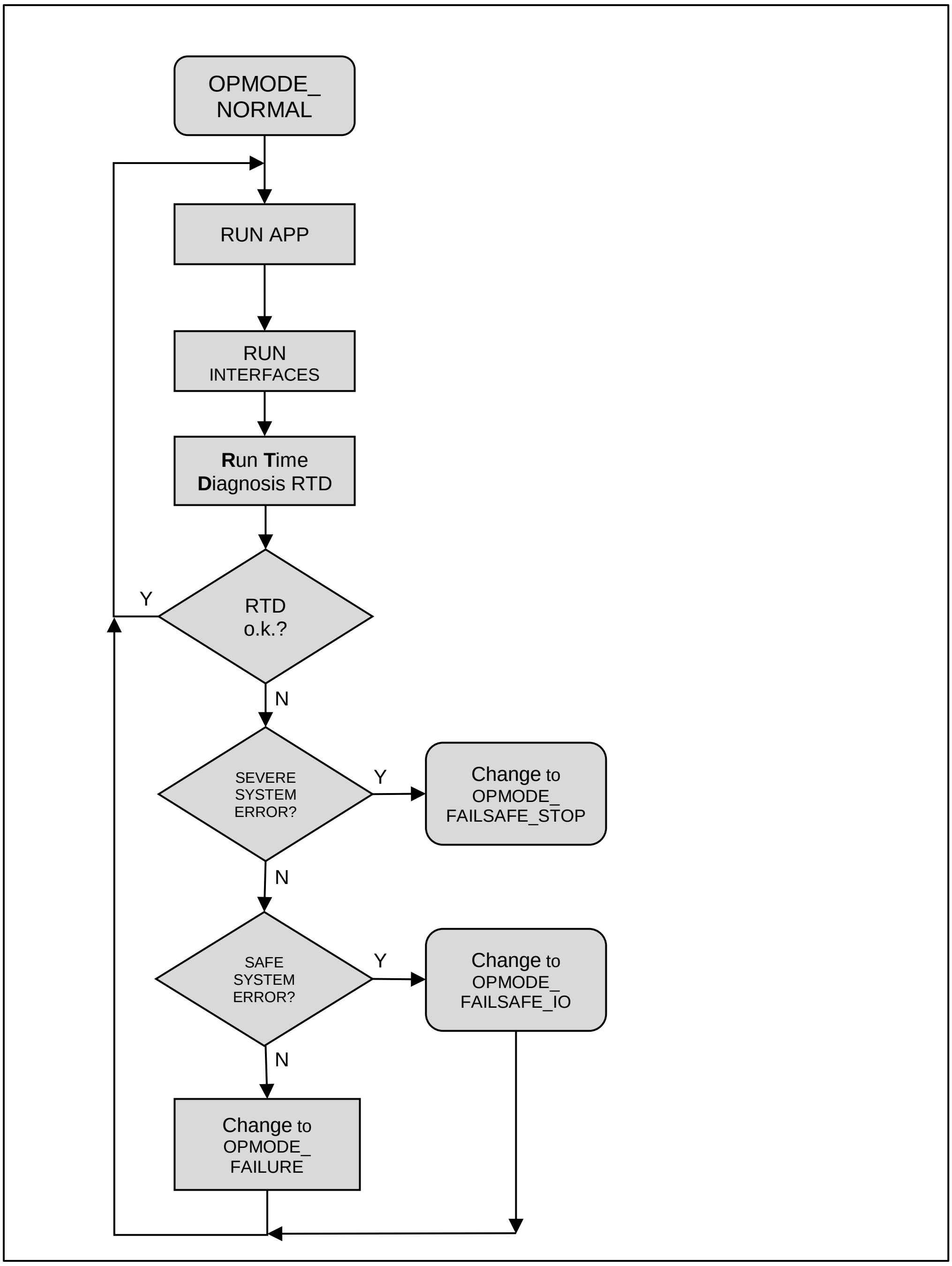


Figure 6-5: OPERATION operating mode

6.1.6 OPMODE_FAILURE operating mode

The OPMODE_FAILURE operating mode indicates a control system state in which an error has been detected.

The application program is active in the OPMODE_FAILURE operating mode. The user can decide in the application program which activities must be performed in order to ensure the functional safety of the machine, depending on its current system state.

Note

Simple error
A simple error can be remedied by means of the application program.

OPMODE_FAILURE operating mode

Like in the OPMODE_ NORMAL operating mode, application programs also run in the OPMODE_ FAILURE operating mode.

In the OPMODE_FAILURE operating mode the outputs are energized and de-energized by the application programs.

HAZARD

Error
An error can produce a hazardous situation for both persons and the machine!

1. An error can be remedied by means of the application program.
2. If possible, the machine can be transferred to a safe state by means of the application program.
3. If an error occurs, the remaining operating time of the control system should not exceed 24h. The machine manufacturer / operator must ensure that this time is not exceeded!

In order to be able to switch from the OPMODE_FAILURE operating mode to the OPMODE_NORMAL operating mode, the control system must first be switched off, the error must be rectified and the control system must be switched back on again.

6.1.7 OPMODE_FAILSAFE_IO operating mode

The OPMODE_FAILSAFE_IO operating mode indicates a control system state in which a serious error has been detected.

The application program is active in the OPMODE_FAILSAFE_IO operating mode. The inputs are passiv and provide the safe state. All the outputs are safely de-energized. The diagnosis interfaces are active.

	HAZARD
	Serious errors A serious error can produce a hazardous situation for both persons and the machine! 1. A serious error cannot be controlled by the means of the application program. 2. If a serious error occurs in the control system, the control system automatically de-energizes all the outputs! When creating a machine safety concept, appropriate measures must be taken which also ensure the functional safety of the machine in case of a serious error. 3. If an error occurs, the remaining operating time of the control system should not exceed 24h. The machine manufacturer / operator must ensure that this time is not exceeded!

	Note
	OPMODE_FAILSAFE_IO operating mode Application programs run in the OPMODE_FAILSAFE_IO operating mode. The control system is still able to communicate for diagnostic purposes.

Proceed as follows in order to transfer the system from the OPMODE_FAILSAFE_IO operating mode to the OPMODE_NORMAL mode:

- An error has occurred in the periphery of the control system: Switch off the control system, rectify the error and switch the control system back on again.
- An error has occurred within the control system (e.g. a programming error): Perform a diagnosis via the programming system and rectify the error.

6.1.8 OPMODE_FAILSAFE_STOP operating mode

The OPMODE_FAILSAFE_STOP operating mode indicates a control system state in which a fatal error has been detected.

The application program is stopped in the OPMODE_FAILSAFE_STOP operating mode. All the outputs are safely de-energized. The diagnosis interfaces are disconnected.

	HAZARD
	Fatal errors A fatal error can produce a hazardous situation for both persons and the machine! 1. A fatal error cannot be controlled by means of the application program! 2. If a fatal error occurs in the control system, the control system automatically de-energizes all the outputs! When creating a machine safety concept, appropriate measures must be taken which also ensure the functional safety of the machine in case of a fatal error.

	Note
	OPMODE_FAILSAFE_STOP operating mode No application program runs in the OPMODE_FAILSAFE_STOP operating mode. The control system is no longer able to communicate, not even for the diagnostic purposes.

Proceed as follows in order to transfer the system from the OPMODE_FAILSAFE_STOP operating mode to the OPMODE_NORMAL mode:

- An error has occurred in the periphery of the control system: Switch off the control system, rectify the error and switch the control system back on again.

	CAUTION
	Restarting the control system in the OPMODE_FAILSAFE_STOP operating mode A fatal error has been detected in a machine control system which has switched to the OPMODE_FAILSAFE_STOP operating mode: In order to avoid hazardous incidents, do not continue to operate the control system! It is strongly advised not to continue to operate the control system!

6.2 Switching the operating mode

The operating mode can be switched as follows:

- Passive switchover by the system-controlled runtime diagnosis,
- Active switchover by calling up functions from the application program.

6.2.1 Passive switchover of the operating mode

Once the runtime diagnosis has detected an error, the passive switchover of the operating mode occurs automatically and without any activity of the application program. The runtime diagnosis is an integral part of the ***digsy***®$_{\text{fusion}}$ S control function.

The application program is able to detect as follows whether a passive switchover of the system state has occurred:

- Evaluating the current operating mode.

The application system can evaluate device or system errors detected during the runtime diagnosis as follows:

- Evaluating the pin error flags,
- Evaluating the system logbook.

6.2.1.1 Evaluating the current operating mode

The current operating mode can be read in the application program using the *SysGetOperationMode* (*DFS_Safe* library) function.

6.2.1.2 Evaluating the pin error flags

A pin error flag is assigned in the corresponding I/O mapping for all the available inputs/outputs: If a malfunction is detected on an input or output during the runtime diagnosis, the related pin error flag is set. This is a simple way to identify and evaluate input/output errors in the application program.

An input whose pin error flag is set always returns the safe state as the input value (e.g. FALSE for a pin configured as a digital input).

The pin error flags are read-only. It is not possible to change the values or reset a pin error flag from the application program. The pin error flags can only be reset during a restart by switching the control system off and on again.

The pin error flags only indicate that an error has occurred on an input/output, but do not show the type of the error. To obtain information on the error type, the system logbook must be evaluated.

6.2.1.3 Evaluating the system logbook

The system logs the following events in the system logbook:

- Start message
- System messages
- System (error) messages resulting in a switchover of the operating mode.

For example, any switchover of the operating mode and any error detected during the runtime diagnosis generate a system logbook entry.

The contents of the system logbook can be read out using the *SysLogRead* (*DFS_Safe* library) function or with an USB device.

6.2.2 Active switchover of the operating mode

If an error has occurred in the periphery of the control system, which is not detected by the controller itself, but can be detected as an error from the application, it may be necessary to switch the control system to a defined state. To do this, the operating mode can be actively switched from the application program.

The current operating mode can be switched in the application program using the *SysSetOperationMode* (*DFS_Safe* library) function.

6.3 System behavior

Table 6-2 and Table 6-3 illustrate the system behavior in the various operating modes.

Operating mode	Access	System behavior
OPMODE_NORMAL	From the INITIALISATION operating mode upon entering the OPMODE_NORMAL mode: - If no error has been detected within the scope of the SUD and RTD.	- The application program is running. - All the inputs are active. - All the outputs are active. - All the interfaces (in particular the diagnosis interfaces) are active. - LEDs see Table 3-6
OPMODE_FAILURE	- After the system self-diagnostics has detected an error condition on one or several inputs. - After the system self-diagnostics has detected an error condition on one or several outputs.	- The application program is running. - All the inputs are active, except for those inputs for which an error has been detected. These show a safe input condition. - All the outputs are active, except for output groups with one or several faulty outputs. These are de-energized via the output itself and the second shutdown path. - All the interfaces (in particular the diagnosis interfaces) are active. - LEDs see Table 3-6 - The switch to the OPMODE_FAILURE operating mode can be detected from the application program.

Operating mode	Access	System behavior
OPMODE_FAILSAFE_IO	- After the internal self-diagnostic function has detected a serious error condition; further operation of the µC micro controller is still possible. - After detecting an error condition in the application program and starting the OPMODE_FAILSAFE_IO operating mode via a function call-up from the application program.	- The application program is running. - All the inputs are passive and show the safe input condition. - All the outputs are passive and definitively de-energized via the output itself and the second shutdown path. - All the interfaces (in particular the diagnosis interfaces) are active. - LEDs see Table 3-6 - The switch over to the OPMODE_FAILSAFE_IO operating mode can be detected from the application program.
OPMODE_FAILSAFE_STOP	- After the system self-diagnostics has detected a fatal error condition which renders the further operation of the µC impossible. - After detecting an error condition in the application program and starting the OPMODE_FAILSAFE_STOP operating mode via a function call-up from the application program. - After the watchdog has responded.	- The application program is stopped. - All the outputs are definitively de-energized via the second shutdown path. - All the interfaces are de-energized (diagnostic interfaces cannot be used!). - LEDs see Table 3-6 - The control system operates no longer.

Table 6-2: ***digsy***®fusion S system states I

Operating mode System behavior	OPMODE_NORMAL	OPMODE_FAILURE	OPMODE_ FAILSAFE_IO	OPMODE_ FAILSAFE_STOP
Application program	Running	Running	Running	Stopped
The operating mode can be detected from the application program	Yes	Yes	Yes	No
Outputs	Active	Output group with faulty output de-energized, all other outputs are active	All output groups de-energized via the second shutdown path	All output groups de-energized via the second shutdown path
Inputs	Active	Faulty inputs show safe state, the others are active	All the inputs show safe state	Passive
Interfaces	Active	All the interfaces are active (in particular the diagnosis interfaces).	All the interfaces are active (in particular the diagnosis interfaces).	All the interfaces are passive (especially the diagnosis interfaces cannot be used!)

Table 6-3: ***digsy***®fusion S system states II

7 Software

7.1 Interfaces

7.1.1 CAN

7.1.1.1 General

digsy®fusion S provides four independent CAN interfaces (CAN1 to CAN4). Via these interfaces the control system communicates with external components, e.g. sensors.

Since ***digsy***®fusion S does not provide internal terminating resistors, external terminating resistors must be connected (see Note).

Note

Wiring of a CAN network

1. The two-wire connection must be twisted. A cable shielding further improves the EMC properties.
2. The CAN bus must be terminated with resistors (120Ω each) on both ends.

   The maximum cable length of a CAN network depends on the selected bit rate (see Table 7-1: CAN bit rates
3.).

Bit rate in Kbits/s	1000	800	500	250	125	50	20	10
Max. cable length in m	25	50	100	250	500	1000	2500	5000

Table 7-1: CAN bit rates

7.1.1.2 CANopen and Layer2 protocols

Thanks to the use of the CODESYS CANopen Libraries (3S_CanOpenxxx.lib) and the CANopen Configurator in the CODESYS development environment, every CAN interface supports the CANopen protocol.

The CAN bus be simultaneously (or alternatively) accessed via a proprietary protocol on the basis of Layer2 (11- and/or 29-bit identifiers).

7.1.1.3 CANopen safety

For more detailed information on the use of CANopen safety refer to the 3S operating manual.

7.1.2 RS232

digsy®fusion S comes with two RS232 interfaces. These two interfaces support baud rates of up to 115,200 baud.

The COM1 interface is reserved as a programming interface towards CODESYS. The default parameter setting is as follows:

Parameter	Value
Baud rate	115,200

Parameter	Value
Data bits	8
Stop bits	1
Parity	none
Handshake	none

Table 7-2: RS232 default setting

The COM2 interface can be freely programmed via the "DFS_Safe" library.

The ***digsy***®fusion S-P(L) also has two RS232 interfaces. Both interfaces support transfer rates up to 115200 baud.

The COM1 interface is the interface for the SCM (SL), which can be configured either as a programming interface to CODESYS or as a freely programmable interface.

Note

Use the COM interface as a freely programmable interface

To use the COM interface as a freely programmable interface, the block must be commented out (with „;") or deleted. (see chapter 7.1.5)

```
;[CmpBlkDrvCom]
;Com.0.Port=1
;Com.0.Baudrate=115200
;Com.0.EnableAutoAddressing=0
;Com.0.Name=MyCom
```

If this block is not commented out in the configuration file, the CODESYS interface is made available as a programming interface.

The COM2 interface is the interface for the GCM-P(L).

7.1.3 Ethernet

The 10/100-Mbit Ethernet interface of ***digsy***®fusion S can be used for the following tasks:

- Program download and online diagnosis via CODESYS V3.5
- Freely programmable data transmission via the TCP/IP and UDP protocols

Default setting:

Parameter	Value
IP	192.168.100.176
Network Mask	255.255.255.0
Gateway	192.168.100.254

Table 7-3: Ethernet default setting

On ***digsy***®fusion S-P both Ethernet interfaces are internaly connected via switch.

7.1.3.1 Program download and online diagnosis via TCP/IP

We recommend to use ethernet interface as the standard interface towards the ***CODESYS V3.5*** programming tool. Both programs and files can be loaded via this interface. In addition, variables can be displayed and the application program can be tested (to mention just the most important features of this tool). Communication is implemented via a so called Gateway (on the PC end) and the CODESYS runtime system (device firmware). Communication between the gateway and the runtime system takes place via UDP telegrams. TCP is not implemented for the programming interface (no access via the Internet gateway).

7.1.4 USB

A V2.0 USB interface is provided for loading the operating data, the firmware and the application program. Furthermore, a USB stick connected to the USB interface can be used as a memory via the file system (see chapter 9). Make sure that the USB stick is connected via a connecting cable provided by INTER CONTROL (item number 4308.36.100000-01) and formatted in FAT.

	Note
	Filesystem More information about the filesystem (max size, format) can be found in chapter 9.

7.1.4.1 Update via USB

You update parts of the ***digsy***®$_{fusion}$ S software via the USB interface. The following software components can be downloaded via USB:

- Safe Application
- Standard Application
- CODESYS configuration file
- Firmware (made available by INTER CONTROL)
- Files to be copied into the internal file system

	Note
	Firmware update After a firmware update, it may occur that the application, which still exists in the system, switches to the *OPMODE_FAILSAFE* mode and no longer functions as required. To rectify this problem the application must be erased and reloaded.

	Note
	Update GCM-P of *digsy*®$_{fusion}$ S-P In addition to the above parts, the GCM-P can also be updated. More details can be found in [MAN-GCM-P].

To perform the update:

Safe Application and Standard Application

You can create a boot application in CODESYS (see Figure 7-1). This function creates an APP file from the application program.

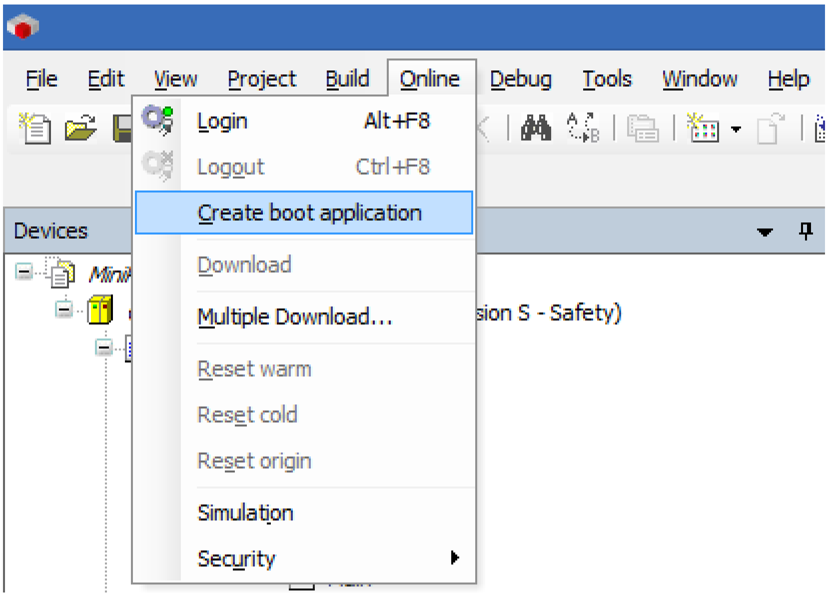


Figure 7-1: Creating a boot application

The Boot Loader tool (provided by INTER CONTROL) can convert the resulting APP file into a format suitable for the ***digsy***®fusion S Boot Loader. Copy the formatted application (*.bin.ic) to the USB stick. It will be loaded into the internal memory and started upon the next booting of the control system.

CODESYS configuration file

The CODESYS configuration file for CODESYS must also be converted into the correct format using the Boot Loader tool. It can be loaded by the Boot Loader via USB into the control system during the next booting.

Firmware

The firmware file required for any necessary firmware updates is provided in the correct format by INTER CONTROL. Copy the firmware file to the USB stick and then perform the update via the Boot Loader.

The booting starts automatically after system start (Power On). During the booting, the Boot Loader browses for a USB stick and loads any update files available. The LEDs indicate when the Boot Loader has completed the update (see Chapter 3.2.6.6).

	Note
	Programming times: The times required for updating several software components add up.

7.1.4.2 File “read_back_script.txt”

With the file “read_back_script.txt” several actions can be triggered on startup:

- Read log files via USB
- Copy of files into internal file system
- Copy of retain data from one device to another
- Determine versions of SCM (SL), SIOM and GIOM

	Note
	read_back_script: A command is accepted by the device only if the line is terminated by “Enter” (CR, carriage return).

Read back the logbooks via USB

The user can read out the internal logbooks (system logbook/ user logbook) via a script on an USB device.

This scriptfile (read_back_script.txt) is avaible at Inter Control. To read a logbook the user has to follow these instructions:

- Copy read_back_script.txt to an USB devise
- Connect the USB device with the PLC
- Restart the PLC (Interrupt of IPON/VIM)
- Wait till the operation mode of the PLC is OPMODE_NORMAL
- The requested files are created on the USB stick in the following format:
  XXXXXX_Syslog.bin
  XXXXXX_Userlog.bin
  XXXXXX corresponds to the serial number of the controller for the returned files.
- Disconnect the USB device from the PLC
- Open logbook (e.g. XXXXXX_Syslog.bin) with the Bootloadertool (available at INTER CONTROL)

The read_back_script.txt is a normal ASCII file. To read back the logbooks the script has to include the following commands:

1. User logbook: GetLog u.0
2. System logbook: GetLog s.0

With the prefix .0 all entries of the user or system logbook are read back.

If another prefix is used, the last n entries are read from the respective logbook.

Example for the read_back_script.txt to get to complete user and system logbook

```
GetLog u.0
GetLog s.0
```

Example for the read_back_script.txt to get the last 10 entries of the user logbook and the last 70 entries of the system logbook:

```
GetLog u.10
GetLog s.70
```

Open logbook

Select "Open Syslogfile" from the menu item "Syslog/Script" in the BootloaderTool.

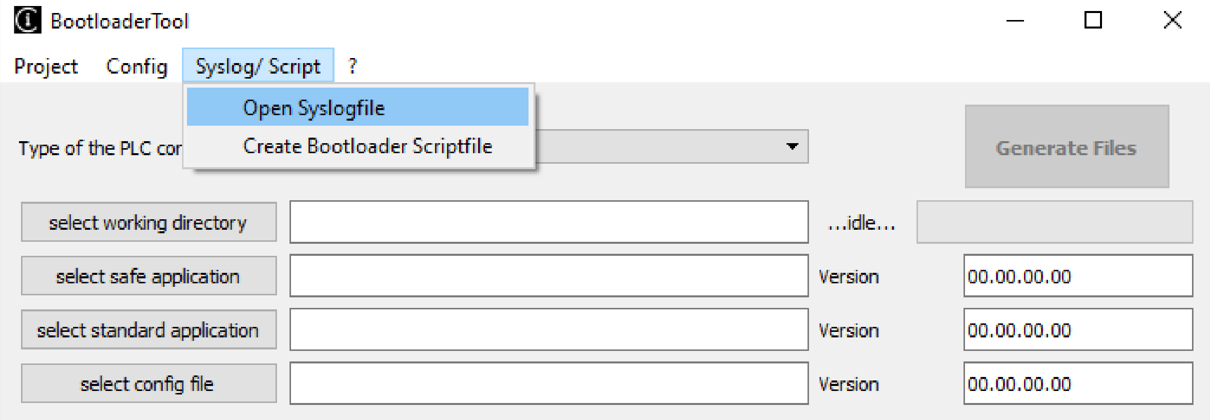


Figure 7-2: BootloaderTool open system logbook

Use "load syslogfile" to open the corresponding logbook file (e.g. XXXXXX_Syslog.bin).

Show logbook

Entries with the description "ERR_TYPE_SYSTEM_STARTUP", indicate a system start-up of the controller. The time stamp for this entry is specified in seconds and indicates the operating time of the controller.

All other time stamps are entered in milliseconds since the last start of the controller.

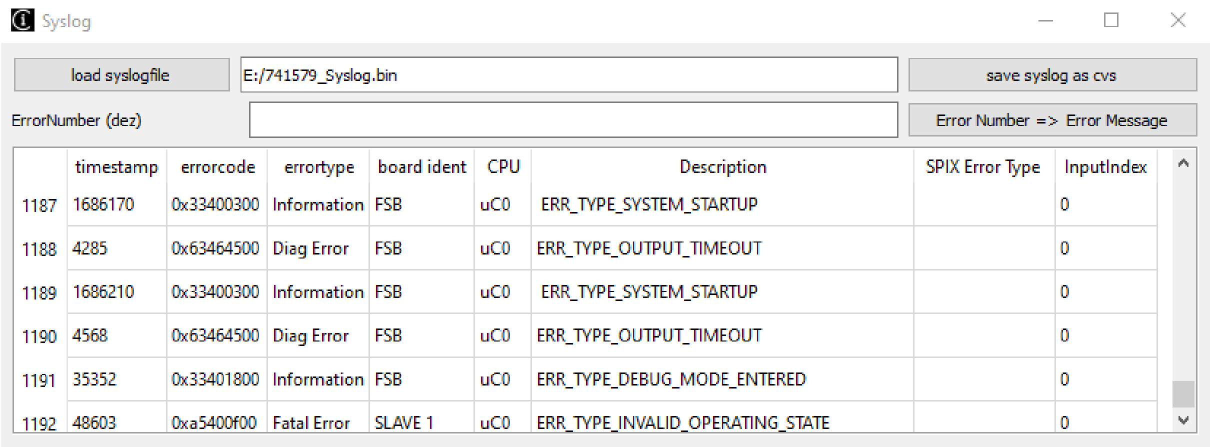


Figure 7-3: BootloaderTool show logbook

Copy from files into the internal filesystem

There are two commands to copy files into the internal filesystem or from the internal filesystem:

- GetFile <file name>
- PutFile <source file name> <destination file name>

Example of a read_back_script.txt to copy a file from the internal filesystem

```
GetFile testFile.txt
```

Example of a read_back_script.txt to copy a file into the internal filesystem:

PutFile testFileSource.txt testFileDestination.txt

Copy retain data from one controller to another controller

Retain data, like the logbooks, can be read out using read_back_script.txt. For this, the following command must be present in the read_back_script:

GetRetain

After restarting the PLC, the retain data should then be in a file on the USB stick. This file should have the following name:

<Serial number> _Retain.ics

To write this retain data to another PLC, the file extension of the read file must be renamed to * .bin.ic.

If this file is on the USB device during the next reboot, the retain data will be loaded onto the PLC.

Whether the loading process was successful can be read in the acknowledgment file "<serial number>_Quittung.txt" (automatically generated on the USB device).

> ⚠ **CAUTION**
>
> **Copy of retain data**
>
> In the software application, the retain data must be checked for validity, for example to prevent incorrect retain data from being loaded.
> See chapter 7.5.7.5 verification of Retain Data.

Determine versions of SCM (SL), SIOM and GIOM

With file „read_back_script.txt" versions of SCM (SL), SIOM and GIOM can be determine. The following command must be available in the file:

CrcFiles

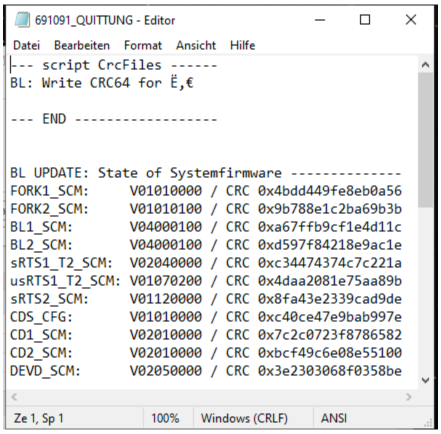


Figure 7-4: Read back versions of SCM (SL), SIOM and GIOM

7.1.5 Setting the interface parameters

Some of the parameters of the interfaces mentioned in Chapter 7.1 can be changed in a CODESYS configuration file. Figure 7-5 illustrates the structure of this configuration file. The user may change all entries highlighted in green. Entries which are not highlighted must not be changed. Furthermore, make sure that the correct formatting is retained.

```
[CmpRouter]
NumRouters=2
EnableParallelRouting=1
0.MainNet=ether x
1.MainNet=MyCom
2.MainNet=BlkDrvShm

[CmpBlkDrvCom]
Com.0.Port=1
Com.0.Baudrate=115200
Com.0.EnableAutoAddressing=0
Com.0.Name=MyCom

[CmpBlkDrvCanServer]
0.Baudrate=250
0.NetId=0
0.NodeId=100
DbgViaCan=0

[CmpAppEmbedded]
Bootproject.RunInFlash=1
Bootproject.CreateOnDownload=1

[SysFlash]
WriteBlockSize=128
EraseBlockSize=0x20000

[CmpBlkDrvUdp]
itf.0.ipaddress=192,168,100,176
itf.0.networkmask=255.255.255.0
itf.0.gateway=192,168,100,254

[FusionName]
Name=Inter Control
```

Figure 7-5: Example of a CODESYS configuration file

In this file you can change data such as the IP address or the control system name. Afterwards, the CODESYS configuration file can be loaded onto ***digsy***®$_{\text{fusion}}$ S via USB (see chapter 7.1.4.1).

⚠	**CAUTION**
	Malfunction of the control system 1. Make sure you only edit the entries highlighted in green in the CODESYS configuration file. 2. Do not change the file formatting. Do not change any blank lines, brackets, names to the left of the equality signs, etc.

7.1.6 Using the FTP server

From controller version 03.30.XX it is possible to access the controller via FTP. Access rights for 2 users can be created.

To access the ***digsy***®fusion S via the FTP server, the following lines must be added in the CODESYS configuration file (chapter 7.1.5):

```
[FTP]
FtpStart=<0/1>
user1=<name of user1>
pass1=<password for user1>
dir1=<start directory for user1>
user2=<name of user2>
pass2=<password for user2>
dir2=<start directory for user2>
```

Figure 7-6: Configuration file FTP settings

parameter	description	value
FtpStart	FTP server inactive/ active	0/1
user1	name User 1	String max 10 characters
pass1	password User 1	String max 10 characters
dir1	Startup folder of user1 on the PLC	z.B A:\ (USB) B:\ (internal filesystem)
user2	name User 2	String max 10 characters
pass2	password User 2	String max 10 characters
dir2	startup folder of user2 on the PLC	for example A:\ (USB) B:\ (internal filesystem)

Table 7-4: FTP parameter

Example:

```
[FTP]
FtpStart=1
user1=userName1
pass1=abc
dir1=A:\
user2=userName2
pass2=efg
dir2=B:\
```

Figure 7-7: Configuration file – FTP example

Note

Use of FTP server

When using the Booloader tool (Article: 04-69293) > Version 3.0, the configuration for the FTP server can be generated automatically.

7.1.6.1 Update via FTP

If the FTP server has been activated, the user has the possibility to access the internal file system B: via an FTP connection. There the identical possibilities of updates exist as described in chapter 7.1.4.

7.2 Structure of the firmware

The software consists of two independent program components, which will be started by a fork and run in sequential operation.

These are:

- Boot loader
- Firmware (one FW for the safe application and one FW for the standard application)

7.2.1 Boot loader

The Boot Loader is run after the Fork. It updates the firmware via the USB interface as required. You can also use the Boot Loader in order to update further software components (see Chapter 7.1.4.1).

7.2.2 Firmware

The firmware is the software component which is run after the booting sequence. The control system operates in this software component. The user can update the firmware. INTER CONTROL provides the necessary update packages.

7.3 Programming (CODESYS)

The safe application and standard application programs are programmed in the CODESYS development environment, version V3.5 SP5 SIL 2 (https://www.intercontrol.de/en/off-highway-electronics/downloads), designed by the company 3S (exact version of CODESYS see chapter 1.6). For general information on how to work with programming system refer to the online help. This chapter explains special aspects to be considered when using this programming system with ***digsy***®fusion S.

HAZARD

In accordance with the 3S Integration Manual the following CODESYS features must not be used for the Safe Application

1. Use of implicit check functions
2. Trace (recording) via the IEC task as a recording context
3. Icon configuration
4. Recipe management
5. Web and target visualization

7.3.1 Installing CODESYS

For more detailed information on the installation and use of the 3S CODESYS development environment, refer to the 3S manual or visit the web site http://www.codesys.com/.

Note

Version oft the 3S CODESYS development environment

The version of the 3S CODESYS development environment named in chapter 1.6 must be used for the respective controller version.

The following installation packages are provided by INTER CONTROL on download area: (https://www.intercontrol.de/en/off-highway-electronics/downloads)

- CODESYS V3.5 SP5 Patch 0 (3.5.5.0)
  (CODESYS_3.5SP5_Release.zip)
- CODESYS V3.5 SP11 Patch 0 (3.5.11.0)
  (CODESYS_3.5_SP11_Install_und_Safety_Package.zip)

Note

Installation of the CODESYS development environment

The CODESYS development environment must be installed as 32bit version.
The 64bit version does not support SIL2 extension and must not be used.
The installation packages provided by Inter Control are 32bit versions.

7.3.2 Installing additional packages

In addition to CODESYS, the CODESYS SIL2 package is required to provide full functionality. This package includes e.g. the SIL2 switch of the development environment which is used to switch the control system to the DEBUG mode.

INTER CONTROL also provides a package for ***digsy***®fusion S for the respective controller version (see chapter 1.6) (*digsy_fusion_VX-X-X-X.package*).

This package includes the device description data and libraries of ***digsy***®fusion S.

The required packages can be downloaded from download area (https://www.intercontrol.de/en/off-highway-electronics/downloads).

The required packages must be installed in CODESYS via "Tools->Package Manager" (see Figure 7-7).

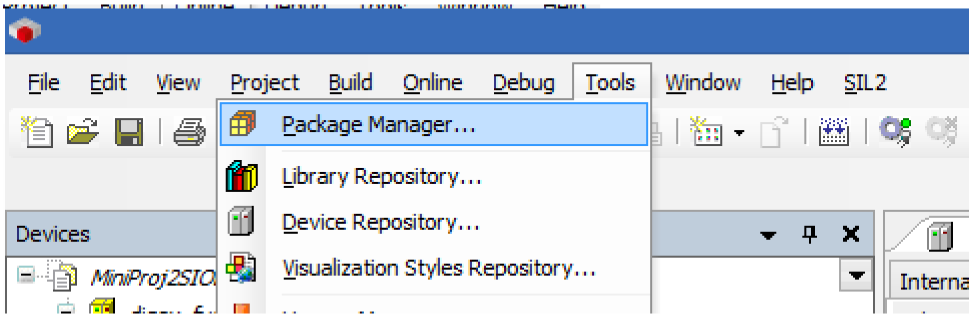


Figure 7-8: Package installation

7.3.3 Installing libraries

INTER CONTROL provides libraries with ready-for-use functions which simplify programming and reduce development times. You can integrate and use the following libraries:

- DFS_Safe
- DFS_Std
- DFS-P_Safe
- FUSIONx01
- Lib_digsyfusion_Util_Safety

Integrate the libraries via "Tools->Library-Repository". For a detailed description of the libraries refer to Chapter 8.

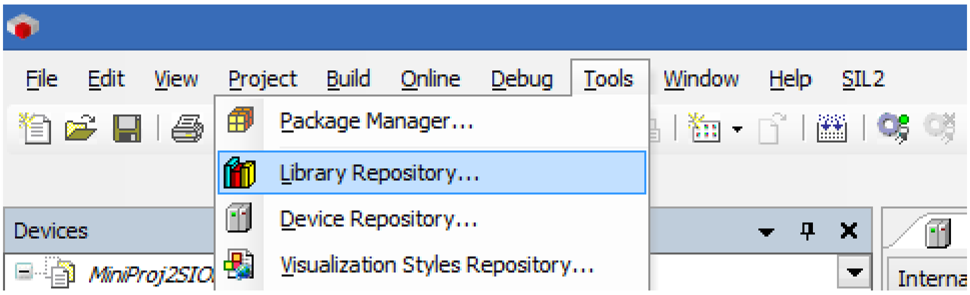


Figure 7-9: Library installation

7.3.4 Usage of the function „Trace“

Since device version V02.10.00 (device description 3.5.5.3) the function “Trace” is available in CODESYS (see help in CODESYS).

To use Trace the user has to add an object (Trace”) to the project. In the trace configuration, the trace must be connected to a task. There are two possibilities:

1. Using the task (Trc10ms (Safety Mode))
2. Using a own separate system task (only in debug mode)

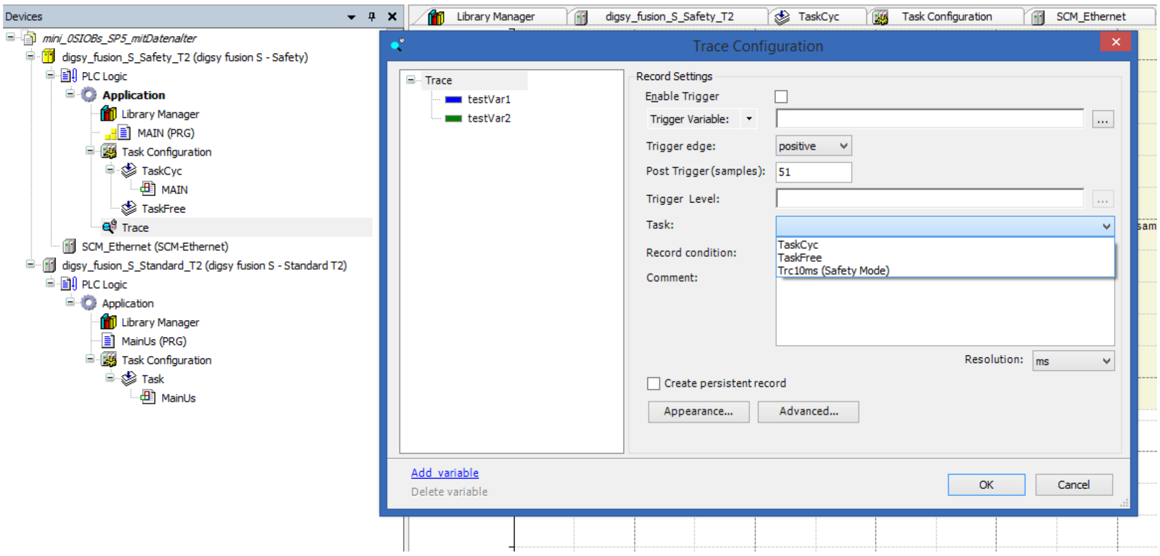


Figure 7-10: Trace

7.3.5 Usage of Breakpoints

From device version V02.15.00 (device description version 3.5.5.4) the ***digsy***®$_{\text{fusion}}$ S in CODESYS supports four breakpoints. These can be used to stay at defined points in the code.

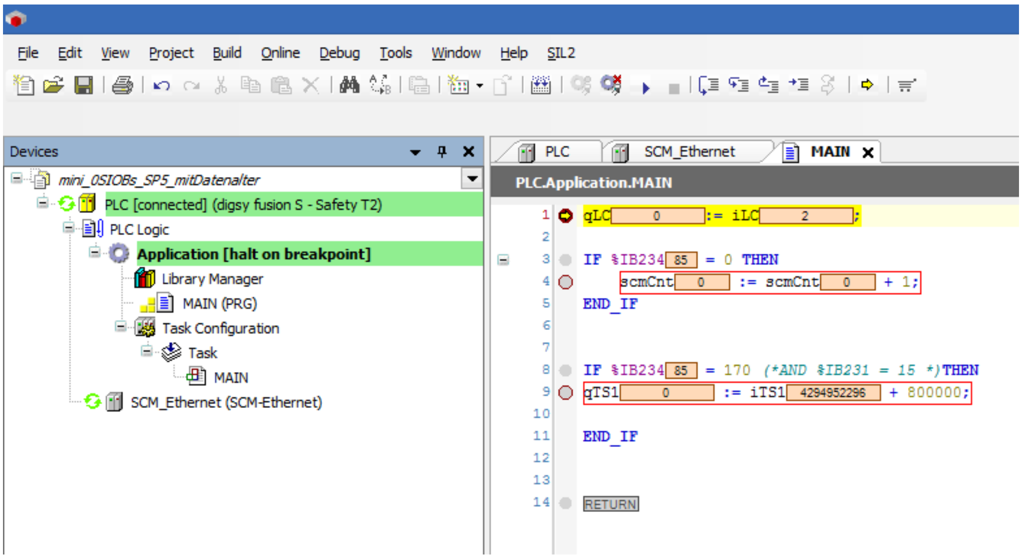


Figure 7-11: Breakpoints

Note: Breakpoints can only be used in debug mode!

7.3.6 Safety notes

Safety information and safety requirements	
The programming guidelines and instructions in the CODESYS Safety Manual V3.5 SIL2 (exact version see chapter 1.6) must be observed and complied with in all cases.	§6977
When programming a Safe Application, only libraries with the SIL flag set may be used.	§6982
digsy®fusion S fulfills the requirements set out in EN ISO 13849-1:2015 ([N1]), Category 3, PL d, provided that input channels in two-channel topology are used.	§5078
An FTP server is running on the controller. This can be reached via Ethernet and must be activated in the CODESYS configuration file.	§8016
The safe runtime and the standard runtime of the PLC supports the feature „CODESYS Trace"	§8004 §8005
The safe runtime and the standard runtime of the PLC supports the feature „CODESYS Breakpoints"	§8006

Table 7-5: Safety notes

7.3.7 Versions of standard application

Version information	
The versions of the device description (devdesc), the system libraries and the CODESYS development environment must be used for the respective PLC version. (see chapter 1.6 for versions of the safety application)	
4888.02.XXX	
PLC version V02.07.XX:	
digsy_fusion_S-Standard.devdesc.xml	Version: 3.5.5.0
digsy_fusion_S-Standard_T2.devdesc.xml	Version: 3.5.5.0
SCM-Ethernet-Std.devdesc.xml	Version: 0.5.0.0
SIOM-Basic-Std.devdesc.xml	Version: 0.5.0.0
DFS_Std.compiled-library	Version: 0.1.0.4
PLC version V02.08.XX:	
digsy_fusion_S-Standard.devdesc.xml	Version: 3.5.5.0
digsy_fusion_S-Standard_T2.devdesc.xml	Version: 3.5.5.0
SCM-Ethernet-Std.devdesc.xml	Version: 0.5.0.0
SIOM-Basic-Std.devdesc.xml	Version: 0.5.0.0
DFS_Std.compiled-library	Version: 0.1.0.4

Version information

PLC version V02.10.XX:

digsy_fusion_S-Standard.devdesc.xml	Version: 3.5.5.0
digsy_fusion_S-Standard_T2.devdesc.xml	Version: 3.5.5.0
SCM-Ethernet-Std.devdesc.xml	Version: 0.5.0.0
SIOM-Basic-Std.devdesc.xml	Version: 0.5.0.0
DFS_Std.compiled-library	Version: 0.1.0.4

PLC version V02.16.XX:

digsy_fusion_S-Standard.devdesc.xml	Version: 3.5.5.1
digsy_fusion_S-Standard_T2.devdesc.xml	Version: 3.5.5.1
SCM-Ethernet-Std.devdesc.xml	Version: 0.6.0.0
SIOM-Basic-Std.devdesc.xml	Version: 0.6.0.0
DFS_Std.compiled-library	Version: 0.1.0.4
DFS_Types.compiled-library	Version: 1.2.0.0

PLC version V03.26.XX:

digsy_fusion_S-Standard.devdesc.xml	Version: 3.5.5.1
digsy_fusion_S-Standard_T2.devdesc.xml	Version: 3.5.5.1
SCM-Ethernet-Std.devdesc.xml	Version: 0.6.0.0
SIOM-Basic-Std.devdesc.xml	Version: 0.6.0.0
DFS_Std.compiled-library	Version: 0.1.0.4
DFS_Types.compiled-library	Version: 1.2.0.0

PLC version V03.30.XX:

digsy_fusion_S-Standard.devdesc.xml	Version: 3.5.11.0
digsy_fusion_S-Standard_T2.devdesc.xml	Version: 3.5.11.0
SCM-Ethernet-Std.devdesc.xml	Version: 0.6.0.0
SIOM-Basic-Std.devdesc.xml	Version: 0.6.0.0
DFS_Std.compiled-library	Version: 0.1.0.4
DFS_Types.compiled-library	Version: 1.2.0.0

PLC version V03.35.XX:

digsy_fusion_S-Standard.devdesc.xml	Version: 3.5.11.0
digsy_fusion_S-Standard_T2.devdesc.xml	Version: 3.5.11.0
SCM-Ethernet-Std.devdesc.xml	Version: 0.6.0.0
SIOM-Basic-Std.devdesc.xml	Version: 0.6.0.0
GIOM-Basic-Std.devdesc.xml	Version: 0.6.0.0
DFS_Std.compiled-library	Version: 0.1.0.4
DFS_Types.compiled-library	Version: 1.3.0.0

PLC version V03.36.XX:

digsy_fusion_S-Standard.devdesc.xml	Version: 3.5.11.0
digsy_fusion_S-Standard_T2.devdesc.xml	Version: 3.5.11.0
SCM-Ethernet-Std.devdesc.xml	Version: 0.6.0.0
SIOM-Basic-Std.devdesc.xml	Version: 0.6.0.0
GIOM-Basic-Std.devdesc.xml	Version: 0.6.0.0

Version information	
DFS_Std.compiled-library	Version: 0.1.0.4
DFS_Types.compiled-library	Version: 1.3.0.0
PLC version V03.37.XX:	
digsy_fusion_S-Standard.devdesc.xml	Version: 3.5.11.0
digsy_fusion_S-Standard_T2.devdesc.xml	Version: 3.5.11.0
SCM-Ethernet-Std.devdesc.xml	Version: 0.6.0.0
SIOM-Basic-Std.devdesc.xml	Version: 0.6.0.0
GIOM-Basic-Std.devdesc.xml	Version: 0.6.0.0
DFS_Std.compiled-library	Version: 0.1.0.4
DFS_Types.compiled-library	Version: 1.3.0.0
4888.03.XXX	
PLC version V01.00.XX:	
GCM-P.devdesc.xml	Version: 3.5.11.1
SCM-Eth-SL-Std.devdesc.xml	Version: 0.6.0.0
SIOM-Basic-Std.devdesc.xml	Version: 0.6.0.0
DFS-P_Std.compiled-library	Version: 1.5.0.0
DFS_Types.compiled-library	Version: 1.2.0.0
PLC version V01.05.XX:	
GCM-P.devdesc.xml	Version: 3.5.11.1
SCM-Eth-SL-Std.devdesc.xml	Version: 0.6.0.0
SIOM-Basic-Std.devdesc.xml	Version: 0.6.0.0
GIOM-Basic-Std.devdesc.xml	Version: 0.6.0.0
DFS-P_Std.compiled-library	Version: 1.6.0.0
DFS_Types.compiled-library	Version: 1.3.0.0
Steuerungsversion V01.06.XX:	
GCM-P.devdesc.xml	Version: 3.5.11.1
SCM-Eth-SL-Std.devdesc.xml	Version: 0.6.0.0
SIOM-Basic-Std.devdesc.xml	Version: 0.6.0.0
GIOM-Basic-Std.devdesc.xml	Version: 0.6.0.0
DFS-P_Std.compiled-library	Version: 1.6.0.0
DFS_Types.compiled-library	Version: 1.3.0.0
Steuerungsversion V01.09.XX:	
GCM-P.devdesc.xml	Version: 3.5.11.1
SCM-Eth-SL-Std.devdesc.xml	Version: 0.6.0.0
SIOM-Basic-Std.devdesc.xml	Version: 0.6.0.0
GIOM-Basic-Std.devdesc.xml	Version: 0.6.0.0
DFS-P_Std.compiled-library	Version: 1.6.0.0
DFS_Types.compiled-library	Version: 1.3.0.0
Steuerungsversion V01.10.XX:	
GCM-P.devdesc.xml	Version: 3.5.11.1
SCM-Eth-SL-Std.devdesc.xml	Version: 0.6.0.0

Version information	
SIOM-Basic-Std.devdesc.xml	Version: 0.6.0.0
GIOM-Basic-Std.devdesc.xml	Version: 0.6.0.0
DFS-P_Std.compiled-library	Version: 1.6.0.0
DFS_Types.compiled-library	Version: 1.3.0.0

Table 7-6: Version list of standard application

7.4 Usage of REAL-Numbers and mathematical functions in the safe application

The following chapter provides the user with information on the special features to be considered when using REAL numbers.

⚠ **WARNING**

Use of REAL / LREAL numbers (FPU) in a safe context

The use of floating point numbers (REAL/ LREAL) and the mathematical functions named in chapter 7.4.10 are allowed in a secure context at extended and system level.

1. By using floating-point numbers (REAL / LREAL), the calculation is performed in the floating-point unit (FPU), whereby the user must observe the special features of floating-point arithmetic. The following chapter provides assistance.
2. The 3S programming guideline „_H2__CODESYS_Safety_SIL2_-_IEC_Programming_Guidelines.pdf“ version 5.0 must be observed.
3. The safety manual 04-68509-010500_CODESYS_Runtime_Math-Lib_Safety_Manual.pdf must be observed when using the mathematical functions.

7.4.1 Precision „REAL“ and „LREAL“

REAL (32Bit single precision): 2^{-23} = approximately 10^{-7} (corresponds to float in C)

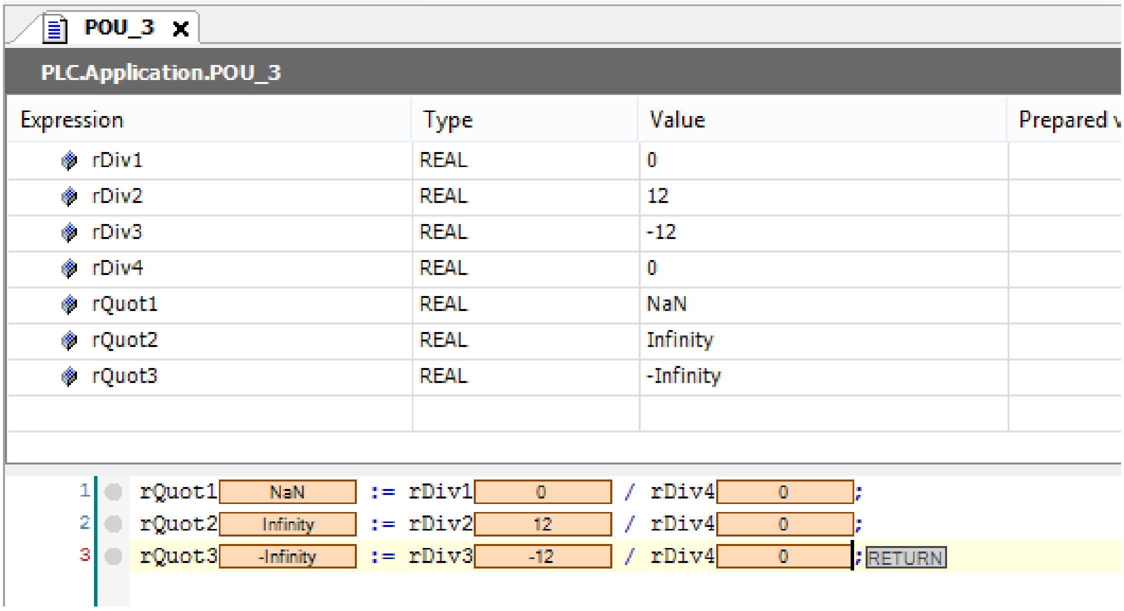
LREAL (64Bit double precision): 2^{-52} = approximately 10^{-15} (corresponds to double in C)

7.4.2 Special REAL_Numbers

Sign	Exponent	Mantissa	Value
0	0000 0000	000 0000 0000 0000 0000 0000	+0
1	0000 0000	000 0000 0000 0000 0000 0000	-0
0	1111 1111	000 0000 0000 0000 0000 0000	+∞
1	1111 1111	000 0000 0000 0000 0000 0000	-∞
?	1111 1111	??? ???? ???? ???? ???? ????	NaN

7.4.3 Division by 0

The division by 0 is defined differently than in the integer arithmetic in the standard IEEE 754. The result depends on the dividend (<0, > 0 or = 0).



7.4.4 Relative and absolute error in the basic operations

v: exact value

ṽ: Approximate value (calculated result)

The absolute maximum error is the difference between the exact value and the approximate value:

$\varepsilon_{abs} = |v - \tilde{v}|$ (REAL: $10^{-7} * v$)

The relative maximum error is the absolute error related to the value size:

$\varepsilon_{rel} = \frac{|v-\tilde{v}|}{v}$ (REAL: 10^{-7})

7.4.5 Example for absolute error

A REAL number is thus accurate to the seventh place. This results, for example, in the following accuracy:

Number	Absolute Precision
1	0,0000001
1000	0,0001
10^7	1
10^9	100

7.4.6 Addition/Subtraction

The subtraction is equivalent to the addition (addition with a negative summand).

7.4.6.1 Absolute error

$\tilde{v} + \tilde{y} = v + \varepsilon_{abs}(v) + y + \varepsilon_{abs}(y)$

$\tilde{v} + \tilde{y} = v + y + \varepsilon_{abs}(v) + \varepsilon_{abs}(y)$

$\boldsymbol{\varepsilon_{abs}}(v + y) = \boldsymbol{\varepsilon_{abs}}(v) + \boldsymbol{\varepsilon_{abs}}(y)$

The absolute error in the addition is the sum of the absolute errors of both summands.

The calculated result thus corresponds (for REAL):

Calculated result = exact result ± $\varepsilon_{abs}(v + y)$

The calculated result considering the absolute maximum error thus corresponds to (for REAL):

Calculated result = exact result ± $(10^{-7} * v + 10^{-7} * y)$

7.4.6.2 Relative error

$\tilde{v} + \tilde{y} = v * (1 + \varepsilon_{rel}(v)) + y * (1 + \varepsilon_{rel}(y))$

$\tilde{v} + \tilde{y} = v + y + v * \varepsilon_{rel}(v) + y * \varepsilon_{rel}(y)$

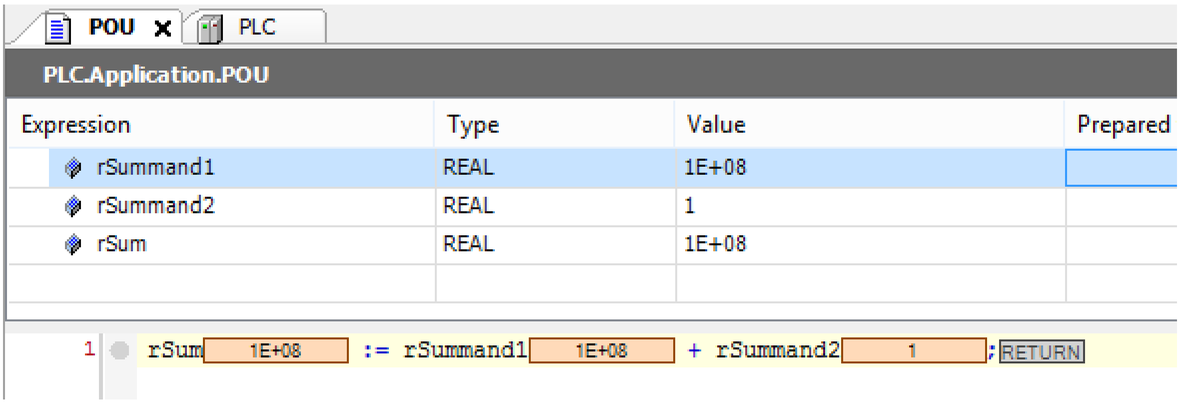
$\tilde{v} + \tilde{y} = (v+y) * \left(1 + \frac{|v|}{|v+y|} * \varepsilon_{rel}(v) + \frac{|y|}{|v+y|} * \varepsilon_{rel}(y)\right)$

$\boldsymbol{\varepsilon_{rel}(v + y) = \frac{|v|}{|v+y|}\varepsilon_{rel}(v) + \frac{|y|}{|v+y|}\varepsilon_{rel}(y)}$

	WARNING
	extinction Problems with | v + y | << 1. The relative error goes against infinity when subtracting approximately equal numbers (extinction).

7.4.6.3 Example Addition

When adding a very large and a very small number, the inaccuracy of REAL numbers can make a summand disappear. In the example, the first addend (rSummand1) is 10^8 and thus has an accuracy of 10. The second addend ((rSummand2) is 1 and therefore significantly smaller than the accuracy of the first addend and thus does not appear in the sum.



7.4.7 Multiplication/Division

The division is equivalent to multiplication (multiplication by the reciprocal).

7.4.7.1 Absolute error

$\tilde{v} * \tilde{y} = (v + \varepsilon_{abs}(v)) * (y + \varepsilon_{abs}(y))$

$\tilde{v} * \tilde{y} = v * y + v * \varepsilon_{abs}(y) + y * \varepsilon_{abs}(v) + \varepsilon_{abs}(v) * \varepsilon_{abs}(y)$

$\boldsymbol{\varepsilon_{abs}(v*y) = y* \varepsilon_{abs}(v) + v* \varepsilon_{abs}(y) + | \varepsilon_{abs}(v) + \varepsilon_{abs}(y)|}$

7.4.7.2 Relative error

$\tilde{v} * \tilde{y} = v * (1+ \varepsilon rel (v)) * y * (1+ \varepsilon rel (y))$

$\tilde{v} * \tilde{y} = v * y * (1+ \varepsilon rel (v) + \varepsilon rel (y) + \varepsilon rel (v) * \varepsilon rel (y))$

mit $\varepsilon rel (v) * \varepsilon rel (y) \ll \varepsilon rel (v) + \varepsilon rel (y)$

$\varepsilon rel (v*y) = \varepsilon rel (v) + \varepsilon rel (y)$

The relative error in multiplication is the sum of the errors of both factors.

The calculated result thus corresponds (for REAL):

Calculated result = exact result * ε_{rel} (vy)

The calculated result considering the absolute maximum error thus corresponds to (for REAL):

Calculated result = exact result * 2 * 10^{-7}

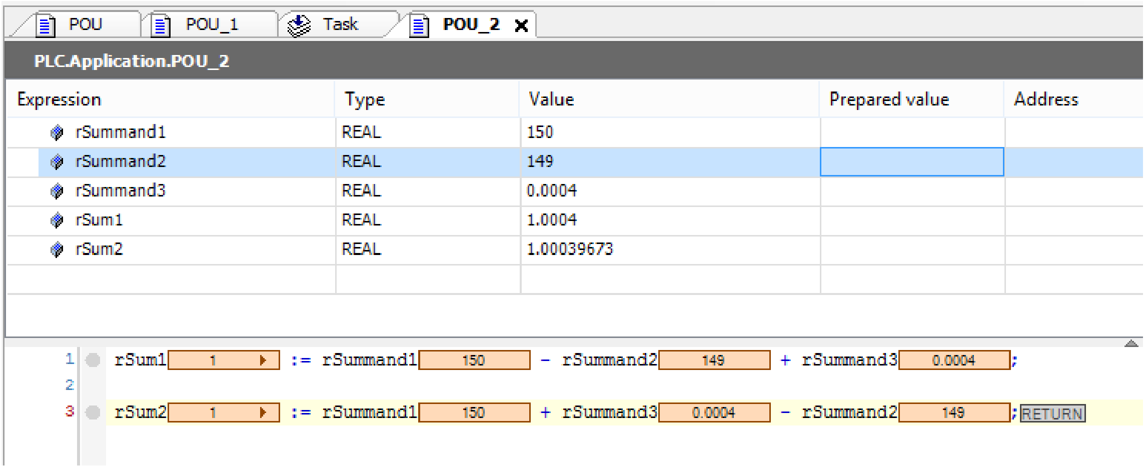
	WARNING
	Division by 0 Control does not go into the state OPMODE_FAILSAFE (see chapter 1.3) When dividing, make sure that the divisor does not become 0 by canceling or rounding.

7.4.8 Associative and Distributive Law

Addition and multiplication are not associative and non-distributive in floating-point numbers.

- $(x + y) + z \neq x + (y + z)$
- $(x * y) * z \neq x * (y * z)$
- $x * (y + z) \neq (x * y) + (x * z)$

To increase the accuracy of a calculation, therefore, the order of operations is important because the errors add up differently. In this case, it is advantageous if the numbers to be calculated are closer together in terms of value.



7.4.9 Comparing REAL numbers

REAL numbers should not be compared by the user with "IF rVar1 = rVar2 THEN". Due to the problems with the accuracy of REAL numbers, an if query is usually answered incorrectly (internally, the FPU calculates with 80-bit precision).

As a better solution, the user is recommended a rel. allow tolerance:

IF (ABS(rVar1 - rVar2) < ε) THEN

END_IF

ε: allowed rel. Tolerance (for example: $\varepsilon = 10^{-5} *$ rVar1)

7.4.10 Usage of mathmatical functions

The following types of calculations can be used in a safe context.

- Basic operations (+, -, *, /)
- MOD(x)
- COS(x), ACOS(x)
- SIN(x), ASIN(x)
- TAN(x), ATAN(x)
- EXPT(x)
- EXP(x)
- ABS(x)
- LN(x)
- LOG(x)
- SQRT(x)

	CAUTION
	Calculation of incorrect values When using the FPU and mathematical functions, attention must be paid to the achievable accuracies of the respective function. These can be found in 04-68509-010400_CODESYS_Runtime_Math-Lib_Safety_Manual.pdf. In addition, the user must be aware of the range of values used, since this is decisive for the absolute accuracy of the calculation.

7.4.11 Special behaviour of CODESYS

As an application developer of CODESYS programs, it is important to become familiar with the peculiarities of this programming language. For example, type casts (e.g. REAL_TO_DINT) are treated differently than, for example, in C.

Since these special features are important when programming with CODESYS, some of them are listed below:

- CODESYS rounds up or down to an integer value for REAL/LREAL type conversions and converts to the corresponding type. For example, x.5 is not always rounded uniformly.

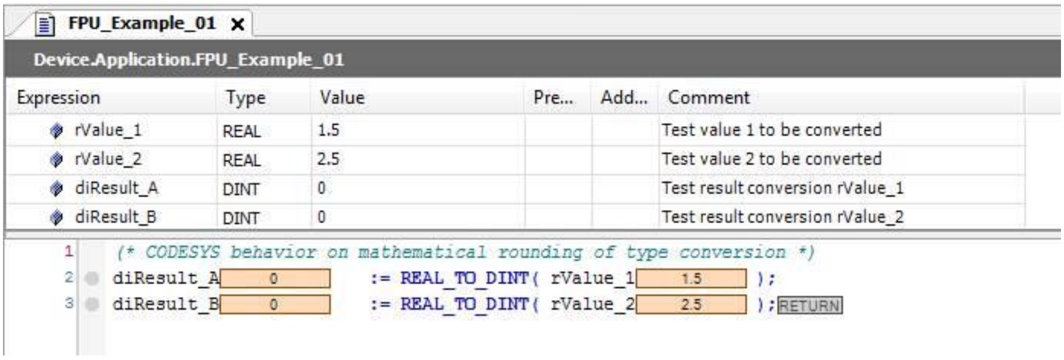


Figure 7-12: Special behaviour of CODESYS – type conversion REAL to Integer

- CODESYS expects either a 32bit or a 64bit operation. This depends on the input operand. This behavior is important for considering the accuracy of the calculation.

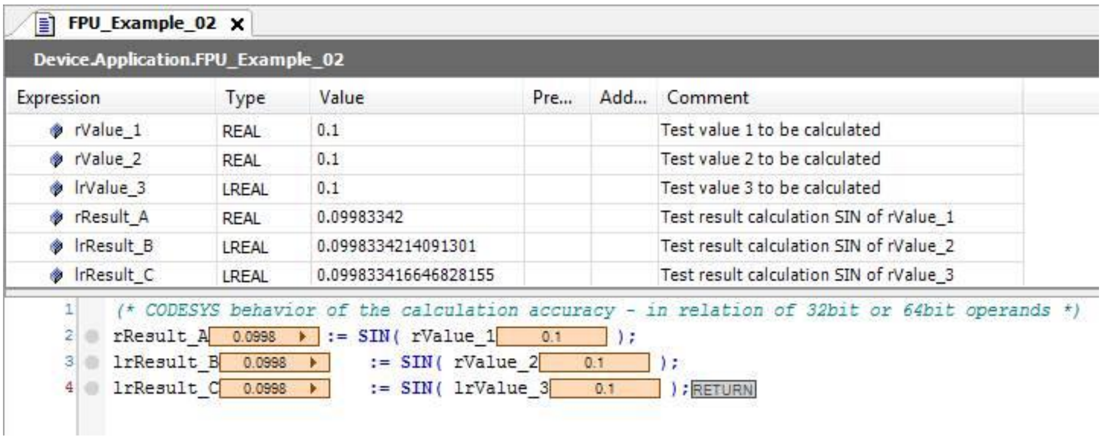


Figure 7-13: Special behaviour of CODESYS – type conversion 32/64bit

- The function TRUNC returns only one DINT. This means that the result is limited to DINT_MAX = 2147483647.

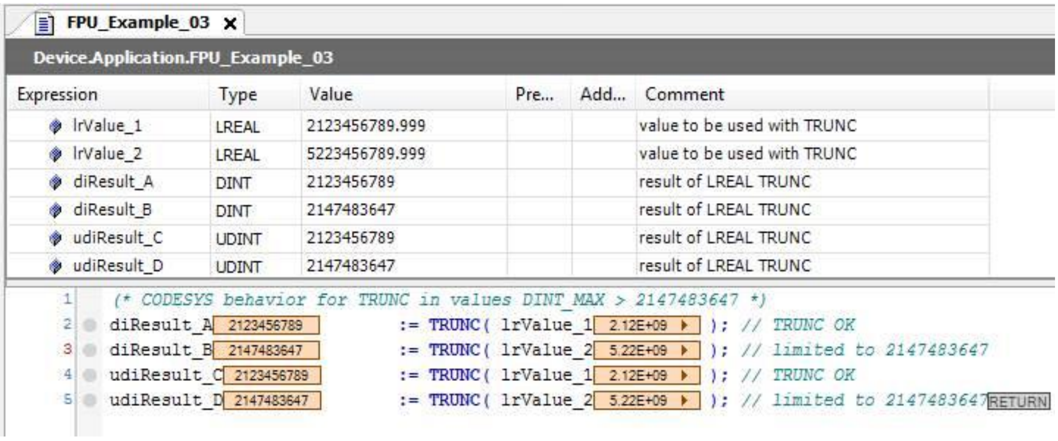


Figure 7-14: Special behaviour of CODESYS – TRUNC

- With integer arithmetic, CODESYS always casts on signed as soon as an operator is signed. Only if both operators are unsigned will also be cast on unsigned.
  The addition is cast to signed and then passed to the value. This differs to the programming language C.

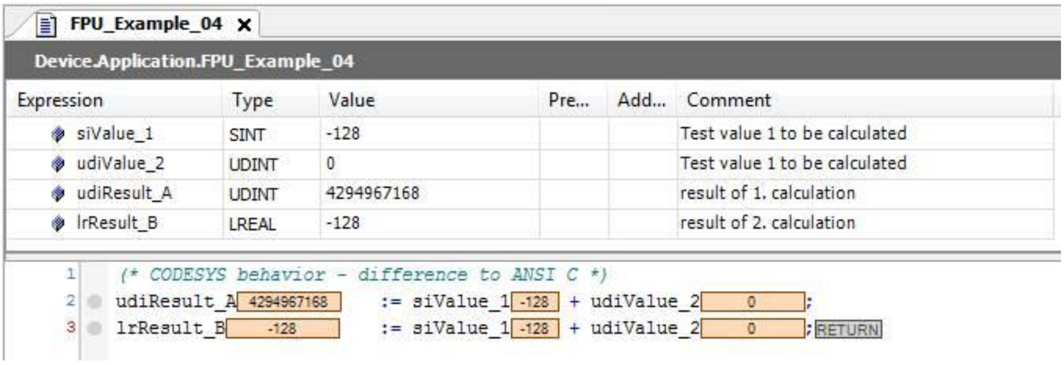


Figure 7-15: Special behaviour of CODESYS – Integer arithmetic

- Using constants may produce different results than using variables. The reason for this is that not the FPU but the compiler performs the calculation.

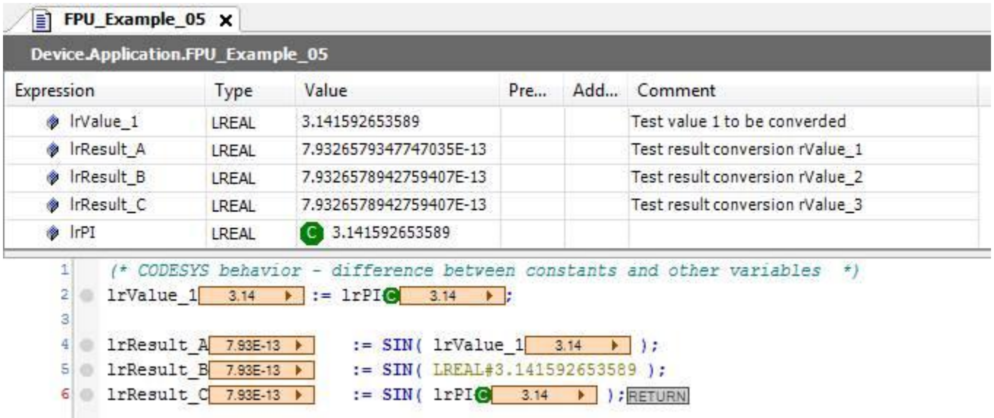


Figure 7-16: Special behaviour of CODESYS – Constants

7.5 Creating a project (CODESYS)

7.5.1 General

Within an *SCM*, ***digsy***®fusion S allows the simultaneous processing of two application programs which operate almost independently from each other. One of the programs runs in an environment which ensures functional safety, provided that the specified programming rules are observed. The second application program executes conventional control tasks and is not suitable for functional safety tasks. It must be ensured that the standard program does not interact with the safe program. Both applications, i.e. the Safe Application and the Standard Application, can be managed either within the same project or in separate projects.

For general information on application programming, refer to the 3S CODESYS programming manual.

When programming an application, the instructions provided in the following 3S manuals must be followed:

- IEC Programming Guidelines in version V5.0 (_H2__CODESYS_Safety_SIL2_-_IEC_Programming_Guidelines.pdf)

7.5.2 Safe Application program sAPP

The following objects must be integrated in order to be able to design a Safe Application:

- ***digsy***®fusion S (*digsy_fusion_safety(digsy fusion S – Safety*)) OR
  digsy®fusion S (*digsy_fusion_safety(digsy fusion S – Safety T2*)) OR
  digsy®fusion S (*digsy_fusion_safety(digsy fusion S – Safety T3*))
- SCM-Ethernet
- SIOM-Basic (where applicable)
- GIOM-Basic (where applicable)

In the case of ***digsy***fusion*S-P(L)*, these are the following objects:

- ***digsy***®fusion S-P (*digsy_fusion_safety(digsy fusion S-P – Safety*))
- SCM-Eth-SL
- SIOM-Basic (where applicable)
- GIOM-Basic (where applicable)

The following components can be added to the project by clicking the right mouse button:

- the program
- the task configuration

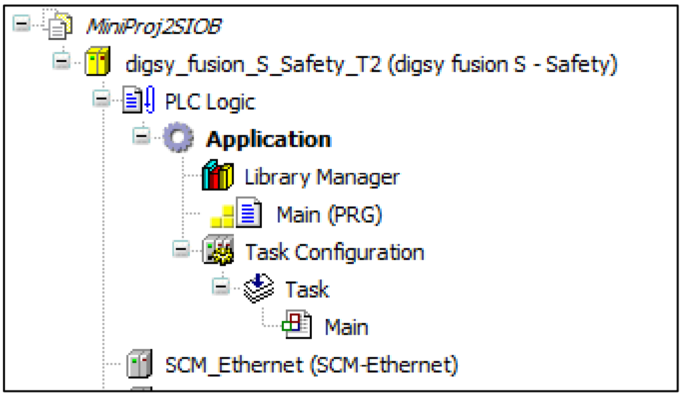


Figure 7-17: Safe application

To load the Safe Application, the control system must be switched to the Debug mode (7.5.4).

7.5.3 Standard Application Program stdAPP

To design a Standard Application, one of the following devices must be integrated into the project:

- digsy_fusion_S_standard (digsy fusion S – Standard) or
- digsy_fusion_S_standard (digsy fusion S – Standard T2)
- GCM_P (GCM_P)

In order to implement a standard application to a ***digsy***® fusion *S*-P, see [MAN-GCM-P].

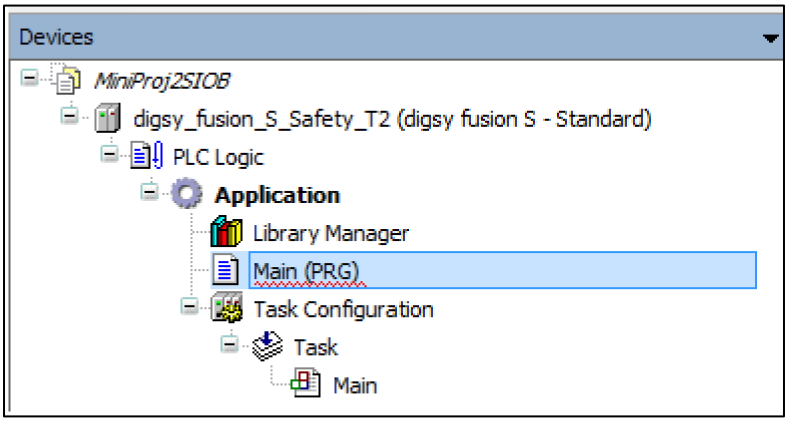


Figure 7-18: Standard application

To load the Standard Application, the control system must be transferred to the Debug mode (7.5.4).

7.5.4 The DEBUG mode

The debug mode is used for following use cases:

- for debugging the application programs

- to load the secure user program sAPP
- to load the standard user program stdAPP

To activate the DEBUG state you need an active connection to the safe device, e.g. digsy_fusion_S_Safety_T2.

In the CODESYS menu bar, below the SIL2 button (see Figure 7-18), there is the option “Enter debug mode...”, which can be used to put the controller into debug mode.

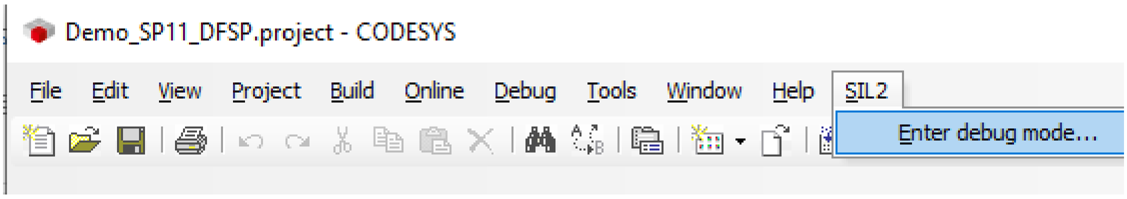


Figure 7-19: SIL2 Enter debug mode

To exit the Debug mode, the digsy®fusionS control system must be reset.

WARNING

The control system is not safe in the Debug mode.
Various safety measures are not active.

1. To exit the Debug mode after use, restart the device.

7.5.5 Task structure

digsy®fusion S enables the user to divide a program into various tasks which can be assigned different processing times. You can run the following number of tasks in ***digsy***®fusion S:

Safe Application:

There are a maximum of four tasks available. These can be divided as follows:

- 2 cyclical tasks
- 2 free-running tasks
- 2 event-controlled tasks

Standard Application:

There are a maximum of six tasks available. These can be divided as follows:

- 1 cyclical task
- 3 free-running tasks
- 3 event-controlled tasks

Note

Assignment of task priorities
Since the cyclical tasks are interrupt-driven, they must be assigned the priorities 0 or 1.
Free-running tasks run in a background loop and therefore have a lower priority.
The task times of cyclical tasks must be a multiple of 10ms.

Every cyclical task can be assigned a cycle time. The tasks are processed depending on their priority. The example (see Figure 7-20) illustrates 3 tasks (with cycle times of 30ms / 70ms) and different priorities. The cyclical task with prio0 needs 10ms, the cyclical task with prio1 needs 20ms and the free running Task needs 20ms. It can be seen that the task with the highest priority is processed first. The cyclic task with the second highest priority is then processed. As a last step, all the free running tasks are passed through.

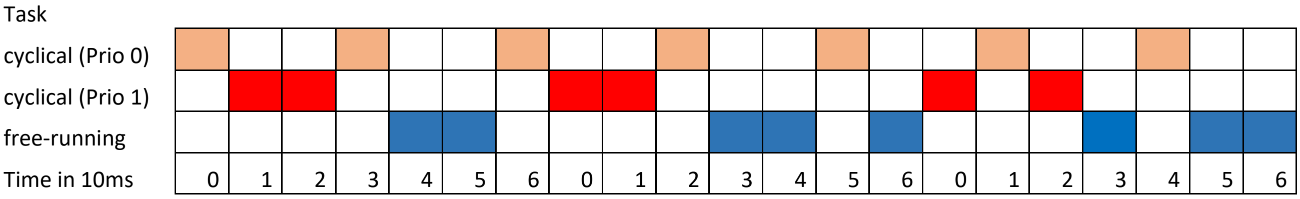


Figure 7-20: Task structure

If an unsuitable task processing time and cycle time (30ms / 30ms) are selected, it may occur that the task is no longer processed. This is illustrated in the example in Figure 7-21. Due to the extended processing time of the second cyclical task, the first task is repeated immediately after the completion of the second task. This means that the cyclical tasks are active all time. As a result, the free-running task is never executed.

To avoid this problem illustrated in the example the cycle time of a task must be reduced by performing an optimization.

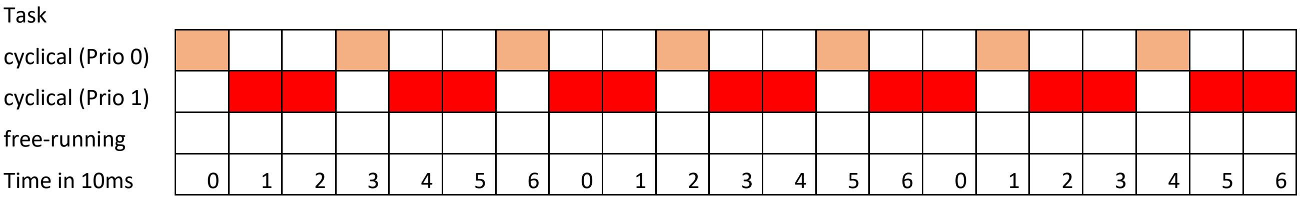


Figure 7-21: Task structure if the processing time is too long

Note

For more detailed information on how to set up and handle tasks refer to the help system of the CODESYS programming system, "Task Configuration".

7.5.6 Data exchange between sAPP and stdAPP

For reasons of functional safety, the firmware component which ensures functional safety must also have control over all hardware resources. This means that the firmware component which controls the Standard application programs does not have immediate access to the inputs/outputs or interfaces.

Note

Access to inputs/outputs and interfaces

Direct physical access to inputs/outputs and interfaces is only and exclusively possible in the safe context, i.e. within the scope of the safe functioning firmware. This firmware makes the input/output data available to both the Safe Application program and the Standard Application program.

7.5.6.1 *digsy*®fusion S

The standard APP thus accesses the inputs/outputs and interfaces indirectly via memory areas in the RAM, also referred to as the Shared Memory. This Shared Memory is subdivided into several areas:

- Flat Area AP: Data exchange between the Safe APP and the Standard APP
- CAN area: Exchange of CAN receive data and CAN transmit data

- RS232 area: Exchange of RS232 receive data and RS232 transmit data
- Ethernet area: Exchange of Ethernet receive data and Ethernet transmit data
- File area: Read/write data exchange from the file system
- Debug area: Data exchange within the scope of system debugging

For more detailed information on the data exchange between the Safe Application and the Standard Application refer to Chapter 8.1.13.

7.5.6.2 *digsy*®fusion S-P(L)

The data exchange between SCM SL (safe application) and GCM-P(L) (standard application) takes place via shared memory functions (SHM). A description of the functions is given in chapter 8.

7.5.7 IEC1131 variables

Several variable types are distinguished for IEC programming:

- Local variables
- Global variables
- Global constants
- Network variables
- Retain variables
- Variables of the I/O configuration
- Input, output or IN_OUT variables
- Implicit variables

7.5.7.1 Local variables

The local variables of a block are declared as 'local variables'. They have no external connection, and direct write access to local variables is therefore not possible. Before the start of the application, local variables can be pre-initialized in the program blocks and function blocks. In functions these only serve as auxiliary variables and do not have a defined call-up value. Local variables are reset to their original value after a reset or download.

7.5.7.2 Global variables

Global variables can be created via "Resources → Global variables". Global variables can be accessed beyond individual projects and tasks.

Global variables are erased upon a start-up, reset, and download, or an initial value is pre-assigned.

WARNING

It must be ensured that DWORD or REAL variables can be accessed consistently beyond the individual task. All variables, arrays or structures must be protected against each other using appropriate mechanisms.

7.5.7.3 (Global) constants

Constants can be created locally and globally. Constants are read-only and always have an initialization value.

7.5.7.4 Network variables

The values of network variables can be exchanged between different controlling devices in a network.

Network variables are only available on the standard application.

Network variable cannot be use on the safety application.

7.5.7.5 Non-volatile Variables (RETAIN)

General

Retain data are non-volatile data and thus are not lost by voltage drops, when handled according to the following instructions. A memory of 16kBytes (2 blocks a 8kByte) in the FRAM (non-volatile memory) is used for the retain data in the ***digsy***®fusion S. For the user, 8kByte are available in the CODESYS application due to the double storage.

Creation of retain data

To create retain data, the user has to assign variables with the key word "RETAIN" when defining it.

Example "Creating a non-volatile variable":

```
VAR RETAIN
    uliRetainVar:   ULINT;   // non-volatile variable
END_VAR
```

	Note
	It is recommended to define retain variables as global data in one block and not in different modules.

Example "Creating a non-volatile structure":

```
VAR_GLOBAL RETAIN
    structRetVar:    retVar;
END_VAR
// with the structure
TYPE retVar:
    STRUCT
        byRetainVar1:   BYTE;
        byRetainVar2:   BYTE;
    END_STRUCT
END_TYPE
```

	CAUTION
	Change of addresses after new compilation by CODESYS When the user program is recompiled in CODESYS, variables may no longer have the same memory addresses as before the compilation. This allows data from the FRAM to be read back and assigned to incorrect variables. In order to avoid this, the following points should be implemented in the application: 1. Packing all retain data in one structure 2. Define all retain data at a single point in the user program (global module)

Pre-initialized retain data:

It is possible to pre-set individual retain variables with values. The retain data are then initialized with these values if one of the following occurs:

- No call of the function LoadRetain()
- Error writing the retain data

```
VAR_GLOBAL RETAIN
    byRetainVar1:   BYTE := 1;
    byRetainVar2:   BYTE := 2;
END_VAR
```

Load retain data

Retain data are not automatically reloaded and made available to the application level at start-up (in contrast to other CODESYS devices). This operation must be started within the user program by calling the function "LoadRetain()" (for more information, see functions in chapter 8.1.10).

Save retain data

Retain data are not automatically saved in the ***digsy***®fusion S. This operation must be started within the user program by calling the function "SaveRetain()" (for more information, see functions from chapter 8.1.10).

Internal function Save/Load retain data

The memory size of the retain data are two blocks of 8kB.

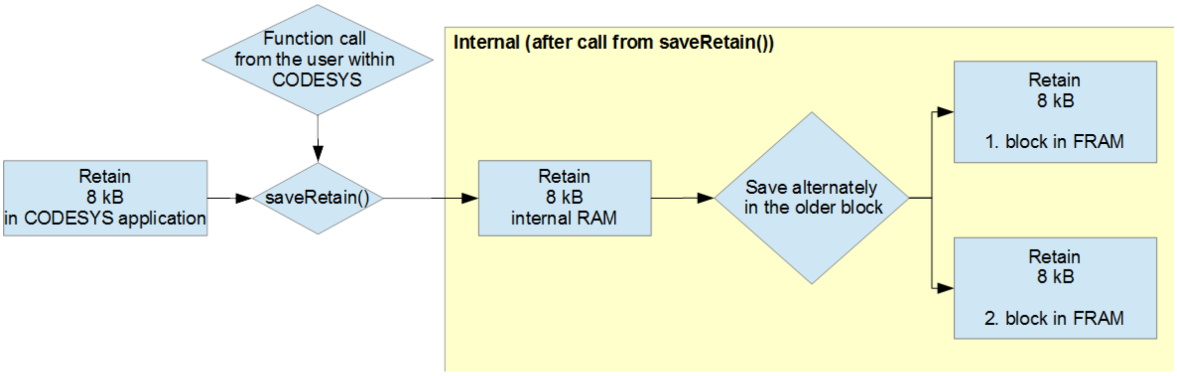


Figure 7-22: Save retain data

Retain data are saved in the ***digsy***®fusion S when the SaveRetain() function is called. During this call the complete 8kB block of the user program (retain data internally protected by a CRC) is written to the internal RAM and then copied in background to the FRAM. CODESYS outputs warnings or errors if the return data range exceeds 8kB.

The next time the SaveRetain() function is called (can not started twice if first SaveRetain() is not done), the complete 8kB block of the user program is written to the internal RAM and then to the other block (FRAM). This always happens alternately and is internally controlled by the firmware. Because of this behaviour, one FRAM block always contains the most recent data. The data in the second block are one save cycle older. This means that the older block can still be accessed if writing of the last block failed.

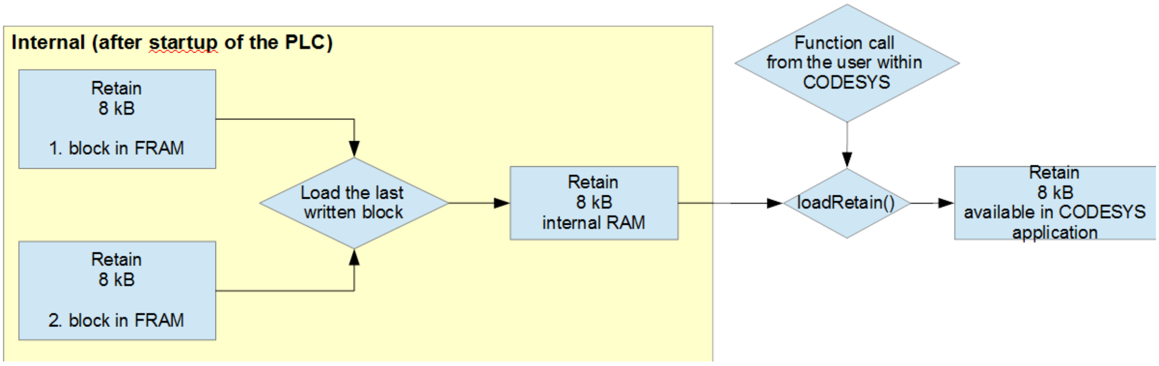


Figure 7-23: Reading of retain data

After the start-up of the PLC, the data are automatically loaded from the FRAM into the RAM (firmware). The last saved 8kB block is always loaded from the FRAM. If this is faulty (CRC error), the other 8kB block is loaded from the FRAM. After calling the LoadRetain() function, the loaded retain data from the RAM are loaded into the application's memory and are available to the user in CODESYS.

If an error occurs (the older block is loaded), the 8kB block in the application is exact one writing cycle older than the last saved block. If there is no valid block available, the “LoadRetain()” function returns an error.

Caution

Load an corrupt block from the FRAM

In this case retain data do not correspond to the last saved data

If it is a problem for the user when older retain data are loaded, the user must call the SaveRetain() function twice each time to save the retain data, in order to ensure that both 8k blocks have the same content.

Verification of the retain data after loading

After loading the retain data (after booting the PLC) using “LoadRetain()”, the user has to care about the return value of the function “LoadRetain()”. This must correspond to the error value ERROR.NO_ERROR if loading was successful.

In addition, the user must verify the read variables (check whether the data is valid and fits to the specific application program). The best way to do this is by saving a unique identifier of the structure additional to the user data.

In the example (see below) the data were saved in a structure with an additional unique version (uliRetainVersion). After loading the retain data this unique version is then compared with the valid version (UliRetainVersionValid) in the user program.

If the user subsequently modifies the structure of the retain data, he can increment this version and ensure that retain data with older structures are not loaded.

An additional method for verifying the data would be the additional storage of the structure length or a CRC calculated via the structure. These two possibilities would have the advantage that they can be executed automatically in the user program.

Example "Creating a non-volatile structure with version":

```
VAR_GLOBAL RETAIN
	uliRetainVersion:	ULINT;
	structRetVar:		retVar;
END_VAR
// with the structure
TYPE retVar:
	STRUCT
		byRetainVar1:	BYTE;
		byRetainVar2:	BYTE;
	END_STRUCT
END_TYPE
```

Example of loading the retain data after booting the controller:

```
// load Retain after Init
IF (LoadRetain( )= ERROR.NO_ERROR) THEN
	// check Version of Retaindata
	IF (uliRetainVersion <> uliRetainVersionValid) THEN
		// Error => new initialisation of the variables
		structRetVar.byRetainVar1 := 0;
		structRetVar.byRetainVar2 := 0;
	END_IF
END_IF
```

⚠ WARNING

Loading false retain data

Use of incorrect values

1. In addition to the retain data, the user should store a unique identifier (e.g., version) of the structure
2. When changing the structure (e.g., adding, moving variables), the user must ensure that this unique identifier changes in the user program.
3. After loading the retain data, the user must compare the loaded ID with the valid ID (user program) and, if incompatible, take appropriate actions to prevent misbehaviour.

⚠ CAUTION

Use of unauthorized data

Misconduct by e.g. division with zero

1. Before using the retain data, the user must check the permissible value range
2. If the retain data are outside this range, the user must ensure that appropriate actions are taken to prevent misbehaviour.

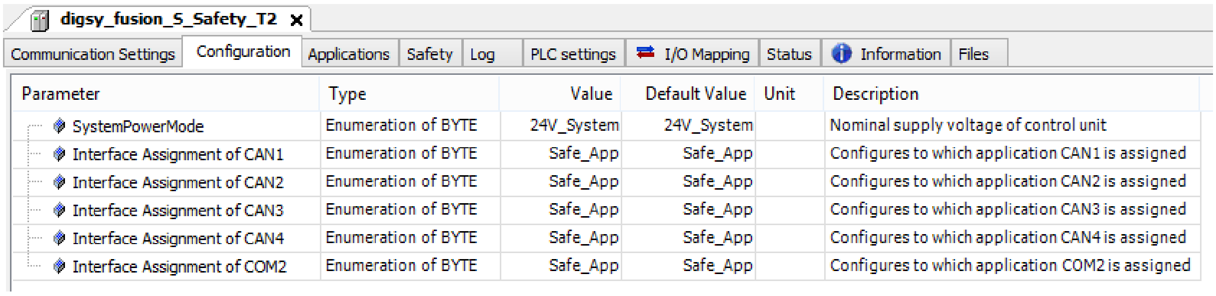
Erasing of the retain data

The retain data can be erased by calling the “SaveRetain()” function twice with pre-initialized (0) retain data, after starting the PLC. It is important that this process takes place before calling the function “LoadRetain()”, otherwise the stored retain data will be written back. After restarting the PLC and calling “LoadRetain()”, the erased retain data are recognized as valid retain data. This means that the values of the pre-initialization are not in the variables, but the variables all have the value zero.

	CAUTION
	Retain data update via USB Retain data can be copied to the controller via USB interface. (see chapter 7.1.4)

7.6 I/O mapping and configuration

7.6.1 General information on *digsy*®fusion S



Parameter	Type	Value	Default Value	Unit	Description
SystemPowerMode	Enumeration of BYTE	24V_System	24V_System		Nominal supply voltage of control unit
Interface Assignment of CAN1	Enumeration of BYTE	Safe_App	Safe_App		Configures to which application CAN1 is assigned
Interface Assignment of CAN2	Enumeration of BYTE	Safe_App	Safe_App		Configures to which application CAN2 is assigned
Interface Assignment of CAN3	Enumeration of BYTE	Safe_App	Safe_App		Configures to which application CAN3 is assigned
Interface Assignment of CAN4	Enumeration of BYTE	Safe_App	Safe_App		Configures to which application CAN4 is assigned
Interface Assignment of COM2	Enumeration of BYTE	Safe_App	Safe_App		Configures to which application COM2 is assigned

Figure 7-24: Configuring ***digsy***®fusion S – General information

7.6.1.1 Configuration

- **SystemPowerMode**

  The underlying rated supply voltage of the internal switching levels can be set to 12V or 24V: Currently only 24V are possible. The setting (configuration) of 12V is not allowed.
  Valid switching levels see chapter 10.13.
  Valid system voltages see also chapter 4.1.3.

	HAZARD
	Selecting 12V for variable SystemPowerMode The user is not allowed to set the variable SystemPowerMode to 12V.

- **Interface assignment of CAN/COM**

  You can define which application uses the corresponding interface.

  The following configuration options are available for the CAN interfaces 1 to 4:

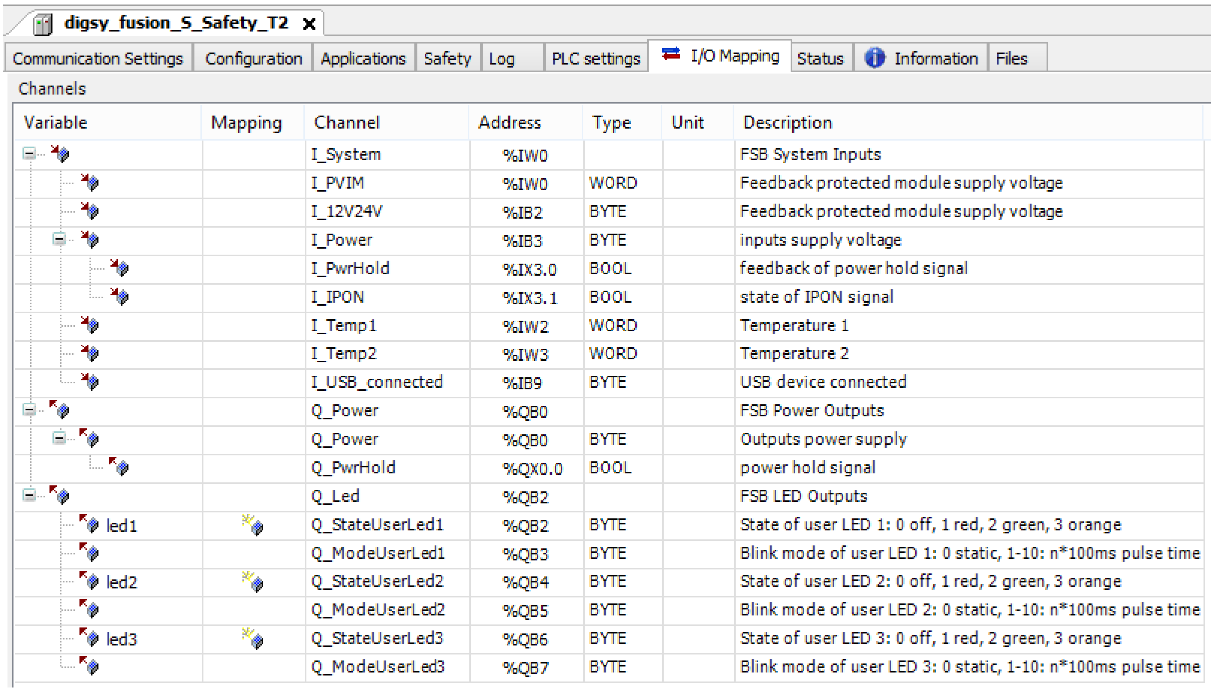
  - Safe_App
    The interface is used in the Safe Application.
  - Safe_App (Std_App listen) (not with *digsy*®fusion S-P(L))

The interface is used in the Safe Application. The Standard Application can access the interface in the LISTEN mode.

- Std_App (not with ***digsy***®fusion S-P(L))
  The interface is used in the Standard Application

- Std_App (Safe _App listen) (not with ***digsy***fusion S-P(L))
  The interface is used in the Standard Application. The Safe Application can access the interface in the LISTEN mode.

7.6.1.2 I/O mapping


digsy_fusion_S_Safety_T2

Communication Settings | Configuration | Applications | Safety | Log | PLC settings | I/O Mapping | Status | Information | Files

Channels

Variable	Mapping	Channel	Address	Type	Unit	Description
		I_System	%IW0			FSB System Inputs
		I_PVIM	%IW0	WORD		Feedback protected module supply voltage
		I_12V24V	%IB2	BYTE		Feedback protected module supply voltage
		I_Power	%IB3	BYTE		inputs supply voltage
		I_PwrHold	%IX3.0	BOOL		feedback of power hold signal
		I_IPON	%IX3.1	BOOL		state of IPON signal
		I_Temp1	%IW2	WORD		Temperature 1
		I_Temp2	%IW3	WORD		Temperature 2
		I_USB_connected	%IB9	BYTE		USB device connected
		Q_Power	%QB0			FSB Power Outputs
		Q_Power	%QB0	BYTE		Outputs power supply
		Q_PwrHold	%QX0.0	BOOL		power hold signal
		Q_Led	%QB2			FSB LED Outputs
led1		Q_StateUserLed1	%QB2	BYTE		State of user LED 1: 0 off, 1 red, 2 green, 3 orange
		Q_ModeUserLed1	%QB3	BYTE		Blink mode of user LED 1: 0 static, 1-10: n*100ms pulse time
led2		Q_StateUserLed2	%QB4	BYTE		State of user LED 2: 0 off, 1 red, 2 green, 3 orange
		Q_ModeUserLed2	%QB5	BYTE		Blink mode of user LED 2: 0 static, 1-10: n*100ms pulse time
led3		Q_StateUserLed3	%QB6	BYTE		State of user LED 3: 0 off, 1 red, 2 green, 3 orange
		Q_ModeUserLed3	%QB7	BYTE		Blink mode of user LED 3: 0 static, 1-10: n*100ms pulse time

Figure 7-25: I/O mapping ***digsy***®fusion S – General information

Name	Explanation
I_PVIM	This variable indicates the supply voltage in mV
I_12V24V	This variable indicates the rated supply voltage configured
I_PwrHold	Read-back of the power hold signal
I_IPON	Read-back of the IPON signal
I_TempX	The I_Temp1 and I_Temp2 variables provide the values of the two temperature sensors installed on the board. The values are indicated in 0.1 * Kelvin. Example: Value = 3200 => 320 Kelvin => 46.85 C
I_USB_connected	This variable indicates whether a USB stick is connected (value = 1) or not connected (value = 0) to the control system.

Name	Explanation
Q_PwrHold	This variable is used to set the Power Hold output. If Power Hold is set, the control system is continued to be supplied independent of the IPON signal. For WRITE operations (e.g. retain or file system) we recommend to set Power Hold in order to ensure the power supply of the control system.
Q_StateUserLedX	Q_StateUserLed1, Q_StateUserLed2 and Q_StateUserLed3 can be used to program the LEDs available in the application program.
Q_ModeUserLedX	Q_ModeUserLed1, Q_ModeUserLed2 and Q_ModeUserLed3 can be used to define the flashing of the LEDs available in the application program. 0: static 1-10: n * 100ms pulse time

7.6.2 SCM/SIOM/GIOM expansion board

7.6.2.1 Configuration

SCM_Ethernet

Internal Configuration | Internal I/O Mapping | Status | Information

Parameter	Type	Value	Default Value	Unit	Description
PinCfg					Pin Configuration
I_PinCfg_Q1_1	Enumeration of BYTE	Digital Output 2.5A	Disabled		Pin Configuration Q1.1
I_PinCfg_Q1_2	Enumeration of BYTE	Digital Output 2.5A	Disabled		Pin Configuration Q1.2
I_PinCfg_Q1_3	Enumeration of BYTE	Digital Output 2.5A	Disabled		Pin Configuration Q1.3
I_PinCfg_Q1_4	Enumeration of BYTE	Digital Output 2.5A	Disabled		Pin Configuration Q1.4
I_PinCfg_Q2_1	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q2.1
I_PinCfg_Q2_2	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q2.2
I_PinCfg_Q2_3	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q2.3
I_PinCfg_Q2_4	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q2.4
I_PinCfg_Q3_1	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q3.1
I_PinCfg_Q3_2	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q3.2
I_PinCfg_Q3_3	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q3.3
I_PinCfg_Q3_4	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q3.4
I_PinCfg_Q4_1	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q4.1
I_PinCfg_Q4_2	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q4.2
I_PinCfg_Q4_3	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q4.3
I_PinCfg_Q4_4	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q4.4
I_PinCfg_Q4_5	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q4.5
I_PinCfg_Q4_6	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q4.6
I_PinCfg_I3_A	Enumeration of BYTE	Disabled	Disabled		Pin Configuration I3.1 - I3.2
I_PinCfg_I3_B	Enumeration of BYTE	Disabled	Disabled		Pin Configuration I3.3 - I3.6
I_PinCfg_I1_1	Enumeration of BYTE	Disabled	Disabled		Pin Configuration I1.1
I_PinCfg_I1_2	Enumeration of BYTE	Disabled	Disabled		Pin Configuration I1.2
I_PinCfg_I1_3	Enumeration of BYTE	Disabled	Disabled		Pin Configuration I1.3
I_PinCfg_I1_4	Enumeration of BYTE	Disabled	Disabled		Pin Configuration I1.4
I_PinCfg_I1_5	Enumeration of BYTE	Disabled	Disabled		Pin Configuration I1.5
I_PinCfg_I1_6	Enumeration of BYTE	Disabled	Disabled		Pin Configuration I1.6
I_PinCfg_I1_7	Enumeration of BYTE	Disabled	Disabled		Pin Configuration I1.7
I_PinCfg_I1_8	Enumeration of BYTE	Disabled	Disabled		Pin Configuration I1.8
I_PinCfg_I2_1	Enumeration of BYTE	Analog Input 10V	Disabled		Pin Configuration I2.1 + I2.9
I_PinCfg_I2_2	Enumeration of BYTE	Analog Input 10V	Disabled		Pin Configuration I2.2 + I2.10
I_PinCfg_I2_3	Enumeration of BYTE	Analog Input 10V	Disabled		Pin Configuration I2.3 + I2.11
I_PinCfg_I2_4	Enumeration of BYTE	Analog Input 32V	Disabled		Pin Configuration I2.4 + I2.12
I_PinCfg_I2_5	Enumeration of BYTE	Analog Input 32V	Disabled		Pin Configuration I2.5 + I2.13
I_PinCfg_I2_6	Enumeration of BYTE	Analog Input 32V	Disabled		Pin Configuration I2.6 + I2.14
I_PinCfg_I2_7	Enumeration of BYTE	Analog Input 10V	Disabled		Pin Configuration I2.7 + I2.15
I_PinCfg_I2_8	Enumeration of BYTE	Analog Input 10V	Disabled		Pin Configuration I2.8 + I2.16
ConfigurationValues					Configuration values
Q_Threshold1	WORD	8000	6250		Configuration value for threshold voltage in mV (Inputs I1.1 - I1.4)
Q_Threshold2	WORD	8000	6250		Configuration value for threshold voltage in mV (Inputs I1.5 - I1.8)
Q_IoTimeInterval	DWORD	400	50		Time interval (in ms) for IO update monitoring (should be equal to time interval of bus cycle t...

Figure 7-26: SCM/SIOM/GIOM configuration

- **PinCfg**

Used to configure the inputs and outputs. For more detailed information refer to chapter 7.7.

- **Q_Threshold1**

  Threshold voltage for the type D inputs I1.1 – I1.4 (see 7.7.4)

- **Q_Threshold2**

  Threshold voltage for the type D inputs I1.5 – I1.8 (see 7.7.4)

- **Q_IoTimeInterval**

  Update time for the inputs/outputs. The value of Q_IoTimeInterval indicates the time window in which the I/O mapping is refreshed. If the time window is e.g. 100 ms, the I/O mapping is refreshed after at least 100 ms.
  Since the refresh of the I/O mapping is monitored by the firmware for safety reasons, the update time must correspond at least to the task time + the task jitter. If the specified time is exceeded, the control system switches to the *OPMODE_FAILSAFE_IO* operating mode. We recommend you define a multiple of the bus cycle task time as a default setting in the "Internal I/O Mapping" tab.

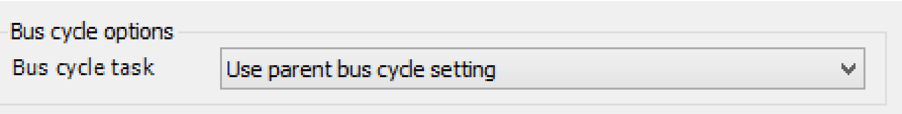


Figure 7-27: Bus cycle

- **Q_PeakCurrDetectEnable**

  Activation / deactivation of peak current detection for outputs Type A1, Type A2 and Type A3.

  The peak current detection detects short circuits for PWM outputs as well as current-controlled PWM outputs. Since a peak current can also occur in the case of large loads (for example halogen lamps) at PWM outputs, and thus there is a problem of availability, these can be switched on / off.

  When the peak current detection of an output is activated (set the bit to TRUE), the following sequence is performed when the short circuit is detected:

As standard configured output

1. Entry of a warning in the error memory
2. Set the associated error flag

The input can be switched on again after eliminating the error without restarting the digsyfusionS after switching off the output.

As safe configured output

1. Entry of an error in the error memory
2. Set the associated error flag

The output can not be restarted after a fault has been cleared without restarting the digsyfusionS.

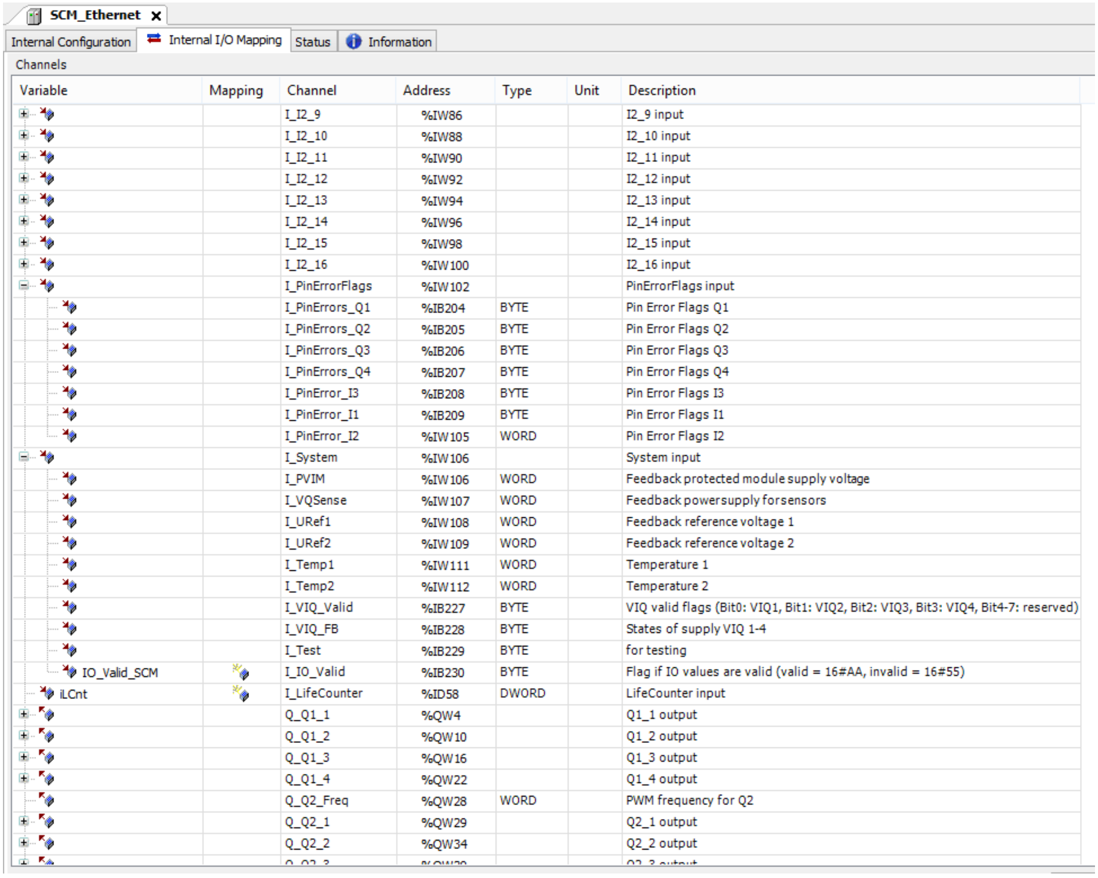
	⚠	**HAZARD**

⚠	**Deenergize wrong outputs** Malfunction of the control version V02.15.00 1. The peak current detection is active as of the device description (SCM-Ethernet / SIOM-Basic) version 1.1.0.0. 2. In PLC version V02.15.00 the peak current detection must not be used, because incorrect outputs can be switched off. 3. The bits of the peak current detection must all be configured with FALSE for PLC version V02.15.00. 4. In the control version V03.25.00, the peak current detection has been revised and may be used since this version.

i	**Note**
	Switch off the outputs Type A2 and A3 for HW version < VXX.40.XX For currents > 30A, the output drivers can be switched off (output switched off). This is possible without the possibility to recognize this in CODESYS application.

i	**Note**
	Switch off the outputs type A1, A2 and A3 for HW version ≥ VXX.40.XX For currents > 30A, the output driver can be switched off (output switched off). This is possible without the possibility to recognize this in CODESYS application.

7.6.2.2 I/O mapping


SCM_Ethernet
Internal Configuration | Internal I/O Mapping | Status | Information
Channels

Variable	Mapping	Channel	Address	Type	Unit	Description
		I_I2_9	%IW86			I2_9 input
		I_I2_10	%IW88			I2_10 input
		I_I2_11	%IW90			I2_11 input
		I_I2_12	%IW92			I2_12 input
		I_I2_13	%IW94			I2_13 input
		I_I2_14	%IW96			I2_14 input
		I_I2_15	%IW98			I2_15 input
		I_I2_16	%IW100			I2_16 input
		I_PinErrorFlags	%IW102			PinErrorFlags input
		I_PinErrors_Q1	%IB204	BYTE		Pin Error Flags Q1
		I_PinErrors_Q2	%IB205	BYTE		Pin Error Flags Q2
		I_PinErrors_Q3	%IB206	BYTE		Pin Error Flags Q3
		I_PinErrors_Q4	%IB207	BYTE		Pin Error Flags Q4
		I_PinError_I3	%IB208	BYTE		Pin Error Flags I3
		I_PinError_I1	%IB209	BYTE		Pin Error Flags I1
		I_PinError_I2	%IW105	WORD		Pin Error Flags I2
		I_System	%IW106			System input
		I_PVIM	%IW106	WORD		Feedback protected module supply voltage
		I_VQSense	%IW107	WORD		Feedback powersupply for sensors
		I_URef1	%IW108	WORD		Feedback reference voltage 1
		I_URef2	%IW109	WORD		Feedback reference voltage 2
		I_Temp1	%IW111	WORD		Temperature 1
		I_Temp2	%IW112	WORD		Temperature 2
		I_VIQ_Valid	%IB227	BYTE		VIQ valid flags (Bit0: VIQ1, Bit1: VIQ2, Bit2: VIQ3, Bit3: VIQ4, Bit4-7: reserved)
		I_VIQ_FB	%IB228	BYTE		States of supply VIQ 1-4
		I_Test	%IB229	BYTE		for testing
IO_Valid_SCM		I_IO_Valid	%IB230	BYTE		Flag if IO values are valid (valid = 16#AA, invalid = 16#55)
iLCnt		I_LifeCounter	%ID58	DWORD		LifeCounter input
		Q_Q1_1	%QW4			Q1_1 output
		Q_Q1_2	%QW10			Q1_2 output
		Q_Q1_3	%QW16			Q1_3 output
		Q_Q1_4	%QW22			Q1_4 output
		Q_Q2_Freq	%QW28	WORD		PWM frequency for Q2
		Q_Q2_1	%QW29			Q2_1 output
		Q_Q2_2	%QW34			Q2_2 output

Figure 7-28: SCM/SIOM/GIOM I/O mapping

For more detailed information on the correct use of the inputs/outputs refer to Chapter 7.7. This chapter describes the system variables.

Q_System

Name	Explanation
Q_VQSense	This variable switches the voltage supply of the sensors ON/OFF
Q_URefX	This variable switches the reference voltages ON/OFF
Q_IRefX	This variable switches the reference currents ON/OFF

Table 7-7: Declaration of Q_System

I_System

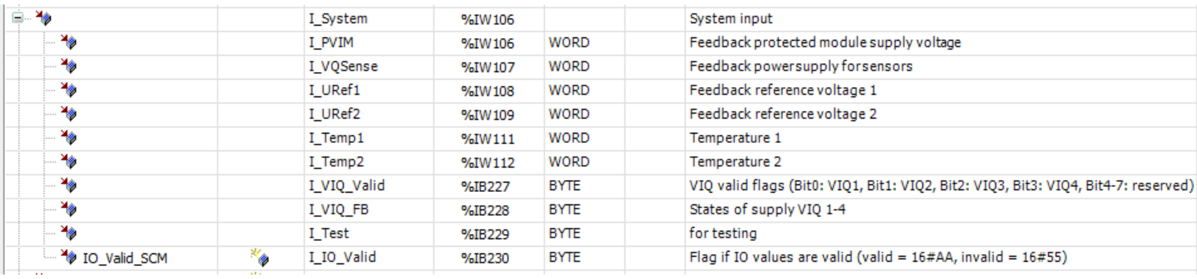
Name	Explanation
I_PVIM	This variable indicates the supply voltage in mV
I_VQSense	Read-back of the voltage supply value of the sensors
I_URefX	Read-back of the reference voltage
I_USys	System voltage about 3.3V (in mV)

Name	Explanation
I_TempX	The variables I_Temp1 and I_Temp2 provide the values of the two temperature sensors on the board. The values are indicated in 0.1 * Kelvin. Example: Value = 3200 => 320 Kelvin => 46.85 °C
I_VIQ_Valid	Validity of the VIQs. Indicates if the supply voltage is present on the VIQs. If the supply voltage is present, the individual VIQ bits are "high".
I_VIQ_FB	State of the second shutdown path of the VIQ output supply voltages. These flags indicate whether the second shutdown path of the VIQs has been enabled. In case of errors, i.e. internal diagnostic errors, I_VIQ_FB is "low" for the corresponding VIQ path although I_VIQ_Valid is "high".
I_Test	This variable is a test variable and not relevant for the user.
I_IO_Valid	See chapter 0
I_LifeCounter	See chapter 7.6.2.6
I_PinErrorFlags	See chapter 7.6.2.7
I_PinConfig	The defined pin configurations described in Chapter 7.6.2.1 can be read back.
I_Config_Values	The configured threshold voltages described in Chapter 7.6.2.1 can be read back.

Table 7-8: Declaration of I_System

7.6.2.3 I_IO_Valid

The value of IO_Valid must be 170 before using the inputs/outputs. Otherwise, the inputs/outputs are not valid. If the IO_Valid flags are 85, the inputs/outputs are invalid.



	I_System	%IW106		System input
	I_PVIM	%IW106	WORD	Feedback protected module supply voltage
	I_VQSense	%IW107	WORD	Feedback power supply for sensors
	I_URef1	%IW108	WORD	Feedback reference voltage 1
	I_URef2	%IW109	WORD	Feedback reference voltage 2
	I_Temp1	%IW111	WORD	Temperature 1
	I_Temp2	%IW112	WORD	Temperature 2
	I_VIQ_Valid	%IB227	BYTE	VIQ valid flags (Bit0: VIQ1, Bit1: VIQ2, Bit2: VIQ3, Bit3: VIQ4, Bit4-7: reserved)
	I_VIQ_FB	%IB228	BYTE	States of supply VIQ 1-4
	I_Test	%IB229	BYTE	for testing
IO_Valid_SCM	I_IO_Valid	%IB230	BYTE	Flag if IO values are valid (valid = 16#AA, invalid = 16#55)

Figure 7-29: IO_Valid

	WARNING
	Input information within the application programs (Safe Application and Standard Application) are only valid if the corresponding *IO_Valid* flags have the value "TRUE" (dec. 170). Otherwise, the data of the corresponding board inputs must not be used in the application programs. The exception is the life counter (I_Lifecounter / Q_Lifecounter), which must be operated immediately after the user program has been started.

	WARNING
	When using the outputs make sure that the flags *I_VIQ_Valid* and *I_VIQ_FB* are evaluated in the application program.

7.6.2.4 I_VIQ_Valid

Check the VIQ flags before energizing the outputs (see Chapter 4.4).

Structure of the VIQ flags:

Name	Bit 7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
VIQ_Valid	-	-	-	-	VIQ4	VIQ3	VIQ2	VIQ1

The VIQ_Valid flags become TRUE if the outputs are not configured as "disabled" and the corresponding VIQ voltage is present.

7.6.2.5 I_VIQ_FB

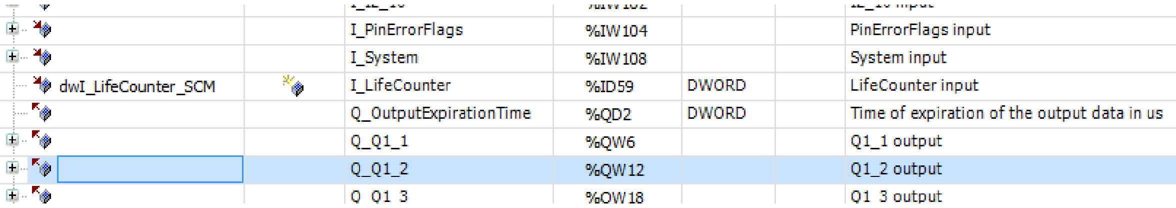
These flags show if there is an external power supply connected to the VIQ pins.

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I_VIQ_FB	-	-	-	-	VIQ4	VIQ3	VIQ2	VIQ1

7.6.2.6 I_LifeCounter / Q_Lifecounter

The LifeCounter must be transmitted within the application program. This ensures that the update of the input/outputs functions correctly. If the LifeCounter is not transmitted in the application program, the control system switches to the OPMODE_FAILSAFE_STOP operating mode.

To transmit the output LifeCounter to the input LifeCounter, the variable must be defined in CODESYS (see Figure).



		I_PinErrorFlags	%IW104		PinErrorFlags input
		I_System	%IW108		System input
dwI_LifeCounter_SCM		I_LifeCounter	%ID59	DWORD	LifeCounter input
		Q_OutputExpirationTime	%QD2	DWORD	Time of expiration of the output data in us
		Q_Q1_1	%QW6		Q1_1 output
		Q_Q1_2	%QW12		Q1_2 output
		Q_Q1_3	%QW18		Q1_3 output

Figure 7-30: LifeCounter

Example to set the life counter of a SCMboard:

dwQ_LifeCounter_SCM := dwI_LifeCounter_SCM;

These steps must be carried out for every I/O expansion board. When using one SCM and two SIOMs, 3 input and 3 output variables must be defined and assigned (Figure 7-31).

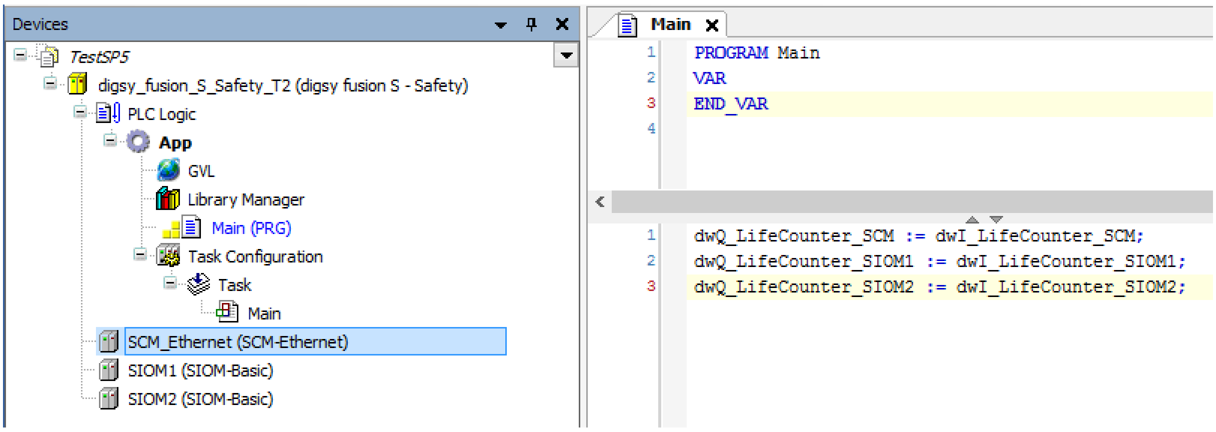


Figure 7-31: Example of LifeCounters

7.6.2.7 I_PinErrorFlags

PinErrorFlags set to "1" indicate an error on the inputs/outputs. The PinErrorFlags are subdivided into the groups:

- I_PinErrors_Q1 (type A1 outputs)
- I_PinErrors_Q2 (type A2 outputs)
- I_PinErrors_Q3 (type A3 outputs)
- I_PinErrors_Q4 (type B outputs)
- I_PinErrors_I1 (type D inputs)
- I_PinErrors_I2 (type E inputs)
- I_PinErrors_I3 (type C inputs)

The bytes / words must be interpreted as follows:

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I_PinErrors_Q1					Q1.4	Q1.3	Q1.2	Q1.1
I_PinErrors_Q2					Q2.4	Q2.3	Q2.2	Q2.1
I_PinErrors_Q3					Q3.4	Q3.3	Q3.2	Q3.1
I_PinErrors_Q4			Q4.6	Q4.5	Q4.4	Q4.3	Q4.2	Q4.1
I_PinErrors_I1	I1.8	I1.7	I1.6	I1.5	I1.4	I1.3	I1.2	I1.1
I_PinErrors_I3			I3.6	I3.5	I3.4	I3.3	I3.2	I3.1

Table 7-9: PinErrors Q1, Q2, Q3, Q4, I1, and I3

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I_PinErrors_I2	I2.8	I2.7	I2.6	I2.5	I2.4	I2.3	I2.2	I2.1
Name	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
I_PinErrors_I2	I2.16	I2.15	I2.14	I2.13	I2.12	I2.11	I2.10	I2.9

Table 7-10: PinError I2

7.6.3 Pin configuration from the Standard Application program

Inputs/outputs configured as "disabled" in the Safe Application program can be configured via the Standard Application program. This is possible for both the SCM inputs/outputs and the expansion board inputs/outputs.

To configure the inputs/outputs via the Standard Application program, the devices (SCM, SIOM, …) must first also be integrated into the standard project analogous to the Safe Application program:

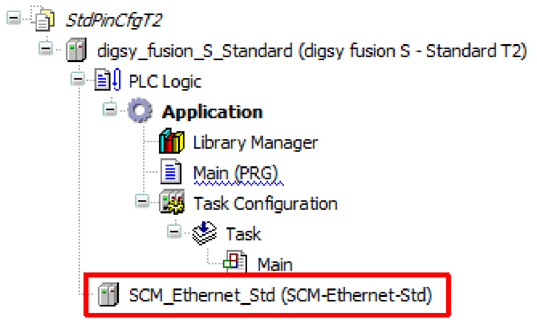
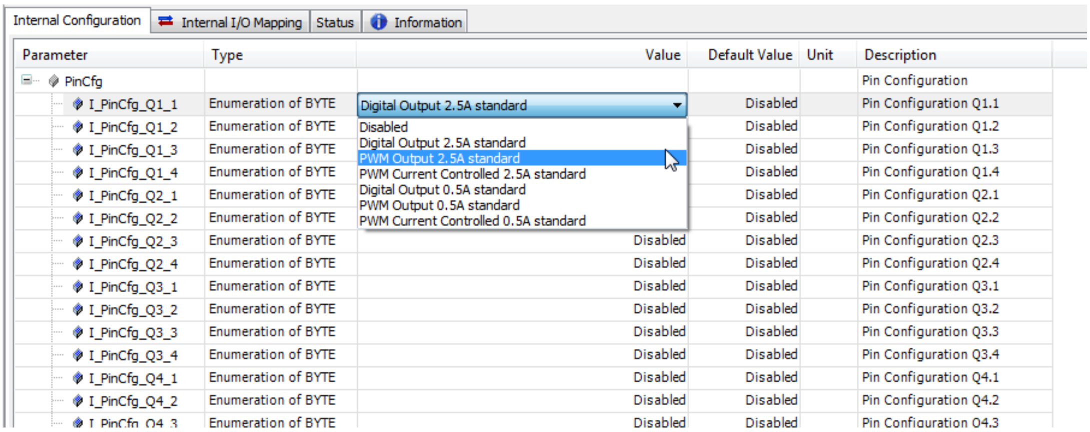


Figure 7-32: Devices in the Standard Application program

The devices intended for the Standard Application program have "-Std" name extension. They cannot be confused with the devices for the Safe Application program, because these devices cannot be integrated into a Standard Application program (and vice-versa).

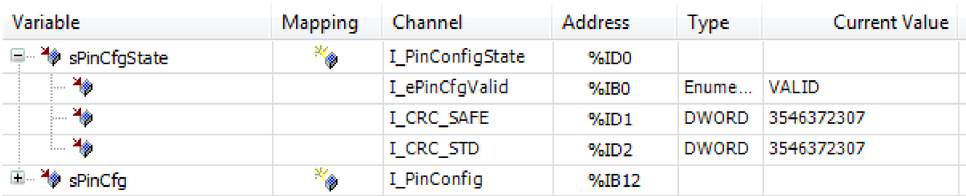
Double-click the device to open its device dialog. In the "Internal Configuration" tab, you can configure the inputs/outputs analogous to procedure applied for the Safe Application program. However, the drop-down list boxes only provide the "standard" configurations:



Parameter	Type	Value	Default Value	Unit	Description
PinCfg					Pin Configuration
I_PinCfg_Q1_1	Enumeration of BYTE	Digital Output 2.5A standard	Disabled		Pin Configuration Q1.1
I_PinCfg_Q1_2	Enumeration of BYTE		Disabled		Pin Configuration Q1.2
I_PinCfg_Q1_3	Enumeration of BYTE		Disabled		Pin Configuration Q1.3
I_PinCfg_Q1_4	Enumeration of BYTE		Disabled		Pin Configuration Q1.4
I_PinCfg_Q2_1	Enumeration of BYTE		Disabled		Pin Configuration Q2.1
I_PinCfg_Q2_2	Enumeration of BYTE		Disabled		Pin Configuration Q2.2
I_PinCfg_Q2_3	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q2.3
I_PinCfg_Q2_4	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q2.4
I_PinCfg_Q3_1	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q3.1
I_PinCfg_Q3_2	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q3.2
I_PinCfg_Q3_3	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q3.3
I_PinCfg_Q3_4	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q3.4
I_PinCfg_Q4_1	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q4.1
I_PinCfg_Q4_2	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q4.2
I_PinCfg_Q4_3	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q4.3

Figure 7-33: Pin configuration in the Standard Application program

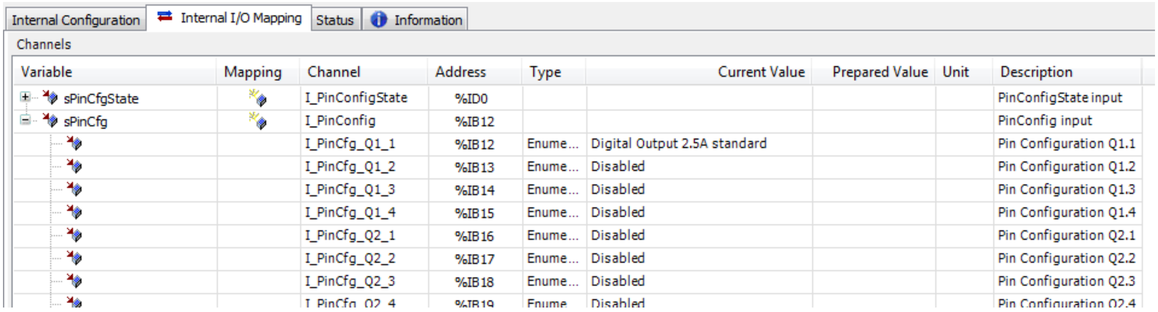
When starting the Safe Application program, the configurations of the inputs/outputs are merged in the safe runtime system. This system first checks whether only inputs/outputs configured as "Disabled" in the Safe Application have been configured in the Standard Application program. Otherwise, the entire configuration of the Standard Application program is rejected. Via the "I_PinConfigState" structure in the "I_ePinCfgValid" field element in the I/O mapping of the Standard Application program (see Figure 7-34), you can check whether the pin configuration has been accepted as valid.



Variable	Mapping	Channel	Address	Type	Current Value
sPinCfgState		I_PinConfigState	%ID0		
		I_ePinCfgValid	%IB0	Enume...	VALID
		I_CRC_SAFE	%ID1	DWORD	3546372307
		I_CRC_STD	%ID2	DWORD	3546372307
sPinCfg		I_PinConfig	%IB12		

Figure 7-34: I/O mapping of the Standard Application program

The merged pin configuration can also be found in the I/O mapping:



Variable	Mapping	Channel	Address	Type	Current Value	Prepared Value	Unit	Description
sPinCfgState		I_PinConfigState	%ID0					PinConfigState input
sPinCfg		I_PinConfig	%IB12					PinConfig input
		I_PinCfg_Q1_1	%IB12	Enume...	Digital Output 2.5A standard			Pin Configuration Q1.1
		I_PinCfg_Q1_2	%IB13	Enume...	Disabled			Pin Configuration Q1.2
		I_PinCfg_Q1_3	%IB14	Enume...	Disabled			Pin Configuration Q1.3
		I_PinCfg_Q1_4	%IB15	Enume...	Disabled			Pin Configuration Q1.4
		I_PinCfg_Q2_1	%IB16	Enume...	Disabled			Pin Configuration Q2.1
		I_PinCfg_Q2_2	%IB17	Enume...	Disabled			Pin Configuration Q2.2
		I_PinCfg_Q2_3	%IB18	Enume...	Disabled			Pin Configuration Q2.3
		I_PinCfg_Q2_4	%IB19	Enume...	Disabled			Pin Configuration Q2.4

Figure 7-35: Merged pin configuration in the I/O mapping

Since the pin configurations are merged upon the start of the Safe Application program, subsequent configuration changes in the Standard Application program can only take effect upon the next start of the Safe Application program. For this reason, the field elements "I_CRC_SAFE" and "I_CRC_STD" are available in the "I_ PinConfigState" structure in the I/O mapping of the Standard Application program (see Figure 7-34) In the "I_CRC_SAFE" field, the safe runtime system stores the CRC check sum of the pin configuration of the Standard Application program which has been used for merging the configurations. The standard runtime system stores the current CRC check sum in the "I_CRC_STD" field. If these two values differ from each other, this means that the pin configuration of the Standard Application program has changed since the last start of the Safe Application program and has not yet been applied by the Safe Application program.

7.6.4 Safety notes

Safety information and safety requirements	
The input/output data for every module (SCM(SL)/SIOM/GIOM) can be cyclically updated at programmable times (depends on the tasks where to I/O mapping is connected) within the application program. In order to ensure that these times are observed, a timeout value for every module (SCM(SL)/SIOM/GIOM) can be programmed via the variable Q_IoTimeInterval [ms].	§6592

Table 7-11: Safety information and safety requirements

7.7 Using the inputs/outputs (SCM(SL)/SIOM/GIOM)

In order to be able to use the *digsy*®fusionS inputs/outputs, these must be integrated as devices. To do this, the corresponding module (SCM(SL),SIOM/GIOM) must be added to the project via "Add Device...".

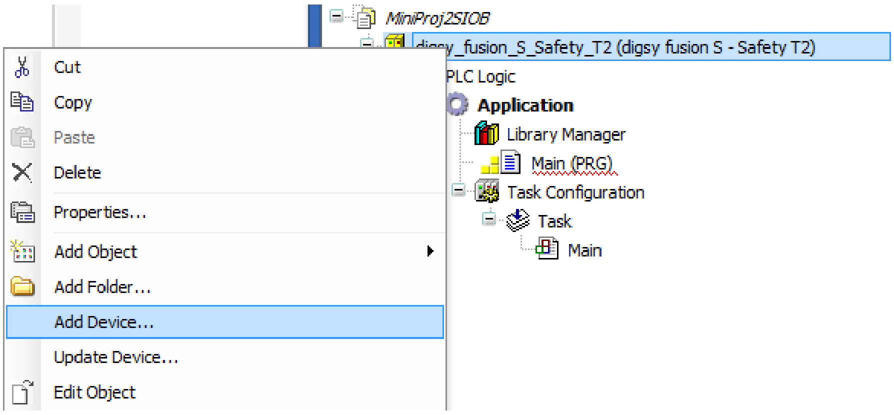


Figure 7-36: Adding a device

Once the I/O devices have been integrated, double-click a device in order to modify its parameters.

Important parameters include the pin configuration, the threshold voltages (for type D inputs) and the I/O time (see Chapter 7.6).

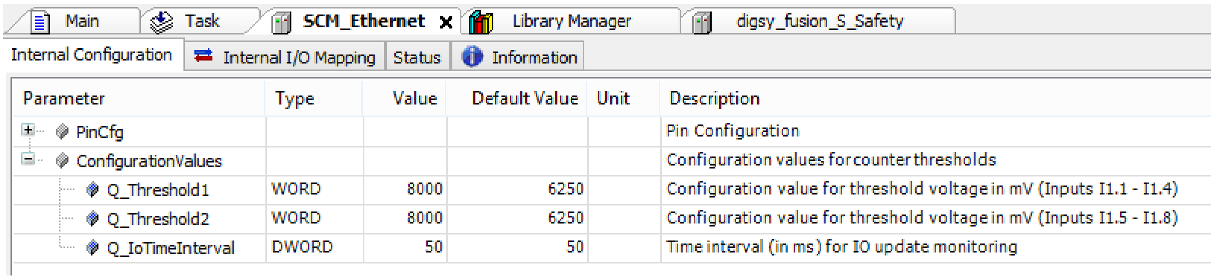


Figure 7-37: I/O parameters

Once the inputs/outputs have been integrated and their parameters have been set, the inputs/outputs are ready for use. The following must be observed:

	CAUTION
	Once a (type A or type B) output of a group has been configured as safe, the other outputs of this group must also be configured as safe (even if these are not used). The "disabled" setting is not permitted in this case. The control system switches to the *OPMODE_FAILSAFE_IO* mode if this rule is not observed.

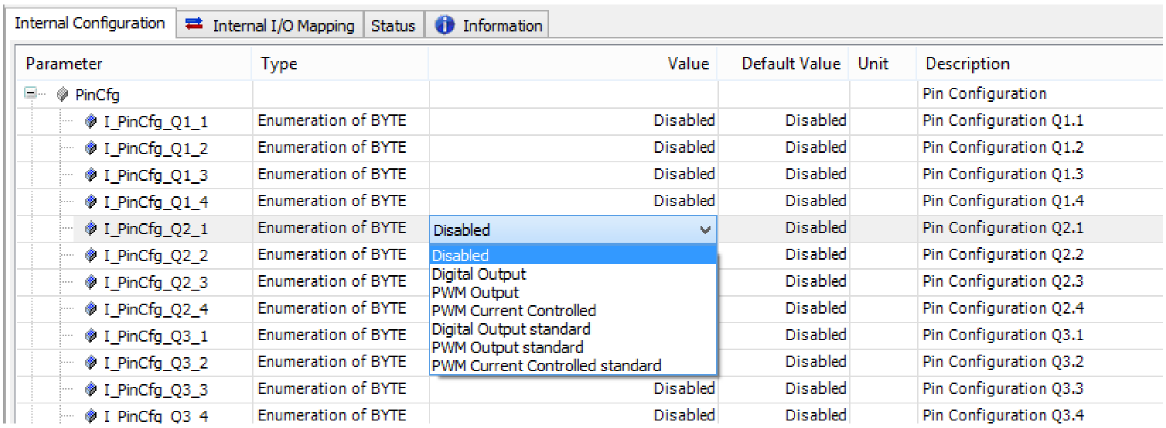
Safety information and safety requirements – Inputs/outputs	
After every task run, the user must transfer an expiration time stamp for every I/O module to the control system via the I/O mapping.	§7287
The CODESYS sets the expiration time stamp to 0 in order to indicate that the value is invalid. This applies until the first valid value (<> 0) has been transmitted.	§7288
The user may only issue an expiration time stamp after IO_Valid = TRUE (dec. 170) has been received from the corresponding I/O module.	§7289
If the cycle time monitoring Q_IoTimeInterval >= 10min, the control system switches to the OPMODE_FAILSAFE_IO mode.	§7310
If the application has received a new expiration time stamp before the current expiration time stamp has expired, and if the I/O mapping can even be issued before the expiration of the current expiration time stamp, the control system switches every output according to the I/O mapping. This ensures that the new output data is issued not later than by the expiration of the expiration time stamp and that this new output data applies the expiration time stamp of the I/O mapping as the new expiration time stamp.	§7295
If the application has not received a new expiration time stamp before the current expiration time stamp has expired, the control system proceeds as follows: All safe outputs are de-energized All error flags are set for the safe outputs of the module Switchover to the OPMODE_FAILSAFE_IO mode This ensures that new output data is issued before the expiration time stamp has expired.	§7296

Table 7-12: Safety information and safety requirements - Inputs / outputs

7.7.1 Type A outputs

7.7.1.1 Configuration via CODESYS

Type A outputs A (Q1.1 – Q3.4) can be configured in the "Internal Configuration" tab of every I/O module (SCM(SL),SIOM,GIOM) used. One configuration per input (see Figure 7-38) can be used.



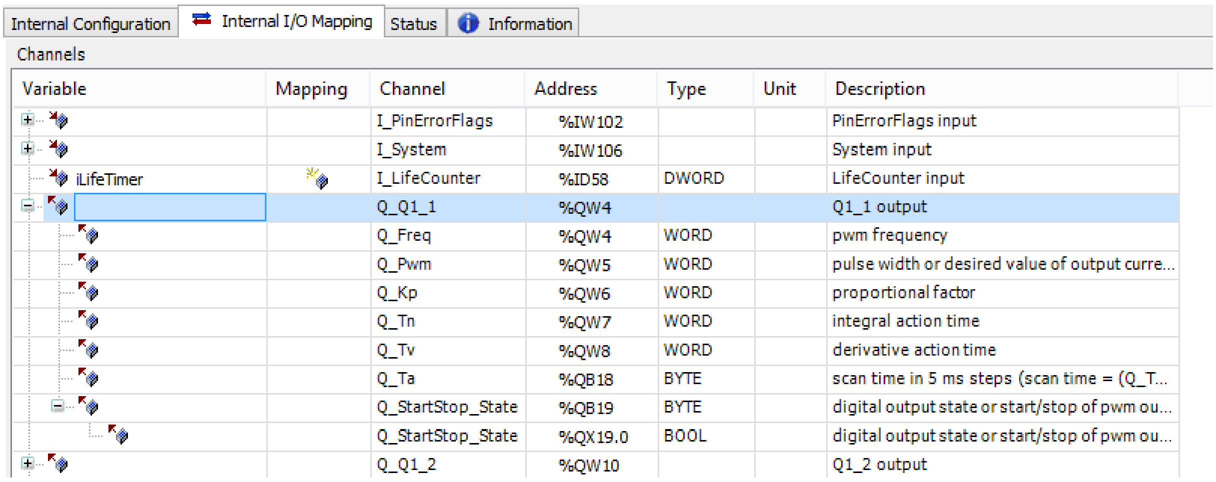
Parameter	Type	Value	Default Value	Unit	Description
PinCfg					Pin Configuration
I_PinCfg_Q1_1	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q1.1
I_PinCfg_Q1_2	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q1.2
I_PinCfg_Q1_3	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q1.3
I_PinCfg_Q1_4	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q1.4
I_PinCfg_Q2_1	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q2.1
I_PinCfg_Q2_2	Enumeration of BYTE		Disabled		Pin Configuration Q2.2
I_PinCfg_Q2_3	Enumeration of BYTE		Disabled		Pin Configuration Q2.3
I_PinCfg_Q2_4	Enumeration of BYTE		Disabled		Pin Configuration Q2.4
I_PinCfg_Q3_1	Enumeration of BYTE		Disabled		Pin Configuration Q3.1
I_PinCfg_Q3_2	Enumeration of BYTE		Disabled		Pin Configuration Q3.2
I_PinCfg_Q3_3	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q3.3
I_PinCfg_Q3_4	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q3.4

Figure 7-38: Configuration of type A outputs

7.7.1.2 Use via CODESYS

Once the configuration is complete, the inputs/outputs can be used via a structure (see Figure 7-39 and Table 7-13). For type A1 inputs/outputs the frequency is included in this structure. This means that a specific PWM frequency can be set for each type A1 output. Only one PWM frequency per type can be set for type A2 and A3 outputs (see Figure 7-40).

Type A output structure (Q_Qx_x)



Variable	Mapping	Channel	Address	Type	Unit	Description
		I_PinErrorFlags	%IW102			PinErrorFlags input
		I_System	%IW106			System input
iLifeTimer		I_LifeCounter	%ID58	DWORD		LifeCounter input
		Q_Q1_1	%QW4			Q1_1 output
		Q_Freq	%QW4	WORD		pwm frequency
		Q_Pwm	%QW5	WORD		pulse width or desired value of output curre...
		Q_Kp	%QW6	WORD		proportional factor
		Q_Tn	%QW7	WORD		integral action time
		Q_Tv	%QW8	WORD		derivative action time
		Q_Ta	%QB18	BYTE		scan time in 5 ms steps (scan time = (Q_T...
		Q_StartStop_State	%QB19	BYTE		digital output state or start/stop of pwm ou...
		Q_StartStop_State	%QX19.0	BOOL		digital output state or start/stop of pwm ou...
		Q_Q1_2	%QW10			Q1_2 output

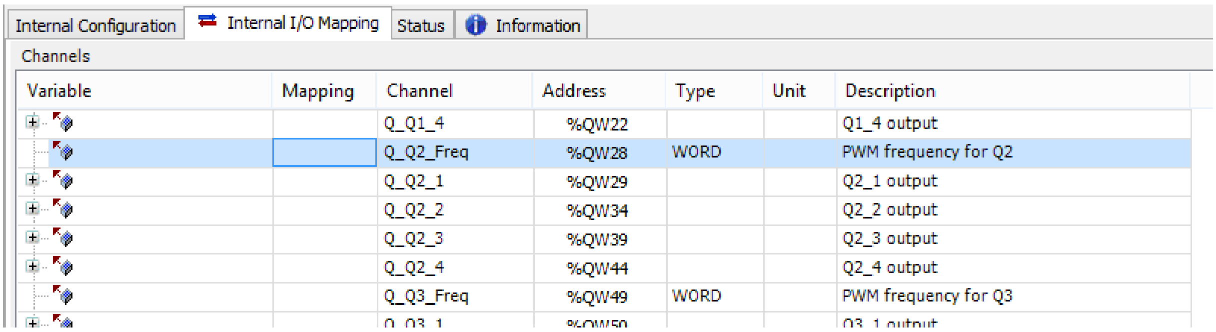
Figure 7-39: Type A output structure (Q_Qx_x)

Configuration	Variable	Description
Digital output	Q_StartStop_State	State of the digital output (TRUE/FALSE)
PWM output	Q_Freq	PWM frequency 0 – 1kHz Standard Application 50Hz – 1kHz Safe Application

Configuration	Variable	Description
	Q_Pwm	PWM pulse width 0 – 4095 units
	Q_StartStop_State	Start/Stop of the PWM output [TRUE/FALSE]
PWM output (controlled)	Q_Freq	PWM frequency [Hz]
	Q_Pwm	Current value [mA]
	Q_Kp	Proportionality factor
	Q_Tn	Reset time
	Q_Tv	Derivative action time
	Q_Ta	Sampling time = (Q_Ta + 1) * 5ms
	Q_StartStop_State	Start/Stop of the PWM output
PID control algorithm (PMW, controlled)		

$$y(n) = y(n-1) + K_P \cdot \left[x_d(n) - x_d(n-1) + \frac{T_{AT}}{T_N} x_d(n) + \frac{T_V}{T_{AT}} \left(x_d(n) - 2x_d(n-1) + x_d(n-2) \right) \right]$$

Table 7-13: Output structure depending on the purpose of use

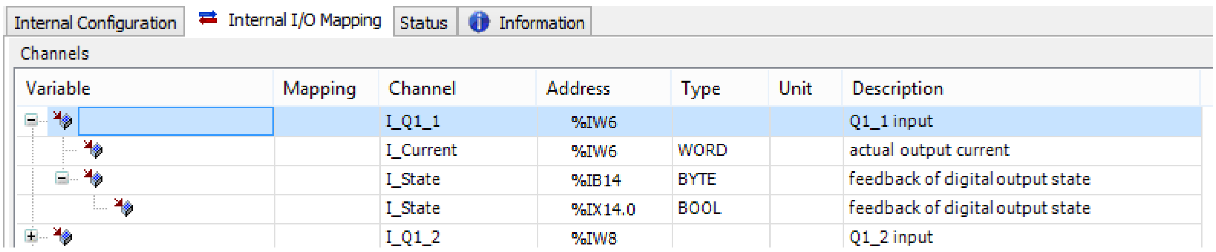

Internal Configuration | Internal I/O Mapping | Status | Information

Channels

Variable	Mapping	Channel	Address	Type	Unit	Description
		Q_Q1_4	%QW22			Q1_4 output
		Q_Q2_Freq	%QW28	WORD		PWM frequency for Q2
		Q_Q2_1	%QW29			Q2_1 output
		Q_Q2_2	%QW34			Q2_2 output
		Q_Q2_3	%QW39			Q2_3 output
		Q_Q2_4	%QW44			Q2_4 output
		Q_Q3_Freq	%QW49	WORD		PWM frequency for Q3

Figure 7-40: Type A2 and A3 frequency setting

Type A input structure (I_Qx_x)


Internal Configuration | Internal I/O Mapping | Status | Information

Channels

Variable	Mapping	Channel	Address	Type	Unit	Description
		I_Q1_1	%IW6			Q1_1 input
		I_Current	%IW6	WORD		actual output current
		I_State	%IB14	BYTE		feedback of digital output state
		I_State	%IX14.0	BOOL		feedback of digital output state
		I_Q1_2	%IW8			Q1_2 input

Figure 7-41: Type A input structure (I_Qx_x)

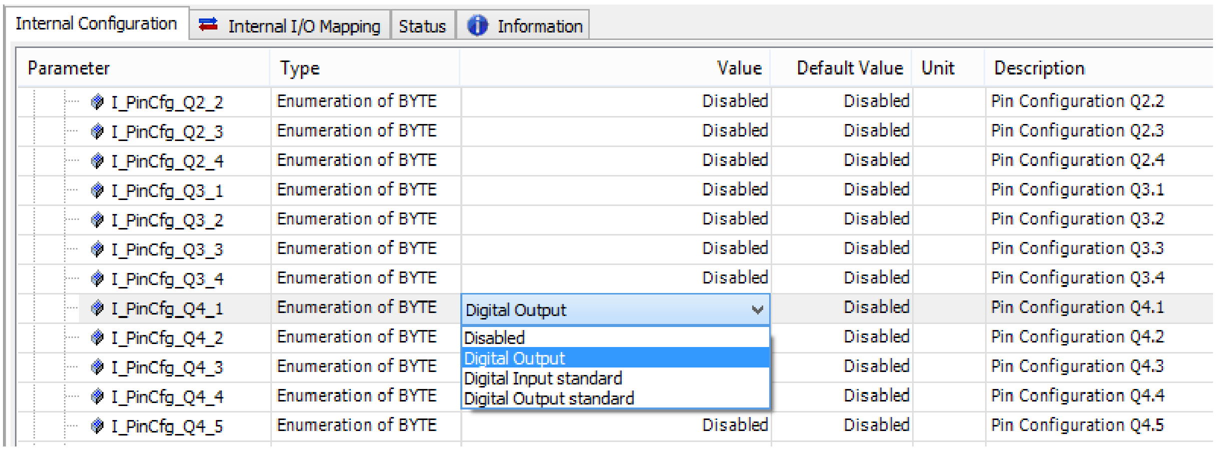
Configuration	Variable	Description
	I_Current	Output current
	I_State	State of the digital output (TRUE/FALSE) **ATTENTION: If the output was parameterized as PWM, the input channel is invalid and must not be used!**

Table 7-14: Type A input structure (I_Qx_x)

7.7.2 Type B inputs/outputs

7.7.2.1 Configuration via CODESYS

The user can configure all type B outputs in the "Internal Configuration" tab. The inputs can be configured as digital inputs, digital outputs, and as safe digital outputs.



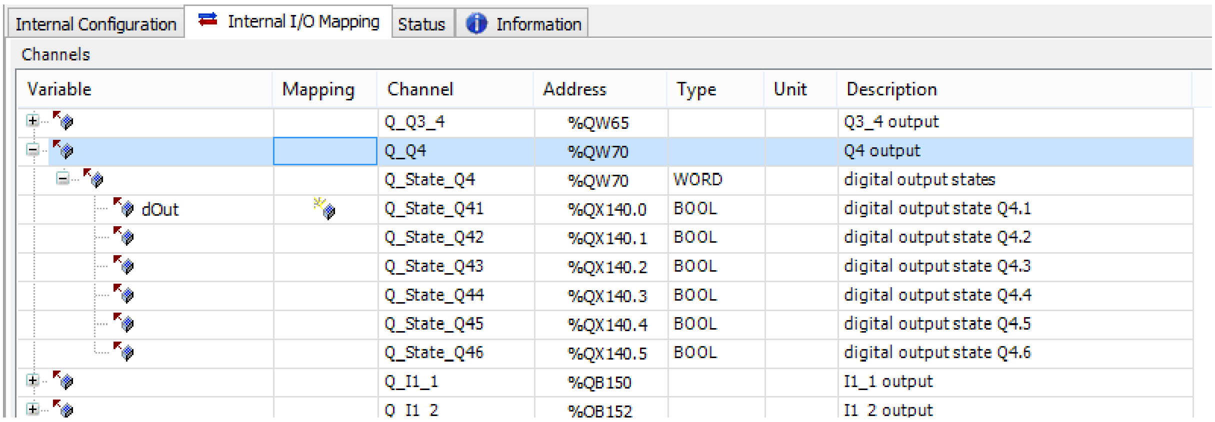
Parameter	Type	Value	Default Value	Unit	Description
I_PinCfg_Q2_2	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q2.2
I_PinCfg_Q2_3	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q2.3
I_PinCfg_Q2_4	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q2.4
I_PinCfg_Q3_1	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q3.1
I_PinCfg_Q3_2	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q3.2
I_PinCfg_Q3_3	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q3.3
I_PinCfg_Q3_4	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q3.4
I_PinCfg_Q4_1	Enumeration of BYTE	Digital Output	Disabled		Pin Configuration Q4.1
I_PinCfg_Q4_2	Enumeration of BYTE		Disabled		Pin Configuration Q4.2
I_PinCfg_Q4_3	Enumeration of BYTE		Disabled		Pin Configuration Q4.3
I_PinCfg_Q4_4	Enumeration of BYTE		Disabled		Pin Configuration Q4.4
I_PinCfg_Q4_5	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q4.5

Figure 7-42: Configuration of type B inputs/outputs

7.7.2.2 Use with CODESYS

The state of the digital outputs is summarized in one byte (Q_Q4). Bits 0 to 5 are available for the six outputs (see Figure 7-43).

Output structure of type B outputs (Q_Q4)



Variable	Mapping	Channel	Address	Type	Unit	Description
		Q_Q3_4	%QW65			Q3_4 output
		Q_Q4	%QW70			Q4 output
		Q_State_Q4	%QW70	WORD		digital output states
dOut		Q_State_Q41	%QX140.0	BOOL		digital output state Q4.1
		Q_State_Q42	%QX140.1	BOOL		digital output state Q4.2
		Q_State_Q43	%QX140.2	BOOL		digital output state Q4.3
		Q_State_Q44	%QX140.3	BOOL		digital output state Q4.4
		Q_State_Q45	%QX140.4	BOOL		digital output state Q4.5
		Q_State_Q46	%QX140.5	BOOL		digital output state Q4.6
		Q_I1_1	%QB150			I1_1 output
		Q_I1_2	%QB152			I1_2 output

Figure 7-43: Output structure of type B outputs (Q_Q4)

Input structure of type B outputs (I_Q4)

The input structure indicates the output current.

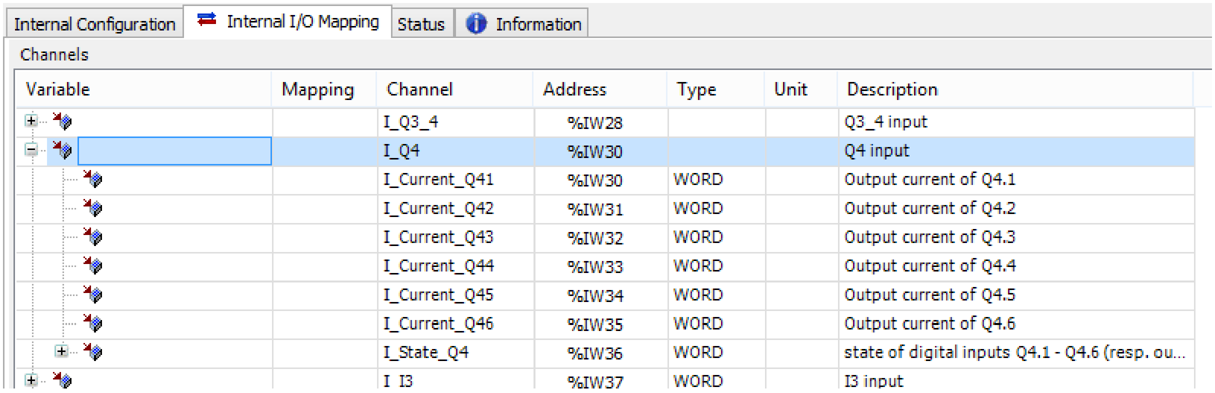


Figure 7-44: Input structure of type B outputs (I_Q4)

7.7.3 Type C inputs

7.7.3.1 Configuration via CODESYS

You can configure the group A inputs via I_PinCfg_I3_A. Group B inputs can be configured via I_PinCfg_I3_B. Type C inputs can be used as digital inputs. A distinction is made between inputs with an internal pull-up resistor (ground switching) and inputs with an internal pull-down resistor (plus switching).

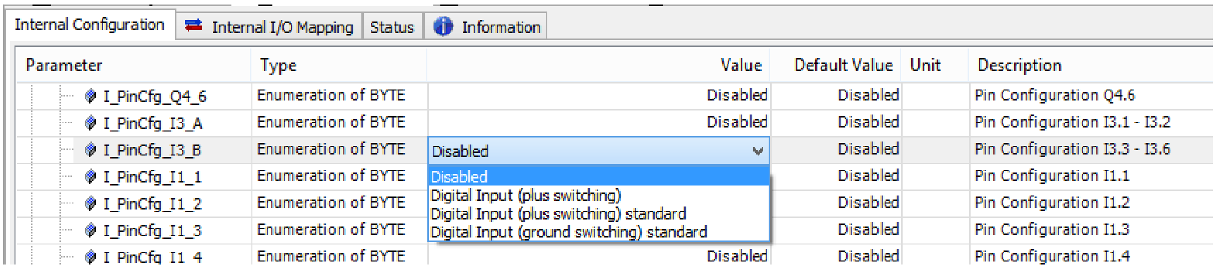


Figure 7-45: Configuration of type C inputs

7.7.3.2 Use with CODESYS

The state of the digital inputs is indicated at "I_I3". Bits 0 to 5 of Byte I_I3 are assigned to one input each.

Input structure of type C inputs (I_I3)

Internal Configuration | Internal I/O Mapping | Status | Information

Channels

Variable	Mapping	Channel	Address	Type	Unit	Description
		I_Q3_3	%IW26			Q3_3 input
		I_Q3_4	%IW28			Q3_4 input
		I_Q4	%IW30			Q4 input
		I_I3	%IW37	WORD		I3 input
		I_State_I31	%IX74.0	BOOL		state of digital input I3.1
		I_State_I32	%IX74.1	BOOL		state of digital input I3.2
		I_State_I33	%IX74.2	BOOL		state of digital input I3.3
		I_State_I34	%IX74.3	BOOL		state of digital input I3.4
		I_State_I35	%IX74.4	BOOL		state of digital input I3.5
		I_State_I36	%IX74.5	BOOL		state of digital input I3.6
		I_I1_1	%ID19			I1_1 input

Figure 7-46: Input structure of type C inputs

7.7.4 Type D inputs

7.7.4.1 Configuration via CODESYS

Type D inputs can be configured in the "Internal Configuration" tab. All type D inputs can be configured as digital inputs, counter inputs, pulse measurement inputs, or encoder inputs (see Figure 7-47). The inputs I1.3, I1.4, I1.7 and I1.8 can also be used for phase measurement.

Internal Configuration | Internal I/O Mapping | Status | Information

Parameter	Type	Value	Default Value	Unit	Description
I_PinCfg_Q4_3	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q4.3
I_PinCfg_Q4_4	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q4.4
I_PinCfg_Q4_5	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q4.5
I_PinCfg_Q4_6	Enumeration of BYTE	Disabled	Disabled		Pin Configuration Q4.6
I_PinCfg_I3_A	Enumeration of BYTE	Disabled	Disabled		Pin Configuration I3.1 - I3.2
I_PinCfg_I3_B	Enumeration of BYTE	Disabled	Disabled		Pin Configuration I3.3 - I3.6
I_PinCfg_I1_1	Enumeration of BYTE	Disabled	Disabled		Pin Configuration I1.1
I_PinCfg_I1_2	Enumeration of BYTE		Disabled		Pin Configuration I1.2
I_PinCfg_I1_3	Enumeration of BYTE		Disabled		Pin Configuration I1.3
I_PinCfg_I1_4	Enumeration of BYTE		Disabled		Pin Configuration I1.4
I_PinCfg_I1_5	Enumeration of BYTE		Disabled		Pin Configuration I1.5
I_PinCfg_I1_6	Enumeration of BYTE		Disabled		Pin Configuration I1.6
I_PinCfg_I1_7	Enumeration of BYTE		Disabled		Pin Configuration I1.7
I_PinCfg_I1_8	Enumeration of BYTE	Disabled	Disabled		Pin Configuration I1.8

Disabled
Digital Input
Counter Input
Pulse Input
Encoder Input
Digital Input standard
Counter Input standard
Pulse Input standard
Encoder Input standard

Figure 7-47: Configuration of type D inputs

The switching thresholds for the inputs I1.1 – I1.4 and I1.5 – I1.8 can be configured via ConfigurationValues (see Figure 7-48).

Internal Configuration | Internal I/O Mapping | Status | Information

Parameter	Type	Value	Default Value	Unit	Description
PinCfg					Pin Configuration
ConfigurationValues					Configuration values for counter thresholds
Q_Threshold1	WORD	8000	6250		Configuration value for threshold voltage in mV (Inputs I1.1 - I1.4)
Q_Threshold2	WORD	8000	6250		Configuration value for threshold voltage in mV (Inputs I1.5 - I1.8)
Q_IoTimeInterval	DWORD	50	50		Time interval (in ms) for IO update monitoring (should be equal to time interval of bus cycle task)

Figure 7-48: Setting Type D switching thresholds

7.7.4.2 Use with CODESYS

The following screenshots illustrate the data structures of the input/output variables of type D inputs. The significance of the individual structure elements basically depends on the configuration of the inputs and is explained in detail in the following paragraphs.

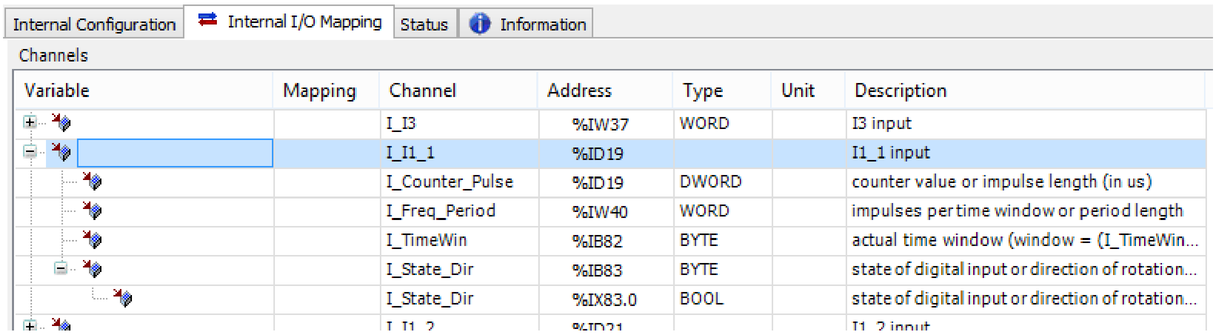


Figure 7-49: I/O structure of type D inputs

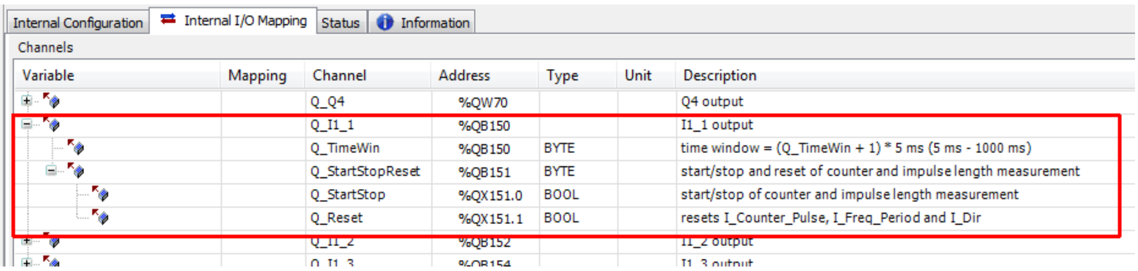


Figure 7-50: I/O structure of type D outputs

7.7.4.3 Use as a digital input

Every type D input can be configured as a digital input (configuration "Digital Input" or "Digital Input standard"). In this case, the input variables have the following functions:

Use	Variable	Description
Digital Input	I_Counter_Pulse	Not applicable
	I_Freq_Period	Not applicable
	I_TimeWin	Not applicable
	I_State_Dir	State of the digital input

If used as a digital input, the output variables have no function:

Use	Variable	Description
Digital Input	Q_TimeWin	Not applicable
	Q_Start_Stop	Not applicable
	Q_Reset	Not applicable

7.7.4.4 Use as a counter input

Every type D input can be configured as a counter input (configuration "Counter Input" or "Counter Input standard"). In this configuration the number of pulses on the input is counted (i.e. the number of low-high edges).

The counting rate is determined in addition to the counter reading. The counting rate is the number of pulses within a time window specified via Q_TimeWin.

The input variables have the following functions:

Use	Variable	Description
Counter Input	I_Counter_Pulse	Counter value
	I_Freq_Period	Counting rate: Number of pulses in the time window (only valid if I_TimeWin is not 0)
	I_TimeWin	Time window of the measured value in I_Freq_Period. The size of the real time window is calculated in milliseconds as follows: Time window (ms) = (I_TimeWin + 1) * 5ms I_TimeWin: 1 – 255: Corresponds to the time window 10ms – 1280ms 0: Time window deactivated or first measuring cycle running. The value in I_Freq_Period is (still) invalid
	I_State_Dir	Not applicable

The output variables have the following functions:

Use	Variable	Description
Counter Input	Q_TimeWin	Time window for counting rate measurement: Time window (ms) = (Q_TimeWin + 1) * 5ms 0: Counting rate measurement deactivated 1 – 255: Corresponds to the time window 10ms – 1280ms
	Q_Start_Stop	Activating/deactivating the counter TRUE: Counter input active FALSE: Counter input deactivated (the input variables retain their previous value)
	Q_Reset	The input variables are reset if a low-high edge is present on Q_Reset

7.7.4.5 Use as pulse measurement input

Every type D input can be configured as a pulse length measurement input (configuration "Pulse Input" or "Pulse Input standard").

The basic sequence applied for pulse length measurement is illustrated in Figure 7-51: Pulse length measurement is triggered by the rising edge (see Marking 1) and terminated and restarted again by the subsequent rising edge (see Marking 3).

The signal period (Marking 1 → 3) and the pulse length (Marking 1 → 2) are captured in µs increments during the measurement. At Marking 3 the measuring result is made available to the application program via the input variables.

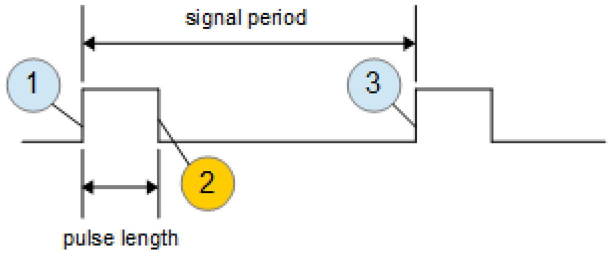


Figure 7-51: Pulse length measurement

The sequence described above corresponds to the pulse length measurement while the time window is deactivated (Q_TimeWin = 255). If the period exceeds its maximum value (65534 µs), the input variable "I_Freq_Period" is set to 16#FFFF (65535 dec). If the maximum pulse length (approx. 1,275 seconds) is exceeded, the input variable "I_Counter_Pulse" is set to 16#FFFFFFFF (4294967295 dec).

To activate the time window monitoring function, set Q_TimeWin to values between 0 and 254 (this corresponds to time window sizes between 5ms and 1275ms). The measuring cycle is not only terminated when Marking 3 is reached, but also if the time window has expired earlier. The input values are set to their maximum values (I_Counter_Pulse = 16#FFFFFFFF and I_Freq_Period = 16#FFFF). This ensures that the application is provided with input values at the defined times.

The input variables have the following functions:

Use	Variable	Description
Pulse Input	I_Counter_Pulse	0 – 1275000 Pulse length in µs 4294967295 Pulse length invalid
	I_Freq_Period	0 – 65534 Period time in µs 65535 Period time invalid
	I_TimeWin	Time window of the measured values (see Q_TimeWin)
	I_State_Dir	Not applicable

The output variables have the following functions:

Use	Variable	Description
Pulse Input	Q_TimeWin	Time window of the pulse length measurement: Time window (ms) = (Q_TimeWin + 1) * 5 ms 0 – 254: Time window monitoring activated 255: No Time window monitoring
	Q_Start_Stop	Activating/deactivating the pulse length measurement TRUE: Pulse length measurement active FALSE: Pulse length measurement deactivated (the input variables retain their previous value)
	Q_Reset	The input variables are reset if a low-high edge is present on Q_Reset and a new measuring cycle is started.

7.7.4.6 Use as an encoder input

Type D inputs can be configured in pairs as encoder inputs (AB counters) – configuration "Encoder Input“ and "Encoder Input standard“. The following inputs form an input pair:

I1.1 (A signal) and I1.2 (B signal)
I1.3 (A signal) and I1.4 (B signal)
I1.5 (A signal) and I1.6 (B signal)
I1.7 (A signal) and I1.8 (B signal)

Please note that both inputs of an input pair must be configured as an "Encoder Input". Only the input and output variables of the A signal are used; the input and output variables of the B signal have no function in this configuration.

Figure 7-52 illustrates the principle of the AB counter. TI1 corresponds to the A signal and TI2 corresponds to the B signal.

The counting rate is determined in addition to the counter reading. For the AB counter the counting rate is indicated as the absolute value of the difference between the counter readings at the start and at the end of the time window.

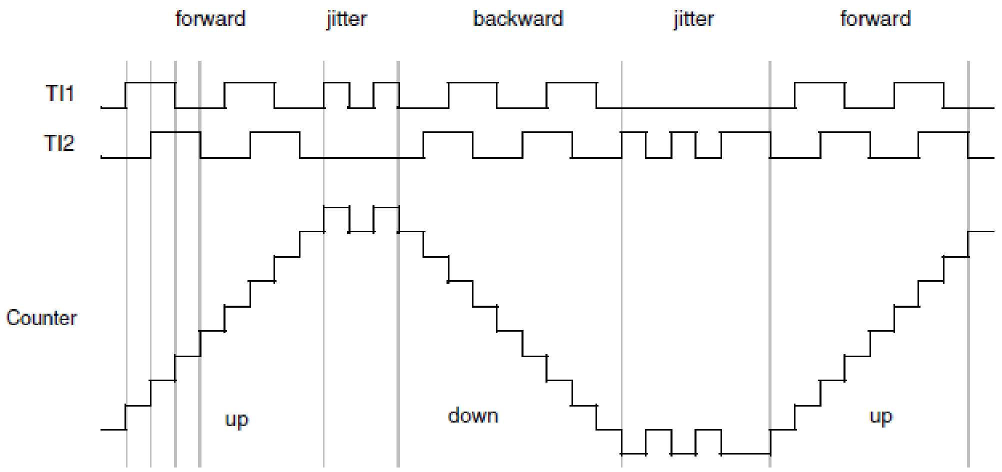


Figure 7-52: Encoder signals

The input variables of the A signal have the following functions:

Use	Variable	Description
Encoder Input	I_Counter_Pulse	Counter value
	I_Freq_Period	Counting rate: Absolute value of the difference between the counter readings at the start and at the end of the time window (only valid if I_TimeWin is NOT 0)
	I_TimeWin	Time window of the measured value in I_Freq_Period. The size of the real time window is calculated in milliseconds as follows: Time window (ms) = (I_TimeWin + 1) * 5ms I_TimeWin: 1 – 255: Corresponds to the time window 10ms – 1280ms 0: Time window deactivated or first measuring cycle running. The value in I_Freq_Period is (still) invalid
	I_State_Dir	Direction of rotation: TRUE: positive (the counter increments) FALSE: negative (the counter decrements)

The output variables of the A signal have the following functions:

Use	Variable	Description
Encoder Input	Q_TimeWin	Time window for counting rate measurement: Time window (ms) = (Q_TimeWin + 1) * 5ms 0: Counting rate measurement deactivated 1 – 255: Corresponds to the time window 10ms – 1280ms
	Q_Start_Stop	Activating/deactivating the counter TRUE: Counter input active FALSE: Counter input deactivated (the input variables retain their previous value)
	Q_Reset	The input variables are reset if a low-high edge is present on Q_Reset

7.7.4.7 Use of the phase measurement input

The inputs I.1.3, I1.4, I1.7 and I1.8 can be configured in pairs as phase measurement inputs (configuration "Phase Measurement standard“). The following inputs form an input pair:

I1.3 (A signal) and I1.4 (B signal)

I1.7 (A signal) and I1.8 (B signal)

Please note that both inputs of an input pair must be configured as phase measurement inputs. Figure 7-53 illustrates the phase measurement:

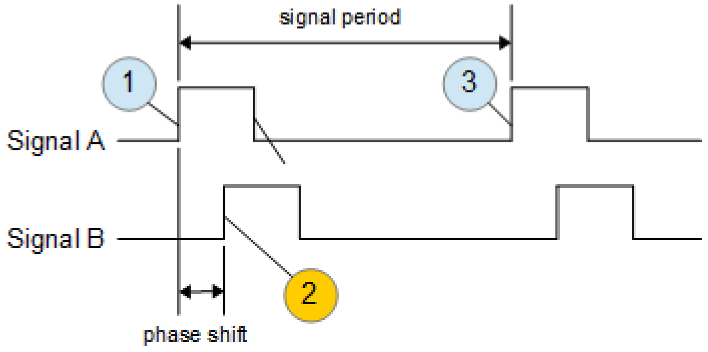


Figure 7-53: Phase measurement

If the phase measurement has started via Q_StartStop = TRUE, the first measuring cycle starts with the rising edge on Channel A (see Marking 1). It is terminated with the following rising edge on Channel A (see Marking 3). (The next measuring cycle starts at the same time, unless the phase measurement is stopped via Q_StartStop = FALSE).

The period of Signal A (Marking 1 → 3) and the time offset between A and B (Marking 1 → 2) in captured in µs increments during the measurement. Once a measuring cycle has been completed, the time offset between A and B is converted into a phase angle based on the period time. The rising edge of the input signals is evaluated. The period of the signal in the input value "I_Counter_Pulse" of Channel A and the phase shift angle are returned to the application program as integer values (in increments of 0.1°) via the input value "I_Freq_Period" of Channel A.

In addition, the elapsed measuring cycle time is captured. This time starts at Marking 1 and is reset to zero as soon as Marking 3 is reached. This value is transmitted to the application program in input "I_Counter_Pulse" of Channel B.

The input variables of the A signal have the following functions:

Use	Variable	Description
Encoder Input	I_Counter_Pulse	Period (of the A signal) in µs
	I_Freq_Period	Phase shift angle in increments of 0.1° (value range 0 – 3600)
	I_TimeWin	Not applicable
	I_State_Dir	Not applicable

The output variables of the A signal have the following functions:

Use	Variable	Description
Encoder Input	Q_TimeWin	Not applicable
	Q_Start_Stop	Activating/deactivating the phase measurement TRUE: Phase measurement active FALSE: Phase measurement deactivated (the input variables retain their previous value)
	Q_Reset	The input variables are reset if a low-high edge is present on Q_Reset

The input variables of the B signal have the following functions:

Use	Variable	Description
Encoder Input	I_Counter_Pulse	Time elapsed since the start of the measuring cycle in µs
	I_Freq_Period	Not applicable
	I_TimeWin	Not applicable
	I_State_Dir	Not applicable

The output variables of the B signal have the following functions:

Use	Variable	Description
Encoder Input	Q_TimeWin	Not applicable
	Q_Start_Stop	Not applicable
	Q_Reset	Not applicable

7.7.5 Type E inputs

Type E inputs can be used as analog inputs with different voltage ranges or as digital inputs.

7.7.5.1 Configuration via CODESYS

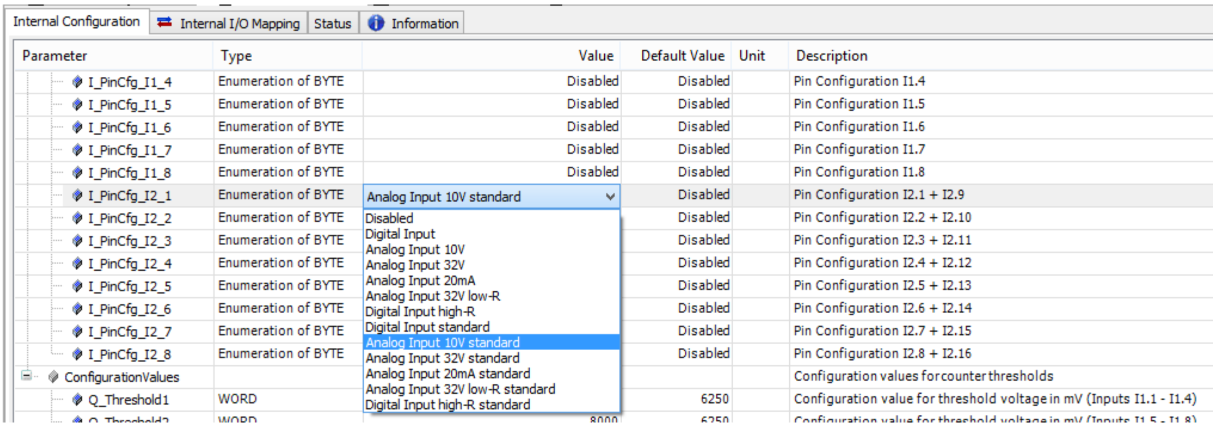
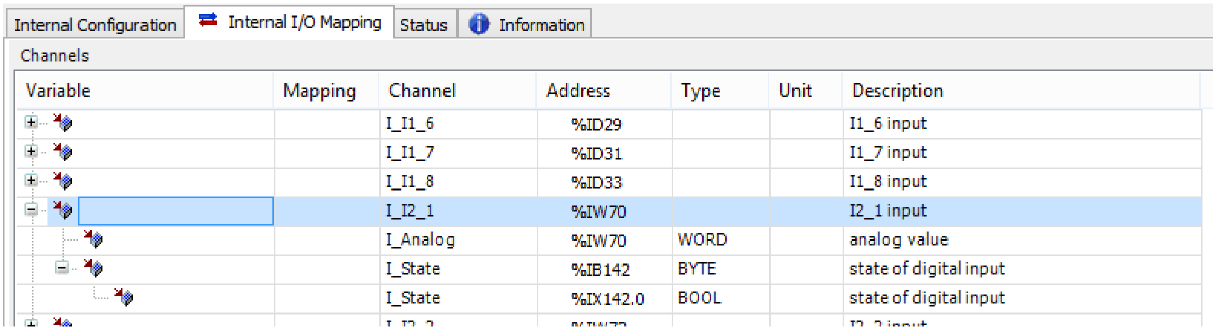


Figure 7-54: Configuration of type E inputs

Use	Description
Digital Input	Digital input
Analog Input 10V	Analog input with a maximum measuring range of 10V
Analog Input 32V	Analog input with a maximum measuring range of 32V
Analog Input 20mA	Analog input with a maximum measuring range of 20mA
Analog Input 32V low-R	Analog input with a maximum measuring range of 32V and input impedance connected (Namur)
Digital Input high-R	Digital input and input impedance disconnected
Digital Input standard	Digital input (standard)
Analog Input 10V standard	Analog input with a maximum measuring range of 10V
Analog Input 32V standard	Analog input with a maximum measuring range of 32V (standard)
Analog Input 20mA standard	Analog input with a maximum measuring range of 20mA (standard)
Analog Input 32V low-R standard	Analog input with a maximum measuring range of 32V and input impedance connected (standard) (Namur)
Digital Input high-R standard	Digital input and input impedance disconnected (standard)

7.7.5.2 Use with CODESYS

Input structure of type D inputs (I_I2_x)



Internal Configuration | Internal I/O Mapping | Status | Information

Channels

Variable	Mapping	Channel	Address	Type	Unit	Description
		I_I1_6	%ID29			I1_6 input
		I_I1_7	%ID31			I1_7 input
		I_I1_8	%ID33			I1_8 input
		I_I2_1	%IW70			I2_1 input
		I_Analog	%IW70	WORD		analog value
		I_State	%IB142	BYTE		state of digital input
		I_State	%IX142.0	BOOL		state of digital input

Figure 7-55: I/O structure of type E inputs

7.8 Error memory and error processing

7.8.1 Error classes

The error classes described in this chapter are used depending on the cause and severity of the error:

- Warnings
- Diagnostic errors
- Serious errors
- Fatal errors

All error messages are entered in the system logbook together with their time stamp.

Depending on the error class, the system must trigger a specific error response and the operating state must be switched if required.

7.8.1.1 Warnings

Warnings are error messages which do not have an impact on the functional safety of the system. A warning can, for example, be issued because a diagnostic error has been detected on an output configured for the Standard Application – i.e. not for the Safe Application.

Warnings are entered in the system logbook and can be evaluated in the application program. Warnings do not have an impact on the operating state of the control system. If, however, the self-diagnostic function detects an error on an I/O configured for the Standard Application, the corresponding error flags are set.

7.8.1.2 Diagnostic errors

Diagnostic errors are error messages which do not compromise the basic functional capability of the control system, and if the detected error can be located. The system generates a diagnostic error if the real-time diagnostic function has detected a malfunction on a safety-relevant input/output. Apart from this error, the overall system can be still be operated in the safe state, and the safety function can still be ensured.

In this case, individual inputs/outputs, respectively input/output groups, for which errors have been diagnosed can be transferred to the safe state. If a diagnostic error has been detected, the control system switches to the OPMODE_FAILURE operating mode.

The error flags of the corresponding inputs/outputs are set. A faulty, safety-relevant output is de-energized immediately; a faulty, safety-relevant input immediately provides a safe value for the application programs in the I/O mapping (e.g. FALSE for a digital input).

The OPMODE_FAILURE operating mode allows further processing for both the Safe Application and the Standard Application. Depending on further environmental conditions within the machine, the user thus has the option to continue working under certain restrictions, or to initiate a program-controlled shutdown of the machine. In the Safe Application, the user can decide whether and when the system is switched to the safe operating mode OPMODE_FAILSAFE_IO or OPMODE_FAILSAFE_STOP.

If an error is diagnosed on an expansion module, this module switches to the OPMODE_FAILURE operating state. Once the SCM has been notified of this operating mode by means of an error message, the entire system is switched to the OPMODE_FAILURE operating mode. This information, including the error message, is made available to the Safe Application.

If an error is diagnosed on an input or output which has not been configured for the Safe Application, but for the Standard Application, the control system does not switch to the OPMODE_FAILURE mode. Only the error flags are set and a warning is entered in the system logbook. In the Standard Application the user can now decide on how to proceed.

7.8.1.3 Serious errors

A serious error is an error which compromises the safety function, even though the central part of the control system continues to operate properly. For example, an incorrect or contradictory pin configuration performed via the application program can cause a serious error.

The location of the serious error is another major factor to be considered.

7.8.1.4 Serious error on a SIOM expansion module

If a serious error is detected on a SIOM expansion module, this module switches to the OPMODE_FAILSAFE_IO operating mode.

The SIOM expansion module is thus switched to the safe state. This means:

- all outputs of the SIOM expansion module are explicitly de-energized,
- all second shutdown paths of the SIOM expansion module are de-energized,
- safe input values towards the SCM are mapped for all inputs of the SIOM,
- the SIOM transmits an error message to the SCM,
- all possible error flags of the SIOM inputs/outputs are set.

The SCM maps the input/output data of the SIOM to the Safe Application and switches to the OPMODE_FAILURE mode. This is possible because the remaining system can still be operated in the safe state. The processing of the application programs is not stopped. In the safe application, the user can decide whether and when the system switches to the safe OPMODE_FAILSAFE_IO or OPMODE_FAILSAFE_STOP mode.

Since the central part of the controller continues to operate, the control system can still be accessed via the diagnostic interfaces. Serious errors can be analysed by means of the CODESYS programming tool.

7.8.1.5 Serious error on an SCM

If a serious error is detected on an SCM, the SCM switches to the OPMODE_FAILSAFE_IO operating mode. This means:

- All outputs of the SCM and of any existing SIOMs/GIOMs are explicitly de-energized,
- all second shutdown paths on the SCM and on any existing SIOMs/GIOMs are de-energized,
- safe input values are mapped to the SCM for all inputs of the SCM and any existing SIOM/GIOM,
- all possible error flags of the SCM inputs/outputs and any existing SIOM/GIOM are set.

Since the central part of the controller continues to operate, the control system can still be accessed via the diagnostic interfaces. Serious errors can be analysed using the CODESYS programming tool.

7.8.1.6 Serious error on a GIOM expansion module

If a serious error is detected on a GIOM expansion module, this module switches to the OPMODE_FAILSAFE_IO operating mode.

The GIOM expansion module is thus switched to the safe state. This means:

- all outputs of the GIOM expansion module are explicitly de-energized,
- all second shutdown paths of the GIOM expansion module are de-energized,
- safe input values towards the SCM are mapped for all inputs of the GIOM,
- the GIOM transmits an error message to the SCM,
- all possible error flags of the GIOM inputs/outputs are set.

The SCM maps the input/output data of the GIOM to the Safe Application and switches to the OPMODE_FAILURE mode. This is possible because the remaining system can still be operated in the safe state. The processing of the application programs is not stopped. In the safe application, the user can decide whether and when the system switches to the safe OPMODE_FAILSAFE_IO or OPMODE_FAILSAFE_STOP mode.

Since the central part of the controller continues to operate, the control system can still be accessed via the diagnostic interfaces. Serious errors can be analysed by means of the CODESYS programming tool.

7.8.1.7 Fatal errors

Fatal errors compromise the safety function because the central part of the control system no longer operates properly. For example, a fatal error is diagnosed if an error is detected during a CPU self-test.

This means that a module of the control system is in an undefined, unsafe state. For this reason, it must be attempted to transfer the corresponding module, or even the entire control system, to the safe state within the shortest possible time.

The monitoring mechanisms of every module ensure that fatal errors are detected and a module reset is initiated.

The location of the fatal error is another major factor to be considered.

7.8.1.8 Fatal error on a SIOM expansion module

If a fatal error is detected on a SIOM expansion module, this module switches to the OPMODE_FAILSAFE_STOP mode.

The SIOM expansion module is thus switched to the safe state. This means:

- all outputs of the SIOM expansion module are explicitly de-energized
- all second shutdown paths of the SIOM expansion module are de-energized
- the communication with the SCM is stopped

The SCM detects the failure of the SIOM and sets all possible input/output error flags of the SIOM concerned for the Safe Application.

The SCM maps safe input/output data of the SIOM concerned to the Safe Application and switches to the OPMODE_FAILURE mode. This is possible because the remaining system can still be operated in the safe state. The processing of the application programs is not stopped. In the safe application, the user can decide whether and when the system switches to the safe OPMODE_FAILSAFE_IO or OPMODE_FAILSAFE_STOP mode.

Since the central part of the control system continues to operate, the control system can still be accessed via the diagnostic interfaces.

The fatal error type is entered in the system logbook.

7.8.1.9 Fatal error on an SCM

If a fatal error is detected on an SCM, the entire control system enters the OPMODE_FAILSAFE_STOP operating mode. This means:

- all outputs of the SCM and of any existing SIOM/GIOM are explicitly de-energized
- all second shutdown paths on the SCM and on any existing SIOM/GIOM are de-energized
- the SCM and all existing SIOMs/GIOMs do not operate

Neither the SCM nor any existing SIOM/GIOM can be accessed via the diagnostic interfaces. Fatal errors <u>cannot</u> be analysed using the CODESYS programming tool.

If possible, the fatal error type is entered in the system logbook.

7.8.1.10 Fatal error on a GIOM expansion module

If a fatal error is detected on a GIOM expansion module, this module switches to the OPMODE_FAILSAFE_STOP mode.

The GIOM expansion module is thus switched to the safe state. This means:

- all outputs of the GIOM expansion module are explicitly de-energized
- all second shutdown paths of the GIOM expansion module are de-energized
- the communication with the SCM is stopped

The SCM detects the failure of the GIOM and sets all possible input/output error flags of the GIOM concerned for the Safe Application.

The SCM maps safe input/output data of the GIOM concerned to the Safe Application and switches to the OPMODE_FAILURE mode. This is possible because the remaining system can still be operated in the safe state. The processing of the application programs is not stopped. In the safe application, the user can decide whether and when the system switches to the safe OPMODE_FAILSAFE_IO or OPMODE_FAILSAFE_STOP mode.

Since the central part of the control system continues to operate, the control system can still be accessed via the diagnostic interfaces.

The fatal error type is entered in the system logbook.

7.8.2 System logbook

The system logbook has an available memory area of 14kB in the FRAM which is implemented as a circular buffer. System errors or warnings (see 10.16) are entered in the system logbook. These errors or warnings can originate from the low-level drivers, from the high level drivers, or from the CODESYS . Every logbook entry is written by the low level drivers and includes a time stamp, a CRC and a Tx flag (indicating that the CODESYS logbook entry has been transmitted). Warnings and errors must be properly considered in order to execute the correct troubleshooting action.

Description	Value
size	14356 Byte
size Header	28 Byte
size per entry	12 Byte
Max number of entries	1194

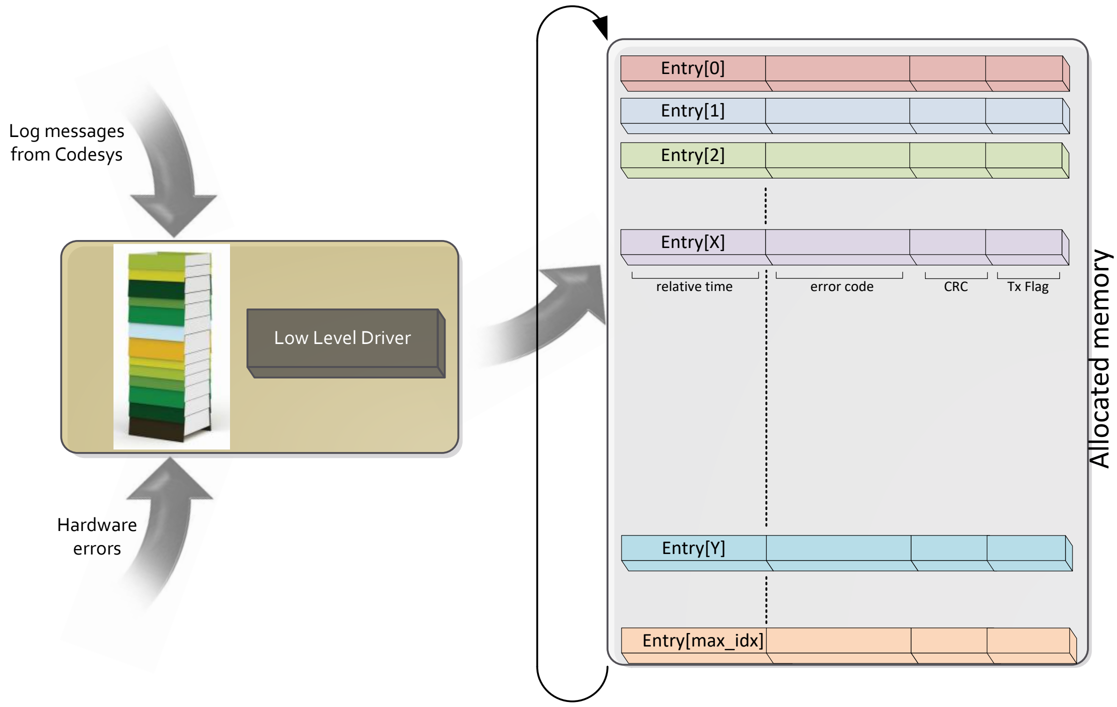


Figure 7-56: Architecture of a system logbook

7.8.3 User logbook

The user logbook consists of a 64 MByte circular buffer in the serial flash. The user can create and save certain logbook entries via CODESYS (see Chapter 8). The logbook entries are provided with a time stamp.

Each entry has a size of 512 Byte, of which 506 Byte are user data, 6 Byte are used for the time stamp and the CRC16 checksum.

The structure of the user logbook entries is user-defined. To ensure compatibility, the size and version is stored in the FRAM.

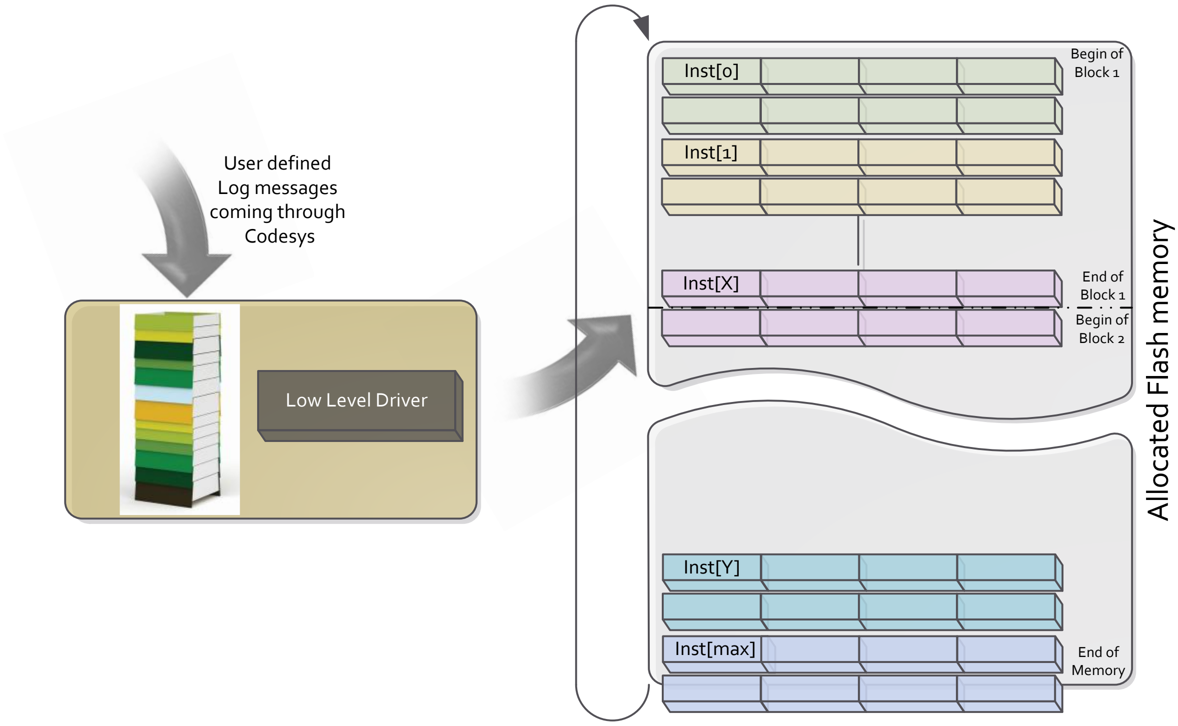


Figure 7-57: Architecture of the user logbook

8 CODESYS Libraries

8.1 DFS_Safe

The "DFS_Safe" CODESYS library provides functions and function blocks for use in the Safe Application program of ***digsy***®fusion S.

The functions (no distinction is made between "functions" and "function blocks" in this chapter) are classified as either "safe" or "non-reactive". Functions marked as "safe" provide safe information or safe functionalities.

"Non-reactive" functions may be used in the Safe Application program, because it is ensured that they do not compromise safe data even though they cannot provide safe function results. To use these functions for safety-relevant tasks, appropriate additional measures must be taken in the application program.

8.1.1 General

8.1.1.1 Interface of function blocks

The interface of the function blocks in the "DFS_Safe" library contains some general input and output parameters which exist in every FB and show the same behavior in every FB. These parameters are explained in this chapter. If a FB differs from this standard behavior, this is explicitly mentioned in the corresponding chapter.

General function inputs:

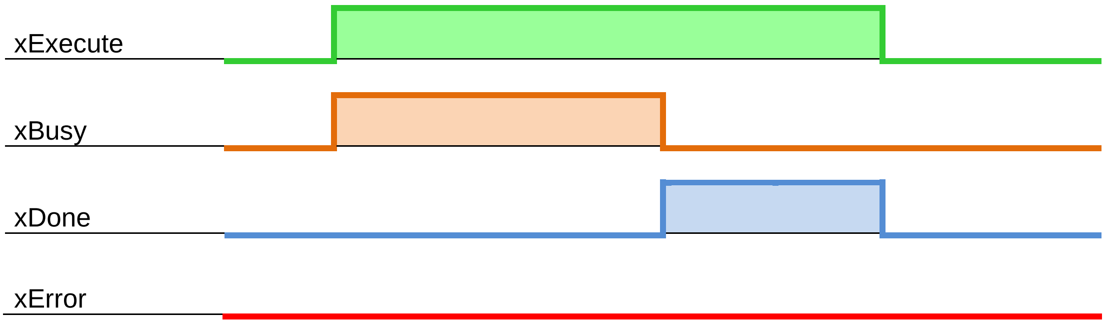
Name	Data type	Description
xExecute	BOOL	Rising edge: The FB starts its action with the rising edge. Falling edge: The outputs of the FB are reset with the falling edge if the FB has already completed its action. If the action is not yet completed, it is now completed first before the outputs are reset.

General function outputs:

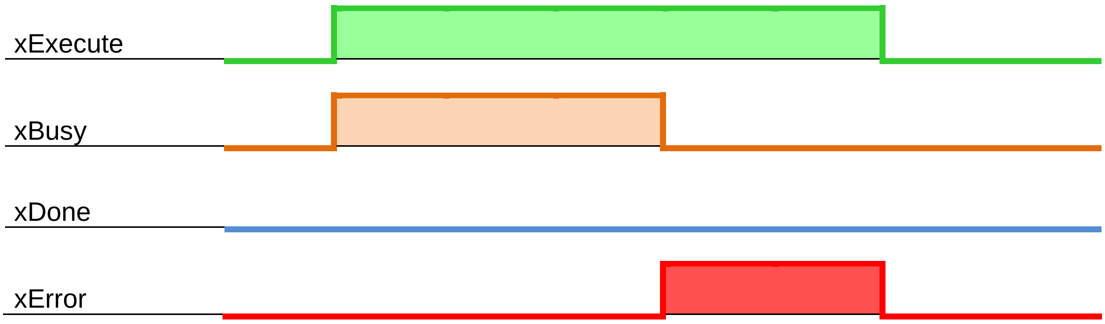
Name	Data type	Description
xBusy	BOOL	xBusy is "TRUE" for as long as the FB is processing its action.
xDone, xError	BOOL	If xDone or xError is "TRUE“, the action is complete. xDone indicates error-free execution. xError indicates that an error has occurred during the action. xDone and xError can never be "TRUE“ at the same time.
eError	ERROR	If an action was completed with an error (xError is TRUE), eError indicates the cause of the error. Otherwise, eError provides the value "NO_ERROR".

The diagrams below illustrate the functioning of the function blocks:

Processing without error:



Processing with error:



8.1.1.2 Error values

The "DFS_Safe“ CODESYS library indicates the cause of the error via the enumeration "ERROR“. The numerical values in the table are only for information and should not be used in application programs. Make sure you always use the symbolic enumeration values in applications programs.

ERROR (ENUM)	Value	Description
NO_ERROR	0	No error
FIRST_ERROR	5100	First library-specific error
TIME_OUT	5101	Time limit exceeded
ABORT	5102	"xAbort“ input activated to abort the task
HANDLE_INVALID	5103	Invalid handle
NOT_EXIST	5104	Directory or file does not exist
EXIST	5105	Directory or file exists
NO_MORE_ENTRIES	5106	No more entries available
NOT_EMPTY	5107	File or directory not empty
READ_ONLY_CAA	5108	File or directory read-only
FILE_WRONG_PARAMETER	5109	Incorrect parameter
ERROR_UNKNOWN	5110	Unknown error
WRITE_INCOMPLETE	5111	Data not completely written
FIRST_MF	5150	First manufacturer-specific error
INTERNAL_RESOURCE	5151	Internal resources not received
GENERAL_SOCKET_ERROR	5152	General socket error

ERROR (ENUM)	Value	Description
BUSY	5153	Function being processed
BUFFER_EMPTY	5154	Buffer is empty
CANLISTEN_NOTCONFIGURED	5155	CAN listen buffer not configured
INVALID_CANCHANNEL	5156	CAN channel number invalid
BUFFER_OVERFLOW	5157	Buffer overflow
COULDNT_OPEN_FILE	5158	File could not be opened
WRITE_FAILED	5159	File writing failed
READ_FAILED	5160	File reading failed
INVALIDDRIVE	5161	Invalid drive
NOTFORMATTED	5162	Drive not formatted
INVALIDDIR	5163	Directory invalid
INVALIDNAME	5164	Invalid file name
DUPLICATED	5165	File or directory already exists
NOTOPEN	5166	Access not possible because the file was not opened
EOF	5167	File end reached
NOTUSEABLE	5168	Invalid parameter
LOCKED	5169	The file has already been opened for Write or Append
ACCESSDENIED	5170	Access not possible
NOTEMPTY	5171	Rename/delete directory failed because not empty
INITFUNC	5172	Initialization failed
CARDREMOVED	5173	Drive removed
ONDRIVE	5174	Non-reparable drive error
INVALIDSECTOR	5175	Invalid sector
ERASE_FAILED	5176	Error during Erase
INVALIDMEDIA	5177	Invalid drive
WRITEPROTECT	5178	Read-only drive
INVFATTYPE	5179	FAT type could not be identified
MEDIATOOSMALL	5180	Drive too small for file system
MEDIATOOLARGE	5181	Drive too large for file system
NOTSUPPSECTORSIZE	5182	Sector size not supported
DRVALREADYMNT	5183	Drive already mounted
TOOLONGNAME	5184	Name too long
DELFUNC	5185	Delete function failed
ALLOCATION	5186	Insufficient memory space
INVALIDPOS	5187	Invalid file position
NOMORETASK	5188	All job tasks used
NOTAVAILABLE	5189	Function not supported by the drive
TASKNOTFOUND	5190	Internal error
UNUSABLE	5191	Drive unusable
FILE_LAST_ERROR	5199	Last library-specific error

8.1.2 Accessing the file system

8.1.2.1 General

The library provides function blocks for various access options to the directories and files of the file system. The standardized input and output variables of the function blocks comply to a large extent with the CAA guidelines set up by the company 3S. The function blocks are explained in detail in the sections below.

Difference to the 3S libraries: No Abort input is available for the function blocks of the IC libraries!

File access is only supported in binary mode. All input variables are stored in local internal variables. This does not apply to data structures. These are handled as input variables via pointers, e.g. for FileRead(FB) or FileWrite(FB).

8.1.2.2 File names and path names

FILENAME is a data type used for file and path names. The names can consist of up to 255 characters.

"/“ must be used as a separator within the path name. The characters to the right of the last separator designate the file name. The characters to the left of the last separator designate the directory path. The directory path must start with the drive designation, separated by a colon.

Example: "B:/root/sub1/sub2/filename.ext“

8.1.2.3 Data types and enumerations

File attributes:

ATTRIB (ENUM)	Value	Description
ARCHIVE	0	Attribute of an archive file
HIDDEN	1	Attribute of a hidden file
NORMAL	2	Attribute of a file for which no other attribute is set
READONLY	3	Attribute of a read-only file

Drive names:

DRIVENAME (ENUM)	Value	Description
A	0	Drive A: External USB stick
B	1	Drive B: Internal Flash Memory

File system formats:

FATTYPE (ENUM)	Value	Description
F_FAT12_MEDIA	1	FAT 12 file system
F_FAT16_MEDIA	2	FAT 16 file system
F_FAT32_MEDIA	3	FAT 32 file system

File access types:

MODE (ENUM)	Value	Description
MWRITE	0	Write access: Overwrites the file or creates a new file
MREAD	1	Read access: Opens the file in read-only mode
MRDWR	2	Read and Write access: Overwrites the file or creates a new file
MAPPD	3	Write access: Opens the file in WRITE mode; data is added at the end of the file

Directory entries:

FILE_DIR_ENTRY (STRUCT)	Variable	Description
sEntry	FILENAME	Name of the directory or file
szSize	SIZE	File size
xDirectory	BOOL	TRUE: - directory FALSE: - file
xExclusive	BOOL	TRUE: - Exclusive access to the file FALSE: - Multiple access to the file is possible
dtLastModification	DT	Date and time of the last modification (not supported)

File names:

FILENAME (TYPE)	Type	Description
FILENAME	STRING(255)	File name incl. path name

8.1.3 Functions for directories

8.1.3.1 DirClose

Classification:

Function block (FB); non-reactive

Function:

Function block for closing a directory

Description:

The entries in a directory can be read via the function blocks "DirListFirst“ and "DirList“. After reading the last entry via "DirList“, the directory must be closed via the "DirClose“ function block in order to ensure that internal firmware resources are freed once again.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs

Name	Data type	Description
hDir	HANDLE	Handle of the directory to be closed

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		!= NO_ERROR	An error has occurred. For an explanation of the error values, refer to section 8.1.1.2.

8.1.3.2 DirCreate

Classification:

Function block (FB); non-reactive

Function:

Function block for creating a directory

Description:

This function block is used to create a subdirectory. An error message pops up if a subdirectory with the desired name already exists. The complete path name must be indicated.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
sDirName	FILENAME	Name of the directory to be created

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		!= NO_ERROR	An error has occurred. For an explanation of the error values, refer to section 8.1.1.2.

8.1.3.3 DirListFirst

Classification:

Function block (FB); non-reactive

Function:

Function block for opening a directory.

Description:

This function block opens a directory with read-only access to the directory entries. The first directory entry is returned via the "deDirEntry“ parameter of the "FILE_DIR_ENTRY“ type. The directory handle is returned via the "hDir“ parameter. With this parameter all other directory entries can be read via the "DirList" function block. DirListFirst (FB) replaces DirOpen (FB).

Files with the SYSTEM attribute are ignored!

If DirListFirst (FB) is called up with the "*.*“ parameters and if the directory is not the root directory, the first entry found is "." – i.e. the current directory – and the second entry found is "..“ – i.e. the higher level directory.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
sDirName	FILENAME	Directory name with filter. Example: "Dir_1/*.*" provides a list of all the files included in the DIR_1 directory

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		!= NO_ERROR	An error has occurred. For an explanation of the error values, refer to section 8.1.1.2.
hDir	HANDLE	Directory handle	
deDirEntry	FILE_DIR_ENTRY	First entry in the directory	

8.1.3.4 DirList

Classification:

Function block (FB); non-reactive

Function:

Function block for reading directory entries

Description:

This function block reads the following directory entry and transmits it to the "deDirEntry“ structure of the "FILE_DIR_ENTRY“ type. DirList (FB) must be called up after the function block DirListFirst (FB) until the last directory entry is displayed. An error message pops up if the function block does not find any further entries. DirClose (FB) must be called up at the end!

Files with the SYSTEM attribute are ignored!

If DirList (FB) is called up with the "*.*“ parameters and if the directory is not the root directory, the first entry found is ".“ – i.e. the current directory – and the second entry found is "..“ – i.e. the higher level directory.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
hDir	HANDLE	Directory handle

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		!= NO_ERROR	An error has occurred. For an explanation of the error values, refer to section 8.1.1.2.
deDirEntry	FILE_DIR_ENTRY	Directory entry read	

8.1.3.5 DirRemove

Classification:

Function block (FB); non-reactive

Function:

Function block for deleting directories

Description:

This function block deletes a directory. The target directory must be empty; otherwise, an error message pops up. An error message also pops up if the target directory has the attribute READONLY.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs

Name	Data type	Description
sDirName	FILENAME	Directory name

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		!= NO_ERROR	An error has occurred. For an explanation of the error values, refer to section 8.1.1.2.

8.1.3.6 DirRename

Classification:

Function block (FB); non-reactive

Function:

Function block for renaming a directory

Description:

This function block renames a directory.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
sDirNameOld	FILENAME	Old directory name, e.g. "A:/olddir"
sDirNameNew	FILENAME	New directory name, e.g. "newdir". No path!

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		!= NO_ERROR	An error has occurred. For an explanation of the error values, refer to section 8.1.1.2.

8.1.4 General functions for the file system

8.1.4.1 DriveFormat

Classification:

Function block (FB); non-reactive

Function:

Function block for formatting a data carrier.

Description:

This function block is used to format a data carrier in the desired format. If a data carrier (e.g. USB stick) is not available, the function block produces an error message. The formatting process destroys all the data on the data carrier. All open files are closed. In addition, the formatting process produces an error message if the desired format is not compatible with the physical data carrier.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
driveName	DRIVENAME	Name of the drive
FATType	FATTYPE	FAT type Relevant for drive A only Not necessary for drive B

Function outputs:

<table>
<tr><th>Name</th><th>Data type</th><th colspan="2">Description</th></tr>
<tr><td>xDone</td><td>BOOL</td><td colspan="2">Processing of the FB completed without error</td></tr>
<tr><td>xBusy</td><td>BOOL</td><td colspan="2">FB is still being processed</td></tr>
<tr><td>xError</td><td>BOOL</td><td colspan="2">Processing of the FB completed with error; error cause see "eError"</td></tr>
<tr><td rowspan="2">eError</td><td rowspan="2">ERROR</td><td>NO_ERROR</td><td>OK</td></tr>
<tr><td>!= NO_ERROR</td><td>An error has occurred. For an explanation of the error values, refer to section 8.1.1.2.</td></tr>
</table>

8.1.4.2 GetProperty

Classification:

Function; non-reactive

Function:

Function for reading out file system properties.

Description:

This function is used to read specific settings and functionalities (properties) of the file system. These refer to the file and the directory system, or to the type of implementation of optional function blocks.

Table of the file system properties:

Number	Constant name (Enumeration "LIBRARY_PROPERTY")	Description	Return value
1	PDIRNAME	Is DirRename (FB) supported?	1 = yes
2	PRENAME	Is FileRename (FB) supported?	1 = yes
3	PCOPY	Is FileCopy (FB) supported?	0 = no
4	PSETATTRIBUTE	Is FileSetAttribute (FB) supported?	1 = yes
100	PFILENAME83	Are only Format 8.3 directory names and file names supported?	0 = no
101	PCAPITALLETTERS	Are only uppercase characters supported for directory names and file names?	0 = no (lowercase characters are permitted; no distinction is made between uppercase and lowercase characters)
200	PABSOLUTEPATH	Are only absolute path names supported?	1 = yes
201	PDRIVELETTER	Are drive identifiers supported?	1 = yes
210	PRESOLVER	Resolver syntax of file names is supported	0 = no
211	PFTPRESOLVER	Are FTP client functions supported?	0 = no
212	PHTTPRESOLVER	Are http client functions supported?	0 = no
300	PNETWORKDRIVES	Are network drives supported?	0 = no

Function inputs:

Name	Data type	Description
wProperty	WORD	Number of the desired property (or element from the "LIBRARY_PROPERTY" enumeration)

Function outputs:

Name	Data type	Description
	DWORD	State of the desired property

8.1.5 Functions for files

8.1.5.1 FileClose

Classification:

Function block (FB); non-reactive

Function:

Function block for closing a file.

Description:

This function block closes a file, i.e. access to this file is terminated. Internal firmware resources are freed once again; the file handle becomes invalid and may no longer be used.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
hFile	HANDLE	Handle of the file to be closed

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		!= NO_ERROR	An error has occurred. For an explanation of the error values, refer to section 8.1.1.2.

To be considered for USB-memory-sticks

	Note
	FileClose When using a USB memory stick, a FileFlush should be executed after a FileWrite before FileClose is executed. This is due to the different behaviour of USB memory sticks. The FileFlush command causes the USB memory stick to complete a previous write operation.

8.1.5.2 FileDelete

Classification:

Function block (FB); non-reactive

Function:

Function block for deleting a file.

Description:

This function block deletes a file. A file with the READONLY attribute or an open file cannot be deleted.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
sFileName	FILENAME	Name of the file to be deleted

Function outputs:

Name	Data type	Description
xDone	BOOL	Processing of the FB completed without error

Name	Data type	Description	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		!= NO_ERROR	An error has occurred. For an explanation of the error values, refer to section 8.1.1.2.

8.1.5.3 FileEOF

Classification:

Function block (FB); non-reactive

Function:

Function block for requesting if the end of file has been reached.

Description:

This function block sets xEOF to TRUE if the current file pointer points to the end of the file. If the file pointer does not point to the end, the FB sets xEOF to FALSE.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
hFile	HANDLE	File handle

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		!= NO_ERROR	An error has occurred. For an explanation of the error values, refer to section 8.1.1.2.
xEOF	BOOL	TRUE: - File pointer to end of file FALSE: - File pointer not to end of file	

8.1.5.4 FileFlush

Classification:

Function block (FB); non-reactive

Function:

Function block for updating a file.

Description:

This function block writes all the contents which are still open (and located e.g. in an internal write buffer) into the file.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
hFile	HANDLE	File handle

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		!= NO_ERROR	An error has occurred. For an explanation of the error values, refer to section 8.1.1.2.

8.1.5.5 FileGetPos

Classification:

Function block (FB); non-reactive

Function:

Function block for determining the current file pointer position.

Description:

This function block reads the current file pointer position. The file must have been opened with FileOpen (FB).

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
hFile	HANDLE	File handle

Function outputs:

Name	Data type	Description
xDone	BOOL	Processing of the FB completed without error
xBusy	BOOL	FB is still being processed
xError	BOOL	Processing of the FB completed with error; error cause see "eError"

Name	Data type	Description	
eError	ERROR	NO_ERROR	OK
		!= NO_ERROR	An error has occurred. For an explanation of the error values, refer to section 8.1.1.2.
udiPos	UDINT	Current position of the file pointer	

8.1.5.6 FileGetSize

Classification:

Function block (FB); non-reactive

Function:

Function block for determining the current file size.

Description:

This function block reads the size of the file indicated with sFileName.

Function inputs:

Name	Data type	Description	
xExecute	BOOL	Rising edge: Falling edge:	Start of the action Reset of the function outputs
sFileName	FILENAME	File name	

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		!= NO_ERROR	An error has occurred. For an explanation of the error values, refer to section 8.1.1.2.
szSize	SIZE	Current file size in bytes.	

8.1.5.7 FileOpen

Classification:

Function block (FB); non-reactive

Function:

Function block for opening a file.

Description:

This function block opens an existing file or creates a new file. The return value is a file handle which is used as an hFile input variable in the function blocks FileWrite (FB), FileRead (FB) and FileClose (FB).

A file can be opened several times in MREAD mode. A file can only be opened once in MWRITE or MRDWR mode. This prevents that the file is opened twice, e.g. for reading and writing. A file opened in MAPPD mode may be opened a second time for reading.

PLEASE NOTE: There is no text mode. The file system exclusively expects data in binary mode.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
sFileName	FILENAME	File name (recommendation: No blank characters, uppercase characters only)
eFileMode	MODE	File mode: MWRITE Write MREAD Read MRDWR Read + Write MAPPD Write with appendices

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		!= NO_ERROR	An error has occurred. For an explanation of the error values, refer to section 8.1.1.2.
hFile	HANDLE	File handle	

8.1.5.8 FileRead

Classification:

Function block (FB); non-reactive

Function:

Function block for reading a file.

Description:

This function block is used to read a file from the current file pointer position onwards. The file must first be opened with FileOpen (FB).

Although less bytes than specified in szBuffer are readable, the execution of the FB is terminated with "xDone = TRUE". In addition, the number of actually read bytes is transferred in szSize.

The size of the data structure of the bytes to be read and the number of bytes to be read are not checked!

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
hFile	HANDLE	File handle
pBuffer	POINTER TO BYTE	Pointer to read buffer
szBuffer	SIZE	Size of the read buffer

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		!= NO_ERROR	An error has occurred. For an explanation of the error values, refer to section 8.1.1.2.
szSize	SIZE	Number of bytes copied to the read buffer	

8.1.5.9 FileRename

Classification:

Function block (FB); non-reactive

Function:

Function block for renaming a file.

Description:

This function block is used to change a file name.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
sFileNameOld	FILENAME	Old file name, e.g. "A:/subdir/oldfile.txt“
sFileNameNew	FILENAME	New file name, e.g. "newfile.txt“. No path!

Function outputs:

Name	Data type	Description
xDone	BOOL	Processing of the FB completed without error

Name	Data type	Description	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		!= NO_ERROR	An error has occurred. For an explanation of the error values, refer to section 8.1.1.2.

8.1.5.10 FileSetAttribute

Classification:

Function block (FB); non-reactive

Function:

Function block for changing file attributes.

Description:

This function block is used to change file attributes. The enumeration "ATTRIB" contains possible values.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
sFileName	FILENAME	File name
sFileAttrib	ATTRIB	File attributes: ARCHIVE HIDDEN NORMAL READONLY

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		!= NO_ERROR	An error has occurred. For an explanation of the error values, refer to section 8.1.1.2.

8.1.5.11 FileSetPos

Classification:

Function block (FB); non-reactive

Function:

Function block for setting the file pointer.

Description:

This function block is used to set the file pointer. The file must have been opened via FileOpen (FB).

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
hFile	HANDLE	File handle
udiPos	UDINT	Desired file pointer position

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		!= NO_ERROR	An error has occurred. For an explanation of the error values, refer to section 8.1.1.2.

8.1.5.12 FileWrite

Classification:

Function block (FB); non-reactive

Function:

Function block for writing a file.

Description:

This function block is used to write into a file. The file must have been opened via FileOpen (FB) in MWRITE, MRDWR or MAPPD mode. The contents of the write buffer, indicated by the pBuffer indicator, must not be changed during a Write operation into the file!

The size of the data structure for the bytes to be written and the number of bytes to be written are not checked!

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
hFile	HANDLE	File handle
pBuffer	PVOID	Pointer points to the write buffer

Name	Data type	Description
szSize	SIZE	Size of the write buffer

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		!= NO_ERROR	An error has occurred. For an explanation of the error values, refer to section 8.1.1.2.

8.1.6 Functions for sockets

8.1.6.1 Structures for socket functions

"SOCKETADDRESS“ structure:

The "SOCKETADDRESS“ structure is used to address the nodes in a network. The values can be directly entered in the structure (e.g. for the port number: sock_addr.sin_port := 32000). Conversion into the "Network Byte Order" is not required because the conversion is performed via the firmware.

Name	Data type	Description
sin_family	INT	The address family value must be set to "SOCKET_AF_INET“.
sin_port	UINT	Port number of the connection
sin_addr	IN_ADDR	32-bit IP address, e.g. 16# C0A864C7 for the address "192.168.100.199“ If the IP address must not be specified (e.g. for a TCP server which is expected to accept connections from any clients), the "SOCKET_INADDR_ANY“ is indicated.
sin_zero	ARRAY [0..7] OF BYTE	Not used

"SOCKETTIMEVAL“ structure:

The "SOCKETTIMEVAL“ structure indicates the timeout times.

Name	Data type	Description
tv_sec	DINT	Part of timeout indication in seconds
tv_usec	DINT	Part of timeout indication in micro seconds (0 – 999999)

"SOCKET_FDSET“ structure:

The "SocketSelect“ allows you to test several sockets for I/O events at the same time. The lists of sockets to be tested is transferred to the function block in the format of the "SOCKET_FDSET“ structure.

Name	Data type	Description
fd_count	UDINT	Number of valid socket handles in "fd_array“.
fd_array	ARRAY [0..4] OF HANDLE	List of socket handles

8.1.6.2 GetEthLinkState

Classification:

Function block (FB); non-reactive

Function:

Function block for querying the connection state of the Ethernet interface.

Description:

This function block is used to determine the connection state of the Ethernet interface. To use the socket functions, the connection state must be CONFIGURED.

Function inputs:

Name	Data type	Description	
xExecute	BOOL	Rising edge: Falling edge:	Start of the action Reset of the function outputs

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
eLinkState	ETHLINK STATE	NOTCONFIGURED CONFIGURED	Configuration of the Ethernet interface not yet completed Ethernet interface ready to operate

8.1.6.3 SocketCreate

Classification:

Function block (FB); non-reactive

Function:

Function block for creating a socket.

Description:

This function block is used to create a socket for TCP or UDP connections.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
diAddressFamily	DINT	Socket Address Family (must be SOCKET_AF_INET)
diType	DINT	Socket type. Values supported: SOCKET_TYPE_STREAM for TCP connections SOCKET_TYPE_DGRAM for UDP connections
diProtocol	DINT	Socket log (not used, should be 0)

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		GENERAL_SOCKET_ERROR	General socket error. The cause can be determined via FB "SocketGetError“.
		ERROR_INTERNAL_RESOURCE	All internal resources are occupied.
hSocket	HANDLE	Socket handle (only valid if not 16#FFFFFFFF)	

8.1.6.4 SocketClose

Classification:

Function block (FB); non-reactive

Function:

Function block for closing a socket.

Description:

This function block is used to close a socket created via "SocketCreate“. This frees all the resources used by the socket.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
hSocket	HANDLE	Socket handle to be released

Function outputs:

Name	Data type	Description
xDone	BOOL	Processing of the FB completed without error
xBusy	BOOL	FB is still being processed

Name	Data type	Description	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		GENERAL_SOCKET_ERROR	General socket error. The cause can be determined via FB "SocketGetError".
		ERROR_INTERNAL_RESOURCE	All internal resources are occupied.

8.1.6.5 SocketAccept

Classification:

Function block (FB); non-reactive

Function:

Function block for accepting a TCP connection from a remote client.

Description:

This function block is used by a TCP server for accepting a TCP connection with a remote client. If the function block has been completed successfully, the "SocketAddr" parameter contains the IP address of the client. The socket handle to be used for communication with the client is returned via the "hClientSocket" parameter. After closing the client connection, this socket handle must be released via the "SocketClose" function block in order to make internal resources available for other connections.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
hSocket	HANDLE	Socket handle of the server

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		GENERAL_SOCKET_ERROR	General socket error. The cause can be determined via FB "SocketGetError".
		ERROR_INTERNAL_RESOURCE	All internal resources are occupied.
		ERROR_WRONG_PARAMETER	Invalid parameter "hSocket"
diSockAddrSize	DINT	Size of the address structure "SocketAddr" in bytes	
SocketAddr	SOCKET ADDRESS	Socket address of the connected client	

Name	Data type	Description
hClientSocket	HANDLE	Socket handle of the client

8.1.6.6 SocketBind

Classification:

Function block (FB); non-reactive

Function:

Function block for linking a socket address to a socket.

Description:

This function block is used to link a socket address to a socket. For UDP communication the UDP port is opened additionally for receiving data.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
hSocket	HANDLE	Socket handle
pSocketAddr	POINTER TO SOCKET ADDRESS	Pointer set to the "SOCKETADDRESS" type memory area which contains the socket address to be connected to the socket handle.
diSockAddrSize	DINT	Size (in bytes) of the memory area which "pSocketAddr“ is pointing to.

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		GENERAL_SOCKET_ERROR	General socket error. The cause can be determined via FB "SocketGetError“.
		ERROR_INTERNAL_RESOURCE	All internal resources are occupied.
		ERROR_WRONG_PARAMETER	Invalid parameter "hSocket“

8.1.6.7 SocketConnect

Classification:

Function block (FB); non-reactive

Function:

Function block for connecting to a TCP server.

Description:

This function block is used by a TCP client to establish a connection to a TCP server.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
hSocket	HANDLE	Socket handle
pSocketAddr	POINTER TO SOCKET ADDRESS	The pointer points towards the "SOCKETADDRESS“ type memory area containing the socket address of the TCP server to which a connection is to be established.
diSockAddrSize	DINT	Size (in bytes) of the memory area which "pSocketAddr“ is pointing to.

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		GENERAL_SOCKET_ERROR	General socket error. The cause can be determined via FB "SocketGetError“.
		ERROR_INTERNAL_RESOURCE	All internal resources are occupied.
		ERROR_WRONG_PARAMETER	Invalid input parameter

8.1.6.8 SocketGetError

Classification:

Function block (FB); non-reactive

Function:

Function block for querying the error codes of the socket function called up last.

Description:

This function block is used to query the error code of the socket function called up last.

Value	Description
0	Invalid socket handle
1	The connection has already been closed
2	The connection was closed by the communication partner
3	General error
4	Invalid parameter

Value	Description
5	Ethernet interface not available
6	The communication partner cannot be contacted
7	Timeout
8	Function currently not available This is not a "real" error, but notifies you that the function is to be executed once again
9	No free memory available
10	Action prohibited
11	Link connection has already started
12	Participant, information, etc. cannot be found

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		ERROR_INTERNAL_RESOURCE	All internal resources are occupied.
diError	DINT	Error code of the socket function called up last	

8.1.6.9 SocketGetOption

Classification:

Function block (FB); non-reactive

Function:

Function block for querying the socket properties

Description:

This function block is used to query the properties of a socket. It currently supports the following socket properties:

- SOCKET_SO_RCVTIMEO: Timeout for waiting for receive data
- SOCKET_SO_SNDTIMEO: Timeout for transmission requests

The timeouts are returned via the data type "SOCKETTIMEVAL“. The structure contains elements for seconds ("tv_sec“) and microseconds ("tv_usec“).

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
hSocket	HANDLE	Socket handle
diLevel	DINT	Must be set to "SOCKET_SOL“
diOption	DINT	Indicates the socket property to be queried. The following properties are supported: SOCKET_SO_RCVTIMEO Receive data timeout SOCKET_SO_SNDTIMEO Transmit data timeout
pdiOptionValue	POINTER TO DINT	Address to which the value of the specified property is written. The memory must be made available by the application. For supported timeout values the address of a variable of the "SOCKETTIMEVAL“ type must be indicated here.
pdiOptionLen	POINTER TO DINT	Address to which the data size (in Bytes) of the specified property is written.

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		GENERAL_SOCKET_ERROR	General socket error. The cause can be determined via FB "SocketGetError“.
		ERROR_INTERNAL_RESOURCE	All internal resources are occupied.
		ERROR_WRONG_PARAMETER	Invalid input parameter

8.1.6.10 SocketInetAtoN

Classification:

Function block (FB); non-reactive

Function:

Function block for converting IP addresses

Description:

This function block is used to convert IP addresses from the ASCII format into the numerical format. Example: The address "192.168.100.199“ is converted into the number 16#C0A864C7.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs

Name	Data type	Description
stIPAddr	STRING(15)	IP address in the ASCII format
pIN_ADDR	POINTER TO IN_ADDR	Address of the memory area the converted IP address is written to. The memory area must be made available by the application.

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		GENERAL_SOCKET_ERROR	General socket error. The cause can be determined via FB "SocketGetError“.
		ERROR_INTERNAL_RESOURCE	All internal resources are occupied.
		ERROR_WRONG_PARAMETER	Invalid input parameter

8.1.6.11 SocketIoctl

Classification:

Function block (FB); non-reactive

Function:

Function block for executing the IOCTL command.

Description:

This function block is used to execute the IOCTL commands. The function block supports the following commands:

- SOCKET_FIONREAD: Returns the number of bytes which are currently available in the receive buffer of the socket.
- SOCKET_FIONBIO: Sets the socket to "blocking" or "non-blocking"

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
hSocket	HANDLE	Socket handle
diCommand	DINT	Specifies the IOCTL command which is being executed. The function block supports the following commands: SOCKET_FIONREAD: Returns the number of bytes which are currently available in the receive buffer of the socket. SOCKET_FIONBIO: Sets the socket to "blocking" or "non-blocking"

Name	Data type	Description
diParameterIn	DINT	Input parameter of the "SOCKET_FIONBIO“ command: 0 blocking 1 non-blocking

Function outputs:

<table>
<tr><th>Name</th><th>Data type</th><th colspan="2">Description</th></tr>
<tr><td>xDone</td><td>BOOL</td><td colspan="2">Processing of the FB completed without error</td></tr>
<tr><td>xBusy</td><td>BOOL</td><td colspan="2">FB is still being processed</td></tr>
<tr><td>xError</td><td>BOOL</td><td colspan="2">Processing of the FB completed with error; error cause see "eError"</td></tr>
<tr><td rowspan="4">eError</td><td rowspan="4">ERROR</td><td>NO_ERROR</td><td>OK</td></tr>
<tr><td>GENERAL_SOCKET_ERROR</td><td>General socket error. The cause can be determined via FB "SocketGetError“.</td></tr>
<tr><td>ERROR_INTERNAL_RESOURCE</td><td>All internal resources are occupied.</td></tr>
<tr><td>ERROR_WRONG_PARAMETER</td><td>Invalid input parameter</td></tr>
<tr><td>diParameterOut</td><td>DINT</td><td colspan="2">Output parameter of the "SOCKET_FIONREAD“ command: Returns the number of bytes which are currently available in the receive buffer of the socket.</td></tr>
</table>

8.1.6.12 SocketListen

Classification:

Function block (FB); non-reactive

Function:

Function block for opening a port which receives connection requests from TCP clients.

Description:

For TCP server applications this function block opens a port which receives connection requests from TCP clients.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
hSocket	HANDLE	Socket handle
diMaxConnections	DINT	Maximum number of simultaneous connections

Function outputs:

Name	Data type	Description
xDone	BOOL	Processing of the FB completed without error
xBusy	BOOL	FB is still being processed

Name	Data type	Description	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		GENERAL_SOCKET_ERROR	General socket error. The cause can be determined via FB "SocketGetError“.
		ERROR_INTERNAL_RESOURCE	All internal resources are occupied.
		ERROR_WRONG_PARAMETER	Invalid input parameter

8.1.6.13 SocketRecv

Classification:

Function block (FB); non-reactive

Function:

Function block for reading the data received via a TCP connection.

Description:

This function block is used to read the data received via a TCP connection. The function block only stops its execution (xDone = TRUE) when receive data is present. This does not mean that the FB blocks, but the execution state remains "xBusy = TRUE“.

To prevent that the execution state of the FB remains "xBusy = TRUE“ if no receive data is present:

- Query via the FB "SocketIoctl“ whether receive data is present and only call up "SocketRecv“ if receive data is present
- Define a receive timeout via "SocketSetOption“: If no receive data is present, the processing of the FB is terminated after this timeout. This is indicated by "xError = TRUE“ and "eError = GENERAL_SOCKET_ERROR“ (xDone remains FALSE). The error value 7 (timeout) would be returned if "SocketGetError“ is called up.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
hSocket	HANDLE	Socket handle
pbyBuffer	POINTER TO BYTE	Address of the receive buffer. The memory area must be made available by the application.
diBufferSize	DINT	Size of the receive buffer (in bytes)
diFlags	DINT	Not used

Function outputs:

Name	Data type	Description
xDone	BOOL	Processing of the FB completed without error
xBusy	BOOL	FB is still being processed
xError	BOOL	Processing of the FB completed with error; error cause see "eError"

Name	Data type	Description	
eError	ERROR	NO_ERROR	OK
		GENERAL_SOCKET_ERROR	General socket error. The cause can be determined via FB "SocketGetError".
		ERROR_INTERNAL_RESOURCE	All internal resources are occupied.
		ERROR_WRONG_PARAMETER	Invalid input parameter
diRecBytes	DINT	Number of bytes received	

8.1.6.14 SocketRecvFrom

Classification:

Function block (FB); non-reactive

Function:

Function block for reading the data received via a UDP connection.

Description:

This function block reads the data received via a UDP connection. The function block only stops its execution (xDone = TRUE) when receive data is present. This does not mean that the FB blocks, but the execution state remains "xBusy = TRUE".

To prevent that the execution state of the FB remains "xBusy = TRUE" if no receive data is present:

- Query via the FB "SocketIoctl" whether receive data is present and only call up "SocketRecv" if receive data is present
- Define a receive timeout via "SocketSetOption": If no receive data is present, the processing of the FB is terminated after this timeout. This is indicated by "xError = TRUE" and "eError = GENERAL_SOCKET_ERROR" (xDone remains FALSE). The error value 7 (timeout) would be returned if "SocketGetError" is called up.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
hSocket	HANDLE	Socket handle
pbyBuffer	POINTER TO BYTE	Address of the receive buffer. The memory area must be made available by the application.
diBufferSize	DINT	Size of the receive buffer (in bytes)
diFlags	DINT	Not used

Function outputs:

Name	Data type	Description
xDone	BOOL	Processing of the FB completed without error
xBusy	BOOL	FB is still being processed
xError	BOOL	Processing of the FB completed with error; error cause see "eError"

Name	Data type	Description	
eError	ERROR	NO_ERROR	OK
		GENERAL_SOCKET_ERROR	General socket error. The cause can be determined via FB "SocketGetError".
		ERROR_INTERNAL_RESOURCE	All internal resources are occupied.
		ERROR_WRONG_PARAMETER	Invalid input parameter
diRecBytes	DINT	Number of bytes received	
diSockAddrSize	DINT	Size of the address structure "SocketAddr"	
SocketAddr	SOCKET ADDRESS	IP address of the client which transmitted the data	

8.1.6.15 SocketSelect

Classification:

Function block (FB); non-reactive

Function:

Function block for testing the sockets for the presence of I/O events.

Description:

This function block is used to check whether I/O events are pending for sockets. The sockets to be checked must be entered in a structure of the "SOCKET_FDSET" type. This structure contains the "fd_array" in which socket handles are entered. Up to 5 handles can be entered. The actual number of handles entered is written into the "fd_count" structure element.

The following I/O events can be monitored:

- The "Socket-Set" must be checked for the presence of receive data. To do this, the address of the socket set structure is transferred to the "pfdRead" input parameter.
- For the sockets in the "Socket-Set" is must be checked whether pending transmit operations have been completed. The address of the socket set structure is transferred to the "pdfWrite" input parameter.

After the processing of the FBs has been terminated due to I/O events (xDone = TRUE), the corresponding socket set only contains the sockets for which the event was valid.

The processing of the FB is only completed if at least one event has occurred or the timeout (input parameter "ptvTimeout") has elapsed. If the FB processing was completed because the timeout has elapsed, "diReady = 0" and the "xDone = TRUE" are set. This socket sets now only contain their original data.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
diNfds	DINT	Number of sockets to be monitored

Name	Data type	Description
pfdRead	POINTER TO SOCKET_ FDSET	Address of the "SOCKET_FDSET“ structure containing the list of sockets to be checked for the presence of receive data. The parameter can be set to pfdRead = 0 if no receive data check is to be performed. Once the processing of the FB has been completed (xDone = TRUE), the structure only contains sockets for which receive data is present.
pfdWrite	POINTER TO SOCKET_ FDSET	Address of the "SOCKET_FDSET“ structure containing the list of sockets to be checked for completion of a transmit process. The parameter can be set to pdfWrite = 0 if it is not required. Once the processing of the FB has been completed (xDone = TRUE), the structure only contains the sockets for which the transmit operation has already been completed.
pfdExcept	POINTER TO SOCKET_ FDSET	Not used
ptvTimeout	POINTER TO SOCKET TIMEVAL	Address of a variable of the "SOCKETTIMEVAL“ type which contains the timeout value. The processing of the function block is terminated at the latest after this timeout ("xDone = TRUE“) has elapsed - irrespective of whether an I/O event is present for one of the specified sockets. If not timeout is to be used, ptvTimeout = 0 can be set.

Function outputs:

<table>
<tr><th>Name</th><th>Data type</th><th colspan="2">Description</th></tr>
<tr><td>xDone</td><td>BOOL</td><td colspan="2">Processing of the FB completed without error</td></tr>
<tr><td>xBusy</td><td>BOOL</td><td colspan="2">FB is still being processed</td></tr>
<tr><td>xError</td><td>BOOL</td><td colspan="2">Processing of the FB completed with error; error cause see "eError"</td></tr>
<tr><td rowspan="4">eError</td><td rowspan="4">ERROR</td><td>NO_ERROR</td><td>OK</td></tr>
<tr><td>GENERAL_SOCKET_ERROR</td><td>General socket error. The cause can be determined via FB "SocketGetError“.</td></tr>
<tr><td>ERROR_INTERNAL_RESOURCE</td><td>All internal resources are occupied.</td></tr>
<tr><td>ERROR_WRONG_PARAMETER</td><td>Invalid input parameter</td></tr>
<tr><td>diReady</td><td>DINT</td><td colspan="2">Number of sockets for which I/O events are present</td></tr>
</table>

8.1.6.16 SocketSend

Classification:

Function block (FB); non-reactive

Function:

Function block for transmitting data via a TCP connection.

Description:

This function block is used to transmit data via a TCP connection. If the number of data transmitted is less (output parameter "diSendBytes“) than the number of data in the "pbyBuffer“ buffer (input parameter "diBufferSize“), the residual data must be transmitted by the application.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
hSocket	HANDLE	Socket handle
pbyBuffer	POINTER TO BYTE	Address of the transmit buffer
diBufferSize	DINT	Number of bytes to be transmitted
diFlags	DINT	Not used

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		GENERAL_SOCKET_ERROR	General socket error. The cause can be determined via FB "SocketGetError“.
		ERROR_INTERNAL_RESOURCE	All internal resources are occupied.
		ERROR_WRONG_PARAMETER	Invalid input parameter
diSendBytes	DINT	Number of bytes actually sent	

8.1.6.17 SocketSendTo

Classification:

Function block (FB); non-reactive

Function:

Function block for transmitting data via a UDP connection.

Description:

This function block is used to transmit data via a UDP connection. If the number of data transmitted is less (output parameter "diSendBytes“) than the number of data in the "pbyBuffer“ buffer (input parameter "diBufferSize“), the residual data must be transmitted by the application.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs

Name	Data type	Description
hSocket	HANDLE	Socket handle
pbyBuffer	POINTER TO BYTE	Address of the transmit buffer
diBufferSize	DINT	Number of bytes to be transmitted
diFlags	DINT	Not used
pSocketAddr	POINTER TO SOCKET ADDRESS	Address of the variable (of the SOCKETADDRESS type) which contains the destination IP address.
diSockAddrSize	DINT	Size (in bytes) of the memory area to which "pSocketAddr" is pointing ("SIZEOF(SOCKETADDRESS)" should always be transferred)

Function outputs:

<table>
<tr><th>Name</th><th>Data type</th><th colspan="2">Description</th></tr>
<tr><td>xDone</td><td>BOOL</td><td colspan="2">Processing of the FB completed without error</td></tr>
<tr><td>xBusy</td><td>BOOL</td><td colspan="2">FB is still being processed</td></tr>
<tr><td>xError</td><td>BOOL</td><td colspan="2">Processing of the FB completed with error; error cause see "eError"</td></tr>
<tr><td rowspan="4">eError</td><td rowspan="4">ERROR</td><td>NO_ERROR</td><td>OK</td></tr>
<tr><td>GENERAL_SOCKET_ERROR</td><td>General socket error. The cause can be determined via FB "SocketGetError".</td></tr>
<tr><td>ERROR_INTERNAL_RESOURCE</td><td>All internal resources are occupied.</td></tr>
<tr><td>ERROR_WRONG_PARAMETER</td><td>Invalid input parameter</td></tr>
<tr><td>diSendBytes</td><td>DINT</td><td colspan="2">Number of bytes actually transmitted</td></tr>
</table>

8.1.6.18 SocketSetOption

Classification:

Function block (FB); non-reactive

Function:

Function block for defining socket properties.

Description:

This function block can be used to define the properties of a socket. The following socket properties are supported:

- SOCKET_SO_RCVTIMEO: Timeout for waiting for receive data
- SOCKET_SO_SNDTIMEO: Timeout for transmission requests

The timeouts are specified via the "SOCKETTIMEVAL" data type. The structure contains elements for seconds ("tv_sec") and microseconds ("tv_usec").

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
hSocket	HANDLE	Socket handle
diLevel	DINT	Must be set to "SOCKET_SOL"
diOption	DINT	Indicates the socket property to be specified. The following properties are supported: SOCKET_SO_RCVTIMEO Receive data timeout SOCKET_SO_SNDTIMEO Transmit data timeout
pdiOptionValue	POINTER TO DINT	Address of the variable which contains the property value. For the timeout values supported, the address of a variable of the "SOCKETTIMEVAL" type must be indicated here.
diOptionLen	DINT	Size of the memory area (in bytes) containing the property value.

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		GENERAL_SOCKET_ERROR	General socket error. The cause can be determined via FB "SocketGetError".
		ERROR_INTERNAL_RESOURCE	All internal resources are occupied.
		ERROR_WRONG_PARAMETER	Invalid input parameter

8.1.7 User logbook management

This chapter describes how to manage the user logbook. For general information on the user logbook refer to chapter 7.8.3.

8.1.7.1 UserLogErase

Classification:

Function block (FB); non-reactive

Function:

Function block for deleting the user logbook

Description:

This function block can be used to remove all entries from the user logbook.

Function inputs:

Name	Data type	Description	
xExecute	BOOL	Rising edge: Falling edge:	Start of the action Reset of the function outputs

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		ERASE_FAILED	An error occurred during the delete action

8.1.7.2 UserLogGetNumOfEntries

Classification:

Function block (FB); non-reactive

Function:

Function block for querying the number of entries in the user logbook.

Description:

This function block is used to determine the current number of entries in the user logbook.

Function inputs:

Name	Data type	Description	
xExecute	BOOL	Rising edge: Falling edge:	Start of the action Reset of the function outputs

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
udiNumOfEntries	UDINT	Number of entries in the user logbook	

8.1.7.3 UserLogGetSizeAndVersion

Classification:

Function block (FB); non-reactive

Function:

Function block for querying the size of an entry and the version of the user logbook.

Description:

This function block is used to determine the size of an entry in the user logbook (in Bytes). It also returns the version specified by the user via the "UserLogSetSizeAndVersion“ function block.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
wVersion	WORD	Version of the user logbook	
wSize	WORD	Size in bytes of an entry in the user logbook	

8.1.7.4 UserLogSetSizeAndVersion

Classification:

Function block (FB); non-reactive

Function:

Function block for initializing the user logbook

Description:

This function block is used to initialize the user logbook. It must be called up before using the user logbook for the first time.

This function block removes all entries written into the logbook by this time. The size of a user logbook entry is set to the value "wSize“ (in Bytes). Via this parameter, the user can determine the size of the logbook entries; however, this size is identical for all entries.

The "wVersion“ input parameter can be used freely, e.g. in order to check whether the user program is compatible with the entries present in the user logbook. The "wVersion“ input parameter is only saved internally to the logbook and can be read out again via the "UserLogGetSizeAndVersion" function block. The value is no longer evaluated by the controller firmware.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
wVersion	WORD	Version of the user logbook

Name	Data type	Description
wSize	WORD	Size in bytes of an entry in the user logbook

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		BUSY	User logbook management busy
		ERROR_UNKNOWN	Internal error

8.1.7.5 UserLogWriteEntry

Classification:

Function block (FB); non-reactive

Function:

Function block for writing user logbook entries.

Description:

This function block is used to write an entry into the user logbook. The start address of the data to be written into the logbook is transferred via the "pbyBuffer“ parameter. The user has defined the number of bytes written into the logbook from this address onwards via the "UserLogSetSizeAndVersion" function block during initialization. The user must ensure that the memory area specified via "pbyBuffer“ corresponds to the size of the logbook entry.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Start of the action Falling edge: Reset of the function outputs
pbyBuffer	POINTER TO BYTE	Address of the data to be written into the user logbook

Function outputs:

Name	Data type	Description	
xDone	BOOL	Processing of the FB completed without error	
xBusy	BOOL	FB is still being processed	
xError	BOOL	Processing of the FB completed with error; error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		BUSY	User logbook management busy
		WRITE_FAILED	Internal error

8.1.7.6 UserLogReadEntry

Classification:

Function block (FB); non-reactive

Function:

Function block for reading a user logbook entry.

Description:

This function block is used to read an entry from the user logbook. The ID of the entry to be read from the logbook is passed with the parameter "dwIdLog". The initial address of the memory area to which the read logbook entry is written is passed with the parameter "pbyBuffer". The number of bytes read from the logbook and stored in this address is passed with the parameter "dwBufferSize". The user must ensure that this number of bytes and the memory area specified by "pbyBuffer" correspond to the size of the logbook entry.

Function inputs:

Name	Data type	Description
xExecute	BOOL	Rising edge: Action start Falling edge: Resetting the function outputs
dwIdLog	DWORD	Userlog Entry ID. 0 - last entry, 1 - penultimate entry, etc.
pbyBuffer	POINTER TO BYTE	Address of the memory area in which the read logbook entry is written. The memory must be reserved by the user program
dwBufferSize	DWORD	Length of the data to be read from the user logbook

Function outputs:

Name	Data type	Description	
xDone	BOOL	Execution of the FB completed error-free	
xBusy	BOOL	FB is still under construction	
xError	BOOL	Execution of the FB terminated with error, error cause see "eError"	
eError	ERROR	NO_ERROR	OK
		BUSY	user logbook administration is currently busy
		WRITE_FAILED	Internal error
pdwTimeStamp	DWORD	Read time stamp of Userlog entry	

8.1.8 CAN_Async

8.1.8.1 CAN_Tra_Async

Classification:

Function; non-reactive

Function:

Function for the cycle-independent transmission of periodical CAN telegrams.

Description:

This function is used to define CAN telegrams at the lowest level (CobId + 8 data bytes) which are sent at regular intervals directly from the runtime system. Up to 10 telegrams can be defined.

Function inputs:

name	data type	description
Index	WORD	Value range 1 to 10. All other values are invalid
Active	BOOL	TRUE➔ a telegram defined as valid is sent regularly.
CanNo	WORD	0, 1, 2, 3 ➔ CAN number. All other values are invalid.
RepeatTime	DWORD	Telegram repetition time in microseconds, value range> = 1000. Every 1ms is checked whether the following condition is fulfilled: (ActTime + 500-LastTime)> = RepeatTime If the condition is fulfilled, the telegram is sent.
LifeTime	WORD	in Microseconds ➔ not used
Count	WORD	Value range 0 to 8. Specifies the number of data bytes to send.
CobId	DWORD	CobId of the CAN telegram (11 or 29 Bit).
pData	POINTER TO ARRAY[1..8] OF USINT	Pointer to the data to be transferred. The data is copied during definition and stored in the runtime system.

Function outputs:

name	data type	description
CAN_Tra_Async	BOOL	TRUE ➔ All input parameters were OK FALSE ➔ One or more input parameters were not OK

8.1.9 CAN LISTEN function

In the "Configuration" tab of the Device Editor of the Safe Application Program, you can define whether a CAN interface is assigned to the Safe Application program or the Standard Application program:

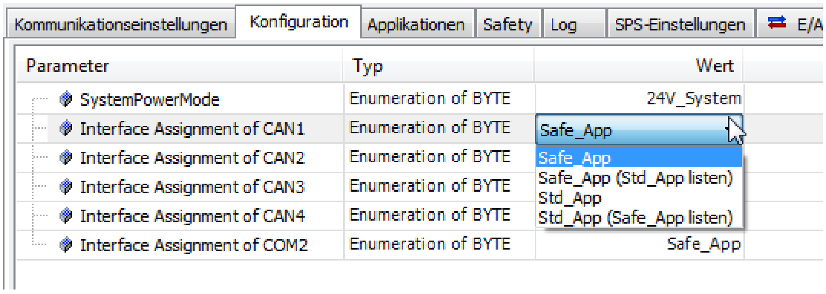


Figure 8-1: Configuration of the interface assignment

The so called "CAN Listen Function" can be enabled as an alternative to the exclusive assignment to an application program. In this mode the CAN interface is e.g. always assigned to the Standard Application program. However, the Safe Application program can "listen" to the messages received via the CAN interface. The messages received via the CAN interface are therefore saved to a separate buffer referred to as the CAN Listen Buffer.

However, CAN messages can only be transmitted via the application program the CAN interface is assigned to.

The following configuration options are available for each CAN interface:

- Safe_App: the CAN interface is assigned to the Safe Application program
- Safe_App (Std_App listen): the CAN interface is assigned to the Safe Application program, the Standard Application program can listen
- Std_App: the CAN interface is assigned to the Standard Application program
- Std_App (Safe_App listen): the CAN interface is assigned to the Standard Application program, the Safe Application program can listen

This chapter describes the functions for managing the CAN Listen Buffer of the Safe Application program. To ensure that the "CanListenMask“ and "CanListenRec“ functions operate properly, the corresponding CAN interface must be assigned to "Std_App (Safe_App listen)".

8.1.9.1 CanListenMask

Classification:

Function; non-reactive

Function:

Function for defining a filter rule for the CAN Listen function.

Description:

This function is used to define a filter rule for the CAN Listen function. This means that only CAN messages whose CAN identifier follows this rule are written into the CAN Listen buffer. The number of CAN messages in the CAN buffer, and thus the probability of a buffer overflow, can thus be reduced.

The filter rule is defined as follows: The CAN identifier is linked to the value "udiMask“ via a bit-wise AND operation. The CAN message is only written into the buffer if the resulting value is identical with "udiFilter“. This rule is implemented as follows in pseudo code:

(CAN identifier AND udiMask) = udiFilter

Example: If the values

udiMask := 16#FFFFFF00

udiFilter := 16#00000500

are set for the filter rule, only the CAN messages in the identifier area between 16#500 and 16#5FF are written into the CAN Listen buffer.

If the "CanListenMask“ has not been called up, both "udiMask“ and "udiFilter“ are set to 0. With this setting all the CAN messages received are written into the CAN buffer.

The values for udiMask and udiFilter must fulfill the condition

(udiFilter AND (NOT udiMask)) = 0

in order to ensure that the filter rule can be followed. Otherwise, the function returns the value "WRONG_PARAMETER“ and the values are not applied according to the filter rule.

Function inputs:

Name	Data type	Description
udiNet	UDINT	CAN interface (0 – 3) to which the filtering rule is to be applied
udiMask	UDINT	Identifier mask
udiFilter	UDINT	Identifier filter

Function outputs:

Name	Data type	Description	
	ERROR	NO_ERROR	OK
		WRONG_PARAMETER	Invalid parameters "udiMask“ or "udiFilter“
		INVALID_CANCHANNEL	Invalid parameter "udiNet“
		CANLISTEN_NOTCONFIGURED	The CAN interface not configured for the "Listen function".

8.1.9.2 CanListenRec

Classification:

Function; non-reactive

Function:

Function for reading out the CAN Listen buffer.

Description:

This function is used to read the CAN message out of the CAN Listen buffer. A CAN message was written into the structure addressed via "pCanMsg“ if the function returns the value "NO_ERROR".

The CAN messages are returned via the "T_CAN_LISTEN“ type. The elements of this structure have the following significance:

Name	Data type	Description
TimeStamp	TIME	Time of receipt of the CAN message (in ms since the start of the controller)
bXtd	BOOL	Extended frame format: FALSE: 11-bit CAN identifier TRUE: 29-bit CAN identifier
bRtr	BOOL	FALSE: Standard data frame TRUE: Remote Frame
bySize	BYTE	Number of valid data bytes in the "Data" array
CobId	DWORD	CAN identifier
Data	ARRAY [0..7] OF BYTE	CAN data

Function inputs:

Name	Data type	Description
udiNet	UDINT	CAN interface (0 – 3)

Name	Data type	Description
pCanMsg	POINTER TO T_CAN_ LISTEN	Address of the "T_CAN_LISTEN“ structure into which the CAN message read out of the CAN Listen buffer is written. The memory must be reserved by the application program.

Function outputs:

Name	Data type	Description	
	ERROR	NO_ERROR	OK
		WRONG_PARAMETER	Invalid parameter “pCanMsg”
		INVALID_CANCHANNEL	Invalid parameter "udiNet“
		CANLISTEN_NOTCONFIGURED	The CAN interface is not configured for the "Listen function".
		BUFFER_EMPTY	The CAN Listen buffer is empty
		BUFFER_OVERFLOW	An overflow of the CAN Listen buffer has occurred

8.1.10 Functions for querying device information

8.1.10.1 GetDeviceInfo

Classification:

Function; safe

Function:

Function for querying device data.

Description:

This function is used to query device data from the control system.

The device data is copied to the application program variable whose address was transferred via the "pDeviceInfo“ function parameter. The variable must be of the "DEVICEINFO“ type. The structure contains the following information:

- udiSerialNr: Serial number of the controller
- abEthMac: Ethernet MAC address
- bNrOfSlaveBoards: Number of slave modules of the control system (not including the programmable controller module; the total number of controller modules is 1 higher than bNrOfSlaveBoards)

Function inputs:

Name	Data type	Description
pDeviceInfo	POINTER TO DEVICEINFO	Address of the "DEVICEINFO“ structure into which the device data is written. The memory must be reserved by the application program.

Function outputs:

Name	Data type	Description	
	ERROR	NO_ERROR	OK
		WRONG_PARAMETER	Invalid parameter “pDeviceInfo”

8.1.10.2 GetFirmwareVersion

Classification:

Function; safe

Function:

Function for querying the firmware versions.

Description:

This function is used to query the firmware versions of the control system.

The CPU whose firmware version is to be queried is addressed via the "eLayer“ and "eDevice“ parameters. The controller module is selected via "eLayer“. The "eDevice“ parameter selects the desired CPU on the controller module (provided that the module consists of several CPUs).

The version is returned in the format of the "FWVERSION“ structure. It contains version information concerning the various firmware components:

- Fork: Firmware component for starting the Boot Loader and the firmware.
- Boot Loader: Firmware component for updating the system.
- Firmware: Firmware component for executing the application program.

The version information for every firmware component is of the "VERSIONINFO“ type and consists of four digits:

V [byMaj].[byMin].[byBugfix].[byPatch]

Every digit can have a value in the 0 – 99 range. Version

V 255.255.255.255

indicates that either the firmware component does not exist or that no valid version information could be determined.

Function inputs:

Name	Data type	Description
eLayer	LAYER	This enumeration is used to select from which controller module firmware versions are to be queried.
eDevice	DEVICE	This enumeration is used to select from which CPU of the controller module (provided that several CPUs exist) firmware versions are to be queried.

Function outputs:

Name	Data type	Description
	FWVERSION	Structure containing the desired firmware versions.

8.1.10.3 GetModuleInfo

Classification:

Function; safe

Function:

Function for querying information concerning a controller module.

Description:

This function is used to query information concerning a controller module.

The desired module is addressed via the "eLayer“ parameter. The module data is copied to the application program variable whose address is transferred via the "pModuleInfo“ function parameter. The variable must be of the "MODULEINFO“ type. The structure contains the following information:

- udiSerialNr: Serial number of the module
- udiHwVersion: Hardware version of the module

Function inputs:

Name	Data type	Description
eLayer	LAYER	This enumeration is used to select the controller module for which information is to be queried.
pModuleInfo	POINTER TO MODULEINFO	Address of the "MODULEINFO“ structure into which information is to be written. The memory must be reserved by the application program.

Function outputs:

Name	Data type	Description	
	ERROR	NO_ERROR	OK
		WRONG_PARAMETER	Invalid parameter “pModuleInfo” or "eLayer“

8.1.10.4 GetFirmwareCrc

Classification:

Function; safe

Function:

Function for querying the firmware checksums (CRC64).

Description:

This function is used to query the firmware checksums (CRC64) of the control system.

The CPU whose firmware checksums are to be queried is addressed via the "eLayer“ and "eDevice“ parameters. The controller module is selected via "eLayer“. The "eDevice“ parameter is used to select an individual CPU on the controller module (if the module consists of several CPUs).

The checksum is returned in the format of the "FWCRC“ structure. It contains the checksums of the various firmware components:

- Crc64Fork: Checksum of the Fork (firmware component for starting the Boot Loader or firmware)
- Crc64Bootloader: Checksum of the Boot Loader (firmware component for performing a system update)
- Crc64Firmware: Checksum of the firmware (firmware component for executing the application program)

This checksum of the "ULINT" data type is a CRC64 calculated over the entire memory space of the firmware component.

Function inputs:

Name	Data type	Description
eLayer	LAYER	This enumeration is used to select from which controller module the firmware checksums are to be queried.
eDevice	DEVICE	This enumeration is used to select from which CPU of the controller module (if several CPUs exist) the firmware checksums are to be queried.
pFwCrc	POINTER TO FWCRC	Address of the structure into which the firmware checksums are to be entered.

Function outputs:

Name	Data type	Description	
	ERROR	NO_ERROR	OK
		WRONG_PARAMETER	Transfer parameter invalid

8.1.10.5 GetOperatingHoursAndStartups

Classification:

Function; non-reactive

Function:

This function is used to query the operating hours and the startup counter.

Description:

With this function you can query the operating hours of the controller and the startup counter (number of switch-on operations) of the controller.

Function inputs:

Name	Data type	Description
pOperatingHours	POINTER TO UDINT	Address of the variable into which the operating hours of the controller are written (in seconds).
pStartups	POINTER TO UDINT	Address of the variable into which the number of switch-on operations of the controller is written.

Function outputs:

Name	Data type	Description	
	ERROR	NO_ERROR	OK

Name	Data type	Description	
		WRONG_PARAMETER	Parameter “pOperatingHours” or "pStartups“ invalid

8.1.11 Functions for retain data

8.1.11.1 LoadRetain

Classification:

Function; safe

Function:

Function for reading the retain data which has been saved previously.

Description:

This function is used to load the previously saved retain data into the retain data segment of the application. Before calling up this function, the application variables declared as 'retain' have their initialization value.

For more detailed information on retain data refer to Chapter 7.5.7.5.

Function outputs:

Name	Data type	Description	
	ERROR	NO_ERROR	OK
		READ_FAILED	Retain data read error This error can occur if no valid retain data block has been saved.

8.1.11.2 SaveRetain

Classification:

Function; safe

Function:

Function for saving retain data.

Description:

This function is used to save the variables declared as retain data in the application program as zero-voltage-safe. The function call only starts the save operation; the actual saving takes place in the firmware "in the background". The "SaveRetainState" function is used to check whether the save operation still persists or has been completed. If the save operation is still active, no further save operation can be started (i.e. the function provides the error value ""BUSY"). A copy of the retain data segment is created when calling up "SaveRetain". This means that normal system operation can be continued during the save operation with retain variables without the risk of saving an inconsistent version.

For more detailed information on retain data refer to Chapter 7.5.7.5.

Function outputs:

Name	Data type	Description	
	ERROR	NO_ERROR	OK
		WRITE_FAILED	Retain data write error
		BUSY	The controller is still busy with the previously started retain data write operation.

8.1.11.3 SaveRetainState

Classification:

Function; safe

Function:

Function for querying the status of a retain data save operation.

Description:

This function is used to check whether the retain data save operation initiated via the "SaveRetain“ has already been completed. See also "SaveRetain“ function.

Function outputs:

Name	Data type	Description	
	ERROR	NO_ERROR	Retain data save operation completed.
		BUSY	Retain data save operation not yet completed.

8.1.12 Functions for the serial interface

8.1.12.1 Init_RS232

Classification:

Function; non-reactive

Function:

Function for initializing the serial interface RS232-2.

Description:

This function is used to initialize the serial interface RS232-2.

Function inputs:

Name	Data type	Description
FreeForUS	BOOL	Here you select by which application program the interface is to be used (all other parameters are ignored if the Standard Application program is selected). FALSE Safe Application program TRUE Standard Application program
ChanalNo	WORD	Selection of the RS232 interface (dfS). Possible values: 1 RS232-2 (free programmable) 2 RS232-1 (if not configured as programming interface) Selection for dfS-P: 2 RS232-1 (free programmable for SCM-Eth-SL side, if not configured as programming interface) Note: For GCM-P library “SysCom” is used
Baud	WORD	Baud rate setting: 48 4800 Baud 96 9600 Baud 192 19200 Baud 384 38400 Baud 576 57600 Baud 1152 115200 Baud
Data	WORD	Number of data bits. Possible values: DATA_8 8 data bits
Parity	WORD	Parity setting. Possible values: PARITY_NONE PARITY_ODD PARITY_EVEN
Stop	WORD	Stop bit setting. Possible values: STOP_1 1 stop bit STOP_2 2 stop bits
Flow control	WORD	Flow control. Possible values: NOTHING No flow control

Function outputs:

Name	Data type	Description	
	INT	0	OK
		-2	Invalid parameter

8.1.12.2 Rec_RS232

Classification:

Function; non-reactive

Function:

Function for reading the receive data of the serial interface RS232-2.

Description:

This function is used to read the receive data on the serial interface. The function does not wait for receive data, but only transmits already received data into the storage area addressed with "pData“.

Function inputs:

Name	Data type	Description
ChanalNo	WORD	Selection of the RS232 interface. Possible values: 1 RS232-2 (free programmable) 2 RS232-1 (if not configured as programming interface) Selection for dfS-P: 2 RS232-1 (free programmable for SCM-Eth-SL side, if not configured as programming interface) Note: For GCM-P library “SysCom” is used
pData	POINTER TO BYTE	Address of the receive buffer
MaxLength	WORD	Size of the receive buffer provided by the application program

Function outputs:

Name	Data type	Description
	WORD	Number of data bytes received and stored in the receive buffer.

8.1.12.3 Tra_RS232

Classification:

Function; non-reactive

Function:

Function for transmitting data via the serial interface RS232-2.

Description:

This function is used to transmit data via the serial interface RS232-2. The function does not wait until data has been transmitted, but only copies the data into the transmit buffer.

Function inputs:

Name	Data type	Description
ChanalNo	WORD	Selection of the RS232 interface. Possible values: 1 RS232-2 (free programmable) 2 RS232-1 (if not configured as programming interface) Selection for dfS-P: 2 RS232-1 (free programmable for SCM-Eth-SL side, if not configured as programming interface) Note: For GCM-P library “SysCom” is used
pData	POINTER TO BYTE	Address of the transmit data buffer

Name	Data type	Description
Length	WORD	Number of bytes to be transmitted

Function outputs:

Name	Data type	Description
	WORD	Number of bytes transferred to the transmit buffer

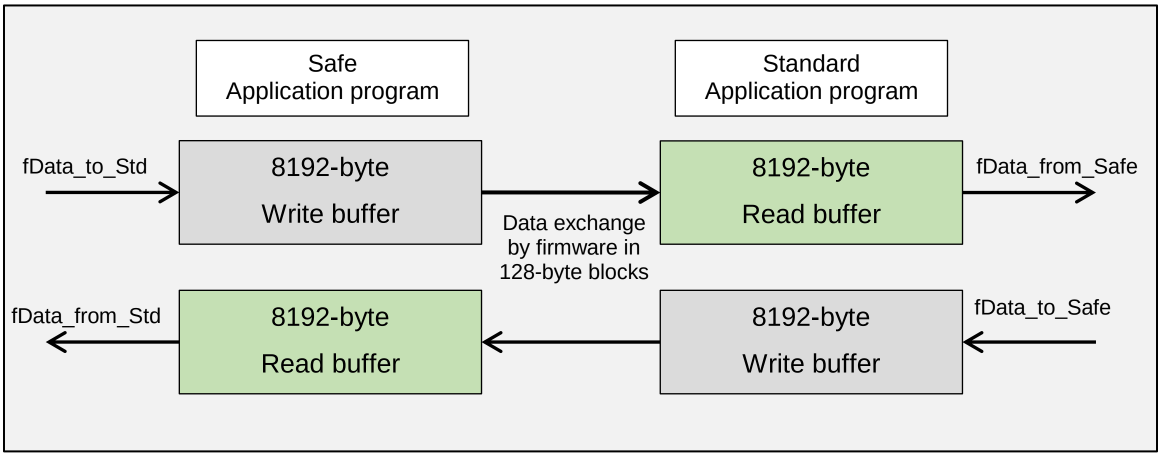
8.1.13 Functions for accessing the Shared Memory

Two 8192 Byte memory areas ("Shared Memory") each are available for the data exchange between the Safe Application program and the Standard Application program.

The Safe Application program writes the data into its write buffer via the "fData_to_Std“ function (library "DFS_Safe“), and reads the data from its read buffer via the "fData_from_Std“ function (library "DFS_Safe“).

The Standard Application program writes the data into its write buffer via the "fData_to_Safe“ function (library "DFS_Std“) and reads the data from its read buffer via the "fData_from_Safe“ function (library "DFS_Std“).

The firmware copies the data in the write buffer of the Safe Application program into the read buffer of the Standard Application program. The data in the write buffer of the Standard Application program is also copied to the read buffer of the Safe Application program. 128 byte blocks are copied together, i.e. it is ensured that these 128 byte blocks contain consistent data.



8.1.13.1 fData_from_Std

Classification:

Function; non-reactive

Function:

Function for reading the read buffer of the Safe Application program.

Description:

This function is used to read the read buffer of the Safe Application program. The buffer has an 8192-byte memory area.

Function inputs:

Name	Data type	Description
pData	POINTER TO BYTE	Address of the memory area to which the data from the shared memory is copied.
OffsetSHM	WORD	Offset in the shared memory from which onwards the data is copied.
max_size	WORD	Number of bytes to be copied

Function outputs:

Name	Data type	Description	
	WORD	0	OK
		2	"pData“ invalid or memory space in the shared memory exceeded (OffsetSHM + max_size must not exceed 8192)

8.1.13.2 fData_to_Std

Classification:

Function; non-reactive

Function:

Function for writing the write buffer into the Safe Application program.

Description:

This function is used to write into the write buffer of the Safe Application program. The buffer has an 8192-byte memory area.

Function inputs:

Name	Data type	Description
pData	POINTER TO BYTE	Address of the memory area from which the data is copied to the shared memory.
OffsetSHM	WORD	Offset in the shared memory from which onwards the data is copied.
size	WORD	Number of bytes to be copied

Function outputs:

Name	Data type	Description	
	WORD	0	OK
		2	"pData“ invalid or memory space of the shared memory exceeded (OffsetSHM + size must not exceed 8192)

8.1.14 System functions

8.1.14.1 SysGetOperationMode

Classification:

Function; safe

Function:

Function for querying the current operating mode of the control system.

Description:

This function is used to determine the current operating mode of the control system.

The return value can have the following values of the enumeration "SYSOPMODE“:

- OPMODE_NORMAL: normal, fault-free operation
- OPMODE_FAILURE: a simple error has been detected
- OPMODE_FAILSAFE_IO: a serious error has been detected

The enumeration value "OPMODE_UNDEFINED“ is not an operating mode. It is used to distinguish the initialization value of variables of the SYSOPMODE type from valid operating modes. This function cannot return the "OPMODE_FAILSAFE_STOP“ mode, because the application program is no longer executed in this state.

For a detailed description of the operating modes refer to Chapter 6.1.

Function outputs:

Name	Data type	Description
	SYSOPMODE	Current operating mode of the control system.

8.1.14.2 SysLogRead

Classification:

Function; safe

Function:

This function can be used to read entries from the system logbook.
The function returns the entries of active errors and warnings since system startup.

Description:

This function is used to read out an entry from the system logbook. The oldest entry which has not yet been read out with this function is always returned.

The system logbook entry is copied to the application program variable whose address is transferred via the function parameter "pEntry“. The variable must be of the "SYSLOG_ENTRY“ type. The system logbook entry contains the following information:

- udiTimeStamp: Time stamp of the entry (in milliseconds since the start of the controller)
- eErrorCode: 32-bit value specifying the cause of the entry

For more detailed information on the system logbook refer to Chapter 7.8.2.

Function inputs:

Name	Data type	Description
pEntry	POINTER TO SYSLOG_ ENTRY	Address of the "SYSLOG_ENTRY“ structure into which the entry read out from the system logbook is written. The memory must be reserved by the application program.

Function outputs:

Name	Data type	Description	
	SYSLOGRES	SYSLOGRES _OK	OK (0) The last entry was read out of the system logbook buffer.
		SYSLOGRES_NEXT	NEXT (1) There is at least one more entry in the system logbook buffer.
		SYSLOGRES_ERR_EMPTY	EMPTY (-1) No unread entry is present in the system logbook.
		SYSLOGRES_ERR_PARAMETER	Invalid parameter "pEntry“ (-2)

8.1.14.3 SysRestart

Classification:

Function; safe

Function:

Function for triggering a restart of the control system.

Description:

This function is used to trigger a restart of the control system. The function no longer returns to the application program.

Function outputs:

Name	Data type	Description
	ERROR	Irrelevant

8.1.14.4 SysSetOperationMode

Classification:

Function; safe

Function:

Function for switching the operating mode of the control system.

Description:

This function is used to switch the operating mode of the control system. The following values of the enumeration "SYSOPMODE“ are permissible for the "sOpMode“ parameter:

- OPMODE_FAILSAFE_IO: All outputs of all controller modules are de-energized. The application program continues operating. The communication interfaces continue to be processed.
- OPMODE_FAILSAFE_STOP: The control system switches to its safe state: All outputs of all controller modules are de-energized. The execution of the Application Program is aborted. In this case, the function call of "SysSetOperationMode“ no longer returns to the application program.

For a detailed description of the operating modes refer to Chapter 6.1.

Function inputs:

Name	Data type	Description
sOpMode	SYSOPMODE	Operating mode to which the control system is to switch Only the values OPMODE_FAILSAFE_IO and OPMODE_FAILSAFE_STOP are valid.

Function outputs:

Name	Data type	Description	
	ERROR	NO_ERROR	OK
		WRONG_PARAMETER	"sOpMode“ parameter invalid

8.1.14.5 SysVIQShutdown

Classification:

Function; safe

Function:

Function for opening the second shutdown paths.

Description:

This function is used to open the second shutdown path of a controller module output group. This means that all the outputs of this output group are de-energized. Once opened, the Application Program cannot shut a second shutdown path. It is only shut upon the next start-up of the control system.

The output group whose second shutdown path is to be opened is addressed via the "eLayer“ and "udiVIQNumber“ parameters. The controller module is selected via "eLayer“. The "udiVIQNumber“ represents the current number of output groups on a controller module (1 corresponds to VIQ1, 2 corresponds to VIQ2 etc.).

Function inputs:

Name	Data type	Description
eLayer	LAYER	This enumeration is used to select from which controller module a second shutdown path is to be opened.
udiVIQNumber	UDINT	Number of the second shutdown path (1 – 4) on which the controller module is to be opened.

Function outputs:

Name	Data type	Description	
	ERROR	NO_ERROR	OK
		WRONG_PARAMETER	Invalid parameter "eLayer“ or "udiVIQNumber“

8.1.14.6 SysGetTimestampsUs

Classification:

Function; safe

Function:

Function for querying the current time stamp in microeconds.

Description:

This function is used to query the current time stamp in microseconds. The time stamp can be used to evaluate or create the time stamp of the input or output data (see Chapter 4.4).

Be aware that the time stamp is a 32-bit value, which means that an overflow occurs after approximately 71 minutes.

Function outputs:

Name	Data type	Description
	UDINT	Current time stamp in microseconds.

8.1.14.7 SetUserOperationTime

WARNING

Overflow of the millisecond counter

If the maximum uninterrupted operating time is set > 49 days, dangerous errors can occur due to the overflow of the millisecond timer and cause material damage or even personal injury.

1. Additional precautions must be taken to handle the timer overflow!
2. The RTC function contained in the supplied "Standard" library must not be used with a power-on time > 49 days, since this function block incorrectly handles the millisecond overflow.

Classification:

Function; safe

Function:

Function for setting the maximum uninterrupted operating time (tReboot) in days. The function must be called within the first hour after switching on the device. The default value of the maximum uninterrupted operating time is 1 day.

Description:

With this function the maximum uninterrupted operation time (tReboot) in days can be set. The time must have a value of 1 day < tReboot <= 90 days.

It must be ensured in the application program that the millisecond timestamp is a 32-bit value, i.e. that an overflow occurs after approx. 49.6 days.

Function inputs:

Name	Datentyp	Description
Days_of_OperationTime	UDINT	The value sets the maximum uninterrupted operation time in days 1 < value <= 90.

Function outputs:

Name	Datentyp	Description
	BOOL	Indicates whether the call was successful. FALSE ➔ new value is not set TRUE ➔ new value is set

8.1.15 Miscellaneous

8.1.15.1 CalcCrc64

Classification:

Function; safe

Function:

Function for calculating a 64-bit CRC checksum.

Description:

This function is used to calculate a 64-bit CRC checksum for the memory area. The generator polynominal

$$x^{64} + x^4 + x^3 + x + 1$$

is used for calculation.

Function inputs:

Name	Data type	Description
pbData	POINTER TO BYTE	Address of the data area whose CRC checksum is to be calculated.
udiLen	UDINT	Length of the data area for the CRC calculation in bytes.
uliStart	ULINT	Initialization value of the CRC value to be calculated.
puliCrc64	POINTER TO ULINT	Address of the storage location into which the calculated CRC value is written.

Function outputs:

Name	Data type	Description	
	ERROR	NO_ERROR	OK
		WRONG_PARAMETER	Invalid parameter "pbData“, "udiLen“, or "puliCrc64“ .

8.2 DFS_Std

The "DFS_Std“ CODESYS library contains functions or function blocks which may be used in the Standard Application program of ***digsy***®fusion S.

The names, functions and transfer parameters of these functions (function blocks) are basically identical to those in the "DFS_Safe“ library. For this reason this section only provides an overview of the scope of functions of the library. For more detailed information refer to the Chapters describing the "DFS_Safe“ library.

8.2.1 Functions for directories

The following function blocks are available for accessing the directories of the file system:

Name	Type	Description
DirClose	FB	Function block for closing a directory
DirCreate	FB	Function block for creating a directory
DirList	FB	Function block for reading directory entries
DirListFirst	FB	Function block for opening a directory
DirRemove	FB	Function block for deleting directories
DirRename	FB	Function block for renaming a directory

For a more detailed description refer to Chapter 8.1.3.

8.2.2 General functions for the file system

The following functions are available for formatting drives and querying file system properties:

Name	Type	Description
DriveFormat	FB	Function block for formatting a data carrier
GetProperty	FUN	Function for reading out file system properties

For a more detailed description refer to Chapter 8.1.4.

8.2.3 Functions for files

The following function blocks provide access to the files:

Name	Type	Description
FileClose	FB	Function block for closing a file
FileDelete	FB	Function block for deleting a file
FileEOF	FB	Function block for querying whether the file end has been reached
FileFlush	FB	Function block for updating a file
FileGetPos	FB	Function block for determining the current position of the file pointer
FileGetSize	FB	Function block for determining the current file size
FileOpen	FB	Function block for opening a file
FileRead	FB	Function block for reading a file
FileRename	FB	Function block for renaming a file

Name	Type	Description
FileSetAttribute	FB	Function block for changing file attributes
FileSetPos	FB	Function block for setting the file pointer
FileWrite	FB	Function block for writing a file

For a more detailed description refer to chapter 8.1.5.

8.2.4 Functions for sockets

The following functions provide network access:

Name	Type	Description
GetEthLinkState	FB	Function block for querying the connection status of the Ethernet interface
SocketCreate	FB	Function block for creating a socket
SocketClose	FB	Function block for closing a socket
SocketAccept	FB	Function block for accepting a TCP connection of a remote client
SocketBind	FB	Function block for linking a socket address to a socket
SocketConnect	FB	Function block for connecting to a TCP server
SocketGetError	FB	Function block for querying the error code of the socket function called up last
SocketGetOption	FB	Function block for querying socket properties
SocketInetAtoN	FB	Function block for converting IP addresses
SocketIoctl	FB	Function block for executing an IOCTL command
SocketListen	FB	Function block for opening a port which accepts connection requests from TCP clients
SocketRecv	FB	Function block for reading the receive data from a TCP connection
SocketRecvFrom	FB	Function block for reading the receive data from a UDP connection
SocketSelect	FB	Function block for checking sockets for I/O events
SocketSend	FB	Function block for transmitting data via a TCP connection
SocketSendTo	FB	Function block for transmitting data via a UDP connection
SocketSetOption	FB	Function block for defining socket properties

For a more detailed description refer to Chapter 8.1.6.

8.2.5 CAN LISTEN function

Analogous to the "CAN Listen function" described in Chapter 8.1.9, this function is also available for the Standard Application program.

If the assignment of a CAN interface in the Safe Application program was set to "Safe_App (Std_App listen)“, the CAN receive messages are also saved to a separate buffer which the Standard Application program can access via the following functions:

Name	Type	Description
CanListenMask	FUN	Function for defining a filter rule for the CAN Listen function
CanListenRec	FUN	Function for reading out the CAN Listen buffer

For a detailed description of the functions refer to Chapter 8.1.9.

8.2.6 Functions for the serial interface

In the "Configuration" tab of the Device Editor of the Safe Application program, you can define whether the serial interface RS232-2 is assigned to the Safe Application program or the Standard Application program:

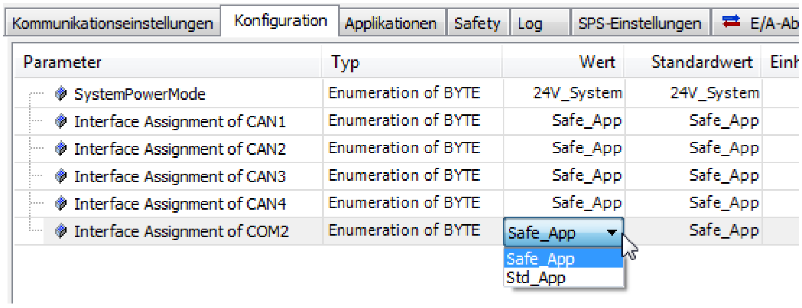


Figure 8-2: Configuration of the serial interface assignment

If the interface is assigned to the Standard Application program, you can access the interface via the following functions:

Name	Type	Description
Init_RS232	FUN	Function for initializing the serial interface RS232-2
Rec_RS232	FUN	Function for reading the receive data of the serial interface RS232-2
Tra_RS232	FUN	Function for transmitting data via the serial interface RS232-2

For a detailed description refer to Chapter 8.1.12.

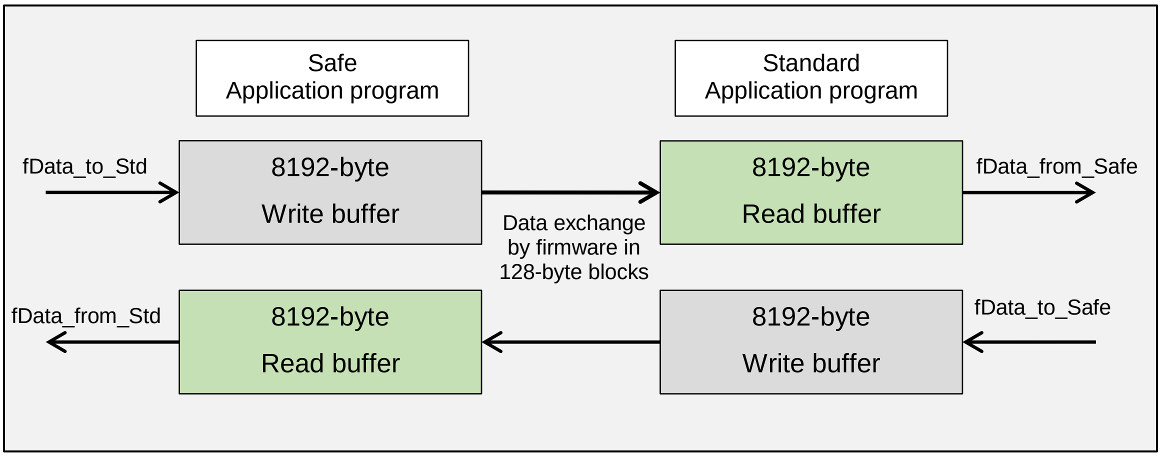
8.2.7 Functions for accessing the Shared Memory

Two 8192 Byte memory areas ("Shared Memory") each are available for the data exchange between the Safe Application and the Standard Application program.

The Safe Application program writes the data into its write buffer via the "fData_to_Std“ function (library "DFS_Safe“), and reads the data from its read buffer via the "fData_from_Std“ function (library "DFS_Safe“).

The Standard Application program writes the data into its write buffer via the "fData_to_Safe“ function (library "DFS_Std“), and reads the data from its read buffer via the "fData_from_Safe“ function (library "DFS_Std“).

The firmware copies the data in the write buffer of the Safe Application program to the read buffer of the Standard Application program. The data in the write buffer of the Standard Application program is also copied to the read buffer of the Safe Application program. 128 byte blocks are copied together, i.e. it is ensured that these 128 byte blocks contain consistent data.



8.2.7.1 fData_from_Safe

Classification:

Function

Function:

Function for reading the read buffer of the Standard Application program.

Description:

This function is used to read the read buffer of the Standard Application program. The buffer has an 8192-byte memory area.

Function inputs:

Name	Data type	Description
pData	POINTER TO BYTE	Address of the memory area to which the data from the shared memory is copied.
OffsetSHM	WORD	Offset in the shared memory from which onwards the data is copied. Must be a multiple of 128.
max_size	WORD	Number of bytes to be copied

Function outputs:

Name	Data type	Description	
	WORD	0	OK
		2	The "pData“ or "OffsetSHM“ parameter is invalid or the memory space in the shared memory has been exceeded (OffsetSHM + max_size must not exceed 8192)

8.2.7.2 fData_to_Safe

Classification:

Function

Function:

Function for writing into the write buffer of the Standard Application program

Description:

This function is used to write into the write buffer of the Standard Application program The buffer has an 8192-byte memory area.

Function inputs:

Name	Data type	Description
pData	POINTER TO BYTE	Address of the memory area from which the data is copied to the shared memory.
OffsetSHM	WORD	Offset in the shared memory from which onwards the data is copied. Must be a multiple of 128.
size	WORD	Number of bytes to be copied

Function outputs:

Name	Data type	Description	
	WORD	0	OK
		2	The "pData“ or "OffsetSHM“ parameter is invalid or the memory space in the shared memory is exceeded (OffsetSHM + size must not exceed 8192)

8.3 DFS-P Safe

The CODESYS library "DFS-P_Safe" contains functions or function blocks which may be used in the ***digsy***®fusion S-P in the safe user program.

The functions (function blocks) are for the most part identical in name, functionality and transfer parameters with those of the "DFS_Safe" library. For this reason, only the differences are explained in this section and the corresponding chapters of the "DFS_Safe" library are referenced for the identical functions.

8.3.1 Functions for accessing the shared memory

For the exchange of data between the secure application program of the SCM-Eth-SL and the standard application program of the GCM-P, there is a memory area that can be written on each board and a memory area that can be read from. The memory areas are each 7680 bytes in size. These memory areas are referred to as "shared memory" in the following, based on the data exchange on the SCM Ethernet, although it is actually not a shared memory.

The data of the write buffer of the SCM-Eth-SL is copied by the firmware into the read buffer of the GCM-P and vice versa. Write and read buffers are divided into 16 data blocks of 480 bytes each, with the 480-byte blocks being copied contiguously. This means it is ensured that these 480-byte blocks have consistent data.

In addition, there is still a 480-byte data block for the read and write direction, which is transmitted with a higher priority than the remaining 16 data blocks.

8.3.1.1 SHM_FastRead

Classification:

Function; non-reactive

Function:

Function for reading the prioritized shared memory data block.

Description:

With this function, the prioritized shared memory data block can be read. The memory area is a maximum of 480 bytes. However, fewer bytes can be read. However, it is always read from the beginning of the data block.

Function inputs:

Name	Data type	Description
pbData	POINTER TO BYTE	Address of the memory area into which the data is copied from the shared memory.
wLen	WORD	Number of bytes to be copied (1 - 480)

Function outputs:

Name	Data type	Description	
	DINT	> 0	OK: Number of copied bytes
		-1	parameter „pbData“ invalid
		-2	parameter „wLen“ invalid

8.3.1.2 SHM_FastWrite

Classification:

Function; non-reactive

Function:

Function for writing the prioritized shared memory data block.

Description:

With this function, the prioritized shared memory data block can be written. The memory area is a maximum of 480 bytes. It is also possible to write fewer bytes. However, it is always written from the beginning of the data block.

Function inputs:

Name	Data type	Description
pbData	POINTER TO BYTE	Address of the memory area from which the data is copied to the shared memory.
wLen	WORD	Number of bytes to be copied

Function outputs:

Name	Data type	Description	
	DINT	> 0	OK: Number of copied bytes
		0	No data was copied because the data block has not been transferred since the last change.
		-1	parameter „pbData“ invalid
		-2	parameter „wLen“ invalid

8.3.1.3 SHM_Read

Classification:

Function; non-reactive

Function:

Function for reading the shared memory.

Description:

With this function, the shared memory can be read. The memory area is 7680 bytes in size. It is organized in 16 data blocks of 480 bytes each, with a 480-byte data block always being transmitted consistently.

Function inputs:

Name	Data type	Description
pbData	POINTER TO BYTE	Address of the memory area into which the data is copied from the shared memory.
wOffset	WORD	Offset in the shared memory from which the data is copied.
wLen	WORD	Number of bytes to be copied

Function outputs:

Name	Data type	Description	
	DINT	> 0	OK
		-1	parameter „pbData“ invalid
		-2	parameters „wLen“ or „wOffset“ invalid ((wOffset + wLen) must be less than or equal to 7680)

8.3.1.4 SHM_Write

Classification:

Function; non-reactive

Function:

Function for writing the shared memory.

Description:

With this function, the shared memory can be written. The memory area is 7680 bytes in size. It is organized in 16 data blocks of 480 bytes each, with a 480-byte data block always being transmitted consistently.

Function inputs:

Name	Data type	Description
pbData	POINTER TO BYTE	Address of the memory area from which the data is copied to the shared memory.
wOffset	WORD	Offset in the shared memory from which the data is copied.
wLen	WORD	Number of bytes to be copied

Function outputs:

Name	Data type	Description	
	WORD	> 0	OK
		0	No data was copied because the data block has not been transferred since the last change.
		-1	parameter „pbData“ invalid
		-2	parameters „wLen“ or „wOffset“ invalid ((wOffset + wLen) must be less than or equal to 7680)

8.3.2 System-Functions

8.3.2.1 SysGetGcmpState

Classification:

Function; non-reactive

Function:

Function to request the system status of the GCM-P(L)

Description:

This function can be used to request the system state of the GCM-P(L). The return value of the function can only get the following values of the enumeration:

GCMP_SYSSTATE_UNDEF, GCMP_SYSSTATE_RUN and GCMP_SYSSTATE_SHUTDOWN.

The remaining values represent internal system states.

The function is intended to ensure a safe shutdown of the ***digsy***®$_{\text{fusion}}$ S-P(L). A possible sequence could be the following:

- The SCM sets the power hold flag
- SCM and GCM-P(L) monitor the IPON signal. If a falling edge is registered at IPON, both boards perform their "clean-up" operations
- The GCM-P(L) is shut down with the "shutdown" command
- The SCM monitors the state of the GCM-P(L) with the "SysGetGcmpState" function. If the state of the GCM-P(L) assumes the value "GCMP_SYSSTATE_SHUTDOWN", the ***digsy***®$_{\text{fusion}}$ S-P(L) can be switched off by resetting the power hold flag

Function outputs:

Name	Data type	Description	
	GCMP_SYSSTATE	GCMP_SYSSTATE_UNDEF	Initial value. The application on the GCM-PL has not yet started.
		GCMP_SYSSTATE_RUN	The application on the GCM-PL has started and has activated UBB communication with the SCM.
		GCMP_SYSSTATE_SHUTDOWN	The GCM-PL has been shut down.

8.4 FUSIONx01

The library FUSIONx01 provides safe function blocks (FB) with which two-channel input information can be compared with each other.

The following function blocks are currently available (no guarantee of completeness).

- SF_Compare2WORD
- SF_Compare2BOOL
- SF_NAMUR_Sensor
- SF_Compare2WordReverse
- SF_Compare2WORD_Circular
- SF_Compare2BOOLReverse
- SF_Compare2INT
- SF_Compare2DWORD

Further information can be found in manual “04-68520_Fusionx01_Manual_E.pdf”.

8.5 Lib_digsyfusion_Util_Safety

The library Lib_digsyfusion_Util_Safety provides safe function blocks (FB) with which the necessary checking of important system parameters is simplified.
The following function blocks are currently available (no guarantee of completeness):

- SF_Monitoring_IO_Valid
- SF_Monitoring_VIQ_Valid
- SF_Monitoring_Temperature
- SF_Switching_QD_Type_A_B
- SF_Monitoring_Analog_Input_10V
- SF_Monitoring_Analog_Input_32V
- SF_Monitoring_Analog_Input_20mA
- SF_Monitoring_Current_Output
- SF_Monitoring_URef
- SF_Monitoring_VIQ_total_current_Typ_A
- SF_Monitoring_VIQ_total_current_Typ_B
- SF_Monitoring_OverCurrent_Qx_y
- SF_Monitoring_VQ_Sense
- SF_Unpack_Syslog_Entry

Further information can be found in manual “04-68512-XXXXXX_digsy-fusion-S_Util_Safety_Manual_E.pdf” (newest version).

9 File System

9.1 Purpose of the file system

A file system can be used to perform various control engineering tasks in a simple manner.

In many cases data is required for setting machine parameters. This data is integrated into the control system during the manufacturing process, read out by the application program at runtime – depending on the system state – and evaluated accordingly. Example: the payload tables for mobile cranes.

A large data volume can be created during the control process and needs to be recorded for subsequent evaluation outside the controller.

A file system is a suitable and flexible means for performing these kinds of tasks in a simple and systematic manner.

9.2 Properties

9.2.1 Scope of application

The file system can be used:

- via the Safe Application program,
- via the Standard Application program.

The function blocks and functions are located in the following libraries:

- *DFS_Safe.Lib* for calls from the Safe Application program,
- *DFS_Std.Lib* for calls from the Standard Application program.

Note

Differences between *DFS_Safe.Lib* and *DFS_Std.Lib*

The call-up interfaces of both libraries are identical, but:

- The FBs and FUNs from the safe context can be called up immediately from the safe context on the system side,
- The FBs and FUNs from the standard context cannot be called up immediately on the system side. It must be ensured on the system side that no memory violations can occur during their execution.

For this reason, the programmer must always make sure that:

- only functions from the *DFS_Safe.Lib* library are used from the safe context
- only functions from the *DFS_Std.Lib* library are used from the standard context.

9.2.2 Data security in case of a power loss

The file system of ***digsy***®fusion provides 100% data loss security in case of a power loss. To ensure this, an earlier version of a file is only completely replaced by a later version once the later file version has been completely written into the file system. The double storage of a File Allocation Table (FAT) and the automatic creation of a CRC of the file contents when closing a file ensure file consistency.

	CAUTION
	Data security in the Safe Application program The term “Data security in case of power loss” – used in the context of the file system – does not relate to the Safe Application program! The file system cannot ensure that the data from a file used in the Safe Application program in the safe context is correct in terms of functional safety. 1. When using data from a file in the Safe Application program in the safe context, it must be ensured that the data is read from the correct file and that the data is correct. 2. To ensure this, e.g. the name can be repeated as a data string within the file concerned. It can thus be verified in the Safe Application program whether the correct file has been read. 3. Furthermore, a CRC can be mapped within the file concerned e.g. via the complete data or a data file section (e.g. a data record). A consistency check can thus be performed for the data or data file section in the Safe Application program. 4. The application programmer is responsible for the correct use of the file system in the Safe Application program. Depending on the individual application, the application programmer must apply appropriate methods in order to ensure that all the functional safety rules are observed!

9.2.3 Drives

Two drives are available for the file system:

- Drive “A:” – External memory (USB memory stick)
- Drive “B:” – Internal memory (NOR Flash)

9.2.3.1 Drive A: external memory

The external memory, i.e. a USB memory stick, can be plugged into the ***digsy***®fusion housing; its storage volume can be variable. Maximum permissible storage volume of a USB memory stick: 4GB.

The external memory can be formatted via the FAT12, FAT16 or FAT32 file systems. FAT32 is recommended for USB memory sticks.

9.2.3.2 Drive B: Internal memory

The internal memory is fixedly integrated into the control system; its storage volume cannot be changed (16MB Flash memory). The memories available for the file system have a capacity of almost 16MB.

The internal memory is formatted in a device-specific format.

CAUTION

A minimum of 100,000 erase/write cycles is guaranteed for the electronic Flash memory of *digsy*$_{fusion}$

An excessive number of write operations into a Flash memory can lead to malfunction. The number of write cycles can be extended by various measures, e.g. Wear Leveling (implemented internally).

1. We strongly recommend that you only execute file write operations upon the shutdown of the machine.
2. The number of erase/write cycles should not exceed 100,000.
3. We strongly recommend that you estimate the number of erase/write cycles over the service life of the machine.
   Example: A machine is operated every day and switched on and off 10 times. If files are only written upon shutdown, 365 x 10 = 3650 save operations would be performed per year. The memory would have a service life of 100,000 / 3,650 = 27.4 years.
4. When writing files onto Drive B: in the Safe Application program in the safe context, the application programmer must always take appropriate measures for the individual application in order to ensure that the functional safety rules are observed!

We recommend you use the VIM in combination with IPON and PWR_HOLD. This ensures that open files are closed and saved securely before the machine is shut down!

9.2.4 File system FAT

When formatting a drive various formats can be selected for the File Allocation Table (FAT).

The table illustrates the basic properties of the various FAT formats:

Format	File name	Properties
FAT12	<Name>.<Extension> = 8.3 characters	Max. number of file or directory entries: 224 in the root directory Max. partition size: 16MB
FAT16	<Name>.<Extension> = 8.3 characters	Max. number of file or directory entries: 512 in the root directory Max. partition size: 2GB
FAT32	<Name>.<Extension> = 256 characters	Max. number of file or directory entries: 2^{28} in the root directory Max. partition size: 2TB Max. file size: 4GB

9.2.5 *digsy*$_{fusion}$ file system

Property	Drive “A:”	Drive “B:”
SafeFLASH file system	No	Yes
Max. nesting depth directories	Only restricted by the path name length	Only restricted by the path name length

Property	Drive "A:"	Drive "B:"
Max. number of simultaneously opened files	2	2
Rules for file and path names	Max. path name length: 65 characters Max. file name length: 63 characters If file names longer than 13 characters are used, the number of file and directory entries is reduced accordingly. Permissible characters: "A"…"Z", "a"…"z", "0"…"9", "-_*!?/.$%#@" Forbidden characters: "+,;[=]"	
Max. path name length	65 characters	65 characters
Max. file name length	63 characters	63 characters
Max. file size	FAT12: 16 MB @ 4 kB Cluster 32 MB @ 8 kB Cluster FAT16: 2 GB @ sector size 512 FAT32: 4 GB	16MB
Flash: Max. number of write access operations	-	100.000
Verifying the file contents via CRC on the system side	No	Yes
File attributes	Read Only, Hidden, System, Archive	Read Only, Hidden, System, Archive
Max. open files	3	3
File time stamp	No time stamp	No time stamp
Wildcards	Yes ("*" + "?")	Yes ("*" + "?")
UniCode16 in file names	No	No
File names case-sensitive	No	No
Hardware Write Protect	Depending on the USB stick	No
Verify function in CODESYS	No	No

9.3 Functions

Frequent file operations can be executed via the application program. Application programmers can use the following functions or function blocks from the DFS_Safe and DFS_Std libraries in the application program:

Function Blocks / Functions	Properties
File directory services:	
DirClose (FB)	Terminates access to a specific file directory
DirCreate (FB)	Creates a subdirectory in the standard directory
DirListFirst (FB)	Opens a file directory and reads the first directory entry
DirList (FB)	Reads further directory entries from a file directory
DirRemove (FB)	Removes a file directory
DirRename (FB)	Renames a file directory
File services:	
FileClose (FB)	Terminates access by closing a file
FileDelete (FB)	Deletes a file
FileEOF (FB)	Checks whether the file pointer is at the end of the file
FileFlush (FB)	Completes a file content write operation without closing the file
FileGetPos (FB)	Reads the file pointer of an open file
FileGetSize (FB)	Determines the size of a file
FileOpen (FB)	Opens an existing file or creates a new file
FileRead (FB)	Reads from an open file
FileRename (FB)	Renames a file
FileSetAttribute (FB)	Sets file or directory attributes
FileSetPos (FB)	Sets the file pointer of an open file
FileWrite (FB)	Writes into an open file
Other services:	
DriveFormat (FB)	Formats a drive
GetProperty (FUN)	Reads information concerning the file and directory system, and its properties and handling

A FileCopy function block does not exist. It can be emulated via the existing function blocks.

A FileVerify function block does not exist. It can be emulated via the existing function blocks.

For a detailed description of the functions, function blocks and the variable types and parameters assigned, refer to the documents DFS_Safe and DFS_Std published by INTER CONTROL, and the description of the individual functions in the CODESYS ***digsy***®fusion development tools, provided that the corresponding libraries have been loaded.

10 Annex

10.1 *digsy®*fusion – Dimensions

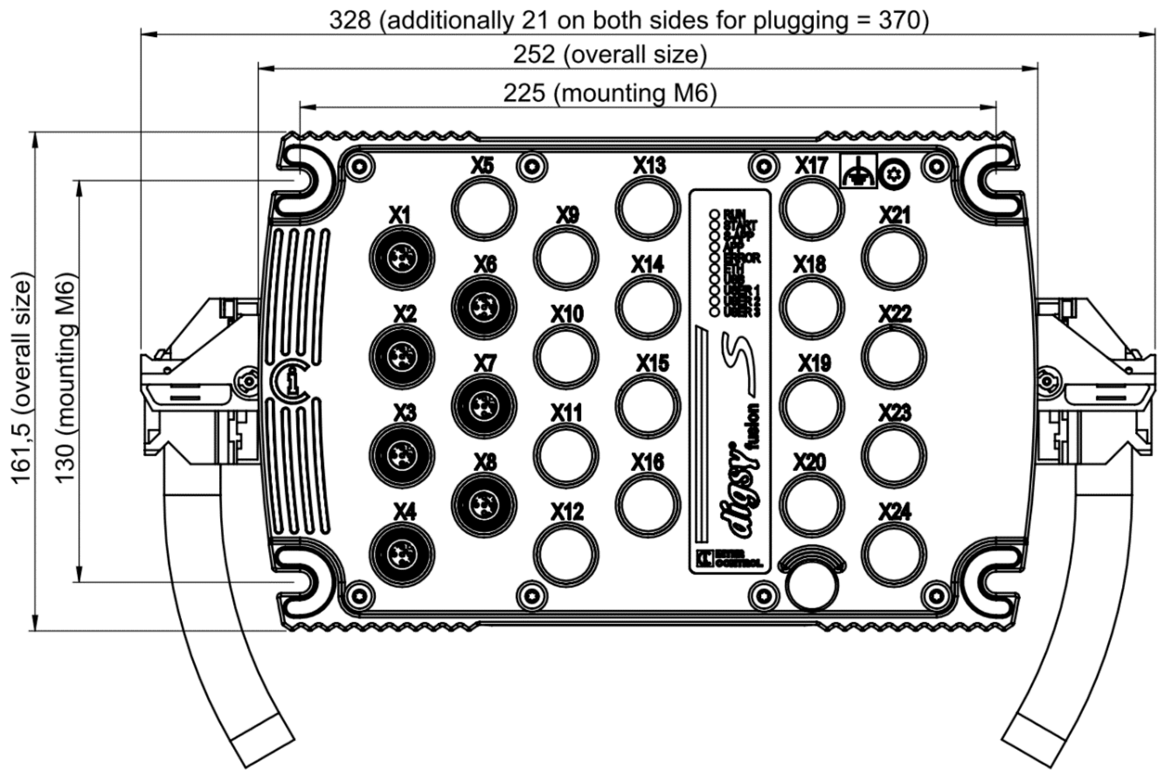


Figure 10-1: Dimensions of the digsy®fusion housing – top view

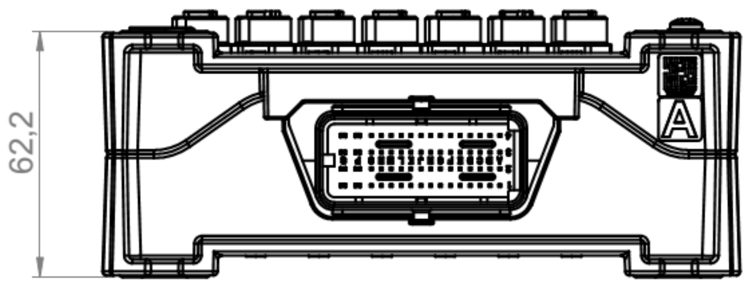


Figure 10-2: Dimensions of the digsy®fusion housing – SMALL – side view

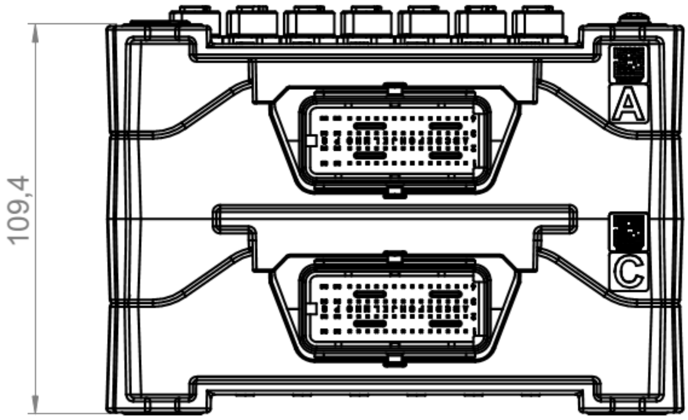


Figure 10-3: Dimensions of the digsy®fusion housing – MEDIUM – side view

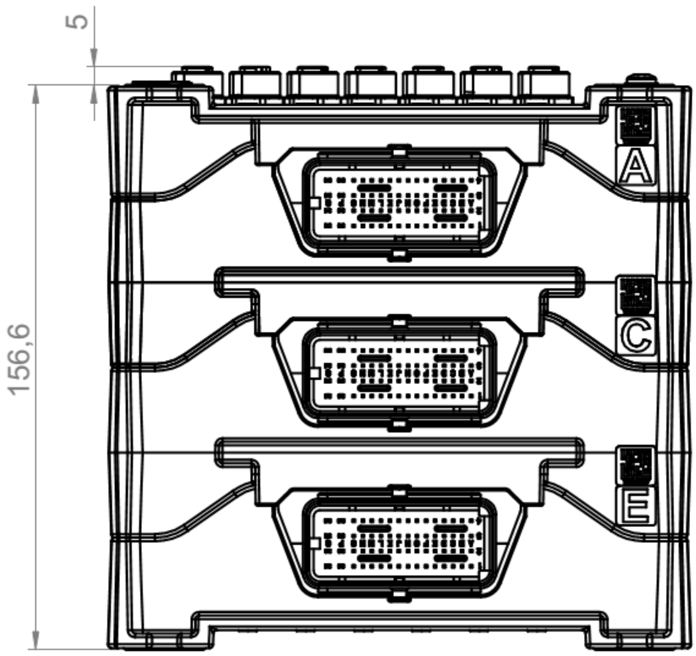


Figure 10-4: Dimensions of the digsy®fusion housing – LARGE – side view

10.2 Rating plate

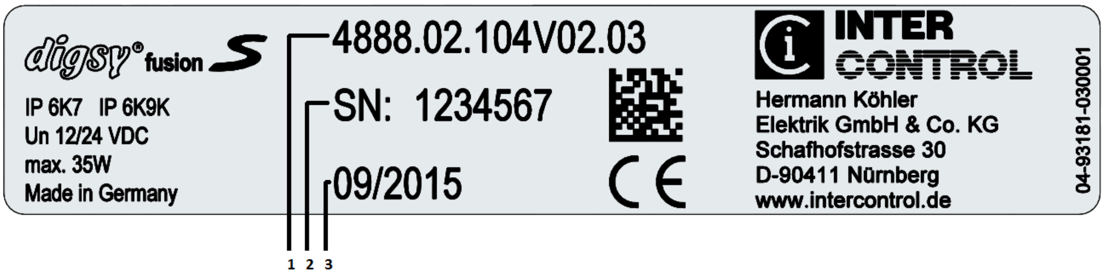


Figure 10-5: Rating plate

1. Versioned product number
2. Serial number
3. Date of manufacture

10.3 Mounting and weight

Housing dimensions (LxWxH)	SMALL: 252 x 162 x 68 MEDIUM: 252 x 162 x 115 LARGE: 252 x 162 x 162 In order to be able to pull and plug the terminal plugs on the sides, an installation space direction of at least 370 mm in longitudinal is required.
Mounting	Using 4 suitable M6 screws, the screw fixing device can be mounted to a mounting level in any direction.
Weight	SMALL: approx. 2 kg MEDIUM: approx. 3.5 kg LARGE: approx. 5 kg

10.4 Maintenance

To get costumer related values please contact the manufacturer.

digsy®fusion S does not need to be serviced by the user within the proof test interval (20 years). In case of malfunction ***digsy***®fusion S must be returned to the manufacturer (INTER CONTROL).

Note

Maintenance

The user is not allowed to open the housing of ***digsy***®fusion S. The warranty will be void in the event of damage caused by failure to observe these safety instructions!

10.5 Pin assignment central plug connector

PIN	PIN NAME	Input Digital +	Input Digital -	Input Count	Input Analog	Output Digital +	Output Digital -	Output PWM	Output PWM IREG	VIX / GND	I/O-Typ	Note
3P	VIQ_1									■		Power Supply Group 1
3B	Q1.1					■		■	■		A1	Safe Out, Group 1, 2,5A
3A	Q1.2					■		■	■		A1	Safe Out, Group 1, 2,5A
4A	Q1.3					■		■	■		A1	Safe Out, Group 1, 2,5A
4B	Q1.4					■		■	■		A1	Safe Out, Group 1, 2,5A
2P	VIQ_2									■		Power Supply Group 2
2A	Q2.1					■		■	■		A3	Safe Out, Group 2, 2,5A
2B	Q2.2					■		■	■		A3	Safe Out, Group 2, 2,5A
1B	Q2.3					■		■	■		A2	Safe Out, Group 2, 4A
1A	Q2.4					■		■	■		A2	Safe Out, Group 2, 4A
2Q	VIQ_3									■		Power Supply Group 3
2O	Q3.1					■		■	■		A3	Safe Out, Group 3, 2,5A
2N	Q3.2					■		■	■		A3	Safe Out, Group 3, 2,5A
1N	Q3.3					■		■	■		A2	Safe Out, Group 3, 4A
1O	Q3.4					■		■	■		A2	Safe Out, Group 3, 4A
1Q	VIQ_4.1									■		Power Supply Group 4
3Q	VIQ_4.2									■		Power Supply Group 4
1J	Q4.1	■				■					B	Safe Out, Std. In, Group 4, 4,0A
1K	Q4.2	■				■					B	Safe Out, Std. In, Group 4, 4,0A
1L	Q4.3	■				■					B	Safe Out, Std. In, Group 4, 4,0A
1M	Q4.4	■				■					B	Safe Out, Std. In, Group 4, 4,0A
2L	Q4.5	■				■					B	Safe Out, Std. In, Group 4, 4,0A
2M	Q4.6	■				■					B	Safe Out, Std. In, Group 4, 4,0A
1C	I1.1	■		■							D	Safe In, Group 1, AB Counter 1A
1D	I1.2	■		■							D	Safe In, Group 1, AB Counter 1B
2C	I1.3	■		■							D	Safe In, Group 1, AB Counter 2A
2D	I1.4	■		■							D	Safe In, Group 1, AB Counter 2B
2E	I1.5	■		■							D	Safe In, Group 1, AB Counter 3A
2F	I1.6	■		■							D	Safe In, Group 1, AB Counter 3B
2G	I1.7	■		■							D	Safe In, Group 1, AB Counter 4A
2H	I1.8	■		■							D	Safe In, Group 1, AB Counter 4B
4F	I2.1	■			■						E	Safe In, Group 2
3F	I2.2	■			■						E	Safe In, Group 2
4G	I2.3	■			■						E	Safe In, Group 2
3G	I2.4	■			■						E	Safe In, Group 2
4H	I2.5	■			■						E	Safe In, Group 2
3H	I2.6	■			■						E	Safe In, Group 2
4J	I2.7	■			■						E	Safe In, Group 2
3J	I2.8	■			■						E	Safe In, Group 2
4K	I2.9	■			■						E	Safe In, Group 2
3K	I2.10	■			■						E	Safe In, Group 2
4L	I2.11	■			■						E	Safe In, Group 2
3L	I2.12	■			■						E	Safe In, Group 2
4M	I2.13	■			■						E	Safe In, Group 2
3M	I2.14	■			■						E	Safe In, Group 2
4N	I2.15	■			■						E	Safe In, Group 2
3N	I2.16	■			■						E	Safe In, Group 2
4C	I3.1	■	■								C	Safe In, Standard In Low, Group 3
3C	I3.2	■	■								C	Safe In, Standard In Low, Group 3
4D	I3.3	■	■								C	Safe In, Standard In Low, Group 3
3D	I3.4	■	■								C	Safe In, Standard In Low, Group 3
4E	I3.5	■	■								C	Safe In, Standard In Low, Group 3
3E	I3.6	■	■								C	Safe In, Standard In Low, Group 3
1P	VIM_1									■		Supply for logic + inputs
4O	IPON_1									■		Connection of terminal 15
3O	GND_1									■		Ground 1 (logic + inputs)
1G	Q_SENS									■		Sensor supply output
1H	IREF									■		Reference current
1E	UREF_1									■		Reference voltage 1
1F	UREF_2									■		Reference voltage 2
2J	GND_A1									■		Ground – analog 1
2K	GND_A2									■		Ground – analog 2
4P	GND_2									■		Ground 2 (leading contact!)
4Q	GND_1									■		Ground 3 (leading contact!)
SUMME	Total: Safe:	36 30	6 0	8 8	16 16	18 18	0 0	12 12	12 12	16	48	

Table 10-1: Pin assignment central plug connector

10.6 MTTFD

The specified MTTFD, PFH and PFH Diag values are calculated for the maximum use of all available IOs.

For a realistic view of the respective implemented safety function, the relevant values can be calculated by Inter Control.

10.6.1 Single-channel usage

Item number	Configuration	MTTFD Single channel	MTTFD Channel A	MTTFD Channel B
1 x SCM / SCM SL		39.3 years	77.1 years	133.0 years
1 x SCM / SCM SL + 1x SIOM		20.5 years	38.6 years	66.5 years
1 x SCM / SCM SL + 2x SIOM		13.8 years	25.7 years	44.3 years
1 x SCM + 3x SIOM		10.4 years	19.3 years	33.2 years

Table 10-2: MTTFD values for single-channel use of all inputs, 85°C inside temperature, 4000m max altitude

Item number	Configuration	PFH	PFH Diag
1 x SCM / SCM SL		6.8802×10^{-7}	2.5041×10^{-7}
1 x SCM / SCM SL + 1x SIOM		1.3718×10^{-6}	4.9716×10^{-7}
1 x SCM / SCM SL + 2x SIOM		2.0573×10^{-6}	7.4392×10^{-7}
1 x SCM + 3x SIOM		2.7445×10^{-6}	9.9068×10^{-7}

Table 10-3: PFH values for single-channel use of all inputs, 85°C inside temperature, 4000m max altitude, 1day maximum uninterrupted operation time

10.6.2 Two-channel usage

Item number	Configuration	MTTFD Single channel	MTTFD Channel A	MTTFD Channel B
1 x SCM / SCM SL		94.4 years	49.0 years	66.9 years
1 x SCM / SCM SL + 1x SIOM		52.1 years	24.5 years	33.5 years
1 x SCM / SCM SL + 2x SIOM		36 years	16.3 years	22.3 years
1 x SCM + 3x SIOM		27.5 years	12.3 years	16.7 years

Table 10-4: MTTFD values for two-channel use of all inputs, 85°C inside temperature, 4000m max altitude

Item number	Configuration	PFH	PFH Diag
1 x SCM / SCM SL		2.8115×10^{-8}	8.3113×10^{-9}
1 x SCM / SCM SL + 1x SIOM		5.5149×10^{-8}	1.3078×10^{-8}
1 x SCM / SCM SL + 2x SIOM		8.7053×10^{-8}	1.7958×10^{-8}
1 x SCM + 3x SIOM		$1{,}2383 \times 10^{-7}$	$2{,}2951 \times 10^{-8}$

Table 10-5: PFH values for two-channel use of all inputs, 85°C inside temperature, 4000m max altitude, 1day maximum uninterrupted operation time

10.6.3 PFH Values for CAN

The PFH values for the CAN interface can be calculated using the following formula:

The residual error rate per hour *Λ* is:

$\Lambda = 3600 \cdot P \cdot v \cdot (m\text{-}1) \cdot 100$

v: safety relevant messages per second

m: number of safety relevant devices = max. 64

P: residual error probability

Figure 10-1: Formular for Residual error rate (Quelle: CiA 304)

The residual error probability is given as $P = (7E\text{-}9)^2 = 4.9E\text{-}17$.

If the calculated residual error rate <1E-8 can be neglected for the overall calculation since it then has less than 1% of the PFH of the overall system.

10.7 Maximum uninterrupted operation time

The specified maximum uninterrupted operating time is calculated for the maximum use of all available IOs.

For a realistic view of the maximum uninterrupted operating time, the relevant value can be calculated by Inter Control.

The user must check whether the parameterized maximum uninterrupted operating time (1-90 days) is sufficient to detect possible errors to the extent required. If necessary, he must use a smaller value for this.

The following table shows the maximum permissible uninterrupted operating time of the possible configurations in relation to the average internal housing temperature.

Konfiguration	**Innentemperatur**		
	40°C	**60°C**	**85°C**
1 x SCM / SCM SL	90 days	90 days	90 days
1 x SCM / SCM SL + 1x SIOM	90 days	90 days	90 days
1 x SCM / SCM SL + 2x SIOM	90 days	90 days	90 days
1 x SCM + 3x SIOM	90 days	90 days	78 days

Table 10-6: Maximum uninterrupted operation time

The CODESYS library "DFS_Safe" contains the function SetUserOperationTime.

This function can be used to set the maximum uninterrupted operation time (tReboot) in days. The default value of the maximum uninterrupted operation time is 1 day.

10.8 List of Tables

10.9 List of Figures

10.10 Declaration of conformity

EC Declaration of Conformity

Manufacturer: INTER CONTROL
Hermann Köhler Elektrik GmbH & Co. KG
Schafhofstraße 30
D-90411 Nürnberg

Product: Programmable 32-bit safety controller *digsy* Fusion **S / S-P**

Product-Numbers:

4888.02.001,	4888.02.101,	4888.02.201,
4888.02.002,	4888.02.102,	4888.02.202,
4888.02.004,	4888.02.104,	4888.02.204
4888.02.005,	4888.02.105,	4888.02.205,
4888.02.007,	4888.02.107,	4888.02.207,
4888.03.301,	4888.03.302,	4888.03.304,
4888.03.321,	4888.03.322,	4888.03.324

The products described above in the form as delivered are in compliance with:

2006/42/EC in conjunction with following Regulations and Standards	DIRECTIVE 2006/42/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 17 May 2006 on machinery, and amending Directive 95/16/EC (NB 0123: TÜV Süd Product Service, D80339 München, Ridlerstr. 57: EC-Type Certificate, No. M6A 16 09 20099 004)
2014/30/EU	DIRECTIVE 2014/30/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 26 February 2014 on the harmonisation of the laws of the Member States relating to electromagnetic compatibility
UN ECE R10 Rev5	REGUALATION 10 OF THE UNITED NATIONS ECONOMIC COMMISSION FOR EUROPE of 16 October 2014: Uniform provisions concerning the approval of vehicles with regard to electromagnetic compatibility
EN ISO 13849 **EN 16590** **ISO 25119**	2015:01 (Cat 3, PL d) 2014:11 (AgPl d) 2018:10 (AgPl d) , NB 0123:TÜV Süd, Certificate No. Z10 020099 0006 Rev.01
ISO 7637	2-2011:03, 3-2016:07
ISO 16750	2-2012:11
IEC 61131-2 **IEC 61326-3**	2-2017:08 1-2017:05
EN 61000	4-2-2009:12, 4-3-2011:04, 4-4-2013:04, 4-5-2014:05, 4-6-2014:08, 4-8-2010:11 4-29: 2001:10
EN 60068	2-1:01-02008:01, 2-02-2008:05, 2-06-2008:10, 2-14-2000:08, 2-27-2008:03, 2-30-2006:06, 2-31-2009-04, 2-64-2009-04
EN 55022 **EN 55016**	2011:12 (CISPR22) : 2011:02 (CISPR16)
EN 13309	2010:12: Construction machinery - Electromagnetic compatibility
EN 50155 **EN 45545-2**	2018:05: Electronic equipment on rail vehicles 2016:06: European railway standard for fire safety
2011/65/EU	DIRECTIVE 2011/65/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 8 June 2011on the restriction of the use of certain hazardous substances in electrical and electronical equipment (ROHS2 directive)

Issuer: INTER CONTROL Hermann Köhler Elektrik GmbH & Co. KG
Place, date: Nuremberg, Feburary 2020

Signature:

ppa. Alexander Holler
General Manager Electronic Dep.

i.A. Martin Lander
Quality Manager Electronic Dep.

- This declaration includes no warranty of properties.
- The safety documentation attached to the product shall be considered in detail.
- The product described above is for use in machines/equipments only. The commissioning is denied until the conformity of the product is approved with commission directive mentioned above.

EG-Konf_DF_S_2020-02_E.docx

10.11 Certificate issued by the German Technical Inspection Agency (TÜV)

ZERTIFIKAT ◆ CERTIFICATE ◆ 認證證書 ◆ СЕРТИФИКАТ ◆ CERTIFICADO ◆ CERTIFICAT

Product Service

CERTIFICATE

No. Z10 020099 0006 Rev. 01

Holder of Certificate: **INTER CONTROL**
Hermann Köhler Elektrik GmbH & Co. KG
Schafhofstraße 30
90411 Nürnberg
GERMANY

Factory(ies): 020099

Certification Mark:

Product: **Safety systems**

Model(s): **digsy® fusion S**

Parameters: Modular programmable system
Supply voltage 8V…32V
Operating temperature -40°C…+80°C

Tested according to: EN ISO 13849-1:2015 (Cat.3, PL d)
EN 16590-1:2014 (AgPl d)
EN 16590-2:2014 (AgPl d)
EN 16590-3:2014 (AgPl d)
EN 16590-4:2014 (AgPl d)
ISO 25119-1:2018 (AgPl d)
ISO 25119-2:2018 (AgPl d)
ISO 25119-3:2018 (AgPl d)
ISO 25119-4:2018 (AgPl d)

The product was tested on a voluntary basis and complies with the essential requirements. The certification mark shown above can be affixed on the product. It is not permitted to alter the certification mark in any way. In addition the certification holder must not transfer the certificate to third parties. See also notes overleaf.

Test report no.: IN87732C

Valid until: 2024-06-27

Date, 2019-06-28 (Peter Weiß)

Page 1 of 1
TÜV SÜD Product Service GmbH • Certification Body • Ridlerstraße 65 • 80339 Munich • Germany

Zertifiziervertrag

Grundlage für die Zertifikatserteilung ist die Prüf- und Zertifizierordnung von TÜV SÜD Product Service.

Mit Erhalt des Zertifikates erkennt der Zertifikatsinhaber die jeweils gültige Fassung der Prüf- und Zertifizierordnung an (www.tuev-sued.de/ps_regulations) und wird somit Partner im Zertifiziersystem von TÜV SÜD Product Service.

Prinzipielle Voraussetzung für die Gültigkeit des Zertifikates:

– Gültigkeit der zitierten normativen Prüfgrundlage(n) ist gegeben

und zusätzlich bei Zertifikaten mit Berechtigung zur Verwendung eines Prüfzeichens bzw. bei Zertifikaten für QM-Systeme:

– Voraussetzungen für vorschriftsmäßige Fertigung werden eingehalten.

– Die Fertigungs- bzw. Betriebsstätten werden regelmäßig überwacht.

Certification contract

Certification is based on the TÜV SÜD Product Service Testing and Certification Regulations.

On receipt of the certificate the certificate holder agrees to the current version of the Testing and Certification Regulations (www.tuev-sued.de/ps_regulations) and thus becomes partner in the TÜV SÜD Product Service Certification System.

Requirements for the validity of the certificate in principle:

– Validity of the quoted test standard(s)

In addition for certificates with the right to use a certification mark and for QM certificates:

– Conditions for an adequate manufacturing are maintained

– Regular surveillance of the facility is performed

Akkreditierungen / Benennungen (Status 14.10.2013) /
Accreditations / notifications (as of 2013-10-14)

Deutschland / Germany

Produktsicherheitsgesetz (ProdSG) /
Product Safety Act (ProdSG)

Europa / Europe

- Niederspannungsrichtlinie 2006/95/EG
- Spielzeugrichtlinie 2009/48/EG
- Richtlinie für aktive medizinische Implantate 90/385/EWG
- Richtlinie für Medizinprodukte 93/42/EWG
- Richtlinie für In-vitro-Diagnostika 98/79/EG
- Richtlinie für Gasverbrauchseinrichtungen 2009/142/EG
- Richtlinie für persönliche Schutzausrüstungen 89/686/EWG
- EMV-Richtlinie 2004/108/EG
- Richtlinie für Sportboote 94/25/EG + 2003/44/EG
- Richtlinie für Maschinen 2006/42/EG
- Richtlinie für Ex-Schutz Geräte 94/9/EG

- Low Voltage Directive 2006/95/EC
- Toys Directive 2009/48/EC
- Directive for Active Implantable Medical Devices 90/385/EEC
- Directive for Medical Devices 93/42/EEC
- Directive on In Vitro Diagnostic Medical Devices 98/79/EC
- Directive for Gas Appliances 2009/142/EC
- Directive for Personal Protective Equipment 89/686/EEC
- EMC Directive 2004/108/EC
- Directive for Recreational Craft 94/25/EC + 2003/44/EC
- Directive for Machinery 2006/42/EC
- Directive for Ex Safe Equipment 94/9/EC

- ENEC Agreement for luminaires, household and IT equipment

USA

- Nationally Recognized Testing Laboratory (NRTL) to 29 CFR 1910.7 by OSHA
- Accredited for FDA 510(k) Third Party Review
- Conformity Assessment Body to the MRA for Medical Devices; FDA QSReg Inspections, FDA 510(k) Third Party Review

Asien-Pazifik Region / Asia Pacific

- Recognized Certification Body to Electrical Products (Safety) Regulation; Hong Kong
- Konformitätsbewertungsstelle / Conformity Assessment Body to the MRA for Medical Devices; Australien / Australia
- Konformitätsbewertungsstelle / Conformity Assessment Body to the MRA for Medical Devices; Neuseeland / New Zealand

Weltweit / Worldwide

- NCB im CB-Scheme des IECEE / NCB in the CB Scheme of IECEE
- ExCB im IECEx-Scheme des IECEE / ExCB in the IECEx Scheme of IECEE
- Zertifizierstellen durch DAkkS akkreditiert DE-ZE-11321-01, DE-ZM-11321-09 und DE-ZM-11321-01. Certification Bodies accredited by DAkkS DE-ZE-11321-01, DE-ZM-11321-09 and DE-ZM-11321-01.

Zertifizierstelle für Produkte / Certification Body for Products • e-mail ps-zert@tuev-sued.de
Zertifizierstelle für Medizinprodukte / Certification Body for Medical Devices • e-mail medical_devices@tuev-sued.de
Kundenservice / Clients Services • Phone +49/89/50 08-42 61 • Fax +49/89/50 08-42 30 • e-mail ps-zert@tuev-sued.de

10.12 Technical data of the housing

Designation	Property
Protection class	Protection class IP6K7/ IP6K9K in accordance with DIN 40050-9:1993
Mechanical protection	Falling object protection Broadband random and guidance (DIN EN 60068-2-64:2009-04): Effective value RMS = 2.2g (22h / axis) Vibration resistance (DIN EN 60068-2-6:2008-10): 1g (20 cycles / axis Shock resistance (DIN EN 60068-2-27:2010-02): 30g, 11 ms, 3 shocks / axis, half-sine 10g, 16 ms, 1000 shocks / axis, half-sine Free fall (DIN EN 60068-2-31:2009-04: 100cm, 5 repetitions
Surface protection	against water, salt water, fuels, oils, greases
Pressure compensation	GORE-TEX membrane
Operating temperature range	-40°C … +80°C in accordance with DIN EN 60068-2-2:2008-05
Air humidity	95% relative humidity at 10°C/80°C over 24h
Further requirements	UL 94 V0, RoHS compliant
Plug systems	Central plug, 64-pin, locking M12 sockets, 4- or 5-pin, a-, b-, d-encoded
Housing dimensions	Small 252.0 x 161.4 x 62.2 (L x W x H, mm) Medium 252.0 x 161.4 x 109.4 (L x W x H, mm) Large 252.0 x 161.4 x 156.6 (L x W x H, mm)
Fastening	see Annex

Table 10-7: Technical data of the housing

10.13 Device data

10.13.1 Scope of application

Scope of application	Description	Req.
	The control system is designed for a maximum altitude of 4,000m.	§7088

10.13.2 Service life

Service life	Description	Req.
	digsy®$_{fusion}$S has a service life of 20 years from the date of manufacture. The proof test interval corresponds to the service life of the control system. The end user is responsible for observing the maximum service life.	§6810

10.13.3 Power consumption

The table below provides a list of power consumption values in idle state (VIM, IPON and VIQ1-VIQ4 connected to the power supply, no interfaces used).

Structure of the control system	Supply voltage	Min.	Typ.	Max.
1 x SCM	8V		2.2W	
	12V		2.4W	
	24V		2.9W	
	32V		3.3W	
1 x SCM + 1 x SIOM/GIOM	8V		3.8W	
	12V		4.0W	
	24V		4.7W	
	32V		5.4W	
1 x SCM + 2 x SIOM/GIOM	8V		5.3W	
	12V		5.5W	
	24V		6.5W	
	32V		7.3W	
1 x SCM + 3 x SIOM/GIOM	8V		6,5W	
	12V		6,6W	
	24V		7,2W	
	32V		7,8W	

Table 10-8: Power consumption digsy fusion S

Structure of the control system	Supply voltage	Min.	Typ.	Max.
1 x SCM SL + 1 x GCM-P	8V		4.8W	
	12V		4.9W	
	24V		5.4W	
	32V		5.7W	
1 x SCM SL + 1 x GCM-P + 1 x SIOM	8V		6.4W	
	12V		6.5W	
	24V		7.2W	

Structure of the control system	Supply voltage	Min.	Typ.	Max.
	32V		7.7W	
1 x SCM SL + 1 x GCM-P + 2 x SIOM	8V		7.9W	
	12V		8.0W	
	24V		9.0W	
	32V		9.6W	
1 x SCM SL + 1 x GCM-P + 3 x SIOM	8V		9.4W	
	12V		9.5W	
	24V		10.8W	
	32V		11.5W	

Table 10-9: Power consumption digsy fusion S-P

Structure of the control system	Supply voltage	Min.	Typ.	Max.
1 x SCM SL + 1 x GCM-P	8V		5,5W	
	12V		5,6W	
	24V		6,1W	
	32V		6,4W	
1 x SCM SL + 1 x GCM-P + 1 x SIOM	8V		7,1W	
	12V		7,2W	
	24V		7,9W	
	32V		8,4W	
1 x SCM SL + 1 x GCM-P + 2 x SIOM	8V		8,6W	
	12V		8,7W	
	24V		9,7W	
	32V		10,3W	

Table 10-10: Power consumption digsy fusion S-PL

Note

Power consumption

The power consumption increases when using e.g. interfaces such as

1. USB
2. Ethernet
3. CAN

Power supplies such as

1. sensor supply
2. reference voltage

or the inputs/outputs

10.13.4 Power supply

The VIM is used to supply the logic and the sensor outputs QSENS, UREF1, UREF2, and IREF. VIQn (n ∈ [1,2,3,4]) is used to supply the various power output groups.

The VIM and the VIQn (n ∈ [1,2,3,4]) are connected to a common ground.

All the ground connections must be wired separately in order to avoid malfunction if one ground connection is lost.

10.13.4.1 Power supply of the logic and electronics

Power supply of the logic and electronics	Note	Min.	Typ.	Max.
VIM	Operating voltage range	+8.0V	+12.0/24.0V	+32.0V
	Overvoltage (permanent)	-60.0V		+60.0V
	Overvoltage detection (OPMODE_FAILSAFE_STOP)	+33.0V	+33.2V	+33.4V
	Overvoltage shutdown	38.5V	40.5V	42.5V
I_VIM (current consumption)[2]	12V (idle, 1x SCM) 24V (idle, 1x SCM)		0.2A 0.12A	

10.13.4.2 Power supply of the power outputs

Power supply of the power outputs	Note	Min.	Typ.	Max.
VIQ1, VIQ2, VIQ3, VIQ4	Operating voltage range	8.0V	12.0 / 24.0V	32.0V
	Overvoltage (permanent)[3]	-60.0V		+60.0V
	Overvoltage shutdown	61.5V	62.0V	62.5V
	Undervoltage shutdown Shutdown threshold			6.,5V
	Minimum voltage for re-activation after undervoltage shutdown	7.0V	7.1V	7.4V
I_VIQ1	Permissible summation current VIQ1[4]		+10A	
I_VIQ2	permissible summation current VIQ2[4]		+10A	
I_VIQ3	permissible summation current VIQ3[4]		+10A	
I_VIQ4	permissible summation current VIQ4[4]		+20A	

[2] For more detailed information refer to Chapter 10.13.3 Power consumption

[3] Reverse voltage protection only in combination with fuse protection of the VIQ cables. See Chapter 4.1

[4] For more detailed information refer to Chapter 0 Safety information and safety requirements – Supply voltage

10.13.4.3 Insulating resistance between supply voltages

Insulating resistance	Note	Min.	Typ.	Max.
VIM – VIQn (n ∈ [1,2,3,4])	Resistance between VIM and VIQn connections	100kΩ		
VIQn – VIQm (n,m ∈ [1,2,3,4], n ≠ m)	Resistance between two VIQn and VIQm connections	40kΩ		

10.13.4.4 IPON input

IPON input	Note	Min.	Typ.	Max.
IPON	Low level			3.0V
	High level	8V		
	Electric strength	-50.0V		+ 60.0V
	Cutoff frequency	70Hz	100Hz	130Hz

10.13.5 Type A outputs

Type A outputs	Note	Min.	Typ.	Max.
Voltage level	OFF state		2.2V	2.3V
	ON state			VIQ
Short-circuit strength	Short circuit against GND		permanent	
	Short circuit against VIM/VIQ		permanent	
Type A leakage current	in OFF state [5] (35ms after switch-off at the latest)	1.0mA	1.1mA	1.2mA
	in the "OFF phase" during PWM operation (concerns types A2 and A3 for HW version < VXX.40.XX)			15.0mA
	in the "OFF phase" during PWM operation (only concerns types A1, A2 and A3 for HW version ≥ VXX.40.XX)			15.0mA
Type A1 switching current	Rated current		2.5A	
	Overcurrent[6]		3.0A	
	Time until overcurrent warning	100ms		250ms
	Time until overcurrent switch-off	900ms		1100ms
	Surge current (t ≤ 10ms)	6.0A		
	Short circuit current (for HW version < VXX.40.XX)			13.5A
	Short circuit current (for HW version ≥ VXX.40.XX)			40,0A

[5] Increased leakage currents may occur sporadically. For more detailed information refer to Chapter 4.6.2.6 Safety information and safety requirements – Type A inputs/outputs

[6] Appropriate measures are initiated automatically if the overcurrent persists for some time (warning, automatic switch-off)

Type A outputs	Note	Min.	Typ.	Max.
Type A2 switching current	Rated current		4.0A	
	Overcurrent[6]		4.8A	
	Time until overcurrent warning	100ms		250ms
	Time until overcurrent switch-off	900ms		1100ms
	Surge current (t ≤ 10ms)	20.0A		
	Short circuit current			40.0A
Type A3 switching current	Rated current		2.5A	
	Overcurrent[6]		3.0A	
	Time until overcurrent warning	100ms		250ms
	Time until overcurrent switch-off	900ms		1100ms
	Surge current (t ≤ 10ms)	6.0A		
	Short circuit current			40.0A
Type A current measurement	Cutoff frequency		3.4Hz	
	Precision[7]		± 1.5%	± 2.0%
	Safety precision[7]			±5.5%
A1 current measurement	Nominal measuring range	0A		2.5A
	Measuring range upper limit	3.050A	3.100A	3.150A
A2 current measurement	Nominal measuring range	0A		4.0A
	Measuring range upper limit	4.830A	4.900A	4.970A
A3 current measurement	Nominal measuring range	0A		2.5A
	Measuring range upper limit	3.050A	3.100A	3.150A
PWM Type A general	Safe frequency range	50Hz		1000Hz
	Safe frequency resolution[8]		1Hz	
	Standard frequency range	1Hz		1000Hz
	Standard Current-controlled frequency range	50Hz		1000Hz
	Standard frequency resolution[8]		1Hz	
	Frequency precision 1 - 49Hz	-0.5Hz	f_{TARGET}	+0.5Hz
	Frequency precision 50 - 200Hz	-2.5Hz	f_{TARGET}	+2.5Hz
	Frequency precision 201 - 300Hz	-5Hz	f_{TARGET}	+5Hz
	Frequency precision 301 - 500Hz	-10Hz	f_{TARGET}	+10Hz
	Frequency precision 501 - 700Hz	-20Hz	f_{TARGET}	+20Hz
	Frequency precision 701 - 1.0kHz	-37.5Hz	f_{TARGET}	+37.5Hz
	Type A duty cycle resolution		12 bit	
PWM Type A uncontrolled	Type A1 duty cycle resolution		5%	
	Type A2 duty cycle resolution		3%	

[7] Reference to the nominal scale end value.

[8] Resolution for setting the frequency parameters in CoDeSys.

Type A outputs	Note	Min.	Typ.	Max.
	Type A3 duty cycle resolution		3%	
Type A Digital output	Maximum switching frequency For more detailed information refer to Chapter 4.6.2.6			10Hz

10.13.6 Type B inputs/outputs

Type B as an output	Note	Min.	Typ.	Max.
Voltage level Type B output	OFF state	0.0V	0.45V	
	ON state			VIQ
Short-circuit strength	Short circuit against GND		permanent	
	Short circuit against VIM/VIQ		permanent	
Leakage current	OFF state (T_a = 25°C +/- 5°C)			50µA
B switching current	Rated current		4.0A	
	Rated current, 2 parallel outputs[9]		6.8A	
	Rated current, 3 parallel outputs[9]		8.8A	
	Type B overcurrent (permanent)	4.8A		4.9A
	Warning in error memory at I_B > 5 A after	0.1s		0.25s
	Output de-energized at I_B > 5 A after	0.9s		1.1s
B switching frequency	For more detailed information refer to Chapter 4.6.3.5			10Hz
Current measurement	Precision (reference to rated scale end value)			± 5.0%

Type B as an input	Note	Min.	Typ.	Max.
Type B input	LOW level			5.0V
	HIGH level	6.8V		
	Electric strength[10]	-60.0V		+60.0V
	Input resistance	3.705kΩ	3.9kΩ	4.095kΩ

10.13.7 Type C inputs

Type C inputs	Note	Min.	Typ.	Max.
Type C input	Electric strength	-60.0V		+60.0V
	Input resistance	3.705kΩ	3.9kΩ	4.095kΩ
	Cutoff frequency[11]		50Hz	
pulled down	LOW level			6.0V
	HIGH level	6.8V		
pulled up	LOW level			6.8V
	HIGH level	6.0V		

[9] Parallel switching is only permissible in standard output configuration.

[10] Only in combination with an external fuse. For more detailed information refer to 4.6.4

[11] 3dB cutoff frequency of the hardware filter. For more information on the maximum frequency which can be processed in the application refer to 4.6.4

10.13.8 Type D inputs

Type D inputs	Note	Min.	Typ.	Max.
general properties (also apply to all type D configurations)	Electric strength	-50.0V		+ 60.0V
	Cutoff frequency[12]		30kHz	
	Input current at V_{IN} = 5V	1.5mA		
	Input current at V_{IN} = 11V	2.0mA		
	Switching threshold setting range	500mV		12,000mV
	Switching threshold precision	-250mV		+ 250mV
	Switching hysteresis	100mV		500mV

Type D inputs	Note	Min.	Typ.	Max.
Configuration Digital input	Switching threshold to be defined		6,400mV	
	LOW level			5.9V
	HIGH level	6.9V		

Type D inputs	Note	Min.	Typ.	Max.
Configuration Counter input	Counting frequency			20kHz
	Counting range		32 bit	
	Counting precision[13]	-1		+1
	Time window setting range[14]	10ms		1,280ms
	Time window setting increment[14]		5ms	

Type D inputs	Note	Min.	Typ.	Max.
Configuration Pulse length measurement	Input signal frequency (use of ≤ 2 inputs)[15]	15.9Hz		1kHz
	Input signal frequency (use of ≤ 4 inputs)[15]	15.9Hz		500Hz
	Pulse length	50µs		1.275s
	Resolution		1µs	
	Deviation from the determined pulse length (period T ≤ 10ms)[16]	-10µs		+10µs

[12] 3dB cutoff frequency of the hardware filter. For more detailed information for the maximum frequency which can be processed in the application refer to 4.6.5.

[13] Deviation from the theoretically expected number of pulses.

[14] Frequency measurement gate time

[15] For more detailed information refer to 4.6.5

[16] Only applies to switching threshold = 12.00Volt.

Type D inputs	Note	Min.	Typ.	Max.
	Deviation from the determined pulse length (period T > 10ms)[16]	-0.1% T	T	+0.1% T
Period measurement (Pulse length measurement configuration)	Measuring range	1µs		65534µs
	Resolution		1µs	
	Deviation from the period duration (period T ≤ 10ms)[17]	-2µs		+2µs
	Deviation from the period duration (Period T > 10ms)[17]	-0.02% T	T	+0.02% T

Type D inputs	Note	Min.	Typ.	Max.
Configuration AB counter	Counting frequency			20kHz
	Counting range		32 bit	
	Counting precision[18]	-1		+1

Type D inputs	Note	Min.	Typ.	Max.
Configuration Phase measurement	Input signal frequency (1 input used for phase measurement)[19]	233µHz		1kHz
	Input signal frequency (2 inputs used for phase measurement)[19]	233µHz		500Hz
	Input signal frequency (1 input used for phase measurement and up to 2 inputs for pulse length measurement)[19]	233µHz		500Hz
	Pulse length	24µs		
	Measuring range	0°		359.9°
	Resolution		1/10°	

10.13.9 Type E inputs

Type E inputs	Note	Min.	Typ.	Max.
general properties (apply to all type E configurations)	Cutoff frequency[20]	70Hz	100Hz	130Hz
	Electric strength	-60.0V		60.0V
	Analog value resolution		12 bit	

[17] Only applies to switching threshold = 12.00Volt.

[18] Deviation from the theoretically expected number of pulses.

[19] For more detailed information refer to 4.6.5

[20] 3dB cutoff frequency of the hardware filter. For more detailed information for the maximum frequency which can be processed in the application refer to 4.6.5.

Type E inputs	Note	Min.	Typ.	Max.
Configuration Digital input	LOW level			5.0V
	HIGH level	6.8V		
	Input resistance	3.51kΩ		3.90kΩ
	Input resistance (digital input high-R configuration)	24kΩ		
	Hysteresis		600mV	

Type E inputs	Note	Min.	Typ.	Max.
Configuration Analog input 0 - 10 Volt	Nominal measuring range	0V		10.0V
	Measuring range upper limit (Error flag is set)[19]		10.75V	
	Measuring range lower limit (safe)[21]	0.5V		
	Measuring range upper limit (safe)[21]			10,0V
	Precisions[22]			± 1.0%
	Safety precision			± 4.5%
	Input resistance	24kΩ		

Type E inputs	Note	Min.	Typ.	Max.
Configuration Analog input 0 - 32 Volt	Nominal measuring range	0V		32.0V
	Measuring range upper limit (Error flag is set)[19]		32.5V	
	Measuring range lower limit (safe)[21]	1.5V		
	Measuring range upper limit (safe)[21]			32,0V
	Precision[22]			± 1.0%
	Safety precision			± 4.5%
	Input resistance	24kΩ		
	Input resistance (analog input low R configuration)	3.51kΩ		3.90kΩ

Type E inputs	Note	Min.	Typ.	Max.
Configuration Analog input 0 - 20 mA	Nominal measuring range	0mA		20mA
	Measuring range upper limit (Error flag is set)[23]		21.75mA	

[21] For more detailed information on the safety measuring range refer to Chapter 4.6.6

[22] Reference to the nominal scale end value.

[23] For more detailed information on the safety measuring range refer to Chapter 4.6.6

Type E inputs	Note	Min.	Typ.	Max.
	Measuring range lower limit (safe)[25]	1.0mA		
	Measuring range upper limit (safe)[25]			20mA
	Current overload shutdown (measuring shunt disconnected)	30.0mA		
	Polarity reversal/overload shutdown (measuring shunt disconnected)	6.0V		
	Polarity reversal/overload shutdown (measuring shunt disconnected)			-2.7V
	Precision[24] (-40°C – 80°C)[25]			± 1.5%
	Precision[22] (-40°C – 25°C)			± 1.0%
	Safety precision			± 6.0%
	Input resistance	112Ω		114Ω

10.13.10 Sensor supply output QSENS

Sensor supply output QSENS	Note	Min.	Typ.	Max.
QSENS output	Output voltage	VIM-1.3V		VIM
	Electric strength	-3.0V		+60.0V
	Output current	400mA		
	Short-circuit strength		permanent	
	Max. deviation from P_VIM (warning is saved)[26]	-1.5V		0.5V
QSENS readback	Precision (Reference to 24V rated current)			± 1.0%

10.13.11 Reference voltage sources UREF1 and UREF2

Reference voltage sources UREFx	Note	Min.	Typ.	Max.
UREFx outputs	Rated output voltage		10.00V	
	Electric strength	-2.0V		+45.0V
	Short-circuit strength		permanent	
	Operating voltage range[27]	11.1V		32.2V
	Output current	50.5mA		

24 Reference to the nominal scale end value.

25 The current values provided do not apply to temperatures > 80°C! For more detailed information refer to 4.6.6

26 For more detailed information refer to Chapter 4.8

27 VIM value required in order to ensure that Uref functions correctly

Reference voltage sources UREFx	Note	Min.	Typ.	Max.
	Output voltage precision[28]			± 1.0%
	Safety precision[28]			± 1.5%
Read back	Precision[28]			± 1.0%
	Resolution		12 bit	

10.13.12 Reference current source IREF

Reference current source IREF	Note	Min.	Typ.	Max.
IREF output	Electric strength	-10.0V		+60.0V
	Short-circuit strength		permanent	
	Output current	49.5mA	50.0mA	50.5mA
	Load resistance (Condition: VIM ≥ 8.0V)			110Ω

10.13.13 Interfaces

10.13.13.1 RS232

RS232	Transmission rate	Unit
COM1, COM2	4800, 9600, 19200, 38400, 57600, 115200	Bit/s

10.13.13.2 Ethernet

Ethernet	Transmission rate	Unit
Eth1	10, 100	Mbit/s

10.13.13.3 USB

USB	Transmission rate	Unit
Standard USB2.0	Low speed (1.5 Mbit/s), Full speed (12 Mbit/s)	Mbit/s

[28] Reference to the nominal output voltage

10.14 List of PLC-Versions

Versioned product number: The third group of the versioned product number (here: XX) is an internal group and is not noted on the article label (no influence on certification).

HW-Version: The difference between HW-Version VZZ.30.XX and VZZ.35.XX consists of an optimized conductor board layout. This change is completely compatible. There is no influence on functionality or security.

The third group of the HW_version (here: XX) is an internal group and has no influence on the certification.

Versioned Product-Nr.	Item	HW version		Module	FW version	
V01.10.XX 4888.04.301 4888.04.302 4888.04.304	*digsy*®$_{\text{fusion}}$S-PL	GCM-PL:	V01.03.XX	GCM-Performance Linux	UBOOT (GCM-PL): Bootloader (GCM-PL): Bootloader 2 (GCM-PL): Firmware (GCM-PL):	V11.06.27 V01.00.01 V01.00.00 V01.00.04
		SCM:	V06.35.XX V06.40.XX	SCM-Eth-SL	Fork (CPU0): Bootloader (CPU0): Firmware safe (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V05.01.00 V02.08.05 V01.01.01 V05.01.00 V01.15.02
		SIOM:	V04.35.XX V04.40.XX	SIOM-Basic	Fork (CPU0): Bootloader (CPU0): Firmware (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V05.01.00 V01.24.00 V01.01.01 V05.01.00 V01.15.00
V01.09.XX 4888.04.301 4888.04.302 4888.04.304	*digsy*®$_{\text{fusion}}$S-PL	GCM-PL:	V01.03.XX	GCM-Performance Linux	UBOOT (GCM-PL): Bootloader (GCM-PL): Bootloader 2 (GCM-PL): Firmware (GCM-PL):	V11.06.27 V01.00.00 V01.00.00 V01.00.03
		SCM:	V06.35.XX V06.40.XX	SCM-Eth-SL	Fork (CPU0): Bootloader (CPU0): Firmware safe (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V05.01.00 V02.08.02 V01.01.01 V05.01.00 V01.15.00
		SIOM:	V04.35.XX V04.40.XX	SIOM-Basic	Fork (CPU0): Bootloader (CPU0): Firmware (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V05.01.00 V01.24.00 V01.01.01 V05.01.00 V01.15.00
V01.08.XX 4888.04.301 4888.04.302 4888.04.304	*digsy*®$_{\text{fusion}}$S-PL	GCM-PL:	V01.03.XX	GCM-Performance Linux	UBOOT (GCM-PL): Bootloader (GCM-PL): Bootloader 2 (GCM-PL): Firmware (GCM-PL):	V11.06.27 V01.00.00 V01.00.00 V01.00.01
		SCM:	V06.35.XX V06.40.XX	SCM-Eth-SL	Fork (CPU0): Bootloader (CPU0): Firmware safe (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V04.01.01 V02.07.00 V01.01.01 V04.00.05 V01.15.00

Versioned Product-Nr.	Item	HW version		Module	FW version	
		SIOM:	V04.35.XX V04.40.XX	SIOM-Basic	Fork (CPU0): Bootloader (CPU0): Firmware (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V04.00.05 V01.24.00 V01.01.01 V04.00.05 V01.15.00
V01.10.XX 4888.03.301 4888.03.302 4888.03.304 4888.03.305 4888.03.321 4888.03.322 4888.03.324 4888.03.325	*digsy*®$_{fusion}$S-P	GCM-P:	V01.02.XX	GCM-Performance	UBOOT (GCM-P): Bootloader (GCM-P): Firmware (GCM-P):	V11.06.27 V01.00.01 V01.03.04
		SCM:	V06.35.XX V06.40.XX	SCM-Eth-SL	Fork (CPU0): Bootloader (CPU0): Firmware safe (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V05.01.00 V02.08.05 V01.01.01 V05.01.00 V01.15.02
		SIOM:	V04.35.XX V04.40.XX	SIOM-Basic	Fork (CPU0): Bootloader (CPU0): Firmware (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V05.01.00 V01.24.00 V01.01.01 V05.01.00 V01.15.00
V01.09.XX 4888.03.301 4888.03.302 4888.03.304 4888.03.305 4888.03.321 4888.03.322 4888.03.324 4888.03.325	*digsy*®$_{fusion}$S-P	GCM-P:	V01.02.XX	GCM-Performance	UBOOT (GCM-P): Bootloader (GCM-P): Firmware (GCM-P):	V11.06.27 V01.00.00 V01.03.03
		SCM:	V06.35.XX V06.40.XX	SCM-Eth-SL	Fork (CPU0): Bootloader (CPU0): Firmware safe (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V05.01.00 V02.08.02 V01.01.01 V05.01.00 V01.15.00
		SIOM:	V04.35.XX V04.40.XX	SIOM-Basic	Fork (CPU0): Bootloader (CPU0): Firmware (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V05.01.00 V01.24.00 V01.01.01 V05.01.00 V01.15.00
V01.06.XX 4888.03.305 4888.03.325	*digsy*®$_{fusion}$S-P	GCM-P:	V01.02.XX	GCM-Performance	UBOOT (GCM-P): Bootloader (GCM-P): Firmware (GCM-P):	V11.06.27 V01.00.00 V01.03.00
		SCM:	V06.35.XX V06.40.XX	SCM-Eth-SL	Fork (CPU0): Bootloader (CPU0): Firmware safe (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V04.00.05 V02.04.01 V01.01.01 V04.00.05 V01.12.01
		SIOM:	V04.35.XX V04.40.XX	SIOM-Basic	Fork (CPU0): Bootloader (CPU0): Firmware (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V04.00.05 V01.22.01 V01.01.01 V04.00.05 V01.12.01
V01.05.XX 4888.03.301	*digsy*®$_{fusion}$S-P	GCM-P:	V01.02.XX	GCM-Performance	UBOOT (GCM-P): Bootloader (GCM-P): Firmware (GCM-P):	V11.06.27 V01.00.00 V01.03.00

Versioned Product-Nr.	Item	HW version		Module	FW version	
4888.03.302 4888.03.304 4888.03.321 4888.03.322 4888.03.324		SCM:	V06.35.XX V06.40.XX	SCM-Eth-SL	Fork (CPU0): Bootloader (CPU0): Firmware safe (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V04.00.05 V02.04.00 V01.01.01 V04.00.05 V01.12.00
		SIOM:	V04.35.XX V04.40.XX	SIOM-Basic	Fork (CPU0): Bootloader (CPU0): Firmware (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V04.00.05 V01.22.00 V01.01.01 V04.00.05 V01.12.00
V01.00.XX		GCM-P:	V01.02.XX	GCM-Performance	UBOOT (GCM-P): Bootloader (GCM-P): Firmware (GCM-P):	V11.06.27 V01.00.00 V01.00.00
4888.03.301 4888.03.302 4888.03.304 4888.03.321 4888.03.322 4888.03.324	*digsy*®$_{fusion}$S-P	SCM:	V06.35.XX V06.40.XX	SCM-Eth-SL	Fork (CPU0): Bootloader (CPU0): Firmware safe (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V04.00.03 V02.03.00 V01.01.01 V04.00.01 V01.11.00
		SIOM:	V04.35.XX V04.40.XX	SIOM-Basic	Fork (CPU0): Bootloader (CPU0): Firmware (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V04.00.01 V01.21.00 V01.01.01 V04.00.01 V01.11.00
V03.37.XX 4888.02.001 4888.02.002 4888.02.004 4888.02.005	*digsy*®$_{fusion}$S	SCM:	V06.30.XX V06.35.XX V06.40.XX	SCM-Ethernet	Fork (CPU0): Bootloader (CPU0): Firmware safe (CPU0): Firmware std (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V05.01.00 V02.08.05 V01.09.00 V01.01.01 V05.01.00 V01.15.02
4888.02.007 4888.02.101 4888.02.102 4888.02.104 4888.02.105	*digsy*®$_{fusion}$S T2	SIOM:	V04.30.XX V04.35.XX V04.40.XX	SIOM-Basic	Fork (CPU0): Bootloader (CPU0): Firmware (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V05.01.00 V01.24.00 V01.01.01 V05.01.00 V01.15.00
4888.02.107 4888.02.201 4888.02.202 4888.02.204 4888.02.205 4888.02.207	*digsy*®$_{fusion}$S T3	GIOM	V04.35.XX V04.40.XX	GIOM-Basic	Fork (CPU0): Bootloader (CPU0): Firmware (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V02.00.00 V05.01.00 V04.02.00 V02.00.00 V05.01.00 V04.02.00
V03.36.XX 4888.02.001 4888.02.002 4888.02.004	*digsy*®$_{fusion}$S	SCM:	V06.30.XX V06.35.XX V06.40.XX	SCM-Ethernet	Fork (CPU0): Bootloader (CPU0): Firmware safe (CPU0): Firmware std (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V05.01.00 V02.08.02 V01.07.02 V01.01.01 V05.01.00 V01.15.00

Versioned Product-Nr.	Item	HW version		Module	FW version	
4888.02.005 4888.02.007 4888.02.101 4888.02.102 4888.02.104 4888.02.105 4888.02.107 4888.02.201 4888.02.202 4888.02.204 4888.02.205 4888.02.207	*digsy*®$_{fusion}$S T2 *digsy*®$_{fusion}$S T3	SIOM:	V04.30.XX V04.35.XX V04.40.XX	SIOM-Basic	Fork (CPU0): Bootloader (CPU0): Firmware (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V05.01.00 V01.24.00 V01.01.01 V05.01.00 V01.15.00
		GIOM	V04.35.XX V04.40.XX	GIOM-Basic	Fork (CPU0): Bootloader (CPU0): Firmware (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V02.00.00 V05.01.00 V04.02.00 V02.00.00 V05.01.00 V04.02.00
V03.35.XX 4888.02.001 4888.02.002 4888.02.004 4888.02.005 4888.02.007 4888.02.101 4888.02.102 4888.02.104 4888.02.105 4888.02.107 4888.02.201 4888.02.202 4888.02.204 4888.02.205 4888.02.207	*digsy*®$_{fusion}$S *digsy*®$_{fusion}$S T2 *digsy*®$_{fusion}$S T3	SCM:	V06.30.XX V06.35.XX V06.40.XX	SCM-Ethernet	Fork (CPU0): Bootloader (CPU0): Firmware safe (CPU0): Firmware std (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V04.00.05 V02.04.00 V01.07.02 V01.01.01 V04.00.05 V01.12.00
		SIOM:	V04.30.XX V04.35.XX V04.40.XX	SIOM-Basic	Fork (CPU0): Bootloader (CPU0): Firmware (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V04.00.05 V01.22.00 V01.01.01 V04.00.05 V01.12.00
		GIOM	V04.35.XX V04.40.XX	GIOM-Basic	Fork (CPU0): Bootloader (CPU0): Firmware (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V02.00.00 V04.00.05 V04.00.02 V02.00.00 V04.00.05 V04.00.02
V03.30.XX 4888.02.001 4888.02.002 4888.02.004 4888.02.101 4888.02.102 4888.02.104 4888.02.201 4888.02.202 4888.02.204	*digsy*®$_{fusion}$S *digsy*®$_{fusion}$S T2 *digsy*®$_{fusion}$S T3	SCM:	V06.30.XX V06.35.XX V06.40.XX	SCM-Ethernet	Fork (CPU0): Bootloader (CPU0): Firmware safe (CPU0): Firmware std (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V04.00.01 V02.02.00 V01.07.01 V01.01.01 V04.00.01 V01.10.00
		SIOM:	V04.30.XX V04.35.XX V04.40.XX	SIOM-Basic	Fork (CPU0): Bootloader (CPU0): Firmware (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V04.00.01 V01.10.00 V01.01.01 V04.00.01 V01.10.00
V03.26.XX 4888.02.001 4888.02.002 4888.02.004	*digsy*®$_{fusion}$S	SCM:	V06.30.XX V06.35.XX V06.40.XX	SCM-Ethernet	Fork (CPU0): Bootloader (CPU0): Firmware safe (CPU0): Firmware std (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V01.09.00 V02.01.01 V01.06.00 V01.01.01 V01.07.00 V01.09.01

Versioned Product-Nr.	Item	HW version		Module	FW version	
4888.02.101 4888.02.102 4888.02.104 4888.02.201 4888.02.202 4888.02.204	*digsy*$^{®}_{fusion}$S T2 *digsy*$^{®}_{fusion}$S T3	SIOM:	V04.24.XX V04.30.XX V04.35.XX V04.40.XX	SIOM-Basic	Fork (CPU0): Bootloader (CPU0): Firmware (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V01.08.00 V01.09.01 V01.01.01 V01.07.00 V01.09.01
V02.16.XX 4888.02.001 4888.02.002 4888.02.004 4888.02.101 4888.02.102 4888.02.104 4888.02.201 4888.02.202 4888.02.204	 *digsy*$^{®}_{fusion}$S *digsy*$^{®}_{fusion}$S T2 *digsy*$^{®}_{fusion}$S T3	SCM:	V06.24.XX V06.30.XX V06.40.XX	SCM-Ethernet	Fork (CPU0): Bootloader (CPU0): Firmware safe (CPU0): Firmware std (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V01.09.00 V01.08.01 V01.05.00 V01.01.01 V01.07.00 V01.08.01
		SIOM:	V04.24.XX V04.30.XX V04.40.XX	SIOM-Basic	Fork (CPU0): Bootloader (CPU0): Firmware (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V01.08.00 V01.08.01 V01.01.01 V01.07.00 V01.08.01
V02.10.XX 4888.02.001 4888.02.002 4888.02.004 4888.02.101 4888.02.102 4888.02.104 4888.02.201 4888.02.202 4888.02.204	 *digsy*$^{®}_{fusion}$S *digsy*$^{®}_{fusion}$S T2 *digsy*$^{®}_{fusion}$S T3	SCM:	V06.24.XX V06.30.XX V06.40.XX	SCM-Ethernet	Fork (CPU0): Bootloader (CPU0): Firmware safe (CPU0): Firmware std (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V01.06.01 V01.06.06 V01.03.04 V01.01.01 V01.06.00 V01.06.06
		SIOM:	V04.24.XX V04.30.XX V04.40.XX	SIOM-Basic	Fork (CPU0): Bootloader (CPU0): Firmware (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V01.06.00 V01.06.06 V01.01.01 V01.06.00 V01.06.06
V02.08.XX 4888.02.001 4888.02.002 4888.02.004 4888.02.101 4888.02.102 4888.02.104	 *digsy*$^{®}_{fusion}$S *digsy*$^{®}_{fusion}$S T2	SCM:	V06.24.XX V06.40.XX	SCM-Ethernet	Fork (CPU0): Bootloader (CPU0): Firmware safe (CPU0): Firmware std (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V01.06.00 V01.05.01 V01.02.01 V01.01.01 V01.06.00 V01.05.01
		SIOM:	V04.24.XX V04.40.XX	SIOM-Basic	Fork (CPU0): Bootloader (CPU0): Firmware (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V01.06.00 V01.05.01 V01.01.01 V01.06.00 V01.05.01
V02.07.XX 4888.02.001 4888.02.002 4888.02.004	 *digsy*$^{®}_{fusion}$S	SCM:	V06.24.XX V06.40.XX	SCM-Ethernet	Fork (CPU0): Bootloader (CPU0): Firmware safe (CPU0): Firmware std (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V01.00.00 V01.05.00 V01.02.01 V01.01.00 V01.00.00 V01.05.00

Versioned Product-Nr.	Item	HW version		Module	FW version	
4888.02.101 4888.02.102 4888.02.104	*digsy*®$_{\text{fusion}}$S T2	SIOM:	V04.24.XX V04.40.XX	SIOM-Basic	Fork (CPU0): Bootloader (CPU0): Firmware (CPU0): Fork (CPU1): Bootloader (CPU1): Firmware (CPU1):	V01.01.00 V01.00.00 V01.05.00 V01.01.00 V01.00.00 V01.05.00

Table 10-11: List of PLC versions

10.15 Index

10.16 Errorcodes

Nibble	1	2	3	4	5	6	7	8	Name	Description	Intern al (1)
	3								Information	This entry is an information	
	5								Warning	Entry of a warning, e.g. In the case of errors of non-safety relevant inputs / outputs	
	6								Diag	Entry of a diagnostic error, e.g. In the event of faults of safety-related inputs / outputs	
	9								Severe Error	Entry of a severe error	
	A								Fatal Error	Entry of a fatal error	
		3							SCM	An error / warning has occurred on the SCM	
		5							Slave 1 (SIOM1/GIOM1)	Error / warning occurred on the 1st SIOM/GIOM	
		6							Slave 2 (SIOM2/GIOM2)	Error / warning occurred on the 2nd SIOM/GIOM	
		9							Slave 3 (SIOM3/GIOM3)	Error / warning occurred on the 3rd SIOM/GIOM	
		A							Slave 4 (SIOM4/GIOM4)	Error / warning occurred on the 4th SIOM/GIOM	
			4						CPU0	Error / warning occurred on CPU1	
			C						CPU1	Error / warning occurred on CPU2	
									Common Errors		
						1			ERR_TYPE_UNKNOWN_ERROR	This error is unknown	X
						2			ERR_TYPE_INVALID_ERRORCODE	This error code is invalid	X
						3			ERR_TYPE_SYSTEM_STARTUP	System StartUp (time stamp = operating hours in s)	
						4			ERR_TYPE_INVALID_BL_AND_FW	Bootloader and firmware are invalid (CRC test failed in fork)	
						5			ERR_TYPE_INVALID_FW	Firmware is invalid (CRC test in fork failed)	
						7			ERR_TYPE_PVIM_OVERLARGE	Voltage VIM is outside the permitted limits	
						8			ERR_TYPE_SAMPLE_TIMEOUT	Sampling the ADC data on a CPU has not been performed in the maximum allowable time.	X
						9			ERR_TYPE_ESM_ERROR_PIN_TIMEOUT	The error pin of the ESM on a CPU is not inactive (start failure).	X
						B			ERR_TYPE_MSG_OVERFLOW	Overflow of the internal error memory	

Nibble	1	2	3	4	5	6	7	8	Name	Description	Intern al (1)
						C			ERR_TYPE_SPIHANDLER_BUFFER_FULL	The buffer of the SPI handler is full	X
						D			ERR_TYPE_CPU0_CORE_VLTG_OUT_OF_RANGE	The supply voltage of the CPU0 is outside its limits	X
						E			ERR_TYPE_PVIM_OUT_OF_RANGE	PVIM is out of range	
						F			ERR_TYPE_INVALID_OPERATING_STATE	The current operating state is invalid	X
					1	0			ERR_TYPE_IOVALIDFLAG_TIMEOUT	The IO valid flag has not become valid within the permissible time	
					1	1			ERR_TYPE_INVALID_IO_DATA_SLAVE	The size of the IO data of an expansion module (SIOM/GIOM) is larger than the allowed size.	
					1	2			ERR_TYPE_INVALID_ADCSPI_DATA	Communication with the ADC block on CPU1 failed	X
					1	3			ERR_TYPE_INVALID_CLK	Wrong clock on CPU1	X
					1	4			ERR_TYPE_SPI_TIMEOUT	The maximum time for exchanging data (sending or receiving) between CPU0 and CPU1 has been exceeded	X
					1	5			ERR_TYPE_START_FIRMWARE_FAILED	Booting the firmware on CPU1 failed	
					1	6			ERR_TYPE_COMM_ERROR_WITH_SLAVE	Communication with a Slavemodule failed.	
					1	7			ERR_TYPE_WRONG_OPERATION_STATE_SLAVE	The slave module has returned an unexpected operating state	
					1	8			ERR_TYPE_DEBUG_MODE_ENTERED	It has gone into the operating state DEBUG via CODESYS. !!! Attention: The control is not safe in the state DEBUG !!!	
					1	9			ERR_TYPE_IO_RESET_TIMEOUT	The reset of the IO has failed	X
					1	A			ERR_TYPE_SYNC_TIME_OUT_OF_RANGE	Synchronization from CPU1 failed	X
					1	B			ERR_TYPE_24H_OPERATING_TIME_EXCEEDED	The PLC runs longer than the maximum permissible operating time (24 hours)	
					1	C			ERR_TYPE_SYSTEM_SHOULD_RESET	Reset request from outside (File * RESET * on internal drive)	
					1	D			ERR_TYPE_25H_OPERATING_TIME_EXCEEDED	More than 1 hour over allowable operating time	
					1	E			ERR_TYPE_UBB_COMMUNICATIONERR	Internal communication error between SCM_SL and GCMP board.	
					1	F			ERR_TYPE_USER_OPERATING_TIME_EXCEEDED	The PLC runs longer than the user-defined maximum permissible operating time.	
					2	0			ERR_TYPE_SYSTIME_OVERFLOW	System time overflow. This occurs after approximately 49.7 days.	
									Input/Output Type A		

Nibble	1	2	3	4	5	6	7	8	Name	Description	Intern al (1)
					6	4			ERR_TYPE_STUCK_AT_TYPE_A Ein Ausgang vom Typ A hat einen Stuck-At-Fehler.	A type A output has a stuck-at fault (high).	
					6	5			ERR_TYPE_CURR_MISMATCH_TYPE_A Der Stromwert zwischen dem Mess- und Diagnosepfad für einen Ausgang vom Typ A liegt außerhalb der Toleranz	The value between the measurement and diagnostic path for a type A output is out of tolerance.	
					6	6			ERR_TYPE_FAST_TOGGLING_TYPE_A	The switching frequency of a Type A output has been exceeded (see device data)	
					6	7			ERR_TYPE_DEFECT_TYPE_A	A type A output is damaged: This error occurs when the output is HIGH, but the diagnostics read LOW (stuck-at-error (low)).	
					6	8			ERR_TYPE_OVERCURRENT_TYPE_A Der Stromwert eines Ausgangs vom Typ A hat den maximalen erlaubten Wert überschritten.	The current value of an output of type A has exceeded the maximum permitted value.	
					6	9			ERR_TYPE_P5V_MEAS_OUT_OF_RANGE_TYPE_A	The supply voltage of the outputs Type A is outside its limits. This can be caused by feedback.	
					6	A			ERR_TYPE_P5V_MEAS_SHORTED_OUTPUT_TYPE_A	The supply voltage of the outputs Type A is short-circuited.	
					6	B			ERR_TYPE_P5V_MEAS_DEFECT_TYPE_A	The supply voltage of outputs Type A is faulty	
					6	C			ERR_TYPE_DISABLED_OUTPUT_SWITCHED_ON_TYPE_A	The user is trying to write to an output configured with "disabled"	
					6	D			ERR_TYPE_PWM_SETPOINT_OUT_OF_RANGE	The PWM value is outside its permissible limits	
					6	E			ERR_TYPE_FREQ_OUT_OF_RANGE	The set PWM frequency is outside its permissible limits	
					6	F			ERR_TYPE_CURR_SETPOINT_TO_HIGH	The value for the current-controlled PWM value is too high	
					7	0			ERR_TYPE_OVERCURRENT_SHUTDOWN_TYPE_A	An overcurrent at output type A has led to shutdown	
					7	1			ERR_TYPE_DISABLED_2ND_CUTOFF_TYPE_A	The 2nd cut-off path type A is defective. It is expected LOW but HIGH	
					7	2			ERR_TYPE_PEAK_CURRENT_TYPE_A	An overcurrent was detected at a PWM output.	
					7	3			ERR_TYPE_PEAK_CURRENT_SHUTDOWN_TYPE_A	An overcurrent was detected at a PWM output and the output switched off.	
									Input/Output Type B		
					C	8			ERR_TYPE_STUCK_AT_TYPE_B Ein Ausgang vom Typ B hat einen Stuck-At-Fehler.	A type B output has a stuck-at fault.	

Nibble	1	2	3	4	5	6	7	8	Name	Description	Internal (1)
					C	9			ERR_TYPE_OVERCURRENT_TYPE_B Der Stromwert eines Ausgangs vom Typ B hat den maximalen erlaubten Wert überschritten	The current value of an output of type B has exceeded the maximum permissible value.	
					C	A			ERR_TYPE_FAST_TOGGLING_TYPE_B	The switching frequency of a Type B output has been exceeded (see device data)	
					C	B			ERR_TYPE_DEFECT_TYPE_B	A Type B input / output is damaged	
					C	C			ERR_TYPE_DISABLE_OUT_SWITCHED_ON_TYPE_B	The user is trying to write to an output configured with "disabled"	
					C	D			ERR_TYPE_OVERCURRENT_SHUTDOWN_TYPE_B	An overcurrent at output type B has led to shutdown	
					C	E			ERR_TYPE_DISABLED_2ND_CUTOFF	The 2nd cut-off path type B is defective. It is expected LOW but HIGH	
									Input/Output Type C		
				1	2	C			ERR_TYPE_STUCK_AT_HIGH_TYPE_C Ein Eingang vom Typ C hat einen Stuck-At-Fehler.	A type C input has a stuck-at fault.	
				1	2	D			ERR_TYPE_SWITCH_STATE_INVALID_TYPE_C	-	
				1	2	E			ERR_TYPE_RUNTIMEDIAG_INTERFERES_SAMPLING_C	Runtime diagnostics has affected scanning of the TypC inputs.	
									Input/Output Type D		
				1	9	0			ERR_TYPE_STUCK_AT_HIGH_TYPE_D Ein Eingang vom Typ D hat einen Stuck-At-Fehler	A type D input has a stuck-at fault.	
				1	9	1			ERR_TYPE_THRESHOLD1_OUT_OF_RANGE_TYPE_D Der Wert der einstellbaren Schaltschwelle 1 (betrifft I1.1, I1.2, I1.3, I1.4) wurde nicht erreicht.	The value of the adjustable switching threshold 1 (concerning I1.1, I1.2, I1.3, I1.4) was not reached.	
				1	9	2			ERR_TYPE_THRESHOLD2_OUT_OF_RANGE_TYPE_D Der Wert der einstellbaren Schaltschwelle 2 (betrifft I1.5, I1.6, I1.7, I1.8) wurde nicht erreicht.	The value of the adjustable switching threshold 2 (concerning I1.5, I1.6, I1.7, I1.8) was not reached.	
				1	9	3			ERR_TYPE_STARTUP_DIAG_WRONG_CNT_TYPE_D	Startup diagnostics type D failed. This error can occur if VIM is not connected to the slave boards.	
				1	9	4			ERR_TYPE_TIME_WINDOW_OUT_OF_RANGE_TYPE_D	The selected time window is outside the permitted range	
				1	9	5			ERR_TYPE_RUNTIMEDIAG_INTERFERES_SAMPLING_D	Runtime diagnostics has affected the scanning of the type D inputs.	
									Input/Output Type E		
				1	F	4			ERR_TYPE_MEASUREMENT_FAILURE_TYPE_E	The value between the measuring and diagnostic path for an input of type E is out of tolerance	

Nibble	1	2	3	4	5	6	7	8	Name	Description	Internal (1)
				1	F	5			ERR_TYPE_IU_DIAG_FAILURE_TYPE_E	The fuse of the current measuring shunt of I / O type E has tripped HW-moderately.	
				1	F	6			ERR_TYPE_ADCSPI_SWITCHING_TIMEOUT	-	
				1	F	7			ERR_TYPE_DIAG_OVERLOAD_TYPE_E	Overload at the input channel was detected	
				1	F	8			ERR_TYPE_DIAG_OVLD_NEIGHBOR_CH_TYPE_E	Overload at the adjacent channel of the entrance was established.	
				1	F	9			ERR_TYPE_DIAG_STARTUP_TEST_FAILED_TYPE_E	The startup type failed	
Startup Tests											
				2	5	8			ERR_TYPE_VOLTAGE_SUPERVISOR_CHECK_FAILED	The test of the supply voltage failed. A HW reset is waiting for. If this does not come this error message is entered.	
				2	5	9			ERR_TYPE_CAN_STARTUP_TEST_FAILED	The CAN test after startup failed	
				2	5	A			ERR_TYPE_INVALID_DEVICE_DESCRIPTION	The internal device description is not valid	
				2	5	B			ERR_TYPE_INVALID_CALIBRATION_DATA	The internal calibration data are not valid	
				2	5	C			ERR_TYPE_HARDCODED_DEVICE_DESCRIPTION_USED	A standard device description is used. There is no valid device description.	
				2	5	D			ERR_TYPE_STARTUP_SFR_TEST_FAILED	The SFR test (internal register test) on CPU0 or CPU1 failed	
				2	5	E			ERR_TYPE_STARTUP_IO_DIAGNOSTICS_TIMEOUT	The startup diagnosis has not been completed within the permitted time	
				2	5	F			ERR_TYPE_INVALID_ASSIGNMENT_FW_NAME_TO_BOARD	The hardware of the controller does not match the software (variant / version) located on it.	
				2	6	0			ERR_TYPE_INCOMPATIBLE_COMPONENTS	The firmware version of CPU0 and CPU1 are not compatible.	
				2	6	1			ERR_TYPE_INCOMPATIBLE_BOARDS	The expansion modules used (SCM/SIOM/GIOM) are not compatible. Perhaps the wrong device description is used or the SW / HW versions of the different modules are not compatible.	
				2	6	2			ERR_TYPE_BOARD_INTEGRITY_MISMATCH	The device descriptions of the different expansion modules are not compatible.	
				2	6	3			ERR_TYPE_INVALID_CALIBRATION_DATA_VERSION	The version of the calibration data on the controller is invalid	X
I/O Diagnostics											
				2	B	C			ERR_TYPE_IO_DIAG_INCOMPLETE	The IO diagnosis was not terminated in the specified time	X
VIQ Diagnostics											

Nibble	1	2	3	4	5	6	7	8	Name	Description	Internal (1)
				2	E	E			ERR_TYPE_VIQ_DIAG_2ND_CUT_OFF_DEFECT	The second cut-off path of the VIQ voltage is defective	X
									Temperature		
				3	2	0			ERR_TYPE_OVER_TEMPERATURE	Exceeding the maximum permissible temperature (100 ° C) on the module.	
				3	2	1			ERR_TYPE_UNDER_TEMPERATURE	Exceeding the minimum permissible temperature (-35 ° C) on the module.	
				3	2	2			ERR_TYPE_MIN_TEMP_EXCEEDED	Exceeding the minimum permissible temperature (-40 ° C) on the module.	
				3	2	3			ERR_TYPE_MAX_TEMP_EXCEEDED	Exceeding the maximum permissible temperature (105 ° C) on the module.	
									Invalid Configuration		
				3	8	4			ERR_TYPE_INVALID_CONFIG_DATA_LENGTH	The length of the transferred configuration structure does not match the expected value.	
				3	8	5			ERR_TYPE_INVALID_CONFIG_TYPE_A	Configuration error for I / O type A (unauthorized settings)	
				3	8	6			ERR_TYPE_INVALID_CONFIG_TYPE_B	Configuration error for I / O type B (unauthorized settings)	
				3	8	7			ERR_TYPE_INVALID_CONFIG_TYPE_C	Configuration error for I / O type C (unauthorized settings)	
				3	8	8			ERR_TYPE_INVALID_CONFIG_TYPE_D	Configuration error for I / O type D (unauthorized settings)	
				3	8	9			ERR_TYPE_INVALID_CONFIG_TYPE_E	Configuration error for I / O type E (unauthorized settings)	
				3	8	A			ERR_TYPE_INVALID_CONFIG	Configuration incorrect. This error can occur if IO modules are disabled, but are still in the project.	
				3	8	B			ERR_TYPE_INVALID_CONFIG_SLAVE_BOARD	An incorrect configuration file is located in the slaveboard	
				3	8	C			ERR_TYPE_INVALID_CONFIG_IO_TIME_INTERVAL	The IO image was not exchanged within 10 minutes.	
									SPIX Errors		
				3	E	8			ERR_TYPE_SPIX_CMD_START_FIRMWARE	Error executing the SPIx command "START_FIRMWARE".	
				3	E	9			ERR_TYPE_SPIX_CMD_ENTER_DEBUG_MODE	Error executing the SPIx command “ENTER_DEBUG_MODE"	
				3	E	A			ERR_TYPE_SPIX_CMD_RELEASE_DEBUG_MODE	Error executing the SPIx command "RELEASE_DEBUG_MODE".	
				3	E	B			ERR_TYPE_SPIX_CMD_ENTER_SAFE_STATE	Error executing the SPIx command "ENTER_FAILSAFE_MODE".	
				3	E	C			ERR_TYPE_SPIX_CMD_GET_OPERATION_MODE	Error executing the SPIx command "GET_OPERATION_MODE".	

Nibble	1	2	3	4	5	6	7	8	Name	Description	Internal (1)
				3	E	D			ERR_TYPE_SPIX_CMD_GET_OPERATION_STATE	Error executing the SPIx command "GET_OPERATION_STATE".	
				3	E	E			ERR_TYPE_SPIX_CMD_SET_SYSTEM_VOLTAGE	Error executing the SPIx command "SET_SYSTEM_VOLTAGE ".	
				3	E	F			ERR_TYPE_SPIX_CMD_SET_IO_PIN_CONFIG	Error executing the SPIx command "SET_IO_PIN_CONFIG".	
				3	F	0			ERR_TYPE_SPIX_CMD_GET_ERROR_LIST	Error executing the SPIx command "GET_ERROR_LIST".	
				3	F	1			ERR_TYPE_SPIX_CMD_START_IO_DIAG	Error executing the SPIx command "START_IO_DIAG ".	
				3	F	2			ERR_TYPE_SPIX_CMD_FINISH_IO_DIAG	Error executing the SPIx command "FINISH_IO_DIAG ".	
				3	F	3			ERR_TYPE_SPIX_CMD_EXCHANGE_IO_DATA	Error executing the SPIx command "EXCHANGE_IO_DATA ".	
				3	F	4			ERR_TYPE_SPIX_CMD_EXCHANGE_CAN_DATA	Error executing the SPIx command "EXCHANGE_CAN_DATA ".	
				3	F	5			ERR_TYPE_SPIX_CMD_IO_RESET	Error executing the SPIx command "IO_RESET ".	
				3	F	6			ERR_TYPE_SPIX_CMD_IO_START	Error executing the SPIx command "IO_START ".	
				3	F	7			ERR_TYPE_SPIX_CMD_SYNC_SLAVE	Error executing the SPIx command "SYNC_SLAVE ".	
				3	F	8			ERR_TYPE_SPIX_CMD_GET_SYS_STATE	Error executing the SPIx command "GET_SYS_STATE ".	
				3	F	9			ERR_TYPE_SPIX_CMD_OPEN_FILE	Error executing the SPIx command "OPEN_FILE ".	
				3	F	A			ERR_TYPE_SPIX_CMD_CLOSE_FILE	Error executing the SPIx command "CLOSE_FILE ".	
				3	F	B			ERR_TYPE_SPIX_CMD_WRITE_FILE	Error executing the SPIx command "WRITE_FILE ".	
				3	F	C			ERR_TYPE_SPIX_CMD_READ_FILE	Error executing the SPIx command "READ_FILE ".	
				3	F	D			ERR_TYPE_SPIX_CMD_DELETE_FILE	Error executing the SPIx command "DELETE_FILE ".	
				3	F	E			ERR_TYPE_SPIX_CMD_STARTUP_CAN_RESULTS	Error executing the SPIx command "STARTUP_CAN_RESULTS ".	
				3	F	F			ERR_TYPE_SPIX_CMD_GET_SW_VERSION	Error executing the SPIx command "GET_SW_VERSION ".	
				4	0	0			ERR_TYPE_SPIX_CMD_GET_SW_VERSION_SLAVE	Error executing the SPIx command "GET_SW_VERSION_SLAVE ".	
				4	0	1			ERR_TYPE_SPIX_CMD_SET_CURR_MEAS_PATH	Error executing the SPIx command "SET_CURR_MEAS_PATH".	
				4	0	2			ERR_TYPE_SPIX_CMD_DISABLE_VIQ	Error executing the SPIx command "DISABLE_VIQ ".	
				4	0	3			RR_TYPE_SPIX_CMD_START_OPERATING	Error executing the SPIx command "START_OPERATING".	
				4	0	4			ERR_TYPE_SPIX_CMD_GET_SYS_ERROR_CODES	Error executing the SPIx command "GET_SYS_ERROR_CODES".	
				4	0	5			ERR_TYPE_SPIX_CMD_GET_CRC_DEV_DESC	Error executing the SPIx command "GET_CRC_DEV_DESC".	
				4	0	6			ERR_TYPE_SPIX_CMD_POLL_SLAVE	Error executing the SPIx command "POLL_SLAVE".	

Nibble	1	2	3	4	5	6	7	8	Name	Description	Internal (1)
				4	0	7			ERR_TYPE_SPIX_CMD_SYS_FW_CHECK	Error executing the SPIx command "SYS_FW_CHECK".	
				4	0	8			ERR_TYPE_SPIX_CMD_DISABLE_MONITORING	Error executing the SPIx command "DISABLE_MONITORING".	
				4	0	9			ERR_TYPE_SPIX_CMD_CAN_INIT	Error executing the SPIx command "CAN_INIT ".	
				4	0	A			ERR_TYPE_SPIX_CMD_CAN_RESET	Error executing the SPIx command "CAN_RESET ".	
				4	0	B			ERR_TYPE_SPIX_IOEX_PROCESS_TIMEOUT	Error executing the SPIx command "PROCESS_TIMEOUT ".	
				4	0	C			ERR_TYPE_SPIX_CMD_UNDEF	Error executing an undefined SPIx command.	
CODESYS Errors											
				4	4	C			ERR_TYPE_SAFE_STATE_REQUESTED	When the bottle type is selected in CODESYS. CODESYS AWP! = Device Description	
				4	4	D			ERR_TYPE_FAILSAFE_STATE_REQUESTED	CODESYS changes the operating state to the safe state	
				4	4	E			ERR_TYPE_RTS_SIL2_EXCEPTION_RUNTIME_INIT	CMInit () could not be executed properly	
				4	4	F			ERR_TYPE_RTS_SIL2_EXCEPTION_LOADBOOTPROJECT	The transfer of the lifecounter is faulty	
				4	5	0			ERR_TYPE_RTS_SIL2_EXCEPTION_LIFE_COUNTER	The transfer of the lifecounter is faulty	
				4	5	1			ERR_TYPE_HDL_LIFECNTFAIL	The transfer of the lifecounter is faulty	
				4	5	2			ERR_TYPE_RTS_SIL2_EXECPTION_GENERAL	An exception of the runtime system was called	
CPU exceptions											
				4	B	0			ERR_TYPE_SVC_FAULT	An "SVC" -Exception on CPU0 has occurred.	
				4	B	1			ERR_TYPE_PHANTOM_INTERRUPT	An undefined interrupt has occurred	
				4	B	2			ERR_TYPE_CPU0_UNDEF_EXCEPTION	Internal processor error	X
				4	B	3			ERR_TYPE_CPU0_PREFETCH_EXCEPTION	Internal processor error	X
				4	B	4			ERR_TYPE_CPU0_ESM_GROUP1_ERROR	Internal processor error	X
				4	B	5			ERR_TYPE_CPU0_ESM_GROUP2_ERROR	Internal processor error	X
				4	B	6			ERR_TYPE_CPU0_DATA_ABORT_EXCEPTION	Internal processor error	X
				4	B	7			ERR_TYPE_CPU1_NMI_FAULT	Internal processor error	X
				4	B	8			ERR_TYPE_CPU1_HARD_FAULT	Internal processor error	X
				4	B	9			ERR_TYPE_CPU1_MEMMANAGE_FAULT	Internal processor error	X
				4	B	A			ERR_TYPE_CPU1_BUS_FAULT	Internal processor error	X

Nibble	1	2	3	4	5	6	7	8	Name	Description	Internal (1)
				4	B	B			ERR_TYPE_CPU1_USAGE_FAULT	Internal processor error	X
				4	B	C			ERR_TYPE_CPU1_DEBUGMON_FAULT	Internal processor error	X
				4	B	D			ERR_TYPE_CPU1_PENDSV_FAULT	Internal processor error	X
				4	B	E			ERR_TYPE_CPU1_SYSTICK_FAULT	Internal processor error	X
				4	B	F			ERR_TYPE_CPU0_ESM_GROUP3_ERROR_DURING_STARTUP	Internal processor error	X
				4	C	0			ERR_TYPE_CPU0_ESM_GROUP3_EFUSE_ERR	Internal processor error	X
				4	C	1			ERR_TYPE_CPU0_ESM_GROUP3_RAM_ECC_ERR	Internal processor error	X
				4	C	2			ERR_TYPE_CPU0_ESM_GROUP3_FLASH_ECC_ERR	Internal processor error	X
				4	C	3			ERR_TYPE_CPU0_ESM_GROUP1_VIM_RAM_PARITY_ERR	Internal processor error	X
				4	C	4			ERR_TYPE_CPU0_ESM_GROUP1_DMA_RAM_PARITY_ERR	Internal processor error	X
				4	C	5			ERR_TYPE_CPU0_ESM_GROUP1_ADC1_RAM_PARITY_ERR	Internal processor error	X
				4	C	6			ERR_TYPE_CPU0_ESM_GROUP1_N2HET_RAM_PARITY_ERR	Internal processor error	X
				4	C	7			ERR_TYPE_CPU0_ESM_GROUP1_DMA_MPU_VIOLATION	Internal processor error	X
				4	C	8			ERR_TYPE_CPU0_ESM_GROUP1_ERR	Internal processor error	X
				4	C	9			ERR_TYPE_CPU0_ESM_GROUP2_ERR	Internal processor error	X
				4	C	A			ERR_TYPE_CPU1_MAINLOOP_STUCK	Internal processor error	X
									Defensive Programming		
				5	1	4			ERR_TYPE_INVALID_PARAM	Verification of a function parameter failed (defensive programming).	
				5	1	5			ERR_TYPE_INVALID_PTR	Checking a pointer failed (defensive programming).	
				5	1	6			ERR_TYPE_INVALID_VALUE	Checking the contents of a variable failed (defensive programming).	
				5	1	7			ERR_TYPE_INVALID_STATE	Verification of the contents of a state variable failed (defensive programming).	
				5	1	8			ERR_TYPE_INVALID_NHET_INTERRUPT	An invalid interrupt of the N2HET coprocessor occurred on CPU0.	
				5	1	9			ERR_TYPE_INVALID_GIO_INTERRUPT	An invalid interrupt of the GPIO component occurred on the CPU0.	
				5	1	A			ERR_TYPE_INVALID_DMA_INTERRUPT	An invalid DMA interrupt occurred on a CPU	
				5	1	B			ERR_TYPE_INVALID_RTI_INTERRUPT	An invalid RTI interrupt occurred on a CPU	
				5	1	C			ERR_TYPE_RDS_ERR	Checking an RDS variable (double-inverted stored variable) failed.	

Nibble	1	2	3	4	5	6	7	8	Name	Description	Intern al (1)
				5	1	E			ERR_TYPE_UNKOWN_DMA_IRQ	An unknown DMA interrupt has occurred	
				5	1	F			ERR_TYPE_STMLIB_ERR	An error has occurred in CPU for CPU1	
				5	2	0			ERR_TYPE_UNUSED_DMA_CHANNEL	-	
				5	2	1			ERR_TYPE_ERRORHANDLER_MAX_RECURSION	The function "errorHandler_AddErrorToList ()" was called too often recursively	
				5	2	2			ERR_TYPE_INVALID_STATE_UBB	Invalid state of the internal UBB connection between SCM_SL and GCMP board.	
MCU selftests errors											
				5	7	8			ERR_TYPE_SFR_TEST_FAILED	The SFR test failed during startup or cyclic execution.	
				5	7	9			ERR_TYPE_SFR_TEST_INVALID_CONFIG	The SFR test failed during startup or cyclic execution.	
				5	7	A			ERR_TYPE_PROGRAMM_FLOW_FAILED	Not all tasks of the CPU (not CODESYS) have been processed	
				5	7	B			ERR_TYPE_CPU0_SELFTEST_FAILED	A CPU self-test failed.	
				5	7	C			ERR_TYPE_CPU1_STHDL_TIMEOUT	A CPU self-test has not gone through in the expected time.	
				5	7	D			ERR_TYPE_CPU1_RAM_TEST_FAILED	The cyclic RAM test on CPU1 failed.	
				5	7	E			ERR_TYPE_CPU1_ROM_TEST_FAILED	The cyclic ROM test on CPU1 failed.	
				5	7	F			ERR_TYPE_CPU1_MARCHC_TEST_FAILED	The startup RAM test (MarchC) on CPU1 failed.	
				5	8	0			ERR_TYPE_CPU1_STACK_TEST_FAILED	The stack test failed.	
				5	8	1			ERR_TYPE_CPU1_SAFECON_TEST_FAILED	The cyclical check of the safety container via a CRC checksum has failed.	
				5	8	2			ERR_TYPE_CPU1_COREREG_TEST_FAILED	The core register test on CPU1 failed.	
				5	8	3			ERR_TYPE_CPU1_OPCODE_TEST_FAILED	The opcode test on CPU1 failed.	
				5	8	4			ERR_TYPE_CPU0_LBIST_TEST_FAILED	Logic Built-In most likely (LBIST) failed	
				5	8	5			ERR_TYPE_CPU0_PBIST_TEST_FAILED	Programmable Built-In Failed most (PBIST)	
				5	8	6			ERR_TYPE_CPU0_CCMR4F_TEST_FAILED	CPU Compare Modultest failed	
				5	8	7			ERR_TYPE_CPU0_RAM_ECC_TEST_FAILED	RAM test failed	
				5	8	8			ERR_TYPE_CPU0_FLASH_ECC_TEST_FAILED	Falsh test failed	
Sensor supplies											

Nibble	1	2	3	4	5	6	7	8	Name	Description	Internal (1)
				5	D	C			ERR_TYPE_VSENS_OUT_OF_RANGE	The value for Q_Sens is outside the permitted range	
				5	D	D			ERR_TYPE_VREF1_OUT_OF_RANGE	The value for URef1 is outside the permitted range	
				5	D	E			ERR_TYPE_VREF2_OUT_OF_RANGE	The value for URef2 is outside the permitted range	
									Watchdog timeouts		
				6	4	0			ERR_TYPE_CLOCK_FREQUENCY_CHECK_FAILED	Monitoring the clock frequency (CPU0 / CPU1) has measured an error	
				6	4	1			ERR_TYPE_MAX_SAMPLE_IRQ_TIMEOUT	Checking the watchdog between CPU0 and CPU1 failed	
				6	4	2			ERR_TYPE_CODESYS_WDT_IRQ_TIMEOUT	Checking the watchdog (CODESYS) failed	
				6	4	3			ERR_TYPE_DIAGNOSTICS_TIMEOUT	Diagnostics of the inputs / outputs failed	
				6	4	4			ERR_TYPE_IO_DATA_EXCHANGE_TIMEOUT	The IO diagnostics are not executed cyclically	
				6	4	5			ERR_TYPE_OUTPUT_TIMEOUT	Faults at outputs	
									Infrastructure		
				6	A	4			ERR_TYPE_FRAM_IS_DAMAGED	The FRAM is damaged	
				6	A	5			ERR_TYPE_SYS_LOG_IS_DAMAGED	The system logbook is corrupted	
				6	A	6			ERR_TYPE_USER_LOG_IS_DAMAGED	User logbook is corrupted	
				6	A	7			ERR_TYPE_OPERATING_HOURS_COUNTER_IS_INVALID	The operating hours counter is invalid	
				6	A	8			ERR_TYPE_STARTUP_COUNTER_IS_INVALID	The startup counter is invalid	
				6	A	9			ERR_TYPE_RETAIN_DATA_IS_DAMAGED	The reta data store is damaged	
							X		further Infomation for SPIX_ERRORS		
								X	Indexnumber: for example: Number of Input (I1.X)		

(1) Internal Error. Usable only for INTER CONTROL