Modicon M580

Safety Manual

Original instructions

10/2020

QGH46982.06

www.schneider-electric.com

The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

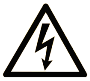
Table of Contents

Safety Information

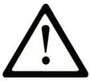
Important Information

NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.



The addition of this symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.



This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER
DANGER indicates a hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠ WARNING
WARNING indicates a hazardous situation which, if not avoided, **could result in** death or serious injury.

⚠ CAUTION
CAUTION indicates a hazardous situation which, if not avoided, **could result** in minor or moderate injury.

NOTICE
NOTICE is used to address practices not related to physical injury.

PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

BEFORE YOU BEGIN

Do not use this product on machinery lacking effective point-of-operation guarding. Lack of effective point-of-operation guarding on a machine can result in serious injury to the operator of that machine.

⚠ WARNING

UNGUARDED EQUIPMENT

- Do not use this software and related automation equipment on equipment which does not have point-of-operation protection.
- Do not reach into machinery during operation.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

This automation equipment and related software is used to control a variety of industrial processes. The type or model of automation equipment suitable for each application will vary depending on factors such as the control function required, degree of protection required, production methods, unusual conditions, government regulations, etc. In some applications, more than one processor may be required, as when backup redundancy is needed.

Only you, the user, machine builder or system integrator can be aware of all the conditions and factors present during setup, operation, and maintenance of the machine and, therefore, can determine the automation equipment and the related safeties and interlocks which can be properly used. When selecting automation and control equipment and related software for a particular application, you should refer to the applicable local and national standards and regulations. The National Safety Council's Accident Prevention Manual (nationally recognized in the United States of America) also provides much useful information.

In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the operator's hands and other parts of the body are free to enter the pinch points or other hazardous areas and serious injury can occur. Software products alone cannot protect an operator from injury. For this reason the software cannot be substituted for or take the place of point-of-operation protection.

Ensure that appropriate safeties and mechanical/electrical interlocks related to point-of-operation protection have been installed and are operational before placing the equipment into service. All interlocks and safeties related to point-of-operation protection must be coordinated with the related automation equipment and software programming.

NOTE: Coordination of safeties and mechanical/electrical interlocks for point-of-operation protection is outside the scope of the Function Block Library, System User Guide, or other implementation referenced in this documentation.

START-UP AND TEST

Before using electrical control and automation equipment for regular operation after installation, the system should be given a start-up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such a check be made and that enough time is allowed to perform complete and satisfactory testing.

⚠ WARNING

EQUIPMENT OPERATION HAZARD

- Verify that all installation and set up procedures have been completed.
- Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices.
- Remove tools, meters, and debris from equipment.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Follow all start-up tests recommended in the equipment documentation. Store all equipment documentation for future references.

Software testing must be done in both simulated and real environments.

Verify that the completed system is free from all short circuits and temporary grounds that are not installed according to local regulations (according to the National Electrical Code in the U.S.A, for instance). If high-potential voltage testing is necessary, follow recommendations in equipment documentation to prevent accidental equipment damage.

Before energizing equipment:

- Remove tools, meters, and debris from equipment.
- Close the equipment enclosure door.
- Remove all temporary grounds from incoming power lines.
- Perform all start-up tests recommended by the manufacturer.

OPERATION AND ADJUSTMENTS

The following precautions are from the NEMA Standards Publication ICS 7.1-1995 (English version prevails):

- Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly operated.
- It is sometimes possible to misadjust the equipment and thus produce unsatisfactory or unsafe operation. Always use the manufacturer’s instructions as a guide for functional adjustments. Personnel who have access to these adjustments should be familiar with the equipment manufacturer’s instructions and the machinery used with the electrical equipment.
- Only those operational adjustments actually required by the operator should be accessible to the operator. Access to other controls should be restricted to prevent unauthorized changes in operating characteristics.

About the Book

At a Glance

Document Scope

This Safety Manual describes the modules of the M580 safety system with special regard as to how they meet the Safety requirements of the IEC 61508 standard. It provides detailed information on how to install, run, and maintain the system correctly in order to protect human beings as well as to prevent damage to environment, equipment, and production.

This documentation is intended for qualified personnel familiar with Functional Safety and Control Expert Safety. Commissioning and operating the M580 Safety System may only be performed by persons who are authorized to commission and operate systems in accordance with established Functional Safety standards.

NOTE:

- The English language version of this manual is the original version.
- In case of a change request or quality issue relating to the M580 Safety offer, please contact your local Customer Care Center for Technical support. You can find more information in the *Support / Contact us* section of your Schneider Electric website at: *http://www.schneider-electric.com/b2b/en/support/*

Validity Note

This document is valid for EcoStruxure™ Control Expert Safety 15.0 or later.

For product compliance and environmental information (RoHS, REACH, PEP, EOLI, etc.), go to *www.schneider-electric.com/green-premium*.

The technical characteristics of the devices described in the present document also appear online. To access the information online:

Step	Action
1	Go to the Schneider Electric home page *www.schneider-electric.com*.
2	In the **Search** box type the reference of a product or the name of a product range. ● Do not include blank spaces in the reference or product range. ● To get information on grouping similar modules, use asterisks (*).
3	If you entered a reference, go to the **Product Datasheets** search results and click on the reference that interests you. If you entered the name of a product range, go to the **Product Ranges** search results and click on the product range that interests you.
4	If more than one reference appears in the **Products** search results, click on the reference that interests you.
5	Depending on the size of your screen, you may need to scroll down to see the datasheet.
6	To save or print a datasheet as a .pdf file, click **Download XXX product datasheet**.

The characteristics that are described in the present document should be the same as those characteristics that appear online. In line with our policy of constant improvement, we may revise content over time to improve clarity and accuracy. If you see a difference between the document and online information, use the online information as your reference.

Related Documents

Title of documentation	Reference number
Modicon M580, Safety System Planning Guide	QGH60283 (English), QGH60284 (French), QGH60285 (German), QGH60286 (Spanish), QGH60287 (Italian), QGH60288 (Chinese)
EcoStruxure™ Control Expert, Safety, Block Library	QGH60275 (English), QGH60278 (French), QGH60279 (German), QGH60280 (Italian), QGH60281 (Spanish), QGH60282 (Chinese)
Modicon Controllers Platform Cyber Security, Reference Manual	EIO0000001999 (English), EIO0000002001 (French), EIO0000002000 (German), EIO0000002002 (Italian), EIO0000002003 (Spanish), EIO0000002004 (Chinese)
Modicon M580 Hot Standby, System Planning Guide for Frequently Used Architectures	NHA58880 (English), NHA58881 (French), NHA58882 (German), NHA58883 (Italian), NHA58884 (Spanish), NHA58885 (Chinese)
Modicon M580, Hardware, Reference Manual	EIO0000001578 (English), EIO0000001579 (French), EIO0000001580 (German), EIO0000001582 (Italian), EIO0000001581 (Spanish), EIO0000001583 (Chinese)
Modicon M580 Standalone, System Planning Guide for Frequently Used Architectures	HRB62666 (English), HRB65318 (French), HRB65319 (German), HRB65320 (Italian), HRB65321 (Spanish), HRB65322 (Chinese)
Modicon M580, System Planning Guide for Complex Topologies	NHA58892 (English), NHA58893 (French), NHA58894 (German), NHA58895 (Italian), NHA58896 (Spanish), NHA58897 (Chjnese)

Title of documentation	Reference number
EcoStruxure™ Automation Device Maintenance, User Guide	EIO0000004033 (English), EIO0000004048 (French), EIO0000004046 (German), EIO0000004049 (Italian), EIO0000004047 (Spanish), EIO0000004050 (Chinese)
Unity Loader, User Guide	33003805 (English), 33003806 (French), 33003807 (German), 33003809 (Italian), 33003808 (Spanish), 33003810 (Chinese)
EcoStruxure™ Control Expert, Operating Modes	33003101 (English), 33003102 (French), 33003103 (German), 33003104 (Spanish), 33003696 (Italian), 33003697 (Chinese)
EcoStruxure™ Control Expert, System Bits and Words, Reference Manual	EIO0000002135 (English), EIO0000002136 (French), EIO0000002137 (German), EIO0000002138 (Italian), EIO0000002139 (Spanish), EIO0000002140 (Chinese)

You can download these technical publications and other technical information from our website at *www.schneider-electric.com/en/download*.

Chapter 1
M580 Safety Function

M580 Safety Function

Introducing the Schneider Electric M580 Safety Function

Using Control Expert with Safety, you can program, configure and maintain a safety application. When designing and programming your safety application, apply safety functions only to components of a safety loop.

NOTE: Include only safety modules, their configuration settings, and their data in a safety loop.

After commissioning, while your M580 safety system is operating in safety mode, the safety system periodically reads safety inputs, processes the application program safety logic, performs diagnostics, and applies the logic results to safety outputs.

If CPU or I/O diagnostics detect an error, the safety system places the affected part of the system into a safe state. Depending on the nature of the detected error, the scope of the response may place a single I/O channel, an I/O module, or the entire system into the safe state.

The safe state is always the de-energized state. For example:

- If the BMXSAI0410 analog input module or the BMXSDI1602 digital input module detects a dangerous internal condition, it sets the value of their inputs to the CPU to “0” (the de-energized state), which remain in that state until the underlying condition has been resolved.
- If the BMXSDO0802 digital output module or BMXSRA0405 digital relay output module detect a dangerous internal condition, it sets its outputs to the de-energized state, which remain in that state until the underlying condition has been resolved and the module is restarted.
- If the BMXSDO0802 digital output module or BMXSRA0405 digital relay output module detects a communication error on a black channel link to the CPU, the output module sets its outputs to their fallback state.
  NOTE: You can use Control Expert Safety to configure the fallback state (energized, de-energized, or maintain last value) in the event black channel communication between the CPU and output module is lost.
- If a BMEP58•040S standalone or a BMEH58•040S Hot Standby CPU detects a communication error on a black channel link to a safety input module, it sets the state of the affected inputs to “0” (the de-energized state) until the black channel again becomes operational and the CPU can again read actual input values.

Safety Loop

A safety loop is the collection of equipment and logic that executes a safety process. A safety project can include multiple safety loops. For each safety loop, you need verify that:

- The process safety time *(see page 150)* is greater than the system reaction time *(see page 150)*.
- The sum of the PFD or PFH values *(see page 143)* for all components in the safety loop does not exceed the maximum permitted value for the intended:
  - safety integrity level (1, 2, 3, or 4)
  - mode of operation (low demand or high demand)
  - proof test interval

Include only safety equipment in a safety loop. Although you can include non-interfering modules *(see page 27)* in your safety project, use them only for non-safe (MAST, FAST, AUX0, or AUX1) tasks.

⚠ WARNING

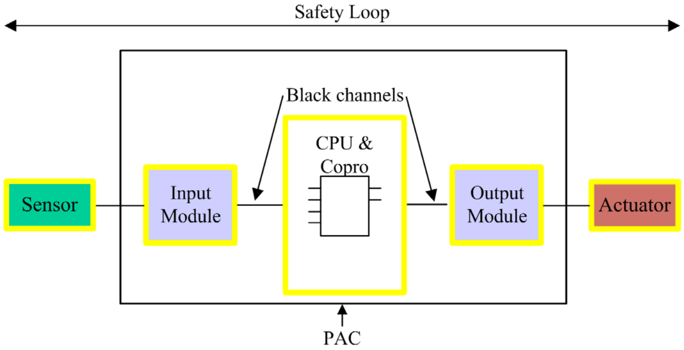
LOSS OF THE ABILITY TO PERFORM SAFETY FUNCTIONS

- Use only safety modules to perform safety functions.
- Do not use inputs or outputs of non-interfering modules for safety-related functions.
- Do not use variables from the Global area for safety-related functions.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Refer to the topic *Data Separation in an M580 Safety Project (see page 166)* for a description of global area variables.

Safety Loop:



Safety equipment includes the following Schneider Electric M580 safety modules:

- BME•58•040S CPU & BMEP58CPROS3 Copro:
  The CPU & Copro together perform the tasks of reading safety inputs, processing safety logic, performing diagnostics, and applying results to outputs. All of these tasks are part of the safety loop. Ports used for black channel communications are also part of the safety loop. However, other CPU components – for example the USB port, SD memory card, and non-volatile static random access memory (nvSRAM) area – are not part of the safety loop.
  NOTE: On both a cold and a warm system start, the CPU & Copro do not load data stored in nvSRAM into the safety task. (nvSRAM data is used only in the non-safe MAST, FAST, and AUX tasks). Instead, the CPU & Copro initially apply default configuration settings from the SD memory card, then apply values received directly from inputs during operation.
- Safety I/O (BMXSAI0410, BMXSDI1602, BMXSDO0802, and BMXSRA0405):
  The functions of sending input signals, receiving output signals, and performing diagnostics are part of the safety loop.
- BMXCPS4002S, BMXCPS4022S, and BMXCPS3522S power supplies:
  These safety power supplies provide over-voltage detection, and this is part of the safety loop. Because each power supply reliability (i.e. its dangerous failure rate) is more than 100 times better than the threshold for the SIL3 standard, these safety power supplies are not included in safety integrity level calculations for the safety loop.

The safety loop also includes the following non-safety equipment:

- Sensors, actuators and the cabling that connects them to safety I/O modules.The safety I/O perform wiring diagnostics for sensors and actuators to help manage the safety loop.
  NOTE: When you design your safety application, you need to identify sensor and actuator characteristics (in particular PFD/PFH values).

Chapter 2
Certification Standards

Introduction

This chapter describes the certification standards that apply to the M580 safety system and its component modules.

What Is in This Chapter?

This chapter contains the following topics:

Certifications

M580 Safety PAC Certification Standards

The M580 safety PAC is certified by TÜV Rheinland Group for use in applications up to and including safety integrity level 3 (SIL3) *(see page 389)*.

Programmable controller specifications

- IEC 61131-2: Programmable controllers- Part 2: Equipment requirements and tests.
- IEC/EN 61010-2-201, UL 61010-2-201, CSA -C22.2 No. 61010-2-201: Safety requirements for electrical equipment - Part 2-201: Particular requirements for control equipment.

Environmental specifications

Refer to "M580 Safety Standards and Certifications" for environment tests levels.

Ex areas specifications

For USA and Canada: Hazardous location class I, division 2, groups A, B, C and D

- CSA 22.2 No213, ANSI/ISA12.12.01 and FM3611

For other countries: CE ATEX (directive 2014/34/EU) or IECEx in defined atmosphere Zone 2 (gas) and/or Zone 22 (dust)

- IEC/EN 60079-0 ; IEC/EN 60079-7; IEC/EN 60079-15

Power utility automation systems specifications

- IEC/EN 61000-6-5: Electromagnetic compatibility - Part 6-5: Generic standards - Immunity for power station and substation environments.
- IEC/EN 61850-3: Communication networks and systems for power utility automation - Part 3: General requirements

Refer to M580 Standards and Certifications *(see page 24)* for installation restrictions.

Railway specifications

- EN 50155 / IEC 60571: Railway applications - Rolling stock - Electronic equipment.
- EN 50121-3-2 / IEC 62236-3-2: Railway applications - Electromagnetic compatibility - Part 3-2: Rolling stock - Apparatus.
- EN 50121-4 / IEC 62236-4: Railway applications - Electromagnetic compatibility - Part 4: Emission and immunity of the signaling and telecommunications apparatus.
- EN 50121-5 / IEC 62236-5: Railway applications - Electromagnetic compatibility - Part 5: Emission and immunity of fixed power supply installations and apparatus.

Refer to M580 Standards and Certifications *(see page 24)* for installation restrictions.

Functional safety specifications

- IEC/EN 61000-6-7: Electromagnetic compatibility - Part 6-7: Generic standards - Immunity requirements for equipment intended to perform functions in a safety-related system (functional safety) in industrial locations.
- IEC 61326-3-1: Electrical equipment for measurement, control and laboratory use - Part 3-1: Immunity requirements for safety-related systems and for equipment intended to perform safety-related functions - General industrial application.
- IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 1-7, edition 2.0.
- IEC 61511-1: Functional safety - Safety instrumented systems for the process industry sector - Part 1: Framework, definitions, system, hardware and software requirements.
- IEC 61511-2: Functional safety - Safety instrumented systems for the process industry sector - Part 2: Guidelines for the application of IEC 61511-1.
- IEC 61511-3: Functional safety - Safety instrumented systems for the process industry sector - Part 3: Guidance for the determination of the required safety integrity levels.

Safety machinery specifications

- IEC/EN 62061: Safety of machinery - Functional safety of safety-related electrical, electronic and programmable electronic control systems.
- ISO EN 13849-1: Safety of machinery - Safety-related parts of control systems - Part 1: General principles for design.

Functional safety in systems specifications

- EN 54-2: Fire detection and fire alarm systems Part 2: Control and indicating equipment.
- EN 50156-1: Electrical equipment for furnaces and ancillary equipment - Part 1: Requirements for application design and installation.
- EN 50130-4: Alarm systems - Part 4: Electromagnetic compatibility - Product family standard: Immunity requirements for components of fire, intruder, hold up, CCTV, access control and social alarm systems.
- EN 298: Automatic burner control systems for burners and appliances burning gaseous or liquid fuels.
- NFPA 85: Boiler and Combustion Systems Hazards Code.
- NFPA 86: Standard for Ovens and Furnaces.
- NFPA 72: National Fire Alarm and Signaling Code.

Notes:

For the complete list of the standards (with their revisions and dates) which are certified by TÜV, please refer to the TÜV certificate in the web site at:

www.certipedia.com or www.fs-products.com.

Standards and Certifications

Download

Click the link that corresponds to your preferred language to download standards and certifications (PDF format) that apply to the modules in this product line:

Title	Languages
Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications	● English: *EIO0000002726* ● French: *EIO0000002727* ● German: *EIO0000002728* ● Italian: *EIO0000002730* ● Spanish: *EIO0000002729* ● Chinese: *EIO0000002731*

Chapter 3
M580 Safety System Supported Modules

Introduction

An M580 safety project can include both safety modules and non-safety modules. You can use:

- Safety modules in the SAFE task.
- Non-safety modules only for the non-safe tasks (MAST, FAST, AUX0, and AUX1).

  NOTE: Only non-safety modules that do not interfere with the safety function can be added to a safety project.

Use only the Control Expert programming software of Schneider Electric for programming, commissioning, and operating your M580 safety application.

- Control Expert L Safety provides all the functionality of Control Expert L and can be used with BMEP582040S and BMEH582040S safety CPUs.
- Control Expert XL Safety provides all the functionality of Control Expert XL and can be used for the entire range of BMEP58•040S and BMEH58•040S safety CPUs.

This chapter lists the safety and non-safety modules supported by the M580 safety system.

What Is in This Chapter?

This chapter contains the following topics:

M580 Safety System Certified Modules

Certified Modules

The M580 safety PAC is a safety-related system certified by TÜV Rheinland Group, according to:

- SIL3/IEC 61508/IEC 61511
- SIL CL3/IEC 62061
- PLe, Cat. 4 / ISO 13849-1
- CIP Safety IEC 61784-3

It is based on the M580 family of programmable automation controllers (PACs). The following Schneider Electric M580 safety modules are certified:

- BMEP584040S standalone CPU
- BMEP582040S standalone CPU
- BMEH582040S Hot Standby CPU
- BMEH584040S Hot Standby CPU
- BMEH586040S Hot Standby CPU
- BMEP58CPROS3 co-processor
- BMXSAI0410 analog input module
- BMXSDI1602 digital input module
- BMXSDO0802 digital output module
- BMXSRA0405 digital relay output module
- BMXCPS4002S power supply
- BMXCPS4022S power supply
- BMXCPS3522S power supply

NOTE: In addition to the safety modules listed above, you can also include non-interfering, non-safety modules *(see page 27)* in your safety project.

You can find the most recent information on the certified product versions on the TÜV Rheinland Group website: www.certipedia.com or www.fs-products.com.

Replacing a CPU

It is possible to replace a BME•58•040S CPU with another BME•58•040S. However, the replacement does not work if the following limitations are exceeded :

- number of I/O
- number of I/O drops
- number of variables
- application memory size

Refer to the topics:

- *Configuration Compatibility* in the *Modicon M580 Hot Standby System Planning Guide for Frequently Used Architectures* for a description of Control Expert applications that are compatible with safety and Hot Standby CPUs.
- *M580 CPU & Copro Performance Characteristic*s in the *Modicon M580 Safety System Planning Guide* for a description of CPU limitations.

Non-Interfering Modules

Introduction

An M580 safety project can include both safety modules and non-safety modules. You can use non-safety modules only for non-safe tasks. Only non-safety modules that do not interfere with the safety function can be added to a safety project.

Definition of a Non-Interfering Module

⚠ CAUTION

INCORRECT USE OF SAFETY-RELATED DATA

Confirm that neither input data nor output data from non-interfering modules are used for controlling safety-related outputs. Non-safety modules can process only non-safety data.

Failure to follow these instructions can result in injury or equipment damage.

A non-interfering module is a module which cannot interfere with the safety function. For in-rack M580 modules (BMEx, BMXx, PMXx, and PMEx), there are two types of non-interfering modules:

- **Type 1**: A type 1 module can be installed in the same rack as safety modules (wherever the safety module is placed, in the main or extension rack).
- **Type 2**: A type 2 non-interfering module cannot be installed in the same main rack as safety modules (wherever the safety module is placed, in the main or extension rack).

NOTE: Type 1 and Type 2 modules are listed on TÜV Rheinland website at https://fs-products.tuvasi.com.
For not in-rack Mx80 modules, all Ethernet equipment (DIO or DRS) can be considered as non-interfering, and therefore can be used as part of an M580 safety system.

Type 1 Non-Interfering Modules for SIL3 Applications

The following non-safety modules can qualify as type-1 non-interfering modules in an M580 safety system.

NOTE: The list of type-1 non-interfering non-safety modules may change from time to time. For the current list, visit the TÜV Rheinland website at https://fs-products.tuvasi.com.

Module type	Module Reference
Backplane 4 slots	BMEXBP0400
Backplane 8 slots	BMEXBP0800
Backplane 12 slots	BMEXBP1200
Backplane 4 slots	BMXXBP0400
Backplane 6 slots	BMXXBP0600
Backplane 8 slots	BMXXBP0800
Backplane 12 slots	BMXXBP1200
Backplane 6 slots with dual slots for redundant power supplies	BMEXBP0602
Backplane 10 slots with dual slots for redundant power supplies	BMEXBP1002
Communication: Performance X80 Ethernet Drop Adapter 1 CH	BMXCRA31210
Communication: Performance X80 Ethernet Drop Adapter 1 CH	BMECRA31210
Communication: Ethernet module with standard web services	BMENOC0301
Communication: Ethernet module with IP Forwarding	BMENOC0321
Communication: Ethernet module with FactoryCast web services	BMENOC0311
Communication: Rack extender module	BMXXBE1000
Communication: AS-Interface	BMXEIA0100
Communication: Global Data	BMXNGD0100
Communication: Fiber Converter MM/LC 2CH 100Mb	BMXNRP0200
Communication: Fiber Converter SM/LC 2CH 100Mb	BMXNRP0201
Communication: M580 IEC 61850 Communication module	BMENOP0300
Communication: Embedded OPC UA Server	BMENUA0100
Counting: SSI module 3 CH	BMXEAE0300
Counting: High speed counter 2 CH	BMXEHC0200
Counting: High speed counter 8 CH	BMXEHC0800
Motion: Pulse Train Output 2 independent CH	BMXMSP0200
Analog: Ana 8 In Current Isolated HART	BMEAHI0812
Analog: Ana 4 Out Current Isolated HART	BMEAHO0412
Analog: Ana 4 U/I In Isolated High Speed	BMXAMI0410
Analog: Ana 4 U/I In Non Isolated High Speed	BMXAMI0800

Module type	Module Reference
Analog: Ana 8 U/I In Isolated High Speed	BMXAMI0810
Analog: Ana 4 In U/I 4 Out U/I	BMXAMM0600
Analog: Ana 2 U/I Out Isolated	BMXAMO0210
Analog: Ana 4 U/I Out Isolated	BMXAMO0410
Analog: Ana 8 Out Current No Isolated	BMXAMO0802
Analog: Ana 4 TC/RTD Isolated In	BMXART0414.2
Analog: Ana 8 TC/RTD Isolated In	BMXART0814.2
Discrete: Dig 8 In 220 Vac	BMXDAI0805
Discrete: Dig 8 In 100 to 120 Vac Isolated	BMXDAI0814
Discrete: Dig 16 In 24Vac/24Vdc Source	BMXDAI1602
Discrete: Dig 16 In 48Vac	BMXDAI1603
Discrete: Dig 16 In 100 to 120 Vac 20 pin	BMXDAI1604
Discrete: Dig 16 Supervised inputs channels 100 to 120 Vac 40 pin	BMXDAI1614
Discrete: Dig 16 Supervised inputs channels 200 to 240 Vac 40 pin	BMXDAI1615
Discrete: Dig 16 Outputs Triacs 100 to 240 Vac 20 pin	BMXDAO1605
Discrete: Dig 16 Outputs Triacs 24 to 240 Vac 40 pin	BMXDAO1615
Discrete: Dig 16 In 24Vdc Sink	BMXDDI1602
Discrete: Dig 16 In 48Vdc Sink	BMXDDI1603
Discrete: Dig 16 In 125Vdc Sink	BMXDDI1604T
Discrete: Dig 32 In 24Vdc Sink	BMXDDI3202K
Discrete: Dig 64 In 24Vdc Sink	BMXDDI6402K
Discrete: Dig 8 In 24Vdc 8Q Source Tr	BMXDDM16022
Discrete: Dig 8 In 24Vdc 8Q Relays	BMXDDM16025
Discrete: Dig 16 In 24Vdc 16Q Source Tr	BMXDDM3202K
Discrete: Dig 16Q Trans Source 0.5A	BMXDDO1602
Discrete: Dig 16 O Trans Sink	BMXDDO1612
Discrete: Dig 32Q Trans Source 0.1A	BMXDDO3202K
Discrete: Dig 64Q Trans Source 0.1A	BMXDDO6402K
Discrete: Dig 8Q 125Vdc	BMXDRA0804T
Discrete: Dig 8Q 24 Vdc or 24 to 240 Vac Isolated Relays	BMXDRA0805
Discrete: Dig 16 non-isolated relay output channels 5 to 125 Vdc or 25 to 240 Vac	BMXDRA0815
Discrete: Dig 16Q Relays	BMXDRA1605
Discrete: Dig NC Output 5 to 125 Vdc or 24 to 240 Vac Relays	BMXDRC0805
Discrete: Dig 16In 24/125Vdc TSTAMP	BMXERT1604

Module type	Module Reference
Mx80 Network Option Switch	BMENOS0300
Turbomachinery Frequency Input 2 CH	BMXETM0200
Profibus DP/DPV1 Master module support	PMEPXM0100

Type 2 Non-Interfering Modules for SIL2/3 Applications

The following in-rack non-safety modules can be considered to be type-2 non-interfering modules in an M580 safety system.

NOTE: The list of type-2 non-interfering non-safety modules may change from time to time. For the current list, visit the TÜV Rheinland website at https://fs-products.tuvasi.com.

Module type	Module Reference
Communication: Standard X80 Ethernet Drop Adapter 1 CH	BMXCRA31200
Standard AC power supply	BMXCPS2000
Standard Isolated DC power supply	BMXCPS2010
High Power Isolated 24 to 48 VDC power supply	BMXCPS3020
Standard Redundant 125VDC power supply	BMXCPS3522
Standard Redundant 24/48VDC power supply	BMXCPS4022
Standard Redundant AC power supply	BMXCPS4002
High Power AC power supply	BMXCPS3500
High Power DC power supply	BMXCPS3540T
Communication: Bus module 2 RS485/232 Port	BMXNOM0200
CANopen X80 Master	BMECXM0100
Weight module	PMESWT0100
Partner diagnostic module	PMXCDA0400

NOTE: All authorized equipment of an M580 system that are linked to safety modules via Ethernet are considered as non-interfering. As a consequence, all modules from Quantum and STB Advantys ranges (not pluggable in the same rack as M580 safety modules) are Type 2 non-interfering modules.

Chapter 4
Cyber Security for the M580 Safety System

Cyber Security for the M580 Safety System

Cyber Security Reference

The purpose of a cyber security policy is to reduce, to the greatest extent possible, the vulnerability of your safety system to cyber attacks. For information on developing a cyber security policy for your M580 safety system, refer to the *Modicon Controllers Platform Cyber Security Reference Manual* (Reference Number EIO0000001999 (EN).

Chapter 5
Application Lifecycle

Application Lifecycle

Introduction

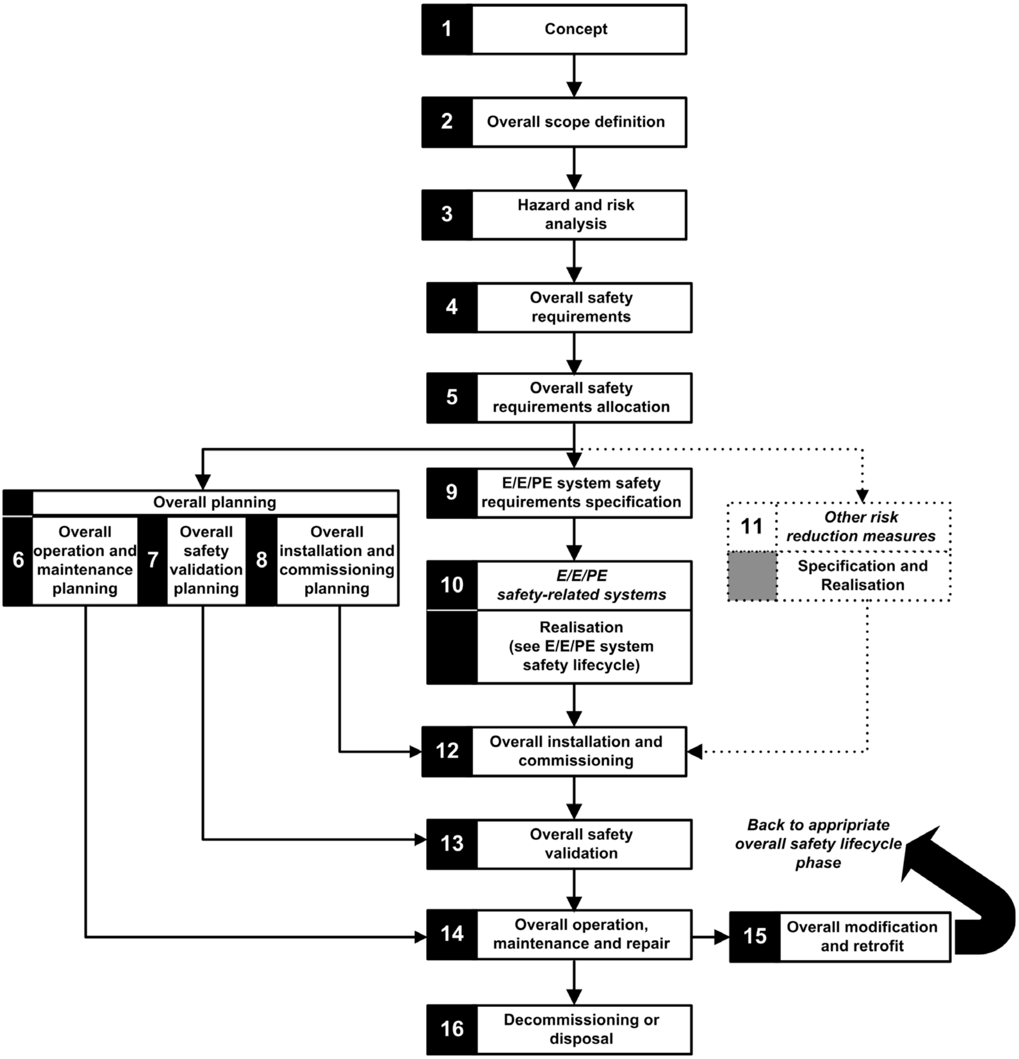
When designing a safe application, you will need to follow the recommendation of one of the safety standard which apply to your application domain. Most of the application standards derive from or are linked to the generic standard IEC 61508 including, for example, the process industry standard (IEC 61511), the machine industry standards (IEC 62061 and ISO 13489), the nuclear industry standard (IEC 61513), and so forth.

IEC 61508 defines an application life cycle with a sequence of steps. Each step has a defined role, needs mandatory input documents, and produces output documents. The decision to use a safety integrated system (SIS) is made at the end of the Safety Requirements Allocation step (step 5).

This topic defines the necessary checks, related to the usage of a M580 safety system, that you need to perform at the following steps:

9. E/E/PE System safety requirements specification
10. E/E/PE Safety related systems realisation
12. Overall installation and commissioning
13. Overall safety validation
14. Overall operation, maintenance and repair
15. Overall modification and retrofit

The following diagram presents the overall safety lifecycle:


1 Concept
2 Overall scope definition
3 Hazard and risk analysis
4 Overall safety requirements
5 Overall safety requirements allocation
Overall planning
6 Overall operation and maintenance planning
7 Overall safety validation planning
8 Overall installation and commissioning planning
9 E/E/PE system safety requirements specification
10 E/E/PE safety-related systems
Realisation (see E/E/PE system safety lifecycle)
11 Other risk reduction measures
Specification and Realisation
12 Overall installation and commissioning
13 Overall safety validation
Back to appripriate overall safety lifecycle phase
14 Overall operation, maintenance and repair
15 Overall modification and retrofit
16 Decommissioning or disposal

Step 9: E/E/PE System safety requirements specification

This step takes place when the risk analysis is completed and has provided, among other things, the following information:

- Definition of the safety integrated functions
- Their required performances (time, risk reduction, SIL…)
- Their failure modes

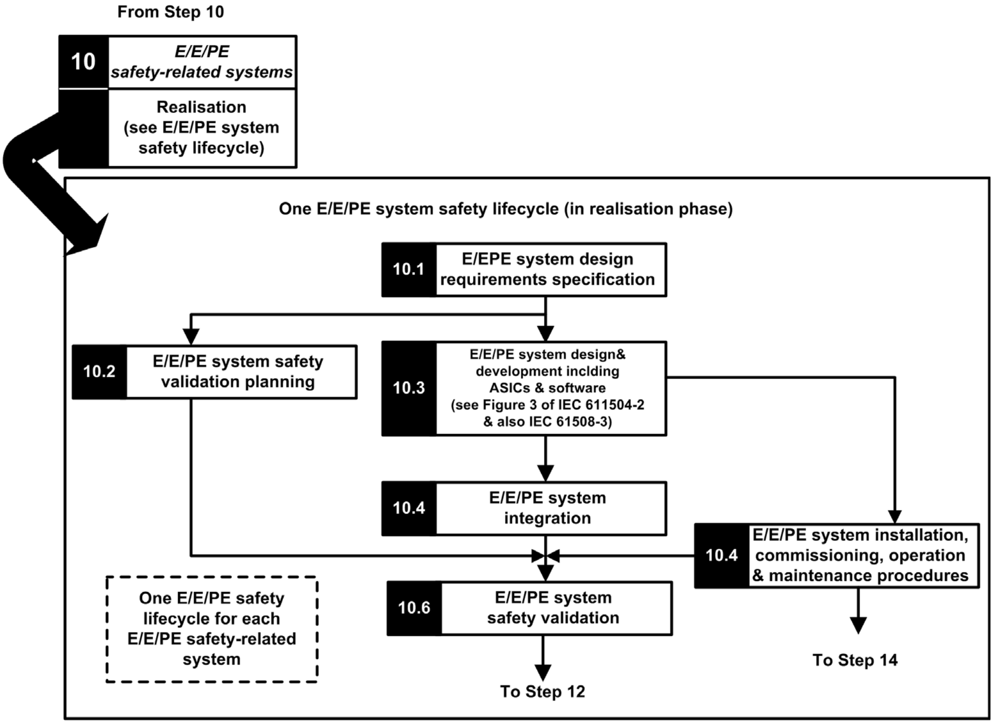
It should produce the safety requirement specifications which will include, at least, the following information necessary to design a safe application using any type of safety PAC:

- Safe state of the safety integrated functions
- SIS operating mode analysis (including the behavior in run, stop, power on sequence, maintenance, repair…)
- Test interval of the SIF
- MTTR of the SIS
- Choice of energized or de-energized SIF
- Performance of the logic solver (reaction time, precision …)
- Performance requirements
  - Fault tolerance
  - Integrity
  - Maximum spurious trip rate
  - Maximum dangerous fault rate
- Environmental specification (EMC, mechanical, chemical, climate…)

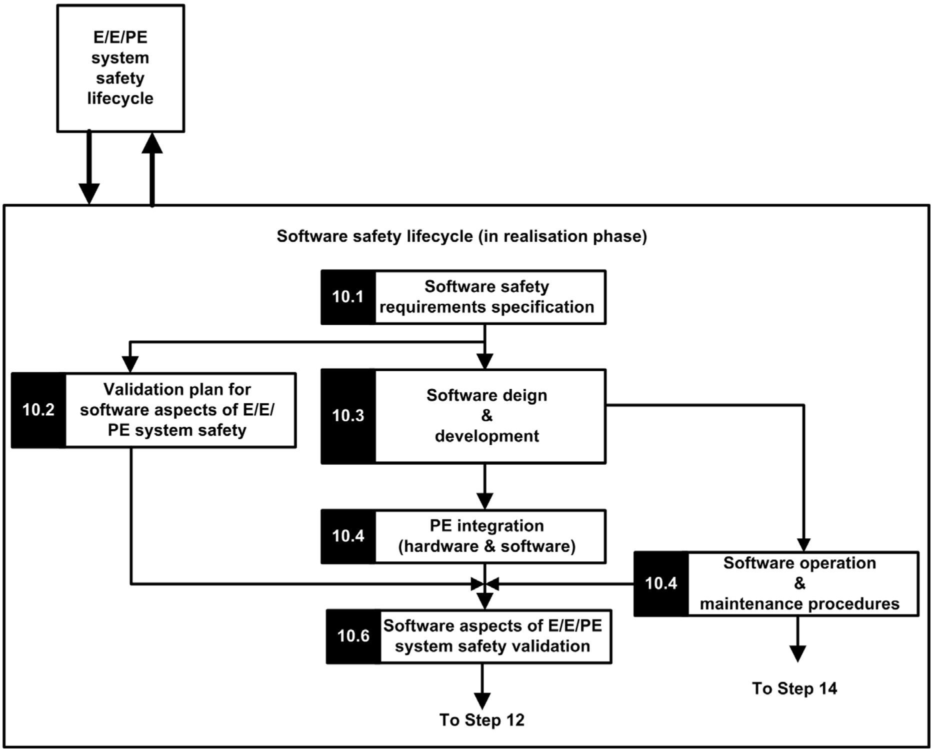
Step 10: E/E/PE Safety related systems realisation

The IEC 61508 divide this step into 2 sub life cycles, one for the system realisation, one for the software realisation.

System realisation:



Software realisation:


E/E/PE system safety lifecycle
Software safety lifecycle (in realisation phase)
10.1 Software safety requirements specification
10.2 Validation plan for software aspects of E/E/PE system safety
10.3 Software deign & development
10.4 PE integration (hardware & software)
10.4 Software operation & maintenance procedures
10.6 Software aspects of E/E/PE system safety validation
To Step 12
To Step 14

The goal of the first sub steps (10.1) is to convert the SIS safety requirements into specification of the hardware design, hardware tests, software design, software tests and integration tests. It should provide at least the following information necessary to design a safe application using safety M580:

- Hardware architecture taking care of:
  - The respect of M580 rules about mixing non safety and safety modules: all the safety modules (safe IO modules and safe CPU/COPRO) are placed in racks where the main rack and its extension are powered by safe power supply and contains only safe modules or non-interfering modules of type 1.
  - Electric consumption per rack.
  - Derating rules.
- Power supply architecture:
  - Only SELV/PELV power supply.
- Software architecture:
  - Including the usage of M580 global variables; a global variable should not prevent a safety action to be triggered unless a “safe application protocol” is used.

- Hardware integration (cabling, cabinet, and so forth):”
  - Fuse protection.
  - Accessories for wire diagnostic.
- Human machine interfaces:
  - Including the usage of M580 global variables; a global variable should not prevent a safety action to be triggered unless a “safe application protocol” is used.
- Electric/numerical interfaces:
  - Safe state.
  - Sensor and actuator.
- Algorithm
- Performances (including task period, watchdog and timeout definition) and prediction of a good behavior using the formula:

  $$\Sigma_{all\ tasks} \frac{Exe_{task}}{Period_{task}} < 80\%$$

  NOTE: The formula is applicable only when MAST task is not in cyclic mode.
- Behavior in case of:
  - Unlock configuration
  - Maintenance mode
  - Maintenance input
  - Invalid channel
  - Wiring failure
  - Channel health
  - Module health
- Management of the UID of the safe IO modules (define when a UID should be changed).
- NTP server:
  - Choice of PAC as NTP server or external NTP server (depending on the usage of I/O time stamping in the process application).
  - Server redundancy
  - Server loss

The next sub steps refine the specifications into technical detailed specification, perform the design itself execute all the test plans and provides the reports.

Step 12: Overall installation and commissioning

The goal of this step is to define the requirements for installation, task planning, tooling, commissioning procedure and then build the system and verify its correctness.

- For Hot Standby applications, verify that the fallback timeout *(see page 153)* of the safety output modules fits the conditions defined for swap *(see page 154)* and switchover *(see page 155)* operations, and verify the CRA hold-up time.
- Verify that fallback safety timeout (S_TO) for the safety output modules is, at least, greater than the greater of 40 ms or (2.5 * T_{SAFE}), where T_{SAFE} equals the configured SAFE task period.
- Clear any pre-existing application inside the PLC, or use an application configured with no CIP safety devices before installing the safety device onto an Ethernet safety network (with CIP safety devices).

In an M580 safety system, the commissioning procedure should include the following points:

- Verify Control Expert integrity, verify Control Expert version.
- Correctness of the CPU and Coprocessor firmware versions by supervising the system words %SW14 (Firmware version of PLC processor) and %SW142 (Firmware version of coprocessor).
- Correctness of each module addresses (position in rack, CRA switches).
- Correctness of the cabling:
  - Point to point verification: from internal variable to IO module and to actuator/sensor.
  - Fuses.
  - Equipment for wiring diagnostic.
- At the end of the procedure, all the safety modules are in “lock” mode (it is recommended that the safe application itself checks this condition).
- Correctness of each module configuration (including the timeouts):”
  - Read the configuration using the Control Expert screen and compare to specification.
- All the safety applications have been rebuilt using the **Rebuild All Project** option, then downloaded to each PLC, and their SAId saved as well as the application archive.
- The task period and task watchdog are correct.
- Module references and version.
- Usage of SELV/PELV only.

- If CIP Safety devices are used in the safety application:
  - The Safety Configuration ID signature (SCID) can be considered to be verified (option enabled in Control Expert CIP Safety DTM) and target configuration locked after user testing.
  - To confirm that the originator configuration created by the user with the Control Expert software tool was correctly sent to and saved in the M580 CIP Safety originator, visually compare all the CIP Safety target configuration parameter values displayed in the target DDDTs (in connected mode with the PAC, using an Animation table) with the parameter values displayed and configured in the target DTM Configuration verification tab *(see page 353)*. All of the values need to be identical.
  - Test all safety connection configurations after they are applied in the M580 CIP Safety originator to confirm that each target connection is operating as intended.
  - Before installing the CIP Safety devices into the safety network, commission all the safety devices with MacId and Baud Rate as necessary.
- User testing is the means by which all application downloads are validated

Step 13: Overall safety validation

The goal of this step is to prove that the safety integrated system fulfills its requirements. It executes all the tests and produce the reports define in the step 7 of the “safety lifecycle”. It should include:

- Verify that there is no overrun condition during any of the system state (verification of the system bit %S19 in the MAST, FAST, AUX0 tasks), and that the max and current SAFE task execution time (%SW42 and %SW43) are below the SAFE task Period.
- Verify the CPU load formula: $\Sigma_{all\ tasks} \frac{Exe_{task}}{Period_{task}} < 80\%$

  NOTE: You can use the system words %SW110 to %SW115 *(see page 396)* to perform a real-time evaluation of the average load for CPU tasks (if all tasks are periodic, %SW116 should be less than 80).
- Verify the special operating modes (module unlock, maintenance input, invalid channel, wiring defect).
- For Hot Standby applications, verify that all tasks are correctly synchronized through the Hot Standby link by checking and using the MAST_SYNCHRONIZED, FAST_SYNCHRONIZED, and SAFE-SYNCHRONIZED bits in the T_M_ECPU_HSBY DDT. Refer to the *Modicon M580 Hot Standby, System Planning Guide for Frequently Used Architectures* for a description of the T_M_ECPU_HSBY DDT.

Step 14: Overall operation, maintenance and repair

- Execute the proof tests at the right period.
- Monitor the SAId see note.

  NOTE: As long as the SAId has not changed, the safety portion of the application has not been changed. Refer to the S_SYST_STAT_MX function block for details on SAId behavior.
- Monitor the configuration lock status of each safety module.
- Record the repair operations.
- If a module is replaced, the replacement device needs to be configured properly and you (the user) need to verify its operation. Execute (at minimum) the commissioning operations related to this module.
- Record the deviations.

Step 15: Overall modification and retrofit

Any modification should be treated as a new design. An impact analysis may help to define the part of the former safety system that can be kept and the part that must be designed again.

NOTE: If an application modification does not concern the SAFE application, you can use the SourceSafeSignature to verify that no unwanted modification has been introduced to the SAFE code. The SourceSafeSignature is an *a priori* verification that the application is unchanged. SourceSafeSignature does not replace the SAId, which is the only measure that reliably confirms a PAC is executing the same SAFE application that was validated.

Chapter 6
M580 Safety I/O Modules

Introduction

This chapter describes the M580 safety I/O modules.

What Is in This Chapter?

This chapter contains the following sections:

Section 6.1
M580 Safety I/O Module Shared Features

Introduction

This section describes the shared or common features of M580 safety I/O modules.

What Is in This Section?

This section contains the following topics:

Introducing the M580 Safety I/O Modules

Introduction

The following four M580 safety I/O modules are certified for use in safety applications:

- BMXSAI0410 (Analog Input)
- BMXSDI1602 (Digital Input)
- BMXSDO0802 (Digital Output)
- BMXSRA0405 (Digital Relay Output)

Use the four safety I/O modules to connect the safety PAC to the sensors and actuators that are part of the safety loop. Each safety I/O module incorporates a dedicated safety processor. You can install these I/O modules in the local backplane or in RIO drops.

Installation and Housing Requirements

Install your M580 safety equipment in a manner that meets:

- The IEC 60950 pollution degree 2 standard for the safety of information technology equipment; and
- IEC 60529 standard for IP54 ingress protection, so that:
  - the presence of dust does not interfere with equipment operation, and
  - splashing water has no harmful effect on the equipment or operations.

Typically these standards are accomplished by placing the safety equipment in a housing, such as a cabinet.

Maximum Operation Altitude

The maximum operating altitude for the M580 safety I/O modules is 2000 m above sea level.

Communication Between the PAC and I/O

The M580 safety CPU and Copro together control all backplane exchanges, while the safety I/O respond to the commands of the CPU and Copro. Safety I/O modules can be installed in either a BMXXBP•••• X Bus rack or a BMEXBP•••• Ethernet rack.

Communications between the safety PAC and safety I/O modules in the local main rack are made via the backplane.

Communications between the safety PAC and safety I/O modules installed in an RIO drop are made through an adapter module installed on the RIO drop, either:

- a BMECRA31210 adapter, for an Ethernet rack, or
- a BMXCRA31210 adapter, for an X Bus rack.

NOTE: With CPU firmware 3.20 or later, the communication between the PAC and safety I/O requires a BM•CRA31210 with a firmware 2.60 or later.

NOTE: A BMXCRA31200 adapter cannot be used to connect safety I/O modules to the M580 safety PAC.

Optionally, you can use BMXNRP0200 or BMXNRP0201 fiber optic repeater modules to extend the physical link between the CPU and Copro in the local rack and the adapter in the RIO drop. Fiber optic repeater modules enhance RIO network noise immunity and increase cabling distance while maintaining the full dynamic range of the network and the safety integrity level.

The communication protocol between the safety I/O and PAC enables their exchanges. It permits both devices to check the accuracy of received data, detect corrupted data, and determine if the transmitting module becomes non-operational. Thus, a safety loop may include any non-interfering *(see page 27)* RIO adapters and backplane.

External Power Supply Used with Digital Safety I/O

The BMXSDI1602 and BMXSDO0802 digital modules require an 24 Vdc protected extra low voltage (PELV) external power supply to provide power to sensors and actuators. The safety I/O modules supervise the non-safety process power supply for overvoltage and undervoltage conditions.

⚠ ⚠ DANGER

PELV OVERVOLTAGE CATEGORY II POWER SUPPLY REQUIRED

Use only a PELV type overvoltage category II power supply, with a maximum output of 60 Vdc, to supply power to sensors and actuators.

Failure to follow these instructions will result in death or serious injury.

Diagnostics Overview for M580 Safety I/O Modules

Introduction

Each M580 safety I/O module presents the following diagnostic features:

- Self-test at module start-up
- Continuous built-in runtime self-test
- Module and channel diagnostic LEDs

In addition, the digital safety I/O modules also perform wiring diagnostics.

Power On Self Test

At power up, the I/O modules perform an extended series of power on self-tests. If the result of these tests are:

- Successful: The modules are considered to be healthy and are operational.
- Unsuccessful: The modules are not considered to be healthy and are not operational. In this case, the inputs are set to 0, and the outputs are de-energized.

NOTE: If the 24 Vdc external power supply is not connected to a digital input or digital output module, the power on self-tests are not performed and the module does not start.

Continuous Built-In Tests

During runtime, the I/O modules continuously perform self-tests. The input modules verify that they are able to read data from the sensors over the complete range. The output modules verify that the actual state of the output is the same as the commanded state.

LEDs

Each safety I/O module provides module and channel LED diagnostics on the front face of the module:

- The top four LEDs (**Run**, **Err**, **I/O**, and **Lck**) together describe the state of the module.
- The bottom two or four (depending on the module) rows of LEDs combine with the top four LEDs to describe the state and health of each input or output channel.

Refer to the LED diagnostics topic for the following safety I/O modules for more information on the how to read the LEDs for that module:

- BMXSAI0410 safety analog input module *(see page 227)*
- BMXSDI1602 safety digital input module *(see page 233)*
- BMXSDO0802 safety digital output module *(see page 240)*
- BMXSRA0405 safety digital relay output module *(see page 246)*

Wiring Diagnostics for Digital Modules

Both the safety digital input module and the safety digital output module can detect the following channel wiring diagnostic conditions:

- Open (or cut) wire.
- Short circuit to the 0 V ground.
- Short circuit to the 24 Vdc.
- Crossed circuits between two channels.

NOTE: The availability of these diagnostic functions depends on the specific wiring design of the module to its field devices. Refer to the application wiring examples for the following safety digital I/O modules for more information:

- BMXSDI1602 safety digital input module *(see page 70)*
- BMXSDO0802 safety digital output module *(see page 98)*

Section 6.2
BMXSAI0410 Analog Input Module

Introduction

This section describes the BMXSAI0410 M580 safety analog input module.

What Is in This Section?

This section contains the following topics:

BMXSAI0410 Safety Analog Input Module

Introduction

The BMXSAI0410 safety analog input module presents the following features:

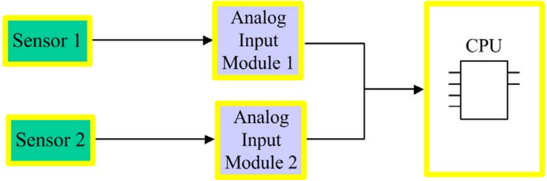
- 4 isolated analog 4...20 mA current input channels.
- 12500 resolution counts, spanning the data range of 0...25 mA.
- Current out of range detection, for current values less than 3.75 mA or greater than 20.75 mA.
- Supports the following SIL3 (IEC61508) standards:
  - Category 2 (Cat2) / Performance Level d (PLd) is achieved using 1 input channel (one-out-of-one (1oo1D) evaluation).
  - Cat4/PLe is achieved using 2 input channels (one-out-of-two (1oo2D) evaluation).
- LED diagnostic *(see page 227)* display provided for the module and for each input channel.
- Module hot swap during runtime.
- Module CCOTF when operating in maintenance mode *(see page 255)*. (CCOTF is not supported in safety mode *(see page 254)*.)

High Availability

You can design your safety application to varying levels of performance and availability, by using single or redundant input channels and modules, as follows:

Design:	Safety Function Levels:			
Input Channels => Modules	SIL	Cat	PL	High Availability?
Single input channel to single input module *(see page 55)*	SIL3	Cat 2	PLd	–
Single input channel to redundant input modules *(see page 56)*	SIL3	Cat 2	PLd	✓
Redundant input channels to single input module *(see page 57)*	SIL3	Cat 4	PLe	–
Redundant input channels to redundant input modules *(see page 58)*	SIL3	Cat 4	PLe	✓
✓ : Provided – : Not provided				

The following figure illustrates the redundant analog input configuration:



The analog input current value from sensor 1 and sensor 2 are sent by input module 1 and input module 2, respectively, over a black channel to a safety CPU. The CPU executes a dedicated function block, (S_AIHA, in each of two separate, compiled logic programs to manage and select data from the two input modules. This function block operates as follows:

- If the health status of the input data coming from module 1 is OK, the input data from this module is used in the safety function.
- If the health status of the input data coming from module 1 is not OK, but the health status of the input data coming from module 2 is OK, the input data from module 2 is used.
- If the health status of the input data from both module 1 and module 2 is not OK, then the system activates the safety function.

BMXSAI0410 Wiring Connector

Introduction

The BMXSAI0410 analog input module includes 4 analog inputs. The module presents two pair of pins – two positive channel (Ch) pins and two negative common (Com) pins – for each input.

For each input:

- the two channel pins (Ch*n*) are internally connected, and
- the two common pins (Com*n*) also are internally connected.

To connect an analog sensor to an input, you can use either channel pin and either common pin for that input.

Terminal Blocks

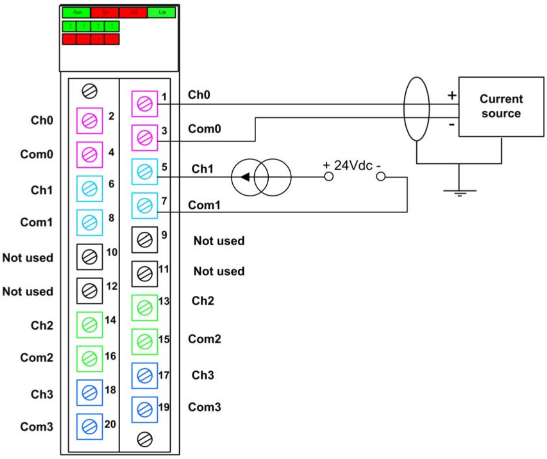
You can use the following Schneider Electric 20-point terminal blocks to fit the 20 pin connector on the front of the module:

- screw clamp terminal block BMXFTB2010
- age clamp terminal block BMXFTB2000
- spring type terminal block BMXFTB2020

NOTE: Terminal blocks can be removed only when power to the module is OFF.

Wiring Connector

The following example presents a generic wiring scheme for inputs on the module:



NOTE: The module detects a cut wire condition and reports it as a current out of range condition (less than 3.75 mA) by setting the `OOR` element of the `T_U_ANA_SIS_CH_IN` *(see page 62)* structure to “1”.

Mapping Inputs to Connector Pins

The following provides a description of each pin on the BMXSAI0410 analog input module:

Pin Description	Pin Number on Terminal Block		Pin Description
Input (+) of channel 0	2	1	Input (+) of channel 0
Input (–) of channel 0	4	3	Input (–) of channel 0
Input (+) of channel 1	6	5	Input (+) of channel 1
Input (–) of channel 1	8	7	Input (–) of channel 1
Not used	10	9	Not used
Not used	12	11	Not used
Input (+) of channel 2	14	13	Input (+) of channel 2
Input (–) of channel 2	16	15	Input (–) of channel 2
Input (+) of channel 3	18	17	Input (+) of channel 3
Input (–) of channel 3	20	19	Input (–) of channel 3

NOTE: Because the two positive pins for each input are internally connected, you need to use only one positive pin for an input channel. Similarly, because the two negative pins for each input are internally connected, you need to use only one negative pin for each input channel.
For example, to connect an analog sensor to input channel 0, you can connect:

- Positive wire of the sensor to either pin 1 or pin 2.
- Negative wire of the sensor to either pin 3 or pin 4.

BMXSAI0410 Input Application Wiring Examples

Introduction

You can wire the BMXSAI0410 safety analog input module to analog sensors to achieve SIL3 compliance in several different ways, depending on:

- the required Category (Cat2 or Cat4) and Performance Level (PLd or PLe) standard
- your application's requirements for high availability

⚠ CAUTION

RISK OF UNINTENDED OPERATION

The maximum safety integrity level (SIL) is determined by the quality of sensor and the length of the proof-test interval to IEC 61508. If you are using sensors that do not meet the quality of the intended SIL standard, always wire these sensors redundantly to two channels.

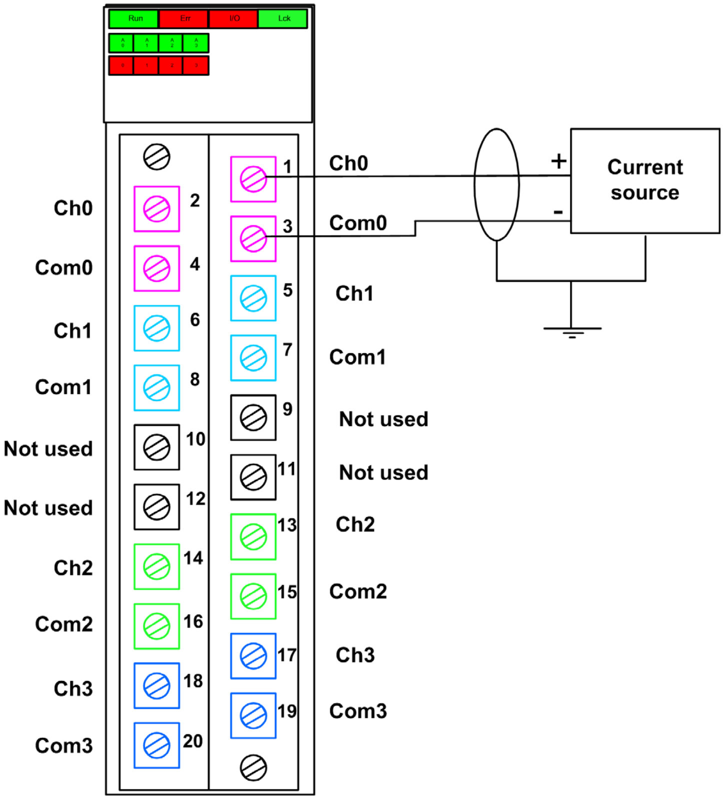
Failure to follow these instructions can result in injury or equipment damage.

The following SIL3 digital input application wiring examples are described, below:

- Cat2/PLd:
  - a single sensor wired to one input.
- Cat2/PLd with high availability:
  - two sensors wired to two input points on different input modules.
- Cat4/PLe:
  - two sensors, each wired to a different input point on the same input module.
- Cat4/PLe with high availability:
  - two pair of sensors (for a total of four sensors): the sensors of the first pair are each wired to a different input point on one module, and the sensors of the second pair are each wired to a different input point on a second module.

SIL3 Cat2/PLd

The following example presents a single sensor wired to one input point on a single input module. The CPU performs 1oo1D evaluation on the single monitored value:



⚠ CAUTION

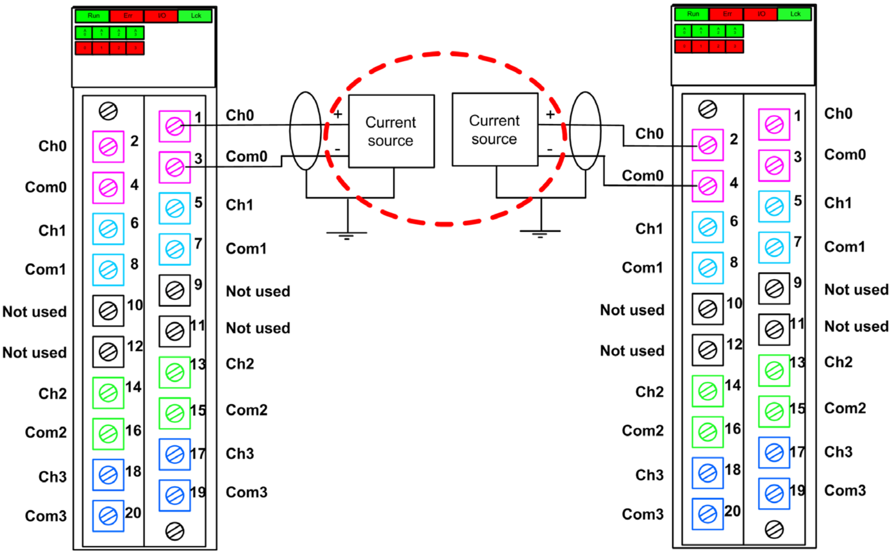
RISK OF UNINTENDED OPERATION

To achieve SIL3 according to IEC61508 and Category 2/Performance Level d according to ISO13849 using this wiring design, you must use a suitable, qualified sensor.

Failure to follow these instructions can result in injury or equipment damage.

SIL3 Cat2/PLd with High Availability

The following example presents two sensors that monitor the same process variable. Each sensor is connected to a single input point on different input modules. The CPU performs 1oo1D evaluation of the single monitored value:



NOTE: In this design, use the `S_AIHA` function block in the SAFE task to manage the two process variable values reported by the two sensors.

⚠ CAUTION

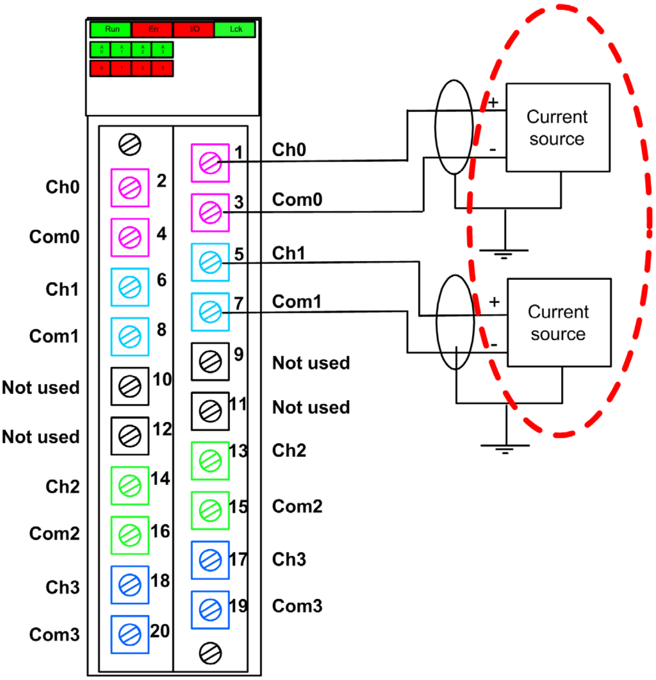
RISK OF UNINTENDED OPERATION

To achieve SIL3 according to IEC61508 and Category 2/Performance Level d according to ISO13849 using this wiring design, you must use a suitable, qualified sensor.

Failure to follow these instructions can result in injury or equipment damage.

SIL3 Cat4/PLe

The following example presents two sensors that monitor the same process variable. Each sensor is connected to a single input point on the same input module. The CPU performs a 1oo2D evaluation of the competing values provided by the two sensors for the same process variable:



NOTE: In this design, use the `S_AI_COMP` function block in the SAFE task to perform a 1oo2D evaluation of the competing values coming from the two sensors.

⚠ CAUTION

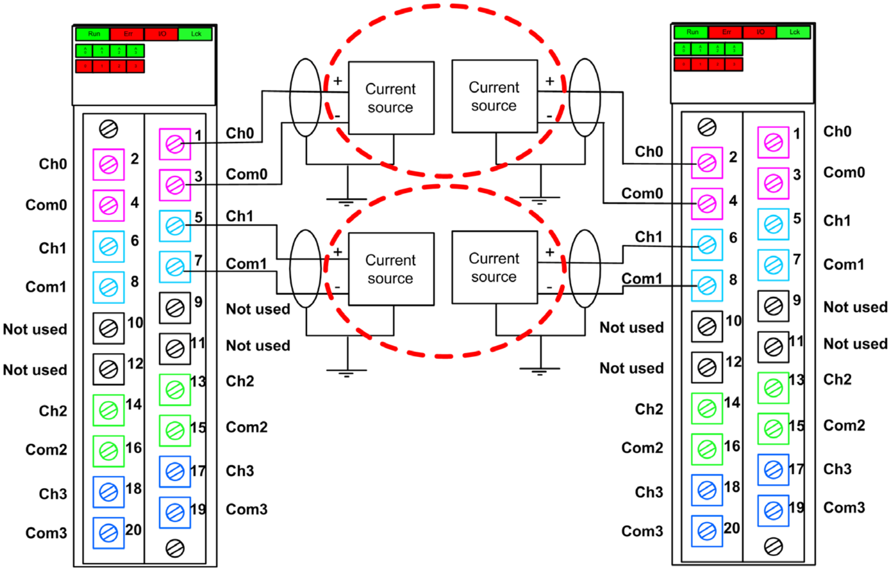
RISK OF UNINTENDED OPERATION

To achieve SIL3 according to IEC 61508 and Category 4/Performance Level e according to ISO13849 using this wiring design, you must use a suitable, qualified sensor.

Failure to follow these instructions can result in injury or equipment damage.

SIL3 Cat4/PLe with High Availability

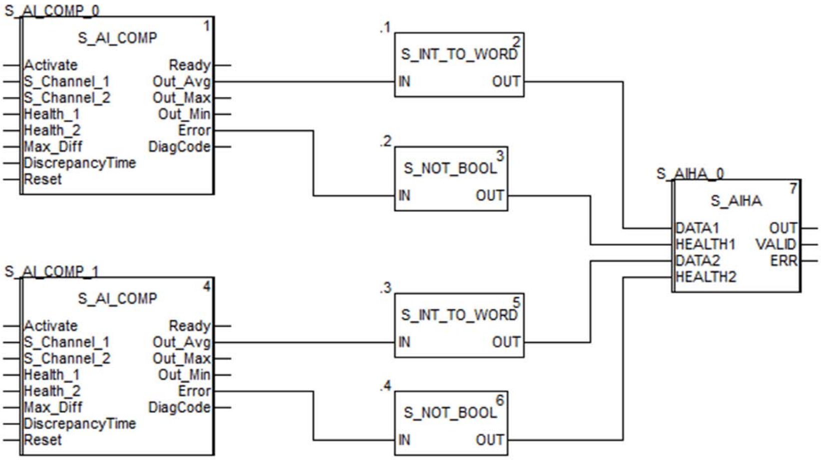
The following example presents two pair of redundant sensors, which monitor the same process variable. Each sensor is connected to a single input point on two different input modules (two inputs on each module). This design makes it possible for the CPU to perform a 1oo2D evaluation:



NOTE: In this design, you need to use the `S_AI_COMP` and `S_AIHA` function blocks inside the SAFE task to manage the four input signals:

- `S_AI_COMP` to perform 1oo2 evaluation of two pairs of values coming from both sensors connected to the same module.
- `S_AIHA` to manage the high availability feature.

The following function block diagram depicts the above referenced code segment design:



⚠ CAUTION

RISK OF UNINTENDED OPERATION

To achieve SIL3 according to IEC 61508 and Category 4/Performance Level e according to ISO13849 using this wiring design, you must use a suitable, qualified sensor.

Failure to follow these instructions can result in injury or equipment damage.

BMXSAI0410 Data Structure

Introduction

The T_U_ANA_SIS_IN_4 device derived data type (DDDT) is the interface between the BMXSAI0410 analog input module and the application running in the CPU. The T_U_ANA_-SIS_IN_4 DDDT incorporates the data types T_SAFE_COM_DBG_IN and T_U_ANA_SIS_CH_IN.

All of these structures are described, below.

T_U_ANA_SIS_IN_4 DDDT Structure

The T_U_ANA_SIS_IN_4 DDDT structure includes the following elements:

Element	Data Type	Description	Access
MOD_HEALTH[1]	BOOL	● 1: The module is operating correctly. ● 0: The module is not operating correctly.	RO
SAFE_COM_STS[1]	BOOL	● 1: Module communication is valid. ● 0: Module communication is not valid.	RO
S_COM_DBG	T_SAFE_COM_DBG_IN	Safe communication debug structure.	RO
CONF_LOCKED	BOOL	● 1: Module configuration is locked. ● 0: Module configuration is not locked.	RO
CH_IN	ARRAY[0...3] of T_U_ANA_SIS_CH_IN	Array of structure of channel.	–
MUID[2]	ARRAY[0...3] of DWORD	Module unique ID (auto-assigned by Control Expert)	RO
RESERVED	ARRAY[0...9] of INT	–	–

1. When the SAFE task on CPU is not in running mode, the data exchanged between the CPU and the module are not updated and MOD_HEALTH and SAFE_COM_STS are set to 0.
2. This auto-generated value can be changed by executing the **Build → Renew Ids & Rebuild All** command in the Control Expert main menu.

T_SAFE_COM_DBG_IN Structure

The T_SAFE_COM_DBG_IN structure includes the following elements:

Element	Data Type	Description	Access[1]
S_COM_EST	BOOL	● 1: Communication with the module is established. ● 0: Communication with the module is not established or corrupted.	RO
M_NTP_SYNC	BOOL	With CPU firmware 3.10 or earlier: ● 1: The module is synchronized with the NTP server. ● 0: The module is not synchronized with the NTP server. **NOTE:** With CPU firmware 3.20 or later, the value is always 1.	RO
CPU_NTP_SYNC	BOOL	With CPU firmware 3.10 or earlier: ● 1: The CPU is synchronized with the NTP server. ● 0: The CPU is not synchronized with the NTP server. **NOTE:** With CPU firmware 3.20 or later, the value is always 1.	RO
CHECKSUM	BYTE	Communication frame checksum.	RO
COM_DELAY	UINT	Communication delay between two values received by the module: ● 1...65534: The time, in ms, since the last communication was received by the CPU from the module. ● 65535: The CPU did not receive a communication from the module.	RO
COM_TO	UINT	Communication time-out value for communications coming from the module. **NOTE:** You may want to edit this read/write value to equal or exceed the actual communication time for the module (for example, in a remote RIO drop).	R/W
STS_MS_IN	UINT	Safe timestamp value for the fraction of a second, to the nearest ms, of the data received from the module.	RO
S_NTP_MS	UINT	Safe time value for the fraction of a second, to the nearest ms, for the current cycle.	RO
STS_S_IN	UDINT	Safe timestamp value in seconds of the data received from the module.	RO

Element	Data Type	Description	Access[1]
S_NTP_S	UDINT	Safe time value in seconds for the current cycle.	RO
CRC_IN	UDINT	CRC value for data received from the module.	RO

T_U_ANA_SIS_CH_IN Structure

The T_U_ANA_SIS_CH_IN structure includes the following elements:

Element	Data Type	Description	Access
FCT_TYPE	WORD	● 1: The channel is enabled. ● 0: The channel is not enabled.	RO
CH_HEALTH[1]	BOOL	● 1: The channel is operational. ● 0: An error has been detected on the channel, which is not operational. **Formula**: CH_HEALTH = not (OOR or IC) and SAFE_COM_STS	RO
VALUE	INT	Analog input value. **Formula**: VALUE = if (SAFE_COM_STS and not (IC)) then READ_VALUE else 0	RO
OOR	BOOL	● 1: The channel input current value is out of range, either: ○ <3.75 mA ○ >20.75 mA ● 0: The channel input current value is not out of range.	RO
IC	BOOL	● 1: Invalid channel detected by the module. ● 0: The channel is declared internally operational by the module.	RO
1. When the SAFE task on CPU is not in running mode, the data exchanged between the CPU and the module are not updated and CH_HEALTH is set to 0.			

Section 6.3
BMXSDI1602 Digital Input Module

Introduction

This section describes the BMXSDI1602 M580 safety digital input module.

What Is in This Section?

This section contains the following topics:

BMXSDI1602 Safety Digital Input Module

Introduction

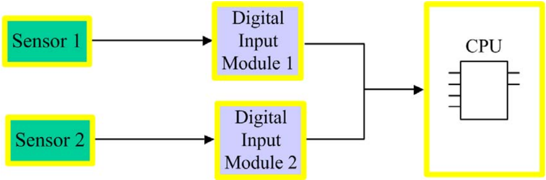
The BMXSDI1602 safety input module presents the following features:

- 16 Type 3 (IEC61131-2) inputs, in two electrically non-isolated groups of 8 inputs.
- 24 Vdc rated input voltage.
- Achieves the following:
  - SIL3 IEC61508, SILCL3 IEC62061.
  - Category 2 (Cat2) / Performance Level d (PLd) ISO13849 using 1 input channel (one-out-of-one (1oo1D) evaluation).
  - Category 4 (Cat4) / Performance Level e (PLe) ISO13849 using 2 input channels (one-out-of-two (1oo2D) evaluation).
- Compatible with 2 or 3 wire proximity sensors.
- Provides optionally two 24 Vdc outputs (VS1 and VS2) for short-circuit to 24 Vdc supervision:
  - VS1 to monitor short-circuit on inputs 0...3 (rank A & B).
  - VS2 to monitor short-circuit on inputs 4...7 (rank A & B).
- Monitor external 24 Vdc sensor supply voltage.
- LED diagnostic display *(see page 233)* provided for the module and for each input channel.
- Configurable (enable/disable) channel wiring diagnostics *(see page 71)* that can detect the following conditions:
  - Open (or cut) wire.
  - Short circuit to the 0 V ground.
  - Short circuit to the 24 Vdc (if sensor power is internally provided).
  - Crossed circuits between two channels (if sensor power is internally provided).
- Module hot swap during runtime.
- Module CCOTF when operating in maintenance mode *(see page 255)*. (CCOTF is not supported in safety mode *(see page 254)*.)

High Availability

You can use two sensors connected two different input channels located on different input modules to monitor the same physical value, and thereby increase system availability.

The following figure illustrates the redundant digital input configurations:



The input state value from sensor 1 and sensor 2 are sent by input module 1 and input module 2, respectively, over a black channel to a safety CPU. The CPU executes a dedicated function block, S_DIHA, to manage and select the data from the two input modules. This function block operates as follows:

- If the health status of the input data coming from module 1 is OK, the input data from this module is used in the safety function.
- If the health status of the input data coming from module 1 is not OK, but the health status of the input data coming from module 2 is OK, the input data from module 2 is used.
- If the health status of the input data from both module 1 and module 2 is not OK, then the state of the input is set to the safe state (“0”) in order to activate the safety function.

Refer to the description of input application wiring examples *(see page 70)* for details on how to wire the module for high availability.

BMXSDI1602 Wiring Connector

Introduction

The BMXSDI1602 digital input module presents 16 inputs in two groups of 8 inputs. The first group consists of inputs 0...3 (rank A & B), the second group consists of inputs 4...7 (rank A & B). There is no isolation between these two groups.

Power can be provided to the sensors either directly from the external power supply, or internally via the VS1 and VS2 power supplies. Each design is presented, below.

Terminal Blocks

You can use the following Schneider Electric 20-point terminal blocks to fit the 20 pin connector on the front of the module:

- screw clamp terminal block BMXFTB2010
- age clamp terminal block BMXFTB2000
- spring type terminal block BMXFTB2020

NOTE: Terminal blocks can be removed only when power to the module is OFF.

Process Power Supply

A 24 Vdc protected extra low voltage (PELV) overvoltage category II process power supply is required. Schneider Electric recommends a power supply that does not automatically restore power after a power interruption.

⚠ DANGER

LOSS OF THE ABILITY TO PERFORM SAFETY FUNCTIONS

Use only a PELV type process power supply with a maximum output of 60 V.

Failure to follow these instructions will result in death or serious injury.

Fuse

A fast blow fuse is required to help protect the external power supply against short-circuit and over voltage conditions.

NOTICE

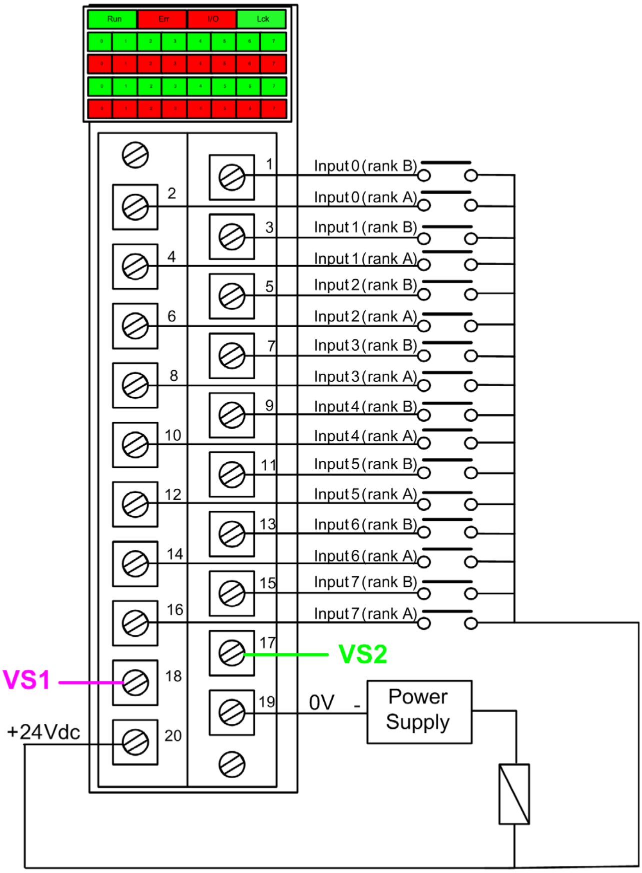
IMPROPER FUSE SELECTION

Use fast acting fuses to help protect the electronic components of the digital input module from an over current condition. Improper fuse selection can result in damage to the input module.

Failure to follow these instructions can result in equipment damage.

Wiring Connector: Sensors Supplied with External Power

In the following design, the sensors are powered directly from an external power supply:



power supply: 24Vdc
fuse: fast blow fuse of 0.5A

NOTE: Powering the sensors externally limits the channel diagnostics the module can perform. In this wiring design, the module can detect:

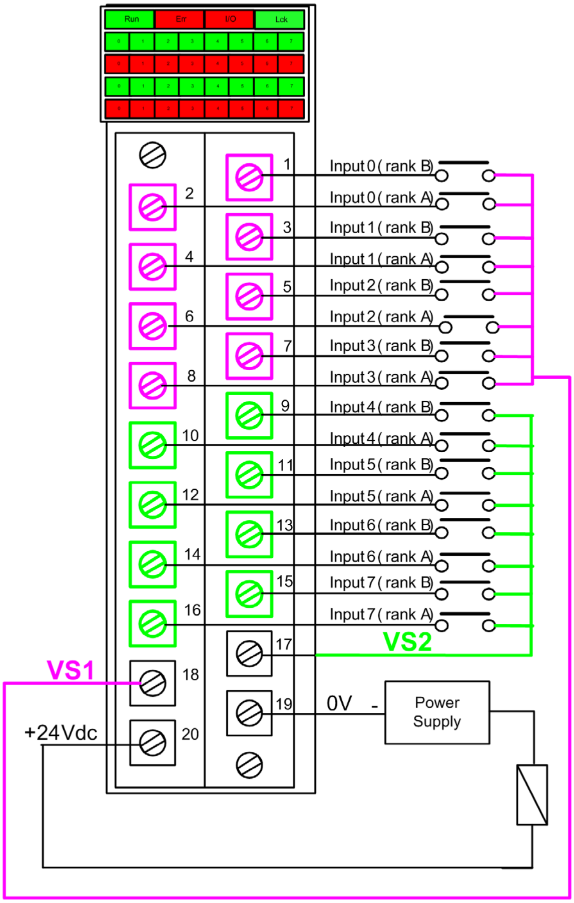
- A cut (or open) wire condition (if enabled for the channel in Control Expert).
- A short circuit to ground condition.

However, in this design, the module does not detect:

- A short circuit to 24 Vdc condition.
- A cross circuit condition with other wiring input.

Wiring Connector: Sensors Supplied with Internal VS Power

In the following design, sensors for channels 0...3 are supplied by the monitored VS1 power supply and sensors for channels 4...7 are supplied by the monitored VS2 power supply:



If you use this design, apply internal power to the channel groups as follows:

- Use VS1 to supply power to channels 0...3 (rank A and B).
- Use VS2 to supply power to channels 4...7 (rank A and B).

NOTE: In this design, the module can detect:

- A short circuit to 24 Vdc condition (if enabled for the channel in Control Expert).
- A cross circuit condition with other wiring input.
- A cut (or open) wire condition (if enabled for the channel in Control Expert).
- A short circuit to ground condition.

Mapping Inputs to Connector Pins and Control Expert Channels

The following provides a description of each pin on the BMXSDI1602 input module, and maps each pin to the channel for that pin as it appears in the channel **Configuration** tab for the module in Control Expert Safety:

Control Expert Channel	Pin Description	Pin Number on Terminal Block		Pin Description	Control Expert Channel
0	Input 0 (rank A)	2	1	Input 0 (rank B)	8
1	Input 1 (rank A)	4	3	Input 1 (rank B)	9
2	Input 2 (rank A)	6	5	Input 2 (rank B)	10
3	Input 3 (rank A)	8	7	Input 3 (rank B)	11
4	Input 4 (rank A)	10	9	Input 4 (rank B)	12
5	Input 5 (rank A)	12	11	Input 5 (rank B)	13
6	Input 6 (rank A)	14	13	Input 6 (rank B)	14
7	Input 7 (rank A)	16	15	Input 7 (rank B)	15
–	VS1 Power Supply	18	17	VS2 Power Supply	–
–	24 Vdc Process Power Supply	20	19	24 Vdc Process Power Supply	–

BMXSDI1602 Input Application Wiring Examples

Introduction

You can wire the BMXSDI1602 safety digital input module to sensors to achieve SIL3 compliance in several different ways, depending on:

- the required Category (Cat2 or Cat4) and Performance Level (PLd or PLe) standard
- your application's requirements for high availability

⚠ CAUTION

RISK OF UNINTENDED OPERATION

The maximum safety integrity level (SIL) is determined by the quality of sensor and the length of the proof-test interval to IEC 61508. If you are using sensors that do not meet the quality of the intended SIL standard, always wire these sensors redundantly to two channels.

Failure to follow these instructions can result in injury or equipment damage.

The following SIL3 digital input application wiring examples are described, below:

- Cat2/PLd:
  - a single sensor wired to one input
- Cat2/PLd with high availability:
  - a single sensor wired to two input points on different input modules
  - two sensors wired to two input points on different input modules
- Cat4/PLe:
  - a single sensor wired to two input points on the same input module
  - two sensors, each wired to a different input point on the same input module
- Cat4/PLe with high availability:
  - two sensors, each wired to two input different input points on different input modules

Configurable Wiring Diagnostics in Control Expert

For the BMXSDI1602 safety digital input module, use its **Configuration** page in Control Expert to:

- Enable **Short circuit to 24V detection** for each energized channel. This test performs the following actuator wiring diagnostics for a channel:
  - ❍ Short circuit to 24 Vdc detection.
  - ❍ Crossed circuit detection between two output channels.

  The principle is to provide power to the sensors, by group of 8 channels (with VS1 for channels 0 to 3 (Rank A & B) and VS2 for channels 4 to 7 (Rank A & B)). A pulse to OFF is applied to these power outputs periodically with a period less than 1 second and a duration less than 1 ms. During this pulse, if the current read into the input is not null, the module considers the input in short-circuit.
- Enable **Open wire detection** for each of eight channels, which performs the following wiring diagnostics for that channel:
  - ❍ Open (or cut) wire detection (i.e. the input channel is not connected to the sensor).
  - ❍ Short circuit wiring detection to the 0 Vdc ground.

  The principle is to create artificially, then measure, a leakage current (Ileakage) on the line (with a resistor in parallel to the sensor) when sensor is opened. If this leakage current (0.4 mA <Ileakage < 1.3 mA) cannot be measured on the input line by the module, the external line is considered as cut (or in a short-circuit to ground condition). The diagnostic is performed with a period less than 10 ms.
  - ❍ For a dry contact sensor, we recommend that you set in parallel with the sensor a 33 K Ω resistor.
  - ❍ Using DDP 2 or 3 wires, the leakage current needs to fall within the limits defined above. You need to define the value of the resistor to set in parallel with the sensor, considering the natural leakage current of the sensor and the internal resistor of the input (7.5 K Ω).

⚠ WARNING

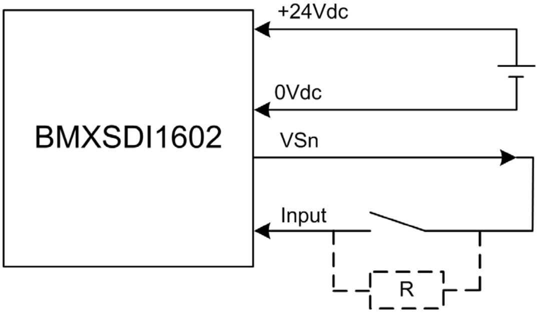
RISK OF UNINTENDED OPERATION

Schneider Electric recommends that you enable the available diagnostics provided in Control Expert to detect or exclude the conditions listed above. If a diagnostic test is not enabled or is not available in Control Expert, you will need to apply another safety measure to detect or exclude these conditions.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

SIL3 Cat2/PLd

Single sensor connected with one input, supplied by internal VS:



In this example, if internal power is supplied via:

- VS1, use channels 0...3 ranks A & B.
- VS2, use channels 4...7 ranks A & B.

Because the sensor is supplied power internally via a VS pin, the following channel wiring diagnostics apply:

Condition	Detectable?	Typical Detection Time
Open (or cut) wire[1]	Yes	< 10 ms
Short circuit to the 0 V ground	Yes	
Short circuit to the 24 Vdc[1]	Yes	< 1 s
Crossed circuits between two channels[1]	Yes	
1. This diagnostic function is performed if enabled in the module **Configuration** tab in Control Expert.		

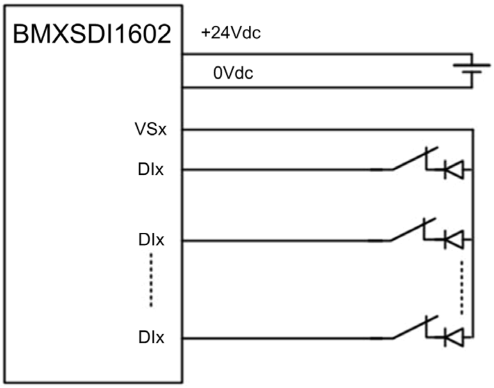
⚠ WARNING

RISK OF CROSSED CIRCUITS BETWEEEN CHANNELS IN THE SAME GROUP

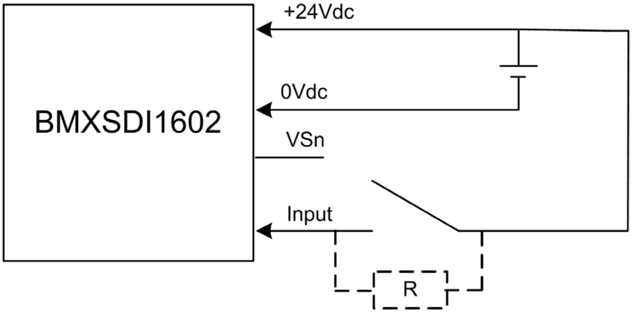
The module cannot detect crossed circuits between two channels in the same channel VS group. You will need to apply another safety measure to detect or exclude this condition.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: Consider adding a Shottky diode to the input loop, between the sensor and the input point, to reduce the likelihood that a short circuit to 24 Vdc condition on one channel might cause the same condition on an adjacent channel.



Single sensor connected with one input, powered by external power:



Because the sensor is supplied power externally, the following channel wiring diagnostics apply:

Condition	Detectable?	Typical Detection Time
Open (or cut) wire[1]	Yes	< 10 ms
Short circuit to the 0 V ground	Yes	
Short circuit to the 24 Vdc	No	–
Crossed circuits between two channels	No	

1. This diagnostic function is performed if enabled in the module **Configuration** tab in Control Expert.

⚠ WARNING

RISK OF CROSSED CIRCUITS BETWEEEN CHANNELS

The module cannot detect crossed circuits between two channels (in the case of a single sensor connected with one input, powered by external power, depicted above). You will need to apply another safety measure to detect or exclude this condition.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

⚠ WARNING

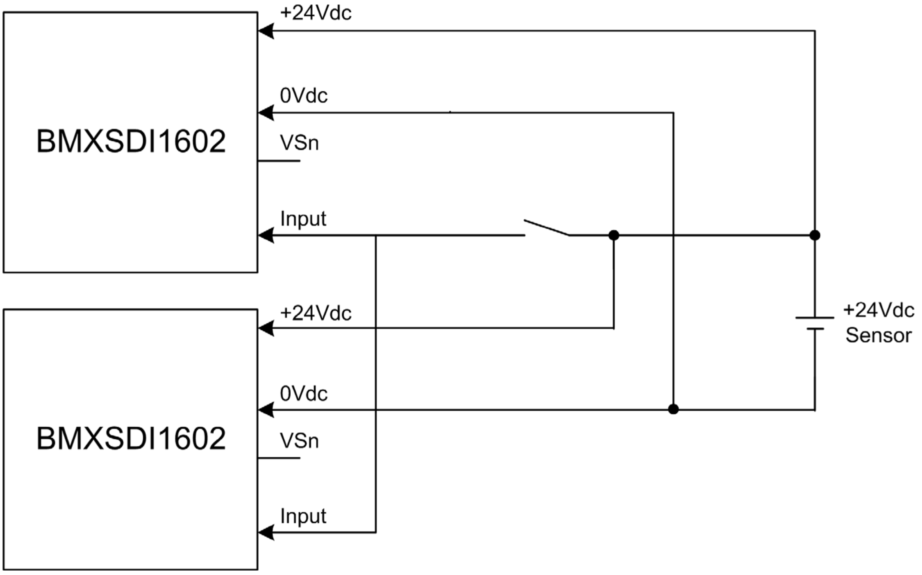
RISK OF SHORT CIRCUIT TO THE 24 Vdc

The module cannot detect a short circuit to the 24 Vdc condition (in the case of a single sensor connected with one input, powered by external power, depicted above). You will need to apply another safety measure to detect or exclude to this condition.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

SIL3 Cat2/PLd with High Availability

Single sensor connected on 2 inputs powered by external power:



Because the single sensor is supplied power externally, the following channel wiring diagnostics apply:

Condition	Detectable?	Typical Detection Time
Open (or cut) wire[1]	No	–
Short circuit to the 0 V ground	No	
Short circuit to the 24 Vdc[1]	No	
Crossed circuits between two channels	No	

[1] This diagnostic function is performed if enabled in the module **Configuration** tab in Control Expert.

⚠ WARNING

RISK OF CROSSED CIRCUITS BETWEEEN CHANNELS

The module cannot detect crossed circuits between two channels (in the case of a single sensor connected on two inputs, powered by external power, depicted above). You will need to apply another safety measure to detect or exclude this condition.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

⚠ WARNING

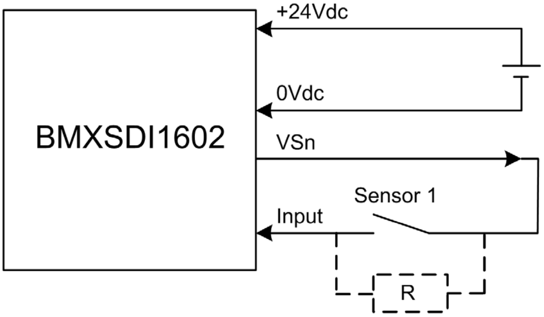
RISK OF SHORT CIRCUIT TO THE 24 Vdc

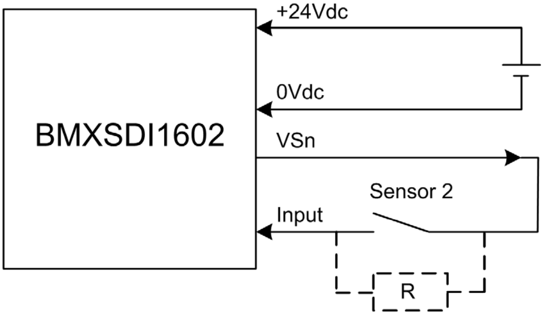
The module cannot detect a short circuit to the 24 Vdc condition (in the case of a single sensor connected on two inputs, powered by external power, depicted above). You will need to apply another safety measure to detect or exclude this condition.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

2 redundant sensors connected on single inputs of 2 modules using VS:

The following example presents two redundant sensors (which may or may not be linked mechanically) that are used to acquire the same process variable. Each sensor is wired to a single input point on a different input module, with power supplied by the monitored VS power supply:





In this example, if internal power is supplied via:

- VS1, use channels 0...3 ranks A & B.
- VS2, use channels 4...7 ranks A & B.

NOTE:

- In this design, you could use the `S_DIHA` function block to manage the two input signals.
- Consider adding a Shottky diode to the input loop, between the sensor and the input point, to reduce the likelihood that a short circuit to 24 Vdc condition on one channel might cause the same condition on an adjacent channel.

Because the sensor is supplied power internally via a VS pin, the following channel wiring diagnostics apply:

Condition	Detectable?	Typical Detection Time
Open (or cut) wire[1]	Yes	< 10 ms
Short circuit to the 0 V ground	Yes	
Short circuit to the 24 Vdc[1]	Yes	< 1 s
Crossed circuits between two channels	Yes	
1. This diagnostic function is performed if enabled in the module **Configuration** tab in Control Expert.		

⚠ WARNING

RISK OF CROSSED CIRCUITS BETWEEEN CHANNELS IN THE SAME GROUP

The module cannot detect crossed circuits between two channels in the same channel VS group. You will need to apply another safety measure to detect and exclude this condition.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

2 redundant sensors connected on single inputs of 2 modules using external power:

NOTE: Alternatively, power could be supplied to the sensors by an external power supply. In this case, a short circuit to the 24 Vdc condition and a crossed circuits between two channels condition would not be detectable.

Because the sensor is supplied power internally via a VS pin, the following channel wiring diagnostics apply:

Condition	Detectable?	Typical Detection Time
Open (or cut) wire[1]	Yes	< 10 ms
Short circuit to the 0 V ground	Yes	
Short circuit to the 24 Vdc	No	–
Crossed circuits between two channels	No	
1. This diagnostic function is performed if enabled in the module **Configuration** tab in Control Expert.		

⚠ WARNING

RISK OF CROSSED CIRCUITS BETWEEEN CHANNELS

The module cannot detect crossed circuits between two channels (in the case of two redundant sensors connected on single inputs of two modules using external power). You will need to apply another safety measure to detect or exclude this condition.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

⚠ WARNING

RISK OF SHORT CIRCUIT TO THE 24 Vdc

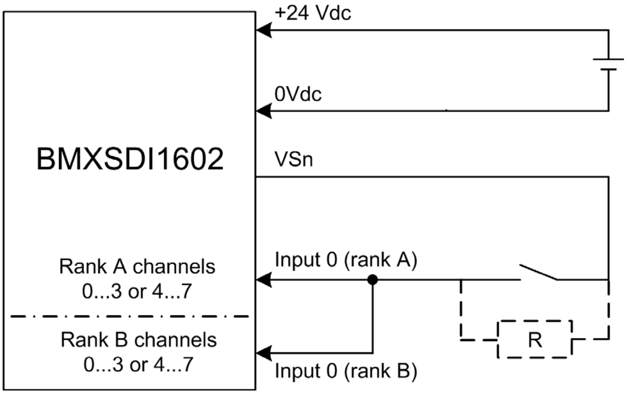
The module cannot detect a short circuit to the 24 Vdc condition (in the case of two redundant sensors connected on single inputs of two modules using external power). You will need to apply another safety measure to detect or exclude this condition.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Cat4/PLe

Single sensor connected on 2 inputs of same module using Vs:

The following example presents a single sensor wired to two input points on the same input module, with power supplied by the monitored VS power supply:



In this example, if internal power is supplied via:

- VS1, use channels 0...3 ranks A & B.
- VS2, use channels 4...7 ranks A & B.

NOTE:

- In this design, you could use the `S_EQUIVALENT` function block to manage the two input signals.
- Consider adding a Shottky diode to the input loop, between the sensor and the input point, to reduce the likelihood that a short circuit to 24 Vdc condition on one channel might cause the same condition on an adjacent channel.

Wiring diagnostic with single sensor connected on two inputs, using power from the VS pin:

Condition	Detectable?	Typical Detection Time
Open (or cut) wire[1]	Yes	< 10 ms
Short circuit to the 0 V ground	Yes	
Short circuit to the 24 Vdc[1]	Yes	< 1 s
Crossed circuits between two channels	Yes	
1. This diagnostic function is performed if enabled in the module **Configuration** tab in Control Expert.		

⚠ WARNING

RISK OF CROSSED CIRCUITS BETWEEEN CHANNELS IN THE SAME GROUP

The module cannot detect crossed circuits between two channels in the same channel VS group. You will need to apply another safety measure to detect and exclude this condition.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Single sensor connected on 2 inputs of same module using external power supply:

NOTE: Alternatively, power could be supplied to the sensors by an external power supply. In this case, a short circuit to the 24 Vdc condition and a crossed circuits between two channels condition would not be detectable.

Wiring diagnostic with single sensor connected on two inputs, using external power:

Condition	Detectable?	Typical Detection Time
Open (or cut) wire[1]	Yes	< 10 ms
Short circuit to the 0 V ground	Yes	
Short circuit to the 24 Vdc[1]	No	–
Crossed circuits between two channels	No	
1. This diagnostic function is performed if enabled in the module **Configuration** tab in Control Expert.		

⚠ WARNING

RISK OF CROSSED CIRCUITS BETWEEEN CHANNELS

The module cannot detect crossed circuits between two channels (in the case of a single sensor connected on two inputs of same module using an external power supply). You will need to apply another safety measure to detect or exclude this condition.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

⚠ WARNING

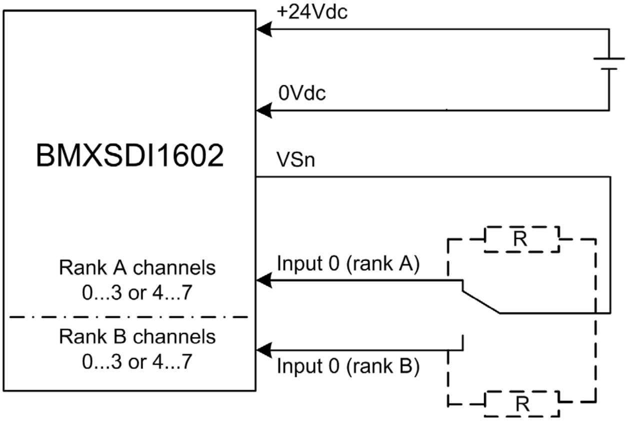
RISK OF SHORT CIRCUIT TO THE 24 Vdc

The module cannot detect a short circuit to the 24 Vdc condition (in the case of a single sensor connected on two inputs of same module using an external power supply). You will need to apply another safety measure to detect or exclude this condition.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Non-equivalent sensor connected on 2 non-equivalent inputs of same module using Vs:

The following example presents a single non-equivalent sensor wired to two input points on the same input module, with power supplied by the monitored VS power supply. The module performs a 1oo2D evaluation:



In this example, if internal power is supplied via:

- VS1, use channels 0...3 ranks A & B.
- VS2, use channels 4...7 ranks A & B.

NOTE:

- In this design, you could use the `S_ANTIIVALENT` function block to manage the two input signals.
- Consider adding a Shottky diode to the input loop, between the sensor and the input point, to reduce the likelihood that a short circuit to 24 Vdc condition on one channel might cause the same condition on an adjacent channel.

Wiring diagnostic with single non-equivalent sensors connected on two inputs, using power from the VS pin:

Condition	Detectable?	Typical Detection Time
Open (or cut) wire[1]	Yes	< 10 ms
Short circuit to the 0 V ground	Yes	
Short circuit to the 24 Vdc[1]	Yes	< 1 s
Crossed circuits between two channels	Yes	
1. This diagnostic function is performed if enabled in the module **Configuration** tab in Control Expert.		

Non-equivalent sensor connected on 2 non-equivalent inputs of same module using external power:

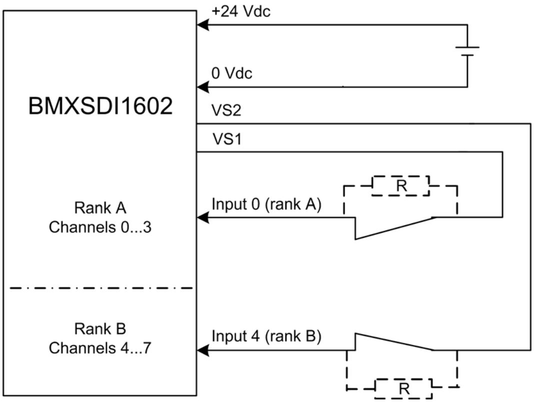
NOTE: Alternatively, power could be supplied to the sensors by an external power supply. In this case, a short circuit to the 24 Vdc condition and a crossed circuits between two channels condition would not be detectable.

Wiring diagnostic with single non-equivalent sensors connected on two inputs, using external power:

Condition	Detectable?	Typical Detection Time
Open (or cut) wire[1]	Yes	< 10 ms
Short circuit to the 0 V ground	Yes	
Short circuit to the 24 Vdc[1]	No	–
Crossed circuits between two channels	No	
1. This diagnostic function is performed if enabled in the module **Configuration** tab in Control Expert.		

Acquisition of the same process variable using two separated sensors (linked mechanically or not) using VS:

The following example presents two redundant sensors (which may or may not be linked mechanically) that are used to acquire the same process variable. Each sensor is wired to a single input point on the same input module, with power supplied by the monitored VS power supply:



NOTE:

- Inputs 0...3 from Rank A are used with Inputs 4...7 from Rank B.
- Inputs 0...3 from Rank B are used with Inputs 4...7 from Rank A.

⚠ WARNING

RISK OF UNINTENDED OPERATION

To achieve SIL3 according to IEC 61508 and Category 4/Performance Level e according to ISO13849 using this wiring design, you must use suitable, qualified sensors.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

In this example, if internal power is supplied via:

- VS1, use channels 0...3 ranks A & B.
- VS2, use channels 4...7 ranks A & B.

NOTE:

- In this design, you could use the `S_EQUIVALENT` function block to manage the two input signals.
- Consider adding a Shottky diode to the input loop, between the sensor and the input point, to reduce the likelihood that a short circuit to 24 Vdc condition on one channel might cause the same condition on an adjacent channel.

Wiring diagnostic with single sensor connected to two inputs, using power from the VS pin:

Condition	Detectable?	Typical Detection Time
Open (or cut) wire[1]	Yes	< 10 ms
Short circuit to the 0 V ground	Yes	
Short circuit to the 24 Vdc[1]	Yes	< 1 s
Crossed circuits between two channels	Yes	

[1] This diagnostic function is performed if enabled in the module **Configuration** tab in Control Expert.

⚠ WARNING

RISK OF CROSSED CIRCUITS BETWEEEN CHANNELS IN THE SAME GROUP

The module cannot detect crossed circuits between two channels in the same channel VS group (in the case of the acquisition of the same process variable using two separated sensors using VS supplied power). You will need to apply another safety measure to detect or exclude this condition.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

⚠ WARNING

RISK OF SHORT CIRCUIT TO THE 24 Vdc

The module cannot detect a short circuit to the 24 Vdc condition (in the case of the acquisition of the same process variable using two separated sensors using VS supplied power). You will need to apply another safety measure to detect or exclude this condition.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Acquisition of the same process variable using two separated sensors (linked mechanically or not) using external power:

NOTE: Alternatively, power could be supplied to the sensors by an external power supply. In this case, a short circuit to the 24 Vdc condition and a crossed circuits between two channels condition would not be detectable.

Wiring diagnostic with single sensor connected to two inputs, using external power:

Condition	Detectable?	Typical Detection Time
Open (or cut) wire[1]	Yes	< 10 ms
Short circuit to the 0 V ground	Yes	
Short circuit to the 24 Vdc[1]	No	–
Crossed circuits between two channels	No	
1. This diagnostic function is performed if enabled in the module **Configuration** tab in Control Expert.		

⚠ WARNING

RISK OF CROSSED CIRCUITS BETWEEEN CHANNELS

The module cannot detect crossed circuits between two channels (in the case of the acquisition of the same process variable using two separated sensors using external power). You will need to apply another safety measure to detect or exclude this condition.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

⚠ WARNING

RISK OF UNINTENDED OPERATION

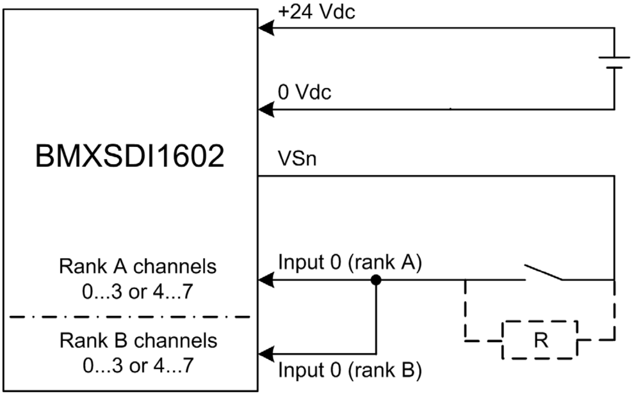
To achieve SIL3/Cat4/PLe using this wiring, you must use a suitably qualified sensor.

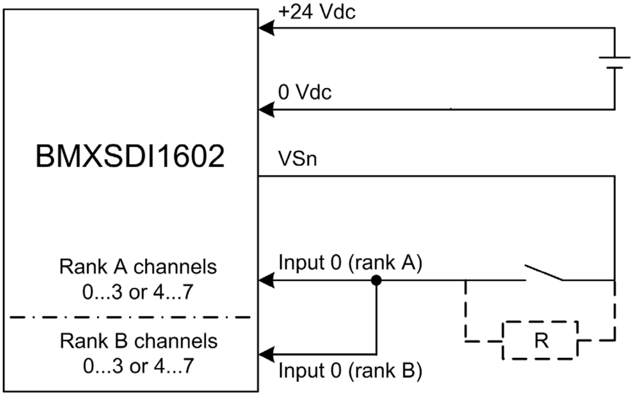
Failure to follow these instructions can result in death, serious injury, or equipment damage.

Cat4/PLe with High Availability

Wiring scheme with single-channel connection of two redundant single-channel sensors using Vs:

The following example presents two redundant single-channel sensors (which may or may not be mechanically linked), each wired to two input points on two different input modules, with power supplied by the monitored VS power supply:





In this example, if internal power is supplied via:

- VS1, use channels 0...3 ranks A & B.
- VS2, use channels 4...7 ranks A & B.

NOTE:

- In this design, you could use the `S_EQUIVALENT` and `S_DIHA` function blocks to manage the four input signals.
- Consider adding a Shottky diode to the input loop, between the sensor and the input point, to reduce the likelihood that a short circuit to 24 Vdc condition on one channel might cause the same condition on an adjacent channel.

Wiring diagnostic with single sensor connected to two inputs, using power from the VS pin:

Condition	Detectable?	Typical Detection Time
Open (or cut) wire[1]	Yes	< 10 ms
Short circuit to the 0 V ground	Yes	
Short circuit to the 24 Vdc[1]	Yes	< 1 s
Crossed circuits between two channels	Yes	
1. This diagnostic function is performed if enabled in the module **Configuration** tab in Control Expert.		

⚠ WARNING

RISK OF CROSSED CIRCUITS BETWEEEN CHANNELS IN THE SAME GROUP

The module cannot detect crossed circuits between two channels in the same channel VS group. You will need to apply another safety measure to detect or exclude this condition.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Wiring scheme with single-channel connection of two redundant single-channel sensors using external power:

NOTE: Alternatively, power could be supplied to the sensors by an external power supply. In this case, a short circuit to the 24 Vdc condition and a crossed circuits between two channels condition would not be detectable.

Wiring diagnostic with single sensor connected to two inputs, using external power:

Condition	Detectable?	Typical Detection Time
Open (or cut) wire[1]	Yes	< 10 ms
Short circuit to the 0 V ground	Yes	
Short circuit to the 24 Vdc[1]	No	–
Crossed circuits between two channels	No	
1. This diagnostic function is performed if enabled in the module **Configuration** tab in Control Expert.		

⚠ WARNING

RISK OF CROSSED CIRCUITS BETWEEEN CHANNELS

The module cannot detect crossed circuits between two channels (in the case of a single-channel connection of two redundant single-channel sensors using external power). You will need to apply another safety measure to detect or exclude this condition.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

⚠ WARNING

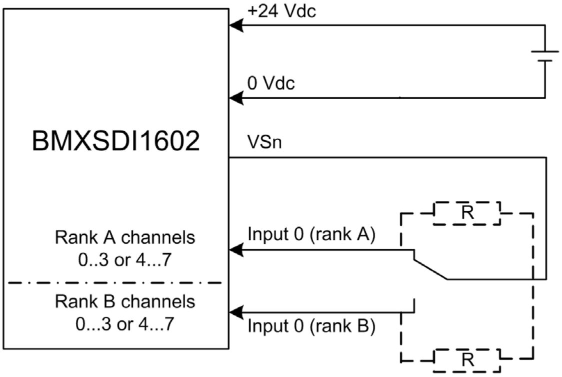
RISK OF SHORT CIRCUIT TO THE 24 Vdc

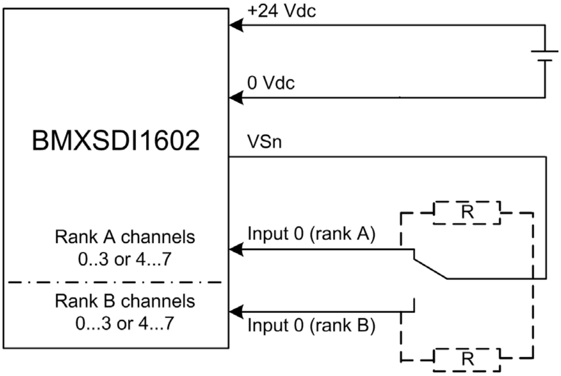
The module cannot detect a short circuit to the 24 Vdc condition (in the case of a single-channel connection of two redundant single-channel sensors using external power). You will need to apply another safety measure to detect or exclude this condition.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Non-equivalent sensor (linked mechanically or not) connected on 2 non-equivalent inputs of two different modules using Vs:

The following example presents two pair of redundant non-equivalent sensors (which may or may not be mechanically linked), each wired to a single input point on two different input modules (two on each module), with power supplied by the monitored VS power supply:





In this example, if internal power is supplied via:
- VS1, use channels 0...3 ranks A & B.
- VS2, use channels 4...7 ranks A & B.

NOTE:
- In this design, you need to use the `S_ANTIVALENT` and `S_DIHA` function blocks to manage the four input signals.
  - `S_ANTIVALENT` to perform 1oo2 evaluation of two pairs of values coming from both sensors connected to the same module.
  - `S_DIHA` to manage the high availability feature.
- Consider adding a Shottky diode to the input loop, between the sensor and the input point, to reduce the likelihood that a short circuit to 24 Vdc condition on one channel might cause the same condition on an adjacent channel.

Because the sensor is supplied power internally via a VS pin, the following channel wiring diagnostics apply:

Condition	Detectable?	Typical Detection Time
Open (or cut) wire[1]	Yes	< 10 ms
Short circuit to the 0 V ground	Yes	
Short circuit to the 24 Vdc[1]	Yes	< 1 s
Crossed circuits between two channels	Yes	
1. This diagnostic function is performed if enabled in the module **Configuration** tab in Control Expert.		

⚠ WARNING

RISK OF CROSSED CIRCUITS BETWEEEN CHANNELS IN THE SAME GROUP

The module cannot detect crossed circuits between two channels in the same channel VS group. You will need to apply another safety measure to detect or exclude this condition.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Non-equivalent sensor (linked mechanically or not) connected on 2 non-equivalent inputs of two different modules using external power:

NOTE: Alternatively, power could be supplied to the sensors by an external power supply (in the case of a non-equivalent sensor connected on two non-equivalent inputs off two different modules using external power). In this case, a crossed circuits between two channels condition would not be detectable.

⚠ WARNING

RISK OF CROSSED CIRCUITS BETWEEEN CHANNELS

The module cannot detect crossed circuits between two channels. You will need to apply another safety measure to detect or exclude this condition.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

⚠ WARNING

RISK OF UNINTENDED OPERATION

To achieve SIL3 according to IEC 61508 and Category 4/Performance Level e according to ISO13849 using this wiring design, you must use suitable, qualified sensors.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

BMXSDI1602 Data Structure

Introduction

The T_U_DIS_SIS_IN_16 device derived data type (DDDT) is the interface between the BMXSDI1602 digital input module and the application running in the CPU. The T_U_DIS_SIS_IN_16 DDDT incorporates the data types T_SAFE_COM_DBG_IN and T_U_DIS_SIS_CH_IN.

All of these structures are described, below.

T_U_DIS_SIS_IN_16 DDDT Structure

The T_U_DIS_SIS_IN_16 DDDT structure includes the following elements:

Element	Data Type	Description	Access
MOD_HEALTH[1]	BOOL	● 1: The module is operating correctly. ● 0: The module is not operating correctly.	RO
SAFE_COM_STS[1]	BOOL	● 1: Module communication is valid. ● 0: Module communication is not valid.	RO
PP_STS	BOOL	● 1: The process power supply is operational. ● 0: The process power supply is not operational.	RO
CONF_LOCKED	BOOL	● 1: Module configuration is locked. ● 0: Module configuration is not locked.	RO
S_COM_DBG	T_SAFE_COM_DBG_IN	Safe communication debug structure.	RO
CH_IN_A	ARRAY[0...7] of T_U_DIS_SIS_CH_IN	Array of structure of channel from rank A.	–
CH_IN_B	ARRAY[0...7] of T_U_DIS_SIS_CH_IN	Array of structure of channel from rank B.	–
MUID[2]	ARRAY[0...3] of DWORD	Module unique ID (auto-assigned by Control Expert)	RO
RESERVED	ARRAY[0...9] of INT	–	–
1. When the SAFE task on CPU is not in running mode, the data exchanged between the CPU and the module are not updated and MOD_HEALTH and SAFE_COM_STS are set to 0. 2. This auto-generated value can be changed by executing the **Build** → **Renew Ids & Rebuild All** command in the Control Expert main menu.			

T_SAFE_COM_DBG_IN Structure

The T_SAFE_COM_DBG_IN structure includes the following elements:

Element	Data Type	Description	Access
S_COM_EST	BOOL	- 1: Communication with the module is established. - 0: Communication with the module is not established or corrupted.	RO
M_NTP_SYNC	BOOL	With CPU firmware 3.10 or earlier: - 1: The module is synchronized with the NTP server. - 0: The module is not synchronized with the NTP server. **NOTE:** With CPU firmware 3.20 or later, the value is always 1.	RO
CPU_NTP_SYNC	BOOL	With CPU firmware 3.10 or earlier: - 1: The CPU is synchronized with the NTP server. - 0: The CPU is not synchronized with the NTP server. **NOTE:** With CPU firmware 3.20 or later, the value is always 1.	RO
CHECKSUM	BYTE	Communication frame checksum.	RO
COM_DELAY	UINT	Communication delay between two values received by the module: - 1...65534: The time, in ms, since the last communication was received by the CPU from the module. - 65535: The CPU did not receive a communication from the module.	RO
COM_TO	UINT	Communication time-out value for communications coming from the module.	R/W
STS_MS_IN	UINT	Safe timestamp value for the fraction of a second, to the nearest ms, of the data received from the module.	RO
S_NTP_MS	UINT	Safe time value for the fraction of a second, to the nearest ms, for the current cycle.	RO
STS_S_IN	UDINT	Safe timestamp value in seconds of the data received from the module.	RO
S_NTP_S	UDINT	Safe time value in seconds for the current cycle.	RO
CRC_IN	UDINT	CRC value for data received from the module.	RO

T_U_DIS_SIS_CH_IN Structure

The T_U_DIS_SIS_CH_IN structure includes the following elements:

Element	Data Type	Description	Access
CH_HEALTH[1]	BOOL	● 1: The channel is operational. ● 0: An error has been detected on the channel, which is not operational. **Formula**: CH_HEALTH = not (OC or IC or SC) and SAFE_COM_STS	RO
VALUE[2]	EBOOL	● 1: The input is energized. ● 0: The input is de-energized. **Formula**: VALUE = if (SAFE_COM_STS and not (IC)) then READ_VALUE else 0	RO
OC	BOOL	● 1: The channel is open or short circuited to ground. ● 0: The channel is connected and not short circuited to ground.	RO
SC	BOOL	● 1: The channel is short circuited to a 24 V source, or cross circuited between two channels. ● 0: The channel is not short circuited to a 24 V source or cross circuited between two channels.	RO
IC	BOOL	● 1: Invalid channel detected by the module. ● 0: The channel is declared internally operational by the module.	RO
V_OC	BOOL	Configuration status of the open or short circuit to ground test: ● 1: Enabled. ● 0: Disabled.	RO
V_SC	BOOL	Configuration status of the short circuit to 24 V source test: ● 1: Enabled. ● 0: Disabled.	RO
1. When the SAFE task on CPU is not in running mode, the data exchanged between the CPU and the module are not updated and CH_HEALTH is set to 0. 2. The VALUE element can be time-stamped by the BMX CRA or the BME CRA.			

Section 6.4
BMXSDO0802 Digital Output Module

Introduction

This section describes the BMXSDO0802 M580 safety digital output module.

What Is in This Section?

This section contains the following topics:

BMXSDO0802 Safety Digital Output Module

Introduction

The BMXSDO0802 safety digital output module presents the following features:

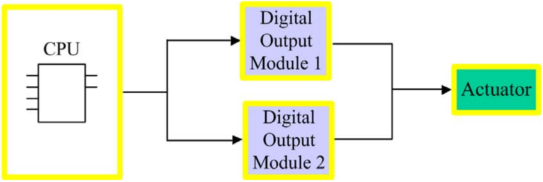
- 8 non-electrically isolated 0.5 A outputs.
- 24 Vdc rated output voltage.
- Achieves the following:
  - SIL3 IEC61508, SILCL3 IEC62061.
  - Category 4 (Cat4) / Performance Level e (PLe) ISO13849.
- Monitors the external pre-actuator power supply.
- LED diagnostic display *(see page 240)* provided for the module and for each output channel.
- Automatically provided channel wiring diagnostics that can detect the following conditions when the output is *energized*:
  - Overload current
  - Short circuit to the 0 Vdc ground
- Configurable (enable/disable) channel wiring diagnostics *(see page 99)* that can detect the following conditions:
  - Open (or cut) wire.
- Configurable (enable/disable) channel wiring diagnostics that can detect the following conditions when the output is *de-energized*:
  - Short circuit to the 0 V ground.
- Configurable (enable/disable) channel wiring diagnostics that can detect the following conditions when the output is *energized* or *de-energized*:
  - Short circuit to the 24 Vdc.
  - Crossed circuits between two channels (if sensor power is internally provided).
- Configurable fallback settings for each channel that are applied if communication between the CPU and output module are lost.
- Module hot swap during runtime.
- Module CCOTF when operating in maintenance mode *(see page 255)*. (CCOTF is not supported in safety mode.) *(see page 254)*

NOTE: A self-test is performed on each output to check its capability to be de-energized and reach its safe state without any impact for the load (off-pulse < 1ms). This self-test is alternatively performed, one output at a time, on each energized output with a period less than 1 second. If the output is connected to a static input of a product, the connected static input may detect this pulse. A filter may be useful to avoid the potential impact of this pulse on the input.

High Availability

You can connect the CPU to two output modules via a black channel, then connect each output module to a single actuator. No function block is needed, because the signal from the CPU is connected to both output channels.

The following figure illustrates the redundant digital output configuration for high availability:



The health of each output module can be read from the elements of its `T_U_DIS_SIS_OUT_8` *(see page 104)* DDDT structure. You can use this data to determine if a module needs to be replaced. If a module ceases to be operational and needs to be replaced, the system continues to run in a SIL3 compliant configuration while the module exchange takes place.

Refer to the high availability output wiring example *(see page 101)* for details on this design.

BMXSDO0802 Wiring Connector

Introduction

The BMXSDO0802 digital output module presents a single group of 8 outputs.

- Both common +24 Vdc power supply pins (18 and 20) are internally connected.
- All common 0 V pins (1, 3, 5, 7, 9, 11, 13, 15, 17 and 19) are internally connected.

Terminal Blocks

You can use the following Schneider Electric 20-point terminal blocks to fit the 20 pin connector on the front of the module:

- screw clamp terminal block BMXFTB2010
- age clamp terminal block BMXFTB2000
- spring type terminal block BMXFTB2020

NOTE: Terminal blocks can be removed only when power to the module is OFF.

Process Power Supply

A 24 Vdc protected extra low voltage (PELV) overvoltage category II process power supply is required. Schneider Electric recommends a power supply that does not automatically restore power after a power interruption.

Fuse

A fast blow fuse, maximum of 6 A, is required to help protect the external power supply against short-circuit and over voltage conditions.

⚠ CAUTION

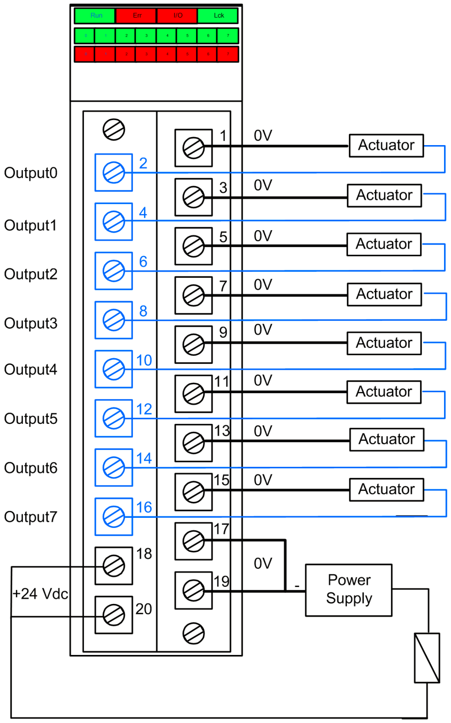
IMPROPER FUSE SELECTION

Use fast acting fuses to help protect the electronic components of the digital output module from an over current condition. Improper fuse selection can result in damage to the module.

Failure to follow these instructions can result in injury or equipment damage.

Wiring Connector Pins

The following wiring diagram presents a single output module connected to 8 actuators:



Mapping Outputs to Connector Pins

The following provides a description of each pin on the BMXSDO0802 output module:

Pin Description	Pin Number on Terminal Block		Pin Description
Output 0	2	1	Common 0V
Output 1	4	3	Common 0V
Output 2	6	5	Common 0V
Output 3	8	7	Common 0V
Output 4	10	9	Common 0V
Output 5	12	11	Common 0V
Output 6	14	13	Common 0V
Output 7	16	15	Common 0V
24 Vdc Process Power Supply	18	17	Common 0V
24 Vdc Process Power Supply	20	19	Common 0V

BMXSDO0802 Output Application Wiring Examples

Introduction

You can wire the BMXSDO0802 safety digital output module to actuators to achieve SIL3 Category 4 (Cat4) / Performance Level e (PLe) compliance in different ways, depending on your requirements for high availability.

⚠ CAUTION

RISK OF UNINTENDED OPERATION

The maximum safety integrity level (SIL) is determined by the quality of actuator and the length of the proof-test interval to IEC 61508. If you are using actuators that do not meet the quality of the intended SIL standard, always wire these actuators redundantly to two channels.

Failure to follow these instructions can result in injury or equipment damage.

The following SIL3 Cat4/PLe digital output application wiring examples are described, below:

- Cat4/PLe:
  - a single output module channel commanding one process variable. A single actuator is employed in this design.
- Cat4/PLe with high availability:
  - two redundant output modules, each with a channel connected to a separate actuator, but commanding the same process variable.

⚠ CAUTION

RISK OF UNINTENDED OPERATION

When the equipment is used in a fire and gas application, or when the demand state of the output is energized:

- Your proof test procedure must include a test that the cut wire detection is effective by removing the terminal block and verifying that the corresponding error bits are set.
- Verify the short circuit to ground detection effectiveness, either by enabling this **Pulse test to energized** diagnostic function in the module's **Configuration** tab, or by implementing another procedure (for example, by setting the output to 1 and checking the diagnostics, and so forth.)
- Avoid using lamp-like actuators, because their impedance is very low when they are switched on, which can result in a risk of detecting a false short-circuit or overload condition.

Failure to follow these instructions can result in injury or equipment damage.

Configurable Wiring Diagnostics in Control Expert

For the BMXSDO0802 safety digital output module, use its **Configuration** page in Control Expert to:

- Enable **Short circuit to 24V detection** for each energized channel. This test performs the following actuator wiring diagnostics for a channel:
  - Short circuit to 24 Vdc detection
  - Crossed circuit detection between two output channels
- Enable **Open wire detection** for each of eight channels, which performs the following wiring diagnostics for that channel:
  - Open (or cut) wire detection (i.e. the output channel is not connected to the actuator)
  - Short circuit wiring detection to the 0 Vdc ground
- Enable the **Pulse test to energized** for each output channel. This test is performed periodically when the output is in the de-energized state, and applies a pulse (duration less than 1 ms) to the output to determine if it can transition to the energized state. If the current exceeds a threshold of 0.7 A, the output is reported as being in a short circuit condition with the 0 Vdc ground. The test period is less than 1 s.

⚠ WARNING

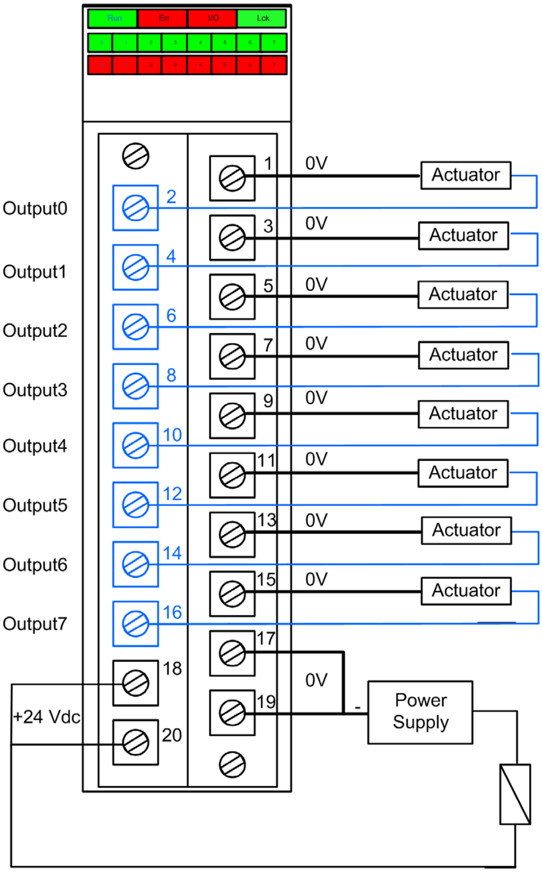
RISK OF UNINTENDED OPERATION

Schneider Electric recommends that you enable the available diagnostics provided in Control Expert to detect and respond to the conditions listed above. If a diagnostic test is not enabled or is not available in Control Expert, you will need to apply another safety measure to detect or exclude these conditions.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

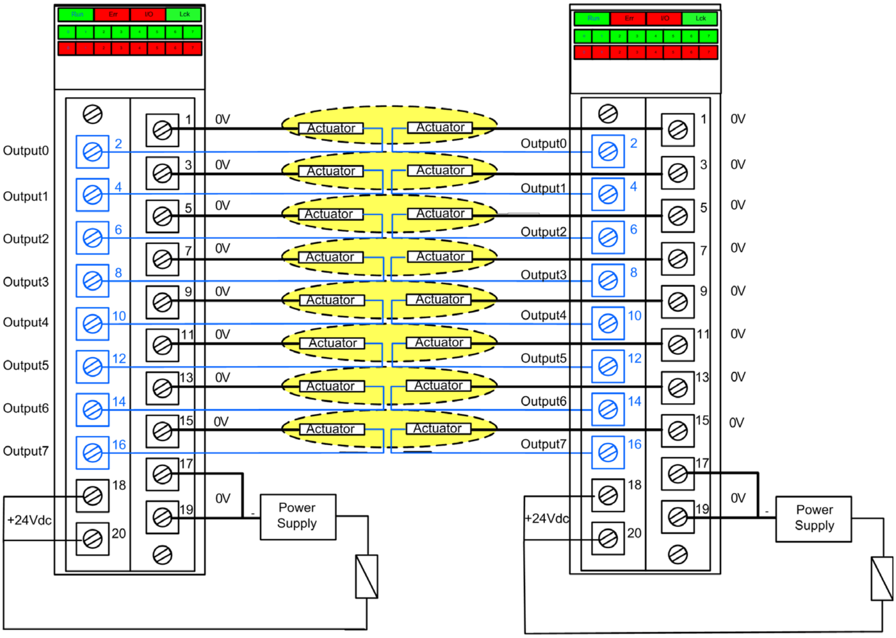
SIL 3 Cat4/PLe - Single Digital Output Module Example

The following example presents one exclusive actuator wired to each output on a single output module. Each loop is SIL 3 Cat4/PLe:



SIL 3 Cat4/PLe - High Availability Example:

In the following wiring diagram, two redundant outputs command the same process variable. As displayed below, each output is connected to separate actuators, then each actuator executes the same command sent over different channels. Alternatively, the two redundant outputs could be wired together to command the same actuator.



Output Wiring Diagnostic Summary

The two designs provide the following wiring diagnostics:

Condition	Diagnostic Provided in Output State?	
	Energized	De-energized
Open (or cut) wire[1]	Yes. Diagnosed each cycle.	Yes. Diagnosed each cycle.
Output in overload[2]	Yes. Diagnosed each cycle.	No.
Short circuit to the 0 V ground	Yes. Diagnosed each cycle.	Yes. Diagnostic period < 1 s.
Short circuit to the 24 Vdc[1]	Yes. Diagnostic period < 1 s.	Yes. Diagnosed each cycle.
Crossed circuits between two channels	Yes. Diagnostic period < 1 s.	Yes. Diagnosed each cycle.
1. This diagnostic function is performed if enabled in the module **Configuration** tab in Control Expert. 2. After the condition is resolved, re-arm the output by de-energizing it.		

⚠ WARNING

RISK OF SHORT CIRCUIT TO 0 Vdc GROUND

For the short circuit to the 0 V ground condition with the output state de-energized, it is recommended that you enable the **Open wire detection** option in the module's **Configuration** tab. Alternatively, you will need to apply another safety measure to detect or exclude this condition.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

⚠ WARNING

RISK OF SHORT CIRCUIT TO THE 24 Vdc

For the short circuit to the 24 Vdc condition with the output state either energized or de-energized, it is recommended that you enable the **Short circuit to 24V detection** option in the module's **Configuration** tab. Alternatively, you will need to apply another safety measure to detect or exclude this condition.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

⚠ WARNING

RISK OF CROSSED CIRCUITS

The module cannot detect the crossed circuits between two channels condition with the output state de-energized and the other channel de-energized. You will need to apply another safety measure to detect or exclude this condition, if it occurs when the output state changes to energized.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

⚠ WARNING

RISK OF CROSSED CIRCUITS

For the crossed circuits between two channels condition with the output state de-energized and the other channel energized, it is recommended that you enable the **Short circuit to 24V detection** option in the module's **Configuration** tab. Alternatively, you will need to apply another safety measure to detect or exclude this condition when the output state changes to energized.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

⚠ WARNING

RISK OF CROSSED CIRCUITS

The module cannot detect the crossed circuits between two channels condition with the output state energized and the other channel de-energized. You will need to apply another safety measure to detect or exclude this condition.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

⚠ WARNING

RISK OF CROSSED CIRCUITS

For the crossed circuits between two channels condition with the output state energized and the other channel energized, it is recommended that you enable the **Short circuit to 24V detection** option in the module's **Configuration** tab. Alternatively, you will need to apply another safety measure to detect or exclude this condition.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

BMXSDO0802 Data Structure

Introduction

The `T_U_DIS_SIS_OUT_8` device derived data type (DDDT) is the interface between the BMXSDO0802 digital output module and the application running in the CPU. The `T_U_DIS_SIS_OUT_8` DDDT incorporates the data types `T_SAFE_COM_DBG_OUT` and `T_U_DIS_SIS_CH_OUT`.

All of these structures are described, below.

`T_U_DIS_SIS_OUT_8` DDDT Structure

The `T_U_DIS_SIS_OUT_8` DDDT structure includes the following elements:

Element	Data Type	Description	Access
MOD_HEALTH[1]	BOOL	● 1: The module is operating correctly. ● 0: The module is not operating correctly.	RO
SAFE_COM_STS[1]	BOOL	● 1: Module communication is valid. ● 0: Module communication is not valid.	RO
PP_STS	BOOL	● 1: The process power supply is operational. ● 0: The process power supply is not operational.	RO
CONF_LOCKED	BOOL	● 1: Module configuration is locked. ● 0: Module configuration is not locked.	RO
S_COM_DBG	T_SAFE_COM_DBG_OUT	Safe communication debug structure.	RO
CH_OUT	ARRAY[0...7] of T_U_DIS_SIS_CH_OUT	Array of structure of channel.	RO
S_TO	UINT	Safety timeout before module enters fallback state.	RO
MUID[2]	ARRAY[0...3] of DWORD	Module unique ID (auto-assigned by Control Expert)	RO
RESERVED_1	ARRAY[0...8] of INT	–	–
RESERVED_2	ARRAY[0...6] of INT	–	–
1. When the SAFE task on CPU is not in running mode, the data exchanged between the CPU and the module are not updated and MOD_HEALTH and SAFE_COM_STS are set to 0. 2. This auto-generated value can be changed by executing the **Build → Renew Ids & Rebuild All** command in the Control Expert main menu.			

T_SAFE_COM_DBG_OUT Structure

The T_SAFE_COM_DBG_OUT structure includes the following elements:

Element	Data Type	Description	Access
S_COM_EST	BOOL	● 1: Communication with the module is established. ● 0: Communication with the module is not established or corrupted.	RO
M_NTP_SYNC	BOOL	With CPU firmware 3.10 or earlier: ● 1: The module is synchronized with the NTP server. ● 0: The module is not synchronized with the NTP server. **NOTE:** With CPU firmware 3.20 or later, the value is always 1.	RO
CPU_NTP_SYNC	BOOL	With CPU firmware 3.10 or earlier: ● 1: The CPU is synchronized with the NTP server. ● 0: The CPU is not synchronized with the NTP server. **NOTE:** With CPU firmware 3.20 or later, the value is always 1.	RO
CHECKSUM	BYTE	Communication frame checksum.	RO
COM_DELAY	UINT	Communication delay between two values received by the module: ● 1...65534: The time, in ms, since the last communication was received by the CPU from the module. ● 65535: The CPU did not receive a communication from the module.	RO
COM_TO	UINT	Communication time-out value for communications coming from the module.	R/W
STS_MS_IN	UINT	Safe timestamp value for the fraction of a second, to the nearest ms, of the data received from the module.	RO
S_NTP_MS	UINT	Safe time value for the fraction of a second, to the nearest ms, for the current cycle.	RO
STS_S_IN	UDINT	Safe timestamp value in seconds of the data received from the module.	RO
S_NTP_S	UDINT	Safe time value in seconds for the current cycle.	RO
CRC_IN	UDINT	CRC value for data received from the module.	RO

Element	Data Type	Description	Access
STS_MS_OUT	UINT	Safe timestamp value for the fraction of a second, to the nearest ms, of the data to be sent to the module.	RO
STS_S_OUT	UDINT	Safe timestamp value in seconds of the data to be sent to the module.	RO
CRC_OUT	UDINT	CRC value for data to be sent to the module.	RO

`T_U_DIS_SIS_CH_OUT` Structure

The `T_U_DIS_SIS_CH_OUT` structure includes the following elements:

Element	Data Type	Description	Access
CH_HEALTH[1]	BOOL	● 1: The channel is operational. ● 0: An error has been detected on the channel, which is not operational. **Formula**: CH_HEALTH = not (SC or OL or IC or OC) and SAFE_COM_STS and not (module in Fallback state)	RO
VALUE	EBOOL	Safe command of output channel: ● 1: Command the output closed (energized). ● 0: Command the output open (de-energized).	R/W
TRUE_VALUE[2]	BOOL	Read back value of the output relay channel: ● 1: The output is closed (energized). ● 0: The output is open (de-energized).	RO
OC	BOOL	● 1: The channel is open or short circuited to ground. ● 0: The channel is connected and not short circuited to ground.	RO
SC	BOOL	● 1: The channel is short circuited to a 24 V source, or cross circuited with another channel. ● 0: The channel is not short circuited to a 24 V source or cross circuited.	RO
1. When the SAFE task on CPU is not in running mode, the data exchanged between the CPU and the module are not updated and CH_HEALTH is set to 0. 2. The TRUE_VALUE element can be time-stamped by the BMX CRA or the BME CRA.			

Element	Data Type	Description	Access
OL	BOOL	● 1: The channel is overloaded or short circuited to 0V. ● 0: The channel is not overloaded or short circuited to 0V.	RO
IC	BOOL	● 1: Invalid channel detected by the module. ● 0: The channel is declared internally operational by the module.	RO
V_OC	BOOL	Configuration status of the open circuit test: ● 1: Enabled. ● 0: Disabled.	RO
V_SC	BOOL	Configuration status of the short circuit to 24 V source test: ● 1: Enabled. ● 0: Disabled.	RO
V_PULSE_ON	BOOL	Configuration status of the pulse test to energized: ● 1: Enabled. ● 0: Disabled.	RO
CH_FBC	BOOL	Configuration of the Channel fallback setting: ● 1: User defined value. ● 0: Hold last value.	RO
CH_FBST	BOOL	Configuration of the channel fallback state when user defined is selected: ● 1: Energized. ● 0: De-energized.	RO
1. When the SAFE task on CPU is not in running mode, the data exchanged between the CPU and the module are not updated and CH_HEALTH is set to 0. 2. The TRUE_VALUE element can be time-stamped by the BMX CRA or the BME CRA.			

Section 6.5
BMXSRA0405 Digital Relay Output Module

Introduction

This section describes the BMXSRA0405 M580 safety digital relay output module.

What Is in This Section?

This section contains the following topics:

BMXSRA0405 Safety Digital Relay Output Module

Introduction

The BMXSRA0405 safety digital relay output module presents the following features:

- 4 relay outputs with 5 A current.
- Rated output voltage of 24 Vdc and 24...230 Vac (over voltage category II).
- Achieves up to SIL3 (IEC61508) Category 4 (Cat4) / Performance Level e (PLe) evaluation.
- Support for 8 pre-defined application wiring configuration selections.
- Configurable automatic self-test monitoring of the relay capacity to execute the commanded output state (depending on the selected application wiring configuration).
- Configurable module settings for fallback mode and fallback timeout (in ms).
- LED diagnostic display *(see page 246)* provided for the module and for each output channel.
- Module hot swap during runtime.
- Module CCOTF when operating in maintenance mode *(see page 255)*. (CCOTF is not supported in safety mode. *(see page 254)*)

BMXSRA0405 Wiring Connector

Introduction

The BMXSRA0405 digital relay output module includes 4 relays and supports up to 4 outputs. The module presents a pair of *a* and *b* pins for each relay. Note that for each relay:

- the two *a* pins are internally connected, and
- the two *b* pins also are internally connected.

Terminal Blocks

You can use the following Schneider Electric 20-point terminal blocks to fit the 20 pin connector on the front of the module:

- screw clamp terminal block BMXFTB2010
- age clamp terminal block BMXFTB2000
- spring type terminal block BMXFTB2020

NOTE: Terminal blocks can be removed only when power to the module is OFF.

Process Power Supply

You need to install the appropriate 24 Vdc or 24 Vac to 230 Vac process power supply.

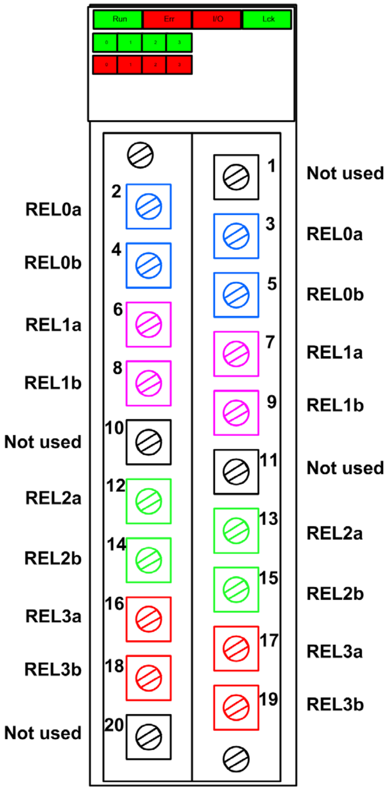
Fuse

A fast blow fuse, maximum of 6 A, that is suitable for the selected application and relay design is required. Always install an external fuse in series with the external power supply, the relay, and the load.

⚠ WARNING
RISK OF UNINTENDED OPERATION It is your responsibility to implement appropriate wiring diagnostics to detect and prevent the occurrence of dangerous faults on the external wiring. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

Wiring Connector

The following example presents the pins on the relay module:



Mapping Inputs to Connector Pins

The following provides a description of each pin on the BMXSRA0405 digital relay output module:

Pin Description	Pin Number on Terminal Block		Pin Description
NO contact, Relay 0a	2	1	Not used
NO contact, Relay 0b	4	3	NO contact, Relay 0a
NO contact, Relay 1a	6	5	NO contact, Relay 0b
NO contact, Relay 1b	8	7	NO contact, Relay 1a
Not used	10	9	NO contact, Relay 1b
NO contact, Relay 2a	12	11	Not used
NO contact, Relay 2b	14	13	NO contact, Relay 2a
NO contact, Relay 3a	16	15	NO contact, Relay 2b
NO contact, Relay 3b	18	17	NO contact, Relay 3a
Not used	20	19	NO contact, Relay 3b

NOTE: Because the two *a* pins for each relay are internally connected, you need to use only one *a* pin for each relay. Similarly, because the two *b* pins for each relay are internally connected, you need to use only one *b* pin for each relay.

BMXSRA0405 Output Application Wiring Examples

Introduction

You can configure the BMXSRA0405 safety digital output relay module to achieve either SIL2 Category 2 (Cat2) / Performance Level c (PLc) or SIL3 Cat4 / PLe compliance in different ways, depending on:

- the number of outputs the module will support, and
- how you want to test the ability of the module to place the actuator into the intended demand state, either:
  - automatically by the module (in this case, there is no state transition for the actuator) or
  - by means of a procedure that performs and checks a daily transition of the signal from the module to the actuator (in this case, the transition impacts the actuator state).

Make this configuration by selecting an application number (described in the tables, below) in the **Function** list in the module **Configuration** tab in Control Expert.

SIL2 Cat2 / PLc wiring design applications:

Function	Demand State	Relays	Outputs	Signal Test?		Wiring Diagram (see below)
				Automatic Signal Test?[1]	Daily Signal Transition?	
Application_1	De-energized	1	4	No	Yes	A
Application_2	De-energized	2	2	Yes	No	B
Application_3	Energized	1	4	No	Yes	A
Application_4	Energized	2	2	Yes	No	C
1. The automatic signal test does not impact the actuator state.						

SIL3 Cat4 / PLe wiring design applications:

Function	Demand State	Relays	Outputs	Signal Test?		Wiring Diagram (see below)
				Automatic Signal Test?[1]	Daily Signal Transition?	
Application_5	De-energized	2	2	No	Yes	C
Application_6	De-energized	4	1	Yes	No	D
Application_7	Energized	2	2	No	Yes	C
Application_8	Energized	2	2	Yes	No	C
1. The automatic signal test does not impact the actuator state.						

Each of these eight application selections are described in the following wiring examples.

Application_1: 4 Outputs, SIL2 / Cat2 / PLc, De-energized State, No Automatic Signal Test

The demand state for this application design is de-energized. If the module detects an internal error for an output, it de-energizes that output.

⚠ CAUTION

LOSS OF THE ABILITY TO PERFORM SAFETY FUNCTIONS

To achieve SIL2 according to IEC61508 and Category 2 / Performance Level c according to ISO 13849 using this wiring design, you need to perform a daily signal transition from the energized state to the de-energized state.

Failure to follow these instructions can result in injury or equipment damage.

Refer to wiring diagram A *(see page 118)*, below, for a depiction of the wiring design for Application_1.

Application_2: 2 Outputs, SIL2 Cat2 / PLc, De-energized State, Automatic Signal Test

The demand state for this application design is de-energized. If the module detects an internal output error on one of the relays used for an output, it de-energizes both relays (Relay 0 and Relay 1 or Relay 2 and Relay 3) for that output.

Your application program needs to command the same output state to all relays that activate the same actuator.

The module sequentially performs an automatic periodic pulse test on each relay. The duration of the test is less than 50 ms. Because of the configuration of the two relays used (in parallel), the test has no impact on the output load (normally *energized*). You can configure the frequency of the test by setting the **Monitoring period** in the **Configuration** tab of the module. Valid test frequency values range from1...1440 minutes.

Refer to wiring diagram B *(see page 119)*, below, for a depiction of the wiring design for Application_2.

Application_3: 4 Outputs, SIL2 / Cat2 / PLc, Energized State, No Automatic Signal Test

The demand state for this application design is energized. If the module detects an internal error for an output, it de-energizes that output, which is the safe state.

⚠ CAUTION

LOSS OF THE ABILITY TO PERFORM SAFETY FUNCTIONS

To achieve SIL2 according to IEC61508 and Category 2 / Performance Level c according to ISO 13849 using this wiring design, you need to perform a daily signal transition from the energized state to the de-energized state.

Failure to follow these instructions can result in injury or equipment damage.

Refer to wiring diagram A *(see page 118)*, below, for a depiction of the wiring design for Application_3.

Application_4: 2 Outputs, SIL2 Cat2 / PLc, Energized State, Automatic Signal Test

The demand state for this application design is energized. If the module detects an internal output error on one of the relays used for an output, it de-energizes both relays (Relay 0 and Relay 1 or Relay 2 and Relay 3) for that output.

Your application program needs to command the same output state to all relays that activate the same actuator.

The module sequentially performs a periodic pulse test on each relay. The duration of the test is less than 50 ms. Because of the configuration of the two relays used (in parallel), the test has no impact on the output load (normally *energized*). You can configure the frequency of the test by setting the **Monitoring period** in the **Configuration** tab of the module. Valid test frequency values range from1...1440 minutes.

Refer to wiring diagram C *(see page 120)*, below, for a depiction of the wiring design for Application_4.

Application_5: 2 Outputs, SIL3 / Cat4 / PLe, De-energized State, No Automatic Signal Test

The demand state for this application design is de-energized. If the module detects an internal output error on one of the relays used for an output, it de-energizes both relays (Relay 0 and Relay 1 or Relay 2 and Relay 3) for that output.

Your application program needs to command the same output state to all relays that activate the same actuator.

⚠ CAUTION

LOSS OF THE ABILITY TO PERFORM SAFETY FUNCTIONS

To achieve SIL3 according to IEC61508 and Category 4 / Performance Level e according to ISO 13849 using this wiring design, you need to perform a daily signal transition from the energized state to the de-energized state.

Failure to follow these instructions can result in injury or equipment damage.

Refer to wiring diagram C *(see page 120)*, below, for a depiction of the wiring design for Application_5.

Application_6: 1 Output, SIL3 / Cat4 / PLe, De-energized State, Automatic Signal Test

The demand state for this application design is de-energized. If the module detects an internal output error on one of the relays used for an output, it de-energizes all relays (Relay 0, Relay 1, Relay 2, and Relay 3) for the module.

Your application program needs to command the same output state to all relays that activate the same actuator.

The module sequentially performs a periodic pulse test on each relay. The duration of the test is less than 50 ms. Because of the configuration of the four relays used (2 pair of two serial relays set in parallel), the test has no impact on the output load (normally *energized*). You can configure the frequency of the test by setting the **Monitoring period** in the **Configuration** tab of the module. Valid test frequency values range from1...1440 minutes.

Refer to wiring diagram D *(see page 121)*, below, for a depiction of the wiring design for Application_6.

Application_7: 2 Outputs, SIL3 / Cat4 / PLe, Energized State, No Automatic Signal Test

The demand state for this application design is energized. If the module detects an internal output error on one of the relays used for an output, it de-energizes both relays (Relay 0 and Relay 1 or Relay 2 and Relay 3) for that output.

Your application program needs to command the same output state to all relays that activate the same actuator.

⚠ CAUTION

LOSS OF THE ABILITY TO PERFORM SAFETY FUNCTIONS

To achieve SIL3 according to IEC61508 and Category 4 / Performance Level e according to ISO 13849 using this wiring design, you need to perform a daily signal transition from the energized state to the de-energized state.

Failure to follow these instructions can result in injury or equipment damage.

Refer to wiring diagram C *(see page 120)*, below, for a depiction of the wiring design for Application_7.

Application_8: 2 Outputs, SIL3 Cat4 / PLe, Energized State, Automatic Signal Test

The demand state for this application design is energized. If the module detects an internal output error on one of the relays used for an output, it de-energizes both relays (Relay 0 and Relay 1 or Relay 2 and Relay 3) for that output.

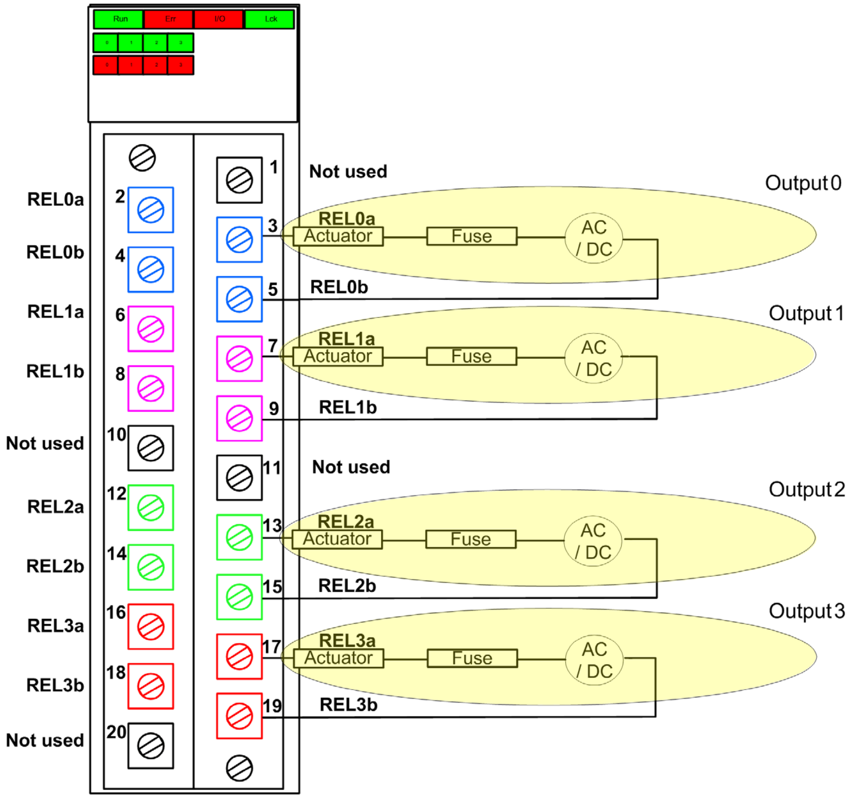
Your application program needs to command the same output state to all relays that activate the same actuator.

The module sequentially performs a periodic pulse test on each relay. The duration of the test is less than 50 ms. Because of the configuration of the two relays used (in serial), the test has no impact on the output load (normally *de-energized*). You can configure the frequency of the test by setting the **Monitoring period** in the **Configuration** tab of the module. Valid test frequency values range from1...1440 minutes.

Refer to wiring diagram C *(see page 120)*, below, for a depiction of the wiring design for Application_8.

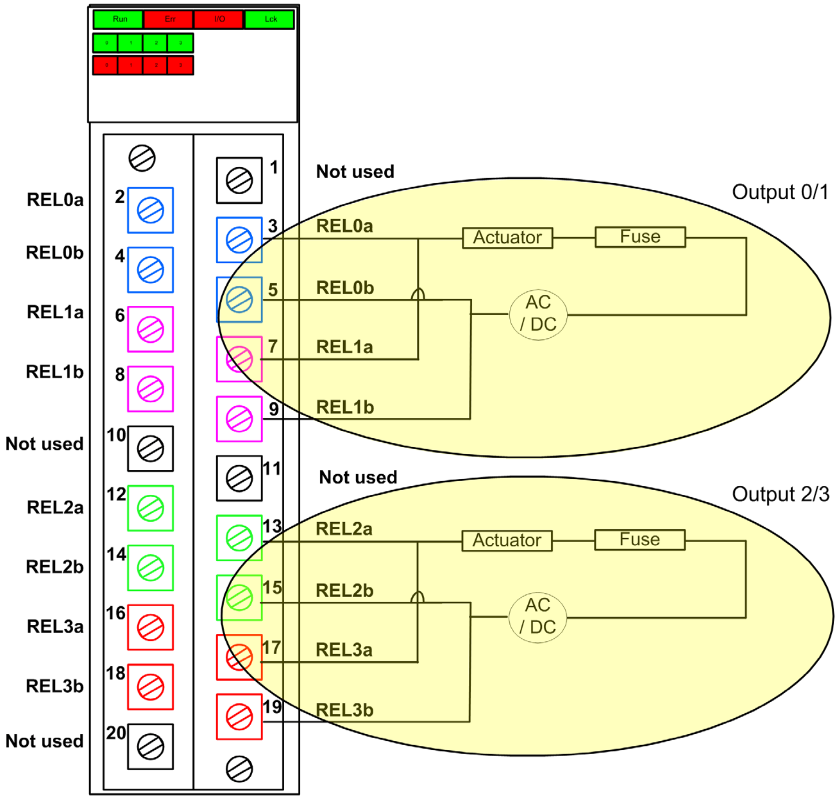
Wiring Diagram A

This wiring diagram applies to Application_1 and Application_3:



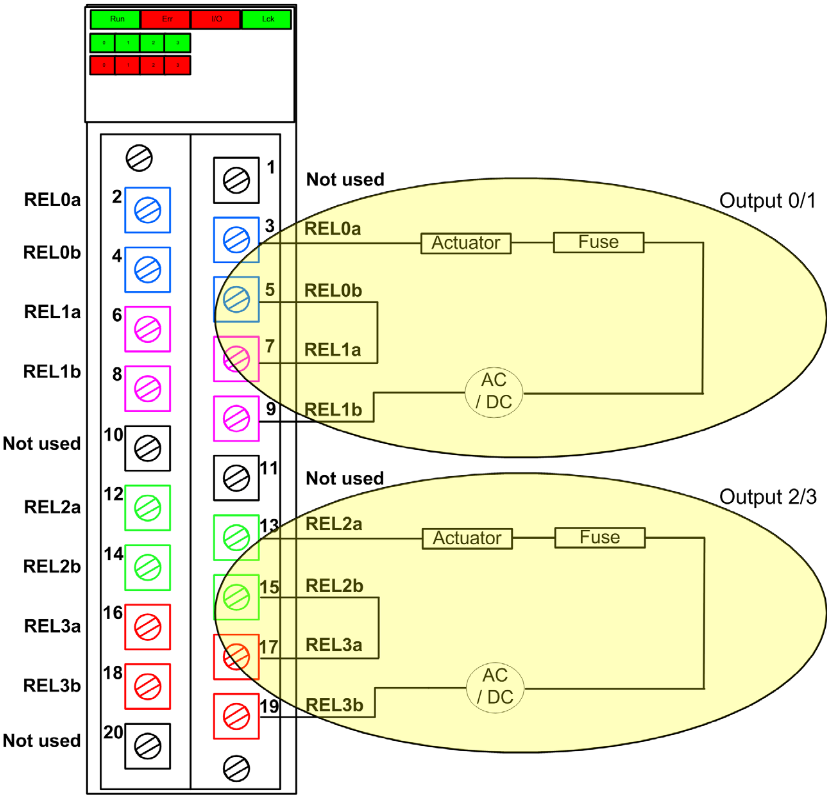
Wiring Diagram B

This wiring diagram applies to Application_2:



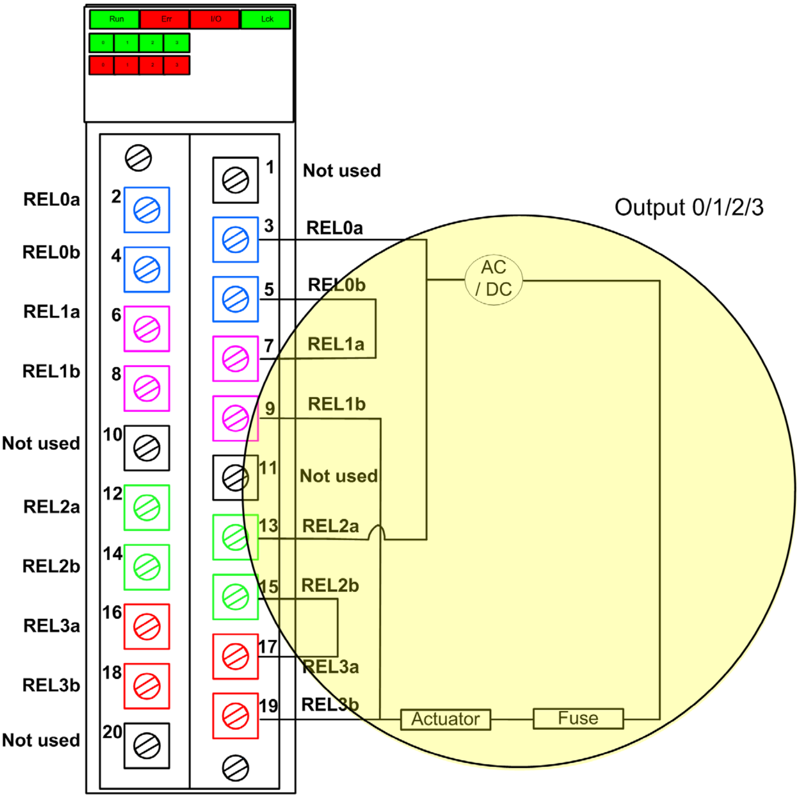
Wiring Diagram C

This wiring diagram applies to Application_4, Application_5, Application_7 and Application_8:



Wiring Diagram D

This wiring diagram applies to Application_6:



BMXSRA0405 Data Structure

Introduction

The `T_U_DIS_SIS_OUT_4` device derived data type (DDDT) is the interface between the BMXSRA0405 relay output module and the application running in the CPU. The `T_U_DIS_SIS_OUT_4` DDDT incorporates the data types `T_SAFE_COM_DBG_OUT` and `T_U_DIS_SIS_CH_ROUT`.

All of these structures are described, below.

`T_U_DIS_SIS_OUT_4` DDDT Structure

The `T_U_DIS_SIS_OUT_4` DDDT structure includes the following elements:

Element	Data Type	Description	Access
MOD_HEALTH[1]	BOOL	● 1: The module is operating correctly. ● 0: The module is not operating correctly.	RO
SAFE_COM_STS[1]	BOOL	● 1: Module communication is valid. ● 0: Module communication is not valid.	RO
CONF_LOCKED	BOOL	● 1: Module configuration is locked. ● 0: Module configuration is not locked.	RO
APPLI	UINT	Relay application configuration: 1, 2, 3, 4, 5, 6, 7 or.	RO
TIME_PERIOD	UINT	Timer period for relay automatic monitoring (in minutes).	RO
S_COM_DBG	T_SAFE_COM_DBG_OUT	Safe communication debug structure.	RO
CH_OUT	ARRAY[0...3] of T_U_DIS_SIS_CH_ROUT	Array of structure of channel.	–
S_TO	UINT	Safety timeout before module enters fallback state.	RO
MUID[2]	ARRAY[0...3] of DWORD	Module unique ID (auto-assigned by Control Expert)	RO
RESERVED_1	ARRAY[0...7] of INT	–	–
RESERVED_2	ARRAY[0...6] of INT	–	–
1. When the SAFE task on CPU is not in running mode, the data exchanged between the CPU and the module are not updated and MOD_HEALTH and SAFE_COM_STS are set to 0. 2. This auto-generated value can be changed by executing the **Build → Renew Ids & Rebuild All** command in the Control Expert main menu.			

T_SAFE_COM_DBG_OUT Structure

The T_SAFE_COM_DBG_OUT structure includes the following elements:

Element	Data Type	Description	Access
S_COM_EST	BOOL	● 1: Communication with the module is established. ● 0: Communication with the module is not established or corrupted.	RO
M_NTP_SYNC	BOOL	With CPU firmware 3.10 or earlier: ● 1: The module is synchronized with the NTP server. ● 0: The module is not synchronized with the NTP server. **NOTE:** With CPU firmware 3.20 or later, the value is always 1.	RO
CPU_NTP_SYNC	BOOL	With CPU firmware 3.10 or earlier: ● 1: The CPU is synchronized with the NTP server. ● 0: The CPU is not synchronized with the NTP server. **NOTE:** With CPU firmware 3.20 or later, the value is always 1.	RO
CHECKSUM	BYTE	Communication frame checksum.	RO
COM_DELAY	UINT	Communication delay between two values received by the module: ● 1...65534: The time, in ms, since the last communication was received by the CPU from the module. ● 65535: The CPU did not receive a communication from the module.	RO
COM_TO	UINT	Communication time-out value for communications coming from the module.	R/W
STS_MS_IN	UINT	Safe timestamp value for the fraction of a second, to the nearest ms, of the data received from the module.	RO
S_NTP_MS	UINT	Safe time value for the fraction of a second, to the nearest ms, for the current cycle.	RO
STS_S_IN	UDINT	Safe timestamp value in seconds of the data received from the module.	RO
S_NTP_S	UDINT	Safe time value in seconds for the current cycle.	RO
CRC_IN	UDINT	CRC value for data received from the module.	RO

Element	Data Type	Description	Access
STS_MS_OUT	UINT	Safe timestamp value for the fraction of a second, to the nearest ms, of the data to be sent to the module.	RO
STS_S_OUT	UDINT	Safe timestamp value in seconds of the data to be sent to the module.	RO
CRC_OUT	UDINT	CRC value for data to be sent to the module.	RO

T_U_DIS_SIS_CH_ROUT Structure

The T_U_DIS_SIS_CH_ROUT structure includes the following elements:

Element	Data Type	Description	Access
CH_HEALTH[1]	BOOL	● 1: The channel is operational. ● 0: An error has been detected on the channel, which is not operational. **Formula**: CH_HEALTH = not (IC) and SAFE_COM_STS and not (module in Fallback state)	RO
VALUE	EBOOL	Safe command of output channel: ● 1: Command the output closed (energized). ● 0: Command the output open (de-energized).	R/W
TRUE_VALUE[2]	BOOL	Read back value of the relay output channel: ● 1: The output is closed (energized). ● 0: The output is open (de-energized).	RO
IC	BOOL	● 1: Invalid channel detected by the module. ● 0: The channel is declared internally operational by the module.	RO

1. When the SAFE task on CPU is not in running mode, the data exchanged between the CPU and the module are not updated and CH_HEALTH is set to 0.
2. The TRUE_VALUE element can be time-stamped by the BMX CRA or the BME CRA.

Element	Data Type	Description	Access
CH_FBC	BOOL	Configuration of the Channel fallback setting: ● 1: User defined value. ● 0: Hold last value.	RO
CH_FBST	BOOL	Configuration of the channel fallback state when user defined is selected: ● 1: Energized. ● 0: De-energized.	RO
1. When the SAFE task on CPU is not in running mode, the data exchanged between the CPU and the module are not updated and CH_HEALTH is set to 0. 2. The TRUE_VALUE element can be time-stamped by the BMX CRA or the BME CRA.			

Chapter 7
M580 Safety Power Supplies

Introduction

This chapter describes the M580 safety power supply modules.

What Is in This Chapter?

This chapter contains the following topics:

M580 Safety Power Supplies

Introduction

The following power supplies can be used with the M580 safety PAC:

- BMXCPS4002S 100...240 Vac redundant safety power supply
- BMXCPS4022S 24/48 Vdc redundant high power safety power supply
- BMXCPS3522S 125 Vdc redundant high power safety power supply

⚠ **WARNING**

LOSS OF THE ABILITY TO PERFORM SAFETY FUNCTIONS

Use only a BMXCPS4002S, BMXCPS4022S, or BMXCPS3522S power supply in any rack that includes an M580 safety module. Check both your physical installation and your project in Control Expert to confirm that only M580 safety power supply modules are used.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Power Supply Functionality

Each M580 safety power supply module converts Vdc or Vac power into two output voltages, 24 Vdc and 3.3 Vdc, as described below:

Features	Power Supply		
	BMXCPS4002S	BMXCPS4022S	BMXCPS3522S
Main input power network	100...240 Vac, 50...60 Hz	24...48 Vdc	100...150 Vdc
Power limit output to backplane	40 Vdc	40 Vdc	40 Vdc
Ambient temperature for power limit	-25° C...+60° C	-25° C...+60° C	-25° C...+60° C
Wire to	● AC network with neutral wired to the earth OR ● AC network with the neutral insulated and impedant against the earth, with AC neutral fused by user.	A DC network 24...48 Vdc	A DC network 125 Vdc

Each power supply detects over voltage, overload, and short-circuit conditions on both the 3.3 Vdc and 24 Vdc backplane lines.

If the 40 Vdc upper threshold is detected, the module takes the following responsive actions:

- A reset is performed, causing the modules that receive power from the power supply to be re-initialized.
- If the upper voltage threshold was detected on the:
  - 24 Vdc backplane line: the PAC is powered down.
  - 3.3 Vdc backplane line: the PAC operation stops, but the PAC continues to receive power.

Refer to the topic *Diagnostics for the 24 Vdc and 3.3 Vdc Backplane Voltages (see page 130)* for information on how to respond to these conditions.

Redundant Power Supply Modules

The BMXCPS4002S, BMXCPS4022S, and BMXCPS3522S are redundant power supply modules. Two of these power supply modules can be installed - one as master one as slave - in a redundant Ethernet rack. The possible configurations include the following:

Configuration	Features		
	Manage Redundancy (Power Control and LED Signals)	Provide Data to the Application	Monitor and Save Power Supply Data
2 power supplies in main rack	✓	✓	✓
2 power supplies in extension rack	✓	X	✓
1 power supply in a legacy rack	X	X	✓
✓ = Supported. X = Not supported.			

For more information on redundant power supplies, refer to chapter *Modicon X80 Power Supply Modules Description (see Modicon X80, Racks and Power Supplies, Hardware Reference Manual)*.

M580 Safety Power Supply Module Diagnostics

Diagnostics for the 24 Vdc and 3.3 Vdc Backplane Voltages

The BMXCPS4002S, BMXCPS4022S, and BMXCPS3522S safety power supplies automatically provide detection for an overvoltage, overload, or short-circuit condition that may occur with respect to both the 24 VDC and 3.3 VDC Backplane Voltages.

If the power supply detects one of these conditions on the 24 Vdc voltage, the following occurs:

- The power conversion function is shut down for the entire backplane.
- A RESET command is issued for all modules in the rack.
- The power supply **OK** LED is turned OFF.
- The entire PAC is powered down.

If the power supply detects one of these conditions on the 3.3 Vdc voltage, the following occurs:

- The power conversion function is shut down for the 3.3 Vdc backplane voltage.
- A RESET command is issued for all modules in the rack.
- The power supply **OK** LED is turned OFF.
- Operation of the entire PAC program is stopped, although some PAC circuits may continue to receive power.

In either case, to recover from these conditions, take the following steps:

1. Power down the primary power line.
2. Check the compatibility between the estimated power supply consumption of the PAC against the capacity of the M580 safety power supply module on the 24 Vdc and 3.3 Vdc backplane lines.
3. Eliminate the cause of the underlying condition.
4. Wait for 1 minute after power down.
5. Apply power on the primary line to restart the M580 safety power supply module.

Alarm Relay Contact Diagnostics

The BMXCPS4002S, BMXCPS4022S, and BMXCPS3522S safety power supplies present a two-pin alarm relay contact that you can use to obtain the following information:

- If the relay is activated (i.e. closed):
  - Both the 24 Vdc and 3.3 Vdc backplane voltages are OK; and
  - RESET is not active.; and
  - If the power supply is placed in the main local rack:
    - the CPU is operational, and
    - the CPU is in RUN mode.
- If the relay is de-activated (i.e. open):
  - Either or both of the 24 Vdc and 3.3 Vdc backplane voltages are not OK; or
  - RESET is active; or
  - If the power supply is placed in the main local rack:
    - the CPU is not operational, or
    - the CPU is in STOP mode.

M580 Safety DDTs

Introduction

The M580 safety power supply modules present two sets of derived data types (DDTs):

- PWS_DIAG_DDT_V2 for diagnostics
- PWS_CMD_DDT for commands

PWS_DIAG_DDT_V2

Byte Offset	Name	Type	Comment
0	Reserved	BYTE	–
1	Reserved	BYTE	–
2	PwsMajorVersion	BYTE	Power supply Major firmware version
3	PwsMinorVersion	BYTE	Power supply Minor firmware version
4	Model	BYTE	Model identifier Model identifier: ● BMXCPS4002S = 01 ● BMXCPS4022S = 02 ● BMXCPS3522S = 03
5	State	BYTE	Power supply state
6	I33BacPos	UINT	Measure current on 3.3V backplane line in nominal role (producer)
8	V33Buck	UINT	Measure voltage of 3.3V Buck
10	I24Bac	UINT	Measure current of 24V backplane line
12	V24Int	UINT	Measure voltage of 24V Int
14	Temperature	INT	Measure of Ambient Temperature
16	OperTimeMasterSincePO	UDINT	Operating Time as Master since last Power ON
20	OperTimeSlaveSincePO	UDINT	Operating Time as Slave since last Power ON
24	OperTimeMaster	UDINT	Operating Time as Master since Manufacturing
28	OperTimeSlave	UDINT	Operating Time as Slave Since Manufacturing
32	Work	UDINT	Work supplied since Manufacturing
36	RemainingLTPC	UINT	Remaining Life Time in percent
38	NbPowerOn	UINT	Number of Power ON since Manufacturing
40	NbVoltageLowFail	UINT	Number of failure detected on Primary Voltage by Low Threshold
42	NbVoltageHighFail	UINT	Number of failure detected on Primary Voltage by High Threshold
44	Reserved	UDINT	–

Byte Offset	Name	Type	Comment
48	Reserved	UDINT	–
52	RemainingLTMO	UINT	Remaining Life Time in month
54	Reserved	BYTE	–
63	Reserved	BYTE	–

PWS_CMD_DDT

Byte Offset	Name	Type	Comment
0	Reserved	BYTE	–
1	Code	BYTE	Command code: ● 1 = swap ● 3 = clear
2	PwsTarget	BYTE	Power supply target: 1 for left& 2 for right& 3 for both Power supply target: ● 1 = left ● 2 = right
3	Reserved	BYTE	–
15	Reserved	BYTE	–

Chapter 8
Validating an M580 Safety System

Introduction

This chapter shows you how to perform calculations that validate your M580 safety system.

What Is in This Chapter?

This chapter contains the following sections:

Section 8.1
M580 Safety Module Architectures

Introduction

This section presents the internal architectures of the safety modules.

What Is in This Section?

This section contains the following topics:

M580 Safety CPU and Coprocessor Safety Architecture

Introduction

The BME•58•040S CPUs and the BMEP58CPROS3 Coprocessor (Copro), acting as a pair of processors, are certified by the TÜV Rheinland Group for use in Safety Integrity Level 3 (SIL3) M580 safety solutions.

Working together, the CPU and Copro provide the following SIL3 safety level functions:

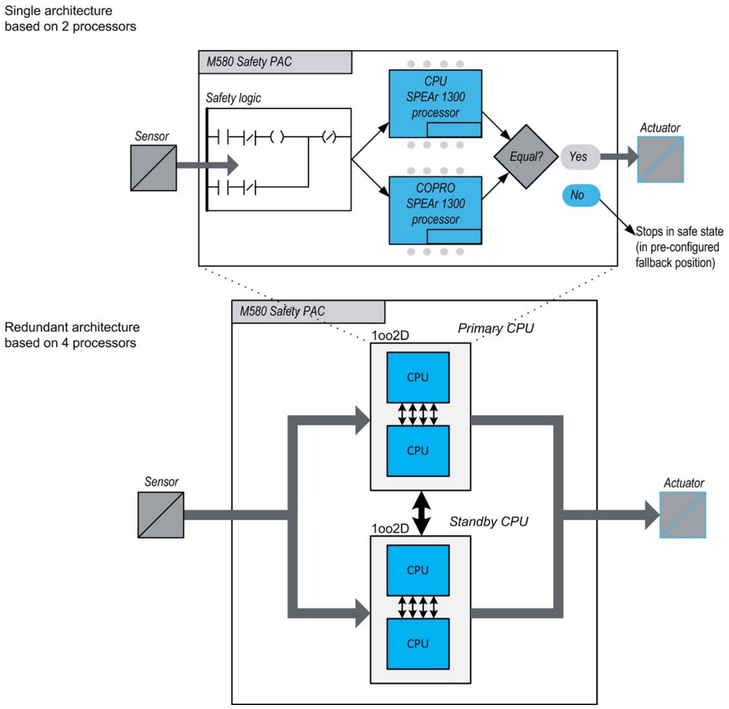
- Independent double execution of the safety task code.
- Comparison of the results of the double code execution.
- Periodic self-tests.
- Support for a 1oo2D (“one out of two”) architecture with diagnostic.

NOTE: In addition to the safety functionality, the BMEP58•040S CPUs provide comparable features of equivalent non-safety standalone M580 CPUs, and the BMEH58•040S CPUs provide comparable features of equivalent non-safety Hot Standby M580 CPUs. Refer to both the *Modicon M580, Hardware, Reference Manual* and the *Modicon M580 Hot Standby, System Planning Guide for Frequently Used Architectures* for information regarding the non-safety features of these safety CPUs.

Description of the Internal CPU & Copro Architecture

The M580 safety CPU and Copro each contains a SPEAr 1300 processor. Each processor executes the safety logic in its own memory area, and compares the results of the execution at the end of the safe task.

The following figures show the internal architecture of the M580 Safety CPU in a single and a redundant configurations:



Double Code Generation and Execution

The two processors inside the M580 safety PAC provide for double code generation and execution. This diversity provides the following advantages in error detection:

- Two executable code programs are generated independently. The use of two independent code compilers aids in the detection of systemic errors in code generation.
- The two generated code programs are executed by two separate processors. Thus, the CPU can detect both systematic errors in the code execution and random errors in the PAC.
- Each of the two processors uses its own independent memory area. Thus, the PAC can detect random errors in the RAM, and a full RAM test is not necessary at every scan.

1oo2D Architecture

The 1oo2D (“one out of two with Diagnostic”) architecture means that two independent channels execute the safety logic and, if an error is detected on either channel, the system goes in its safe state.

Single Architecture

The Single M580 Safety PAC architecture is based on a 1oo2D architecture made of dual processors providing safety integrated level (SIL3) compliance even in a non Redundant architecture.

Redundant Architecture

The M580 Safety PAC in Redundant architecture provides maximum system availability and process uptime by adding full redundancy (Quadruple Structure, i.e. four CPU) on the control, power supply and communication.

One of the CPU (pair of processors) acts as the Primary, runs the application by executing program logic and operating IO. The Primary CPU (pair of processors) updates the secondary CPU (pair of processors) so that it is ready to assume IO control.

The system monitors itself continuously. In case of the Primary CPU control failure, the system switches control to the secondary CPU. In this degraded mode, the system remains SIL3. In case primary and secondary CPU fails, the system goes in fail safe state.

The redundant M580 Safety PAC, based on a quadruple architecture (4 processors) allows to increase the system availability and provides safety integrated level (SIL3) compliance.

Watchdog

A hardware and a firmware watchdog check the PAC activity and the time required to execute the safety program logic.

NOTE: Configure the software watchdog (in the **Properties of SAFE** dialog) to allow for:

- application execution time
- filtering of any detected I/O communication errors
- process safety time.

For more information, refer to the topic *Process Safety Time (see page 150)*.

Memory Check

The integrity of the content of static memory is tested via cyclic redundancy check (CRC) and the double code execution. The integrity of the content of dynamic memory is tested by double code execution, by a periodic memory test, and by an error correcting code (ECC) mechanism that detects and corrects the most common instances of corrupted internal data. At cold start, these tests are re-initialized and fully performed before the CPU goes into Stop or Run mode.

Over Voltage Monitoring

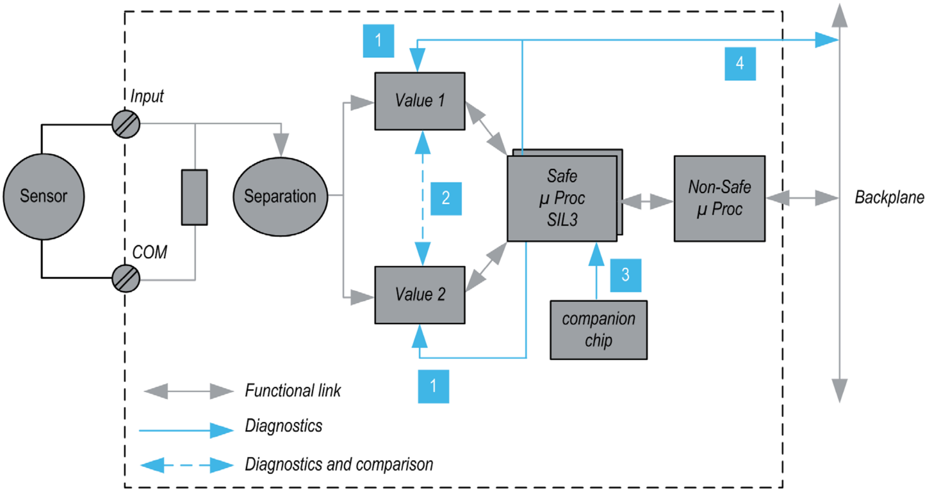
The CPU receives power from the dedicated M580 safety power supply module over the backplane line. The power supply module provides a regulated 24V with an absolute maximum voltage in the range 0...36V.

Embedded in the CPU is an embedded function that checks the internal power supplies. If an undervoltage or overvoltage condition is detected, the PAC shuts down.

BMXSAI0410 Analog Input Module Safety Architecture

Safety Function Architecture

The BMXSAI0410 module internal architecture performs its safety function as follows:



1 The measuring devices are regularly monitored for their ability to measure, without a detected error, 10 analog values between 4 and 20 mA. The linearity of the measuring stages is verified at the same time.

2 Each input value is acquired by 2 identical circuits. The measuring values are compared by the safety processor. If values are different, that channel is determined to be not valid. A maximum discrepancy of 0.35% of the 20 mA full scale range is tolerable between the two values.

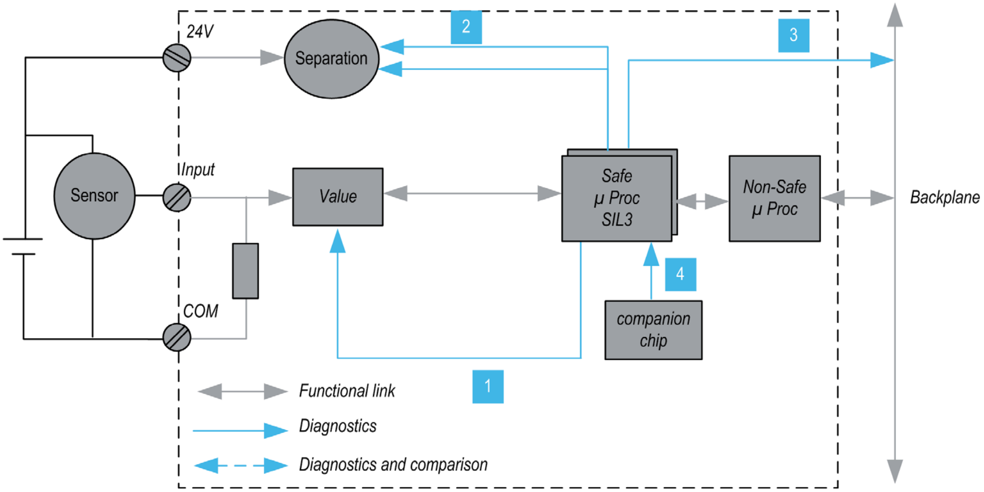
3 The companion chip supplies the safety processor, continuously diagnoses the safety processor, and monitors backplane voltage.

4 The supply voltage from the backplane is monitored to detect if an over voltage or under voltage condition occurs.

BMXSDI1602 Digital Input Module Safety Architecture

Safety Function Architecture

The BMXSDI1602 module internal architecture performs the safety function as follows:



1 The measuring devices are continuously monitored for their ability to measure a "1" and a "0".

2 The external 24 Vdc power supply is continuously monitored by the safety processor. Each input value is acquired by two identical circuits. The acquired values are compared by the safety processor. If values are different, the channel is declared not valid.

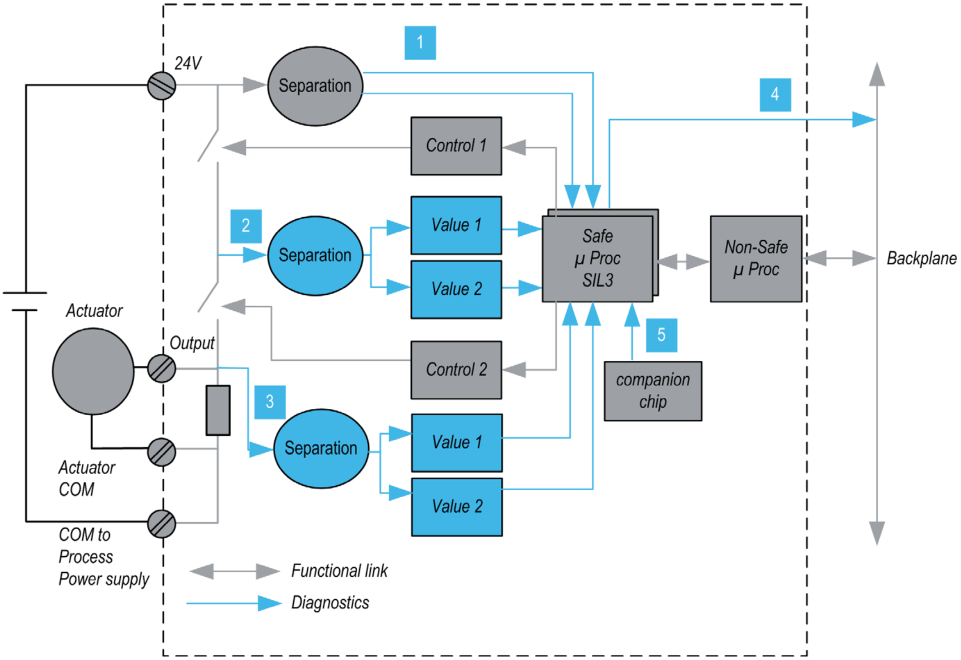
3 The supply voltage from the backplane is monitored to detect an over or under voltage condition.

4 The companion chip supplies the safety processor, continuously diagnoses the safety processor, and monitors backplane voltage.

BMXSDO0802 Digital Output Module Safety Architecture

Safety Function Architecture

The BMXSDO0802 module internal architecture performs the safety function as follows:



1 The external 24 Vdc power supply is continuously monitored by the safety processor.

2 Each output consists of 2 switches in series between the external +24 Vdc power supply and the ground. The mid-point value (2) is redundantly read and sent to safety processor. The mid-points measuring values are compared by the safety processor. If the values are not the expected ones, the channel is declared not valid.

3 The low-point value (3) is also supervised in order to perform external wiring diagnostic.

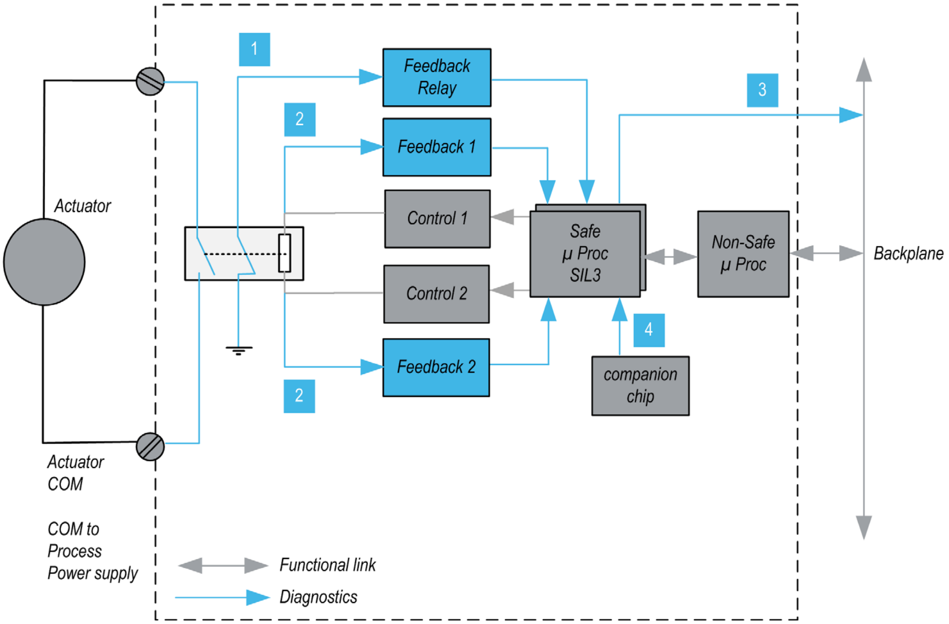
4 The supply voltage from the backplane is monitored to determine if there is an over voltage or under voltage condition.

5 The companion chip supplies the safety processor, continuously diagnoses the safety processor, and monitors backplane voltage.

BMXSRA0405 Digital Relay Output Module Safety Architecture

Safety Function Architecture

The BMXSRA0405 module internal architecture performs the safety function as follows:



1 The state of the relay is continuously monitored by the safety processor, which reads the state of an NC contact mechanically linked to the NO contact, and which in turn is linked to the actuator.

2 The state of the relay command is continuously monitored. Each input is received by 2 identical circuits. The measured values are compared by the safety processor. If values are different, the channel is declared to be not valid.

3 The supply voltage from the backplane is monitored to determine if there is an over voltage or under voltage condition.

4 The companion chip supplies the safety processor, continuously diagnoses the safety processor, and monitors backplane voltage.

Section 8.2
M580 Safety Module SIL & MTTF Values

Safety Integrity Level Calculations

Classification of the Schneider Electric Products

The M580 safety PAC can consist of:

- Safety modules, which can perform safety functions, including:
  - CPU and coprocessor
  - I/O modules
  - power supply
- Non-interfering modules *(see page 27)*, which do not perform safety functions, but enable you to add non-safety elements to your safety project.

NOTE:

- Because non-interfering modules are not part of the safety loop, they are not part of safety integrity level calculations.
- An error detected in a non-interfering module does not negatively impact the execution of the safety functions.
- The BMXCPS4002S, BMXCPS4022S, and BMXCPS3522S power supplies are certified. Because they present a negligible dangerous failure rate (<1% of the SIL3 target), the power supply is not included in safety integrity level calculations for the safety loop. As a consequence, neither PFH nor PFD are provided for the power supply modules.

PFD/PFH Values for M580 Safety Modules

Schneider Electric offers the following safety modules certified for use in safety applications. The safety modules are listed with their corresponding probabilities of failure *(see page 146)* (PFD/PFH) values for different proof test intervals *(see page 148)* (PTIs). The PFD/PFH are expressed as values that contribute to the overall PFD/PFH of the entire safety loop *(see page 18)*.

The tables below list the safety modules and their PFD/PFH values for SIL2 and SIL3 applications, where applicable:

Product Type	Product Reference	SIL	PTI = 1 year	
			PFD_G	PFH_G
CPU with Copro	BME•58•040S & BMEP58CPROS3	SIL3[1]	4.26E-07	9.73E-11
Analog input	BMXSAI0410	SIL3[2]	5.76E-06	1.31E-09
Digital input	BMXSDI1602	SIL3[2]	6.81E-06	1.56E-09
Digital output	BMXSDO0802	SIL3[1]	5.75E-06	1.31E-09
Digital relay output	BMXSRA0405	SIL2 [3]	5.85E-06	1.58E-09
		SIL3[4]	5.84E-06	1.34E-09
		SIL3[5]	–	1.35E-09
Power supply	BMXCPS4002S, BMXCPS4022S, and BMXCPS3522S	SIL3	–	–
1. 1 output @ 80° C 2. 1 input @ 80° C 3. 1 relay per output @ 80° C 4. 2 relays per output @ 80° C 5. 4 relays per output @ 80° C				

Product Type	Product Reference	SIL	PTI = 5 years	
			PFD_G	PFH_G
CPU & Copro	BME•58•040S & BMEP58CPROS3	SIL3 [1]	2.14E-06	9.81E-11
Analog input	BMXSAI0410	SIL3[2]	2.88E-05	1.31E-09
Digital input	BMXSDI1602	SIL3[2]	3.41E-05	1.56E-09
Digital output	BMXSDO0802	SIL3[1]	2.88E-05	1.31E-09
1. 1 output @ 80° C 2. 1 input @ 80° C 3. 1 relay per output @ 80° C 4. 2 relays per output @ 80° C 5. 4 relays per output @ 80° C				

Product Type	Product Reference	SIL	PTI = 5 years	
			PFD_G	PFH_G
Digital relay output	BMXSRA0405	SIL2 [3]	2.92E-05	1.68E-09
		SIL3[4]	2.92E-05	1.34E-09
		SIL3[5]	–	1.35-09
Power supply	BMXCPS4002S, BMXCPS4022S, and BMXCPS3522S	SIL3	–	–
1. 1 output @ 80° C 2. 1 input @ 80° C 3. 1 relay per output @ 80° C 4. 2 relays per output @ 80° C 5. 4 relays per output @ 80° C				

Product Type	Product Reference	SIL	PTI = 10 years	
			PFD_G	PFH_G
CPU & Copro	BME•58•040S & BMEP58CPROS3	SIL3[1]	4.37E-06	9.91E-11
Analog input	BMXSAI0410	SIL3[2]	5.76E-05	1.31E-09
Digital input	BMXSDI1602	SIL3[2]	6.81E-05	1.56E-09
Digital output	BMXSDO0802	SIL3[1]	5.75E-05	1.31E-09
Digital relay output	BMXSRA0405	SIL2 [3]	5.84E-05	1.68E-09
		SIL3[4]	5.84E-05	1.34E-09
		SIL3[5]	–	1.35E-09
Power supply	BMXCPS4002S, BMXCPS4022S, and BMXCPS3522S	SIL3	–	–
1. 1 output @ 80° C 2. 1 input @ 80° C 3. 1 relay per output @ 80° C 4. 2 relays per output @ 80° C 5. 4 relays per output @ 80° C				

Product Type	Product Reference	SIL	PTI = 20 years	
			PFD_G	PFH_G
CPU & Copro	BME•58•040S & BMEP58CPROS3	SIL3[1]	8.74E-06	7.01E-10
Analog input	BMXSAI0410	SIL3[2]	1.15E-04	1.31E-09
Digital input	BMXSDI1602	SIL3[2]	1.36E-04	1.56E-09
Digital output	BMXSDO0802	SIL3[1]	1.15E-04	1.31E-09
Digital relay output	BMXSRA0405	SIL2 [3]	1.17E-04	1.68E-09
		SIL3[4]	1.17E-04	1.34E-09
		SIL3[5]	–	1.35E-09
Power supply	BMXCPS4002S, BMXCPS4022S, and BMXCPS3522S	SIL3	–	–
1. 1 output @ 80° C 2. 1 input @ 80° C 3. 1 relay per output @ 80° C 4. 2 relays per output @ 80° C 5. 4 relays per output @ 80° C				

Probabilities of Failure for SIL3 Applications

For SIL3 applications, the IEC 61508 defines the following probabilities of failure on demand (PFD) and probabilities of failure per hour (PFH) for each safety loop, depending on the mode of operation:

- PFD ≥ 10^{-4} to < 10^{-3} for low demand mode of operation
- PFH ≥ 10^{-8} to < 10^{-7} for high demand mode of operation

The M580 Safety PAC is certified for use in low and high demand systems.

Safety Integrity Level Sample Calculation

This sample calculation shows you how to determine:

- The risk contribution of the Schneider Electric safety modules to your safety application; and
- The remaining amount of risk that other devices in the safety loop (for example, sensors and actuators) can contribute to your safety application for a given safety integrity level and mode of operation.
  NOTE: When calculating the risk contribution of sensors and actuators to your safety application, contact the manufacturers of these devices and obtain the PFD/PFH values for the appropriate proof test interval.

The following Schneider Electric safety modules are included in this example:

- 1: BMEP584040S CPU
- 1: BMEP58CPROS3 Copro
- 1: BMXSAI0410 Analog input
- 1: BMXSDO0802 Digital output
- 1: BMXCPS4002S Power supply

The following calculation employs PFH_G values for a high demand mode of operation for a SIL3 safety loop with a PTI of 20 years. The maximum permissible PFH value for this safety application is 10^{-7} (or 1.0E-7):

Safety module		Contribution (Scientific Notation)	Remaining Contribution for Sensors & Actuators
CPU with Copro		7.01E-10	–
Analog input		1.31E-09	
Digital output		1.31E-09	
Power Supply		–	
Total	numeric	**2.72E-09**	**97.28E-09**
	% max	**2.72%**	**97.28%**
note 1: The relay output uses four relays to support one output.			

Values for M580 Safety Modules for Machinery

Schneider Electric offers the following safety modules certified for use in safety machinery applications according to ISO13849-1 standard. The table below list the safety modules and their values, category and level where applicable:

Product Type	Product Reference	Configuration	Category	Performance Level	MTTFd (years)	DCav
CPU with Copro	BME•58•040S & BMEP58CPROS3	NA	4	e	235	High (>99%)
Analog input	BMXSAI0410	using 1 channel	2	d	255	99.66%
		using 2 channels	4	e	255	99.66%
Digital input	BMXSDI1602	using 1 channel	2	d	231	99.69%
		using 2 channels	4	e	231	99.69%
Digital output	BMXSDO0802	NA	4	e	253	99.63%
Digital relay output	BMXSRA0405	using 1 channel	2	c	156	99.77%
		using 2 channels	4	e	156	99.77%

Safety Times Description

The M580 safety PAC has a minimum PAC cycle time of 10 ms, which is necessary for processing the signals from the I/O modules, executing the program logic, and setting the outputs. For calculating the maximum PAC reaction time, you need to know the maximum reaction time of the sensors and actuators that are being used. In addition, the maximum PAC reaction time depends on the process safety time (PST) *(see page 150)* required for your process.

Proof Test Interval

The proof text is a periodic test you need to perform to detect failures in a safety-related system so that, if necessary, the system can be restored to a like new condition or as close as practical to this condition. The time period between these tests is the proof test interval.

The proof test interval depends on the targeted safety integrity level, the sensors, actuators and the PAC application. The M580 safety system is suitable for use in a SIL3 application and a proof test interval of 20 years.

Section 8.3
M580 Safety System Performance and Timing Calculations

Introduction

This section shows you how to calculate PAC reaction time, system reaction time, and process safety time for your M580 safety system.

What Is in This Section?

This section contains the following topics:

Process Safety Time

Description of the Process Safety Time

The process safety time (PST) is an essential measure of a process executed by a safety loop. It is defined as the period between the occurrence of a failure in equipment under control (EUC) and the occurrence of a hazardous event if the safety function is not performed (i.e. if the safe state is not achieved).

NOTE: The process safety time is determined by your specific safety process. You need to verify that your safety-related system can perform its safety functions within the process safety time.

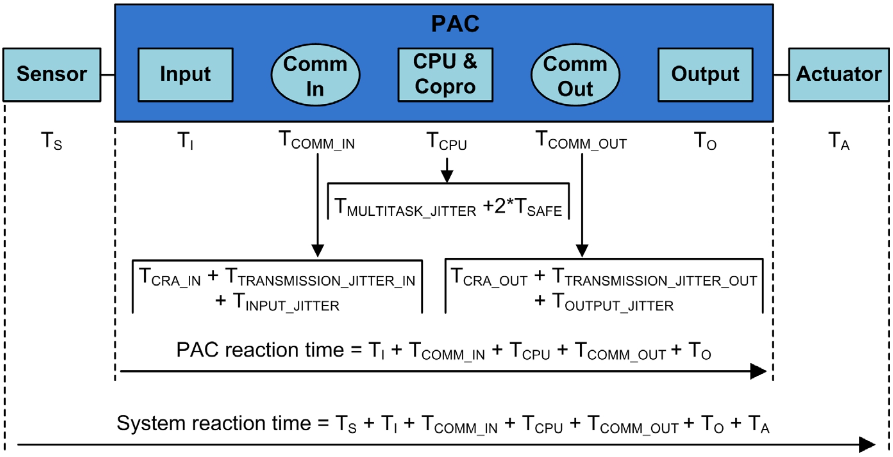
Description of the System Reaction Time

The system reaction time is the sum of the PAC reaction time, plus the reaction times for both the selected sensor (T_S) and the selected actuator (T_A).

NOTE: T_S and T_A are device specific.

For each safety loop, verify that the system reaction time is less than the process safety time.

System reaction time is illustrated below:



PAC
Sensor
Input
Comm In
CPU & Copro
Comm Out
Output
Actuator
T_S
T_I
T_{COMM_IN}
T_{CPU}
T_{COMM_OUT}
T_O
T_A
$T_{MULTITASK\_JITTER} + 2*T_{SAFE}$
$T_{CRA\_IN} + T_{TRANSMISSION\_JITTER\_IN} + T_{INPUT\_JITTER}$
$T_{CRA\_OUT} + T_{TRANSMISSION\_JITTER\_OUT} + T_{OUTPUT\_JITTER}$
PAC reaction time = $T_I + T_{COMM\_IN} + T_{CPU} + T_{COMM\_OUT} + T_O$
System reaction time = $T_S + T_I + T_{COMM\_IN} + T_{CPU} + T_{COMM\_OUT} + T_O + T_A$

The components of the system reaction time can include the following:

Component	Description	Estimated Worst Case Value
T_{S}	Reaction time required by the selected sensor to react to a process event.	Device specific.
T_{I}	Maximum time required by the input module to sample and confirm a sensor event. It includes: ● One input module sampling period. ● Multiple input module sampling periods for filtering.	6 ms
T_{COMM_IN}	Input communication delay. Its components are described in the topic *Application Response Time* in the *Modicon M580 Standalone System Planning Guide for Frequently Used Architectures*, and include the following (numbers refer to the ART calculation in the referenced topic): ● T_{CRA_IN}: CRA_Drop_Process (2) + CRA Input RPI (3) ● T_{JITTER_IN}: Network_In_Time (4) + Network_In_Jitter (5) + CPU_In_Jitter (6)	–
T_{CPU}	The CPU and coprocessor reaction time, which equals the sum of the delay caused by pending higher priority tasks (the FAST task) plus two SAFE task scan times – the first being a missed scan, the second being a successful scan: $T_{MULTITASK\_JITTER}$ +2* T_{SAFE}.	
$T_{MULTITASK\_JITTER}$	The maximum delay caused by execution of pending tasks with higher priority. In this case the FAST task. $T_{MULTITASK\_JITTER} = T_{FAST}$.	–
T_{SAFE}	The configured SAFE task period.	–
T_{FAST}	This value is included because the FAST task execution takes priority over the SAFE task. **NOTE:** To simplify the formula, it is assumed that no system task is in an overrun condition. Thus, this value equals the configured FAST task period, or 0 if the FAST task is not configured.	–
T_{COMM_OUT}	Output communication delay. Its components are described in the topic *Application Response Time* in the *Modicon M580 Standalone System Planning Guide for Frequently Used Architectures*, and include the following (numbers refer to the ART calculation in the referenced topic): ● T_{CRA_OUT}: CRA_Drop_Process (12) ● T_{JITTER_IN}: CPU_Out_Jitter (9) + Network_Out_Time (10) + Network_Out_Jitter (11)	–

Component	Description	Estimated Worst Case Value
T_O	Equal to the sum of the following times: ● Delay time between reading and applying the CPU output value (0...3 ms). ● Time required by the safety output module to modify the physical output, i.e. to propagate the change from X ram to the physical output (between 0...3 ms).	6 ms
T_A	Reaction time for the selected actuator.	Device specific.

Description of the PAC Reaction Time

For I/O placed in the local main rack (with the CPU) the PAC reaction time is the sum of the related reaction times for both the selected input module (T_I) and the selected output module (T_O), plus the CPU & Copro reaction time (T_{CPU}):

PAC reaction time (local) = $T_{CPU} + T_I + T_O$

If the I/O are located in a remote rack, the PAC reaction time also includes input communication delay (T_{COMM_IN}) and output communication delay (T_{COMM_OUT}) times:

PAC reaction time (remote) = $T_{CPU} + T_{COMM\_IN} + T_I + T_{COMM\_OUT} + T_O$

Description of the CPU & Copro Reaction Time

The CPU & Copro reaction time is directly impacted by both the SAFE task period and the FAST task period. Verify that safety logic will be executed within the SAFE task period.

Because a signal may appear just at the beginning of the execution cycle when the signals have already been processed, two SAFE task cycles may be necessary to react to the signal.

Because the FAST task takes priority over the SAFE task, you also need to consider the time to execute the FAST task when estimating jitter.

This leads to the following equation for the maximum (i.e. worst case) reaction time:

CPU & Copro reaction time = 2 x $T_{SAFE} + T_{FAST}$

NOTE: If you are using peer-to-peer safe communication *(see page 179)* to perform the safety function, the CPU reaction time estimation is different.

Description of the Time for Input Modules

The maximum times (worst case) for the safety digital input module and for the safety analog input module T_I are 6 ms.

Description of the Time for Output Modules

The maximum time T_O for the safety digital output module is estimated to be 6 ms.

A fallback safety timeout S_TO needs to be configured for both the digital output module *(see page 104)* and the digital relay output module *(see page 122)*. Depending on the configured SAFE task period (T_{SAFE}), the value for S_TO needs to be configured as follows:

- If (2.5 * T_{SAFE}) ≤ 40 ms, set S_TO to a minimum of 40 ms.
- If (2.5 * T_{SAFE}) > 40 ms, set S_TO to a minimum of (2.5 * T_{SAFE}) ms.

NOTICE

RISK OF UNINTENDED EQUIPMENT OPERATION

Set the fallback safety timeout (S_TO) for a safety output module to, at least, a value greater than the greater of 40 ms or (2.5 * T_{SAFE}), where T_{SAFE} equals the configured SAFE task period.

Failure to follow these instructions can result in equipment damage.

For Hot Standby applications, consider the impact on the fallback safety timeout (S_TO) parameter of additional time (T_{SWAP}) required by a swap *(see page 154)*, and of additional time T_{SWITCH} required by a switchover *(see page 155)*.

Computation of System Reaction Time

Knowing the required process safety time (PST) and the maximum reaction time of the sensors and actuators, you are able to calculate the maximum system reaction time (SRT) tolerable in your process.

The maximum (i.e. worst case) system reaction time can be computed as follows:

For systems with I/O in remote drops:

Max SRT = $T_S + T_I + 2 \times T_{CRA} + T_{RPI} + 2 \times T_{SAFE} + T_{FAST} + T_O + T_A$.

or

Max SRT = 16 ms + $T_S + 2.5 \times T_{SAFE} + T_{FAST} + T_A$.

For systems with local I/O:

Max SRT = $T_S + T_I + 2.5 \times T_{SAFE} + T_{FAST} + T_O + T_A$.

or

Max SRT = 15 ms + $T_S + 2.5 \times T_{SAFE} + T_{FAST} + T_A$.

NOTE: For Hot Standby PACs, for calculation of the maximum safety reaction time, the additional components to the above calculations have to be considered:

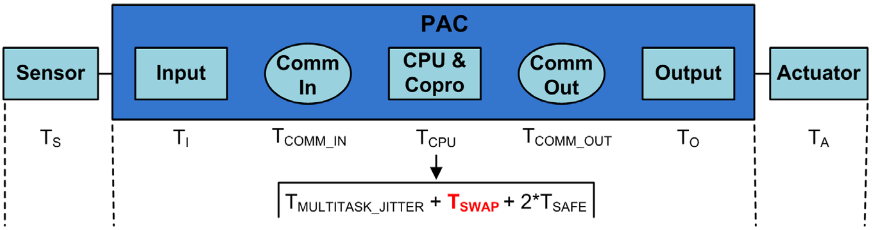
- While an unexpected event and a switchover occurs, maximum safety reaction time could increase by adding the component *(see page 155)* T_{SWITCH} to the above calculations.
- While the system operator performs a swap, maximum safety reaction time could increase with an additional component *(see page 154)* T_{SWAP} to the above calculations.

System Reaction Time During a Swap

A swap is the operator-initiated action on a Hot Standby system, which causes the primary and standby PACs to exchange roles. A swap consumes additional time, because during the swap no information may be lost and all system outputs need to be safely timed out.

The added swap time component is added to the T_{CPU} time following the normal T_{JITTER} component, as shown below:

The T_{SWAP} time component is added to the T_{CPU} time following the normal T_{JITTER} component. This sequence is displayed below. Except for the inclusion of the swap component, the system reaction time description is the same as described above *(see page 150)*:



The T_{SWAP} time component is the sum of the following:

$T_{ADDITIONAL\_JITTER} + T_{TRANSFER}$

The swap-specific components are described as follows:

Component	Description	Estimated Worst Case Value
$T_{ADDITIONAL\_JITTER}$	Jitter introduced by the multi-task system to restart the task on the new PAC. Hence, $T_{ADDITIONAL\_JITTER} = T_{SAFE}$.	–
$T_{TRANSFER}$	During the diagnostics of the MAST task, the PAC accepts the Swap command and begins to perform the transfer of all the latest data for each task.	Refer to the formula, below.

$T_{TRANSFER}$ can be calculated as follows:

$$K3 \text{ x } (MAST_{KB} + 2 \text{ x } SAFE_{KB} + FAST_{KB}) + K4 \text{ x } (MAST_{DFB} + 2 \text{ x } SAFE_{DFB} + FAST_{DFB}) / 1000$$

Where:

- $TASK_{KB}$ = Size of the data (in Kbytes) exchanged for the TASK between the primary PAC and standby PAC.
- $MAST_{DFB}$ = The number of DFBs declared in the TASK.
- K3 and K4 are constants with values determined by the specific CPU module used in the application, as follows:

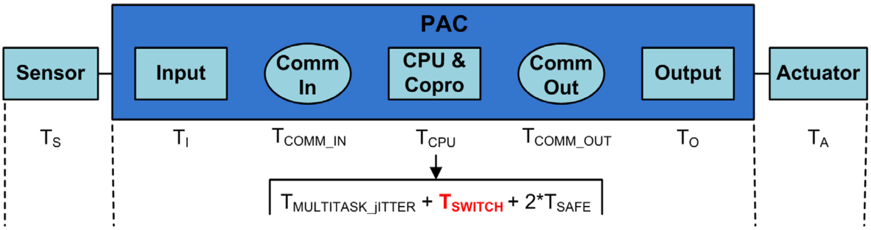
Coefficient	BMEH582040S	BMEH584040S or BMEH586040S
K3	46.4 µs/kB	14.8 µs/kB
K4	34.5 µs/DFB instance	11.0 µs/DFB instance

If the system operator wants to perform a swap without safety module outputs going into their fallback state, set the fallback safety timeout parameter of the safety output modules (S_TO) to, at least, a value greater than: $T_{MULTITASK\_JITTER} + T_{SWAP} + T_{SAFE}$.

System Reaction Time During a Switchover

A switchover occurs when the standby PAC in a Hot Standby system becomes the primary PAC in response to an unexpected event, for example, when hardware in the primary PAC suddenly becomes non-operational. The goal of the switchover is for the new primary PAC to seamlessly replace the old one, and begin operations at the point where the old primary PAC ceased to function. Nevertheless, the last cycle may be re-executed. The system target is to achieve the fastest possible recovery.

The T_{SWITCH} time component is added to the T_{CPU} time following the normal T_{JITTER} component. This sequence is displayed below. Except for the inclusion of the switchover component, the system reaction time description is the same as described above *(see page 150)*:



The T_{SWITCH} time component is the sum of the following:

$T_{DETECT} + T_{ADDITIONAL\_JITTER}$

The switchover-specific components are described as follows

Component	Description	Estimated Worst Case Value
T_{DETECT}	Time used by the standby PAC to detect and confirm the primary PAC has become non-operational.	15 ms
$T_{ADDITIONAL\_JITTER}$	Jitter introduced by the multi-task system to restart the task on the new PAC. Hence, $T_{ADDITIONAL\_JITTER} = T_{SAFE}$.	–

Unlike a swap, no additional time is needed to perform a data transfer.

To allow the system to respond to an unexpected event and perform a switchover without safety module outputs going into their fallback state, set the fallback safety timeout parameter of the safety output modules (S_TO) to, at least, a value greater than: $T_{JITTER} + T_{SWITCH} + T_{SAFE}$.

Configuring the Maximum CPU SAFE and FAST task Periods

The M580 safety PAC can perform only periodic execution for the SAFE and FAST tasks (cyclic execution is not supported for these tasks).

The SAFE task **Period** and the maximum allowed CPU **Watchdog** settings are configured in the **General** tab of the **Properties of SAFE** dialog. The safety digital output **Fallback Timeout** settings are configured in the **Configuration** tab for the output module *(see page 99)*.

Similarly, the FAST task **Period** and the maximum allowed CPU **Watchdog** settings are configured in the **General** tab of the **Properties of FAST** dialog.

NOTE:

- Permissible SAFE task period settings range is 10...255 ms, with a default value of 20 ms.
- Permissible FAST task period settings range is 1...255 ms, with a default value of 5 ms.
- Permissible watchdog settings range is 10...500 ms, with a default value of 250 ms.
- Permissible digital output fallback timeout settings range is 0...65535 ms, with a default value of 500 ms.

Verify that the watchdog setting is greater than the SAFE task period.

Check your CPU SAFE task period setting when commissioning your project. At this time, Control Expert Safety provides the real time values from the PAC.

You can find this information in Control Expert Safety in the **Task** tab using the menu entry **Tools → PLC Screen**.

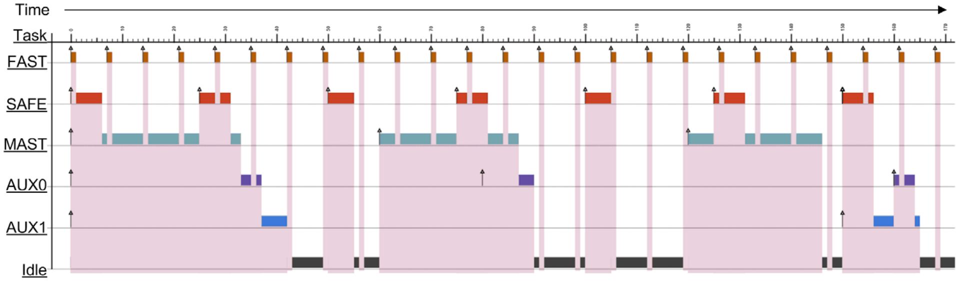
⚠ WARNING

RISK OF EXCEEDING THE PROCESS SAFETY TIME

Set the maximum CPU SAFE task period by taking into account your process safety time. Your CPU SAFE task period must be less than your project process safety time.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

The following drawing illustrates the execution of each task in a multi-task system, and depicts the preemption of CPU resources depending on the task priority:



FAST			SAFE			MAST			AUX0			AUX1			IDLE
Period 7 ms	Exe 1 ms	Bandwidth 14%	Period 25 ms	Exe 5 ms	Bandwidth 20%	Period 60 ms	Exe 18 ms	Bandwidth 30%	Period 80 ms	Exe 3 ms	Bandwidth 4%	Period 150 ms	Exe 5 ms	Bandwidth 3%	Free Bandwidth 29%

NOTE: When the MAST task is not in cyclic mode and for optimal CPU performance, Schneider Electric recommends that the 20% of CPU bandwidth remain idle.

Calculating the Impact of Task Execution Periods on CPU Bandwidth

Each configured task consumes a portion of CPU processing time, or bandwidth. The estimated percentage of CPU bandwidth consumed by a task is the result (or quotient) of the estimated execution time required by a task (E_{TASK}) divided by the configured execution period for that task (T_{TASK}), and can be presented as follows:

Task bandwidth = E_{TASK} / T_{TASK}.

Thus, the total percentage of CPU bandwidth consumed by an application is the sum of consumed CPU bandwidth percentages for all tasks.

NOTE: When the MAST task is not in cyclic mode and for optimal CPU performance, Schneider Electric recommends that the total percentage of CPU bandwidth consumed by an application is not exceeding 80%.

The following table presents two applications, and indicates the impact of high priority tasks (FAST and SAFE) on total CPU bandwidth usage:

#	FAST			SAFE			MAST			AUX0			Total
	Per	Exe	BW%	Per	Exe	BW%	Per	Exe	BW%	Per	Exe	BW%	
1	5 ms	1 ms	20%	20 ms	5 ms	25%	50 ms	18 ms	35%	200 ms	30 ms	15%	96%
2	7 ms	1 ms	14%	25 ms	5 ms	20%	60 ms	18 ms	30%	200 ms	30 ms	15%	79%
Per = Task period (T_{TASK}) Exe = Execution time required for the task (E_{TASK}) BW% = Task bandwidth.													

Impact of CIP Safety Communications on Safety System Reaction Time

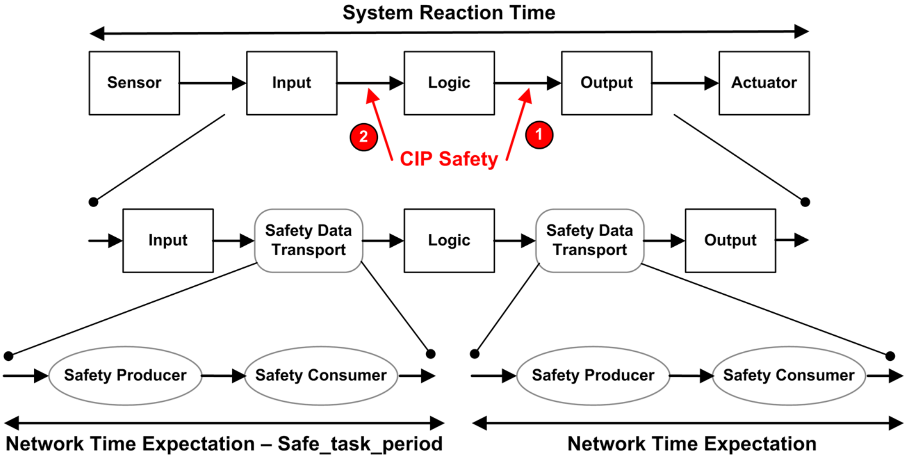
Introduction

Time consumed by CIP safety communication, called the *network time expectation*, is added to and becomes part of the *system reaction time (see page 150)*. The network time expectation represents the maximum, or worst case, time period starting when the data is captured by the safety data producer, and ending when the consuming application recognizes a safety state. This also includes errors during production and consumption.

If the CIP Safety communication is between an input and the logic, replace the term variable TCOMM_IN in the process safety time calculation *(see page 150)* with *Network Time Expectation - Safe_task_period*. If the CIP Safety communication is between the logic and an output, replace the variable TCOMM_OUT in the process safety time calculation with *Network Time Expectation*.

Default measures of the Network Time Expectation vary, depending on the role of the M580 safety CPU as producer or consumer.

The elements of network time expectation, and its placement within the context of system reaction time, is set forth in the following diagram:



1 CIP Safety CPU as producer
2 CIP Safety CPU as consumer

Calculating Network Time Expectation

The Network Time Expectation can be calculated using the following formula:

Network Time Expectation = Network_Time_Expectation_Multiplier * 128 μSec >
(EPI * Timeout_Multiplier + Safety_Message_Time(max) + Time_Coord_Message_Time(max) +
Connection_Correction_Constant*128 μSec)

Where:

- **Safety_Message_Time(max)** is the actual time from the data being captured by the safety data producer until the time that the safety data is passed to the consuming application for use.
- **Time_Coord_Message_Time(max)** is the maximum time it could take for the time coordination information to be sent from the consumer to the producer.
- **Timeout_Multiplier** is a parameter used in CIP safety protocol processing, which determines the number of messages that may be lost before declaring a connection error. A Timeout_Multiplier of 1 indicates that no messages may be lost.
- **Connection_Correction_Constant** is a value in 128 μSec increments that is subtracted from the time stamp to represent the worst case error due to time drift, the asynchronous nature of the producer and consumer clocks, and the minimum time for the Time Coordination Message to traverse from the consumer to the producer.
- **EPI** is the expected packet interval, and is based on the configured SAFE task period.
- **Network_Time_Expectation_Multiplier** and **Timeout_Multiplier** are CIP communication parameters configured for the SafetyOpen Type 2 connection frame *(see page 361)*.

Default Network Time Expectation Values

The default calculation for the network time expectation value depends on the role of the CIP Safety CPU as consumer (case 2 in the preceding diagram) or producer (case 1).

CPU as consumer (case 2):

- Timeout_Multiplier = 2
- EPI = SAFE task period / 2
- Safety_Message_Time(max) = Safe task period + 20 ms (worst case)
- Time_Coord_Message_Time(max) = Safe task period + 20 ms (worst case)
- Connection_Correction_Constant = 0 ms

Network Time Expectation = 1.5 * minimum_Network_Time_Expectation =
1.5 * (3 * Safe task period + 40 ms) = 4.5 * Safe task period + 60 ms

CPU as producer (case 1):

- Timeout_Multiplier = 2
- EPI = SAFE task period
- Safety_Message_Time(max) = Safe task period + 20 ms (worst case)
- Time_Coord_Message_Time(max) = Safe task period + 20 ms (worst case)
- Connection_Correction_Constant = 0 ms

Network Time Expectation = 1.5 * minimum_Network_Time_Expectation =
1.5 * (4 * Safe task period + 40 ms) = 6 * Safe task period + 60 ms

Chapter 9
Safety Library

Safety Library

Introducing the Safety Library

When you install Control Expert Safety, a safety library of elementary functions (EFs), elementary function blocks (EFBs), and derived function blocks, (DFBs) are automatically included. These EFs, EFBs, and DFBs are identified by the prefix “S_” and are reserved for use in code sections managed by the SAFE task.

NOTE: Also installed is an additional collection of EFs, EFBs and DFBs. This is the same collection of data objects used by non-safety M580 PACs. These EFs, EFBs, and DFBs can be used only in code sections managed by process namespace tasks (MAST, FAST, AUX0, and AUX1).

For a description of the blocks included in the M580 safety library, refer to the *Control Expert Safety Block Library* document.

Certified Safety Functions and Function Blocks

⚠ **WARNING**

UNEXPECTED APPLICATION BEHAVIOR

- Do not use V1.00 of the S_GUARD_LOCKING derived function block in your application.
- In Unity Pro 13.0 XLS or later, update the S_GUARD_LOCKING function block in your application with V1.01 or later, and rebuild the application.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: Unity Pro is the former name of Control Expert for version 13.1 or earlier.

These are the subset of EFs and Functions Blocks, which can be used inside safety logic. These are provided in the Safety Library:

Family	Group or Name	Type	Description
Logic	S_AND_*, S_OR_*, S_XOR_*, S_NOT_*, S_SHL_*, S_SHR_*, S_ROR_*, S_ROL_*	EF	Type specific, e.g. S_AND with 2 to 32 inputs (inline code)
Logic	S_RS, S_SR, S_F_TRIG, S_R_TRIG	EFB	–
Mathematics	S_ADD_*, S_MUL_*, S_SUB_*, S_DIV_*, S_ABS_*, S_SIGN_*, S_NEG_*, S_MOVE, S_SQRT_REAL	EF	Type specific detected error handling (e.g. overflow) to be considered (inline code)

Family	Group or Name	Type	Description
Comparison	S_GT_*, S_GE_*, S_LT_*, S_LE_*, S_NE_*, S_EQ_*	EF	Type specific (inline code)
Statistical	S_LIMIT_*, S_MAX_*, S_MIN_*, S_MUX_*, S_SEL	EF	Type specific (inline code)
Type To Type	S_BIT_TO*, S_BOOL_TO_*, S_BYTE_TO_*, S_DINT_TO_*, S_DWORD_TO_*, S_INT_TO_*, S_REAL_TO_*, S_TIME_TO_*, S_UDINT_TO_*, S_UINT_TO_*, S_WORD_TO_*	EF	Type specific (inline code)
Timers & Counters	S_CTU_*, S_CTD_*, S_CTUD_*	EFB	Type specific
Timers & Counters	S_TON, S_TOF, S_TP	EFB	–
Peer to peer	S_RD_ETH_MX, S_WR_ETH_MX, S_RD_ETH_MX2, S_WR_ETH_MX2	DFB	Functions to perform a Safety peer to peer communication
Actuator Connection	S_EDM, S_ENABLE_SWITCH, S_ESPE, S_OUTCONTROL, S_GUARD_LOCKING, S_GUARD_MONITORING, S_MODE_SELECTOR	DFB	Machine Safety Function Blocks linked to actuators
Sensor Connection	S_EQUIVALENT, S_ANTIVALENT, S_EMERGENCYSTOP, S_TWO_HAND_CONTROL_TYPE_II, S_TWO_HAND_CONTROL_TYPE_III, S_MUTING_SEQ, S_MUTING_PAR, S_AI_COMP	DFB	Machine Safety Function Blocks linked to sensors
System	S_SYST_STAT_MX, S_SYST_TIME_MX, S_SYST_CLOCK_MX, S_SYST_RESET_TASK_BIT_MX, S_SYST_READ_TASK_BIT_MX	EFB	System function blocks

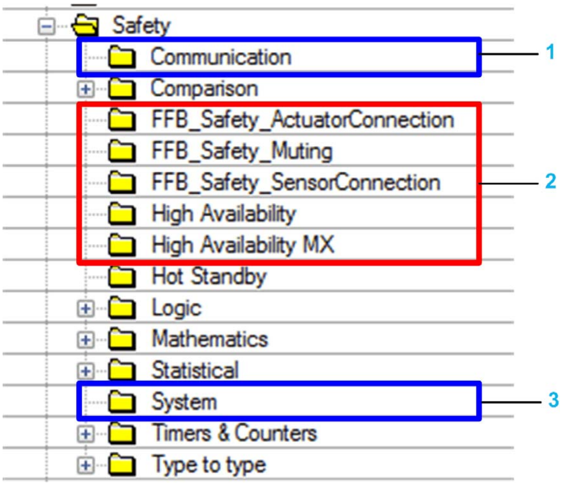
Non-Certified Safety Functions and Function Blocks

These are the subset of Derived Functions Blocks (DFBs), which can be used inside safety logic. These function blocks are not certified. Their purpose is to provide you sample safety function blocks that can be easily reused and adapted. You can copy and paste these function blocks into your application and change them to meet the requirements of your application.

Family	Group or Name	Type	Description
High Availability MX	S_DIHA, S_AIHA	DFB	Function for high availability SIL2 or SIL3 digital input modules (inline code)
Sensor Connection	AI_COMP	DFB	Machine Safety Function Blocks linked to sensors

Viewing the Safety Library in Control Expert

You can access the safety library only from the SAFE task. When you open the safety library in the **FBD-Editor**, the safety library presents groups of EFs, EFBs, and DFBs. Some of these groups include safety versions of functions and blocks found in non-safety tasks. Others groups, noted below, contain functions and blocks unique to the SAFE task:



1 Blocks for reading and writing safety data values.
2 Blocks for performing safety-specific tasks.
3 Blocks for reading and writing safety system values.

For an example of how some of the safety blocks are implemented, refer to the PAC-to-PAC communication configuration example *(see page 182)*, which includes S_RD_ETH_MX and S_WR_ETH_MX.

Also refer to the *EcoStruxure™ Control Expert Safety Block Library* for a description of each available safety function and block.

Chapter 10
Data Separation in an M580 Safety System

Introduction

This chapter presents the division of data in an M580 safety system.

What Is in This Chapter?

This chapter contains the following topics:

Data Separation in an M580 Safety Project

Data Separation and Scope

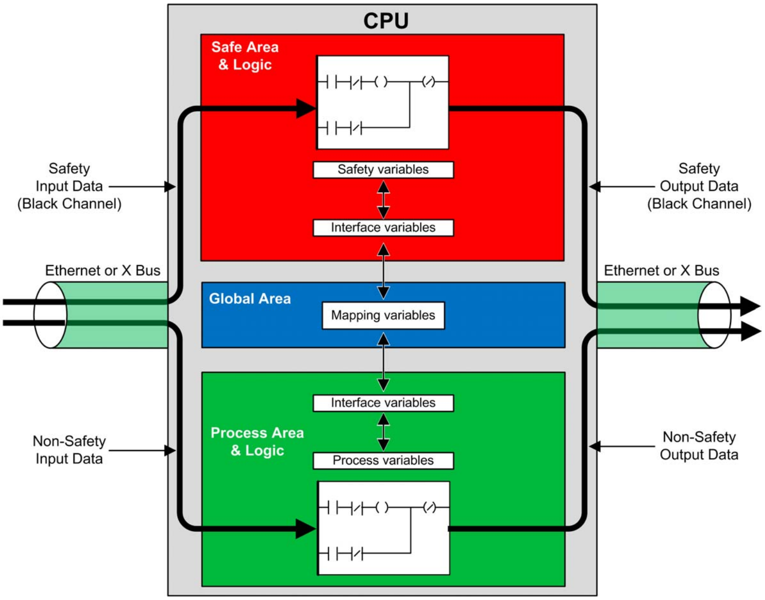
An M580 safety project includes both a safety program and a process (non-safety) program. Control Expert isolates the logic and data used by the safety program from the logic and data used by the process program. Control Expert accomplishes this by placing each part of the project into its own namespace (also called an area), either *safe* or *process*.

As a result of this design, the scope of a safety variable is restricted to the safe area, and the scope of a process variable is restricted to the process area. This becomes apparent when you add program logic to your application:

- When you configure an EF or EFB in the SAFE task, only variables created in the safe area are visible. Variables created in the process area are not visible.
- When you configure an EF or EFB in a non-safe (MAST, FAST, AUX0 or AUX1) task, only variables created in the process area are visible. Variables created in the safe area are not visible.

To permit communication between the safe area and the process area, Control Expert also provides a *global* area. The global area serves as a pass-through for data transmissions between the safe area and the process area. This is accomplished by declaring interface variables in both the safety and process areas, then linking these interface variables to mapping variables declared in the global area.

This data separation in the M580 safety CPU and coprocessor is graphically described below:



Safe, Process and Global Area Properties

The three data areas of an M580 safety project present the following properties:

Area	Supported Variable Types	Scope	External Access
Global	Unlocated variables only. **NOTE:** Located variables cannot be used to map to a safety or process interface variable.	Can access: ● Safety variables, via namespace addressing. ● Process variables, via namespace addressing. ● Other global variables.	Variables from all three areas can be accessed by HMI, SCADA, or FactoryCast applications. (See Note, below.)
Safe	Unlocated variables only.	Can access only other safety variables.	
Process	Both: ● Located variables ● Unlocated variables	Can access only other process variables.	

When an external viewer seeks to read a process variable, the addressing format depends on whether the **Usage of Process Namespace** setting has been selected in the **Scope → common** area of the **Tools → Project Settings...** window. If the **Usage of Process Namespace** setting is

- Selected: the operator screen can read process area variables only by using the format “PROCESS.<variable name>”.
- De-selected: the operator screen can read process area variables only by using the format “<variable name>” without the PROCESS prefix. In this case, verify that each process variable name is unique, and is not the same as any global variable name.

NOTE: If the **Usage of Process Namespace** setting is de-selected, verify that each process variable name is unique, and is not the same as any global variable name. If a variable name is common to both the global and process areas, an error will be detected by Control Expert when you build the project.

How to Transfer Data Between Namespace Areas

Introduction

The M580 safety PAC includes three different data editors:

- a **Safety Data Editor** to manage data used in the safe namespace.
- a **Process Data Editor** to manage data used in the process namespace.
- a **Global Data Editor** to manage global variables and data types used throughout the application.

Both the **Safety Data Editor** and the **Process Data Editor** include an **Interface** tab. Use the **Interface** tab to create unlocated variables in that process namespace. The **Interface** tab presents two groups of unlocated variables:

- <inputs>: A variable created in this group can be linked to, and receive data from, a globally scoped pass-through variable in the **Global Data Editor**.
- <outputs>: A variable in this group can be linked to, and send data to, a globally scoped pass-through variable in the **Global Data Editor**.

NOTE: A variable created in either **Interface** tab needs to be all of the following:

- An EDT or DDT category variable.
- Of the same data type as the variable to which it is linked.
- Not a variable linked to an extracted bit of a located variable (for example, not %MW10.1).

Unlocated variables created in the **Interface** tab groups of the **Safety Data Editor** and **Process Data Editor** can be linked as follows:

A process variable in this group in the Process Data Editor...	Can be linked to a safety variable in this group in the Safety Data Editor...
<inputs>	<outputs>
<outputs>	<inputs>

Using these three data editors, you can configure the transfer of data between the safe namespace and the process namespace.

Transferring Data Between Namespaces

The process for passing data from the safe to the process namespace, and from the process to the safe namespace are the mirror image of each other. The following example shows you how to pass data from the process to the safe area:

Step	Action
1	Open the **Process Data Editor**, click on the program **Interface** tab and then create a new variable in the **<outputs>** part of the data editor.
2	Open the **Safety Data Editor**, click on the program **Interface** tab and then create a new variable with the same type of the one created in step1 in the **<inputs>** part of the data editor. Then, double-click the **Effective Parameter** field. The **Data Scope Editor: Variable Selection** dialog opens.
3	In the drop-down menu at the top-right of the dialog, select the target namespace **PROCESS**. The variables in the selected namespace PROCESS in the <outputs> part are displayed.
4	Select the process variable created in step1 to be linked to the safe variable you created in step2, then click **OK**. The selected target variable appears in the **Effective Parameter** field.
5	**Save** your edits.

After you compile, download and run the edited application program, the value is transferred as follows:

- The data from the **Interface** tab created in the **<outputs>** are published at the end of the corresponding task execution.
- The data from **Interface** tab created in the **<inputs>** are subscribed at the beginning of the corresponding task execution.

Chapter 11
M580 Safety System Communications

Introduction

This chapter describes communications within the M580 safety system.

What Is in This Chapter?

This chapter contains the following sections:

Section 11.1
Time Synchronization

Introduction

For PAC with CPU firmware 3.10 or earlier: The NTP service configuration is required to allow a safe communication. Both safe senders and receivers need to be time synchronized using NTP services.

For PAC With CPU firmware 3.20 or later: The safe time synchronization is based on an internal and "monotonic" time clock. The safe communication does not need an NTP time synchronization:

- The Safety CPU is sharing its safe time with all its local and remote IOs.
- The remote IO head communication module BM•CRA31210 need a firmware 2.60 or later.
- For a peer to peer communication, CPU are sharing their safety time.

What Is in This Section?

This section contains the following topics:

Configuring the Time Synchronization with CPU firmware 3.10 or earlier

Introduction

If you are installing safety I/O modules in an RIO drop, the current time needs to be configured for the PAC. This can be accomplished in three designs with CPU firmware 3.10 or earlier:

1. **Remote NTP server design with CPU as NTP client**: Configure a device in the Control Network as an NTP server, then configure the safety CPU as the NTP client.
2. **Local NTP server design**: Configure the safety CPU as the NTP server for devices on the Ethernet RIO network.
3. **Remote NTP server design with eNOC or eNOP**: Configure a device in the control network as an NTP server, then configure a module - either BMENOP0300 or BMENOC0301/11 communications modules - in the local main rack and enable the optional feature **CPU Time Update → Update CPU time with this module** in the corresponding DTM. If RIO drop with safety devices are configured, configure the safety CPU as an NTP server as described in case 2 above.

In either design, you also need to:

- Enable the NTP service.
- Set the NTP polling period to 20 s.

If the safety CPU is not configured as either an NTP server or an NTP client, as described above, the time settings of the remote safety I/O modules and CPU will not be synchronized and black channel communication will not operate properly. Inputs and outputs of safety I/O modules in RIO drops will enter the safe (de-energized) or fallback state.

⚠ CAUTION

RISK OF UNINTENDED OPERATION

If you are placing safety I/O modules in an RIO drop, the current time needs to be configured for the PAC with firmware 3.10 or earlier. Enable the NTP service for your M580 system and configure the safety CPU as an NTP server or an NTP client.

Failure to follow these instructions can result in injury or equipment damage.

Two NTP sources are recommended. These can be configured as redundant servers, with one configured as the primary time server and the other as the standby. In this design the standby takes over if the primary server becomes non-operational, or if communications to the primary server is lost. It is recommended that both servers be time synchronized. A time adjustment equal to or greater than 2s in any 20s update causes the CPU and the safety I/O modules to enter a fault state.

Changing the NTP Time Setting During Operations

⚠ CAUTION
RISK OF SAFETY SYSTEM SHUTDOWN
Using Control Expert V13 or V13.1 or using CPU firmware 2.70 or earlier, do not change the time setting in the NTP server or the CPU.
Changing the time during operations can cause a loss of communication and a safety system shutdown.
Failure to follow these instructions can result in injury or equipment damage.

Changing the time during operation can cause a time de-synchronization with the reference clock. It can also trigger a loss of the safety communication causing the I/O to enter their fallback or safe state. Monitor your system for the occurrence of de-synchronization, and if it occurs restore synchronization to avoid communication loss. If such a de-synchronization occurs, use the following procedure *(see page 175)* to re-synchronize the system.

If you are using Control Expert V14 or later and using CPU firmware 2.80, 2.90, or 3.10: It is possible to change the time setting in the NTP server or the CPU during operation without a negative impact. Perform this operation by following the procedure set forth below immediately after a time modification.

Refer to the topic *NTP Tab* in the *Modicon M580 Hardware Reference Manual* for information on how to configure the NTP service for an M580 CPU.

Procedure for Synchronizing the NTP Time Settings

When power is cycled to the CPU, or the CPU is reset, and the CPU initially receives a time setting from an external NTP server, use the following procedure to synchronize CPU time.

⚠ CAUTION

RISK OF INOPERABLE EQUIPMENT

When using the **Update CPU time with this module** optional feature on a BMENOP0300 or BMENOC0301/11 module to update the PAC time, after the time from the external NTP server becomes operational (when %SW152 changes from 0 to 1), synchronize the Safe time with the external NTP server by using %SW128. Follow the procedure for synchronizing the NTP time set forth below.

Failure to follow these instructions can result in injury or equipment damage.

The following procedure is valid with the SAFE task in RUN state, using Control Expert V14.0 or later and CPU firmware 2.80, 2.90, or 3.10:

Step	Action
1	Check that CPU or external NTP server time is valid, healthy and stable.
2	If the configuration includes one or more eRIO drops, after the NTP service is operating again or after the time modification (which led to the de-synchronization), wait for 2 NTP polling periods to allow the new reference time value to be sent to all CRA modules.
3	Synchronize the system time on the reference clock using the %SW128 system word: ● Set %SW128 to 16#1AE5 for at least 500 ms. ● Then set %SW128 to #E51A for at least 500 ms.
4	Check that the time is synchronized, by verifying that the parameter values for CPU_NTP_SYNC and M_NTP_SYNC in safe IO DDDT are true (1)

If this sequence of synchronization is not properly executed, execute it again.

NOTICE

RISK OF SYSTEM SAFETY SHUTDOWN

- If you use Control Expert V14.0 or later and CPU firmware 2.80 or later to perform a PAC time modification, you need to follow that modification by performing the synchronization procedure previously described.
- If you do not perform a synchronization procedure, the safety I/O can enter their safe or fallback state after the clock drifts for about a communication delay timeout.

Failure to follow these instructions can result in equipment damage.

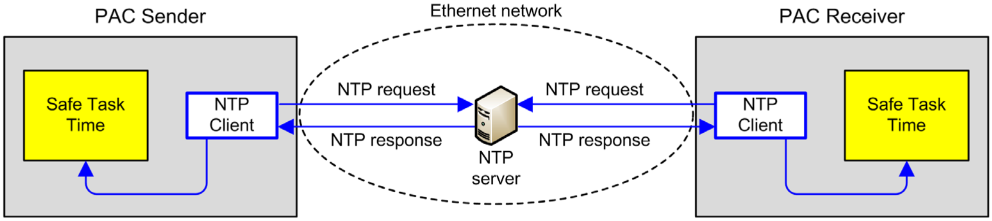
During the step 3 time synchronization operations, some diagnostics of the safe communication are disabled for a duration of 500 ms. Schneider Electric recommends a maximum of one time modification and synchronization per day.

NTP Service for Peer-to-Peer Communication

The safe Ethernet PAC-to-PAC communication requires the synchronization of the time base of both the sender and receiver PACs.

NOTE: Schneider Electric recommends that you configure in each PAC – either the safety CPU, a BMENOP0300 communications module, or a BMENOC0301/11 communications module – an NTP client, and configure another network device as NTP server.

The following figure describes the sender and receiver PACs time base synchronization principle:



In Control Expert, configure the NTP service parameters for each client as follows:

- Select **NTP Client**.
- Set the **Primary NTP Server IP address** to the IP address setting for the remote NTP server.
- Schneider Electric recommends a **Polling period** value set to 20 seconds.

NTP Server Time Consistency and System Bits

NTP server time consistency:

- If the NTP server time is consistent with the internal PAC time displayed by the EF `S_SYST_CLOCK` with less than 2 seconds difference, then the time value in the EF `S_SYST_CLOCK` is updated with the last NTP server time received filtered with a slope of 1ms/s.
- If the received NTP server time differs from the internal PAC time displayed by the EF `S_SYST_CLOCK` by more than 2 seconds, then:
  - the last received NTP server time is ignored by the PAC,
  - the time value displayed by the EF `S_SYST_CLOCK` is refreshed internally,
  - the `status` parameter of `S_SYST_CLOCK` is set to 0, and
  - the output parameter `SYNCHRO_NTP` from the `S_RD_ETH_MX` and the `S_WR_ETH_MX` DFBs is set to 0 to indicate this condition.

  In this case, you can reset the internal PAC time by taking one of the following actions:
  - reinitialize the application by a cold start
  - download the application
  - restart the PAC
  - follow the steps for changing NTP time settings *(see page 175)*.

  NOTE: If the NTP synchronization is lost on one of the two PACs (`SYNCHRO_NTP` parameter set to 0), both sender and receiver PACs time base can be de-synchronized. In this case, the safe peer-to-peer communication may cease to be operational (`S_RD_ETH_MX` DFB `health` output parameter is set to 0).

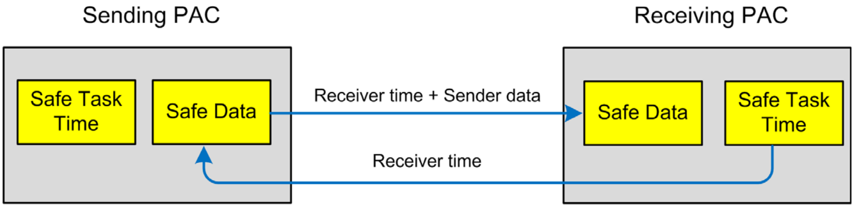
Time Synchronization for CPU firmware 3.20 or later

Time synchronization for Peer-to-Peer Communication

NOTE: With CPU firmware 3.20 or later the NTP service is not used for time synchronization.

The safe Ethernet PAC-to-PAC communication requires both the sender and receiver PACs to share a common safe time.

The following figure describes the sender and receiver PACs time sharing principle:



In Control Expert, configure:

- a communication for Sender to Receiver for Data transmission
- a communication for Receiver to Sender for Safe Time transmission

Time Consistency

An internal safety time (NTP independent) is distributed by the CPU to its local and remote safety IO modules.

Section 11.2
Peer to Peer Communications

Introduction

This section describes peer-to-peer communications between M580 safety PACs.

What Is in This Section?

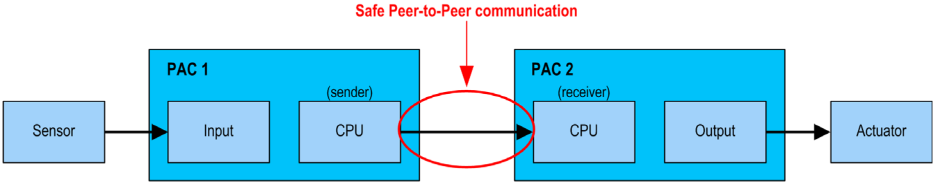
This section contains the following topics:

Peer-to-Peer Communication

Introduction

You can configure two M580 safety PACs to perform peer-to-peer safe communications over Ethernet. The configuration is based on Modbus TCP scanner communication, embedded in a black channel.

The safety peer-to-peer communication functional overview is as follows:



The communication is performed by two elementary function blocks from the M580 safety block library, that manage the safety loop at a SIL3 level. The protocol detects transmission errors – including omissions, insertions, disordered sequence, delays, inaccurate addressing, and masquerade bits – and manages retransmissions.

This safe peer-to-peer communication is possible only between:

- two M580 safety PACs with both CPU firmware 3.10 or earlier,
- two M580 safety PACs with both CPU firmware 3.20 or later.

NOTE: The safe peer-to-peer communication is also possible between a Modicon Quantum Safety PLC and a M580 Safety PLC with CPU firmware 3.10 or earlier.

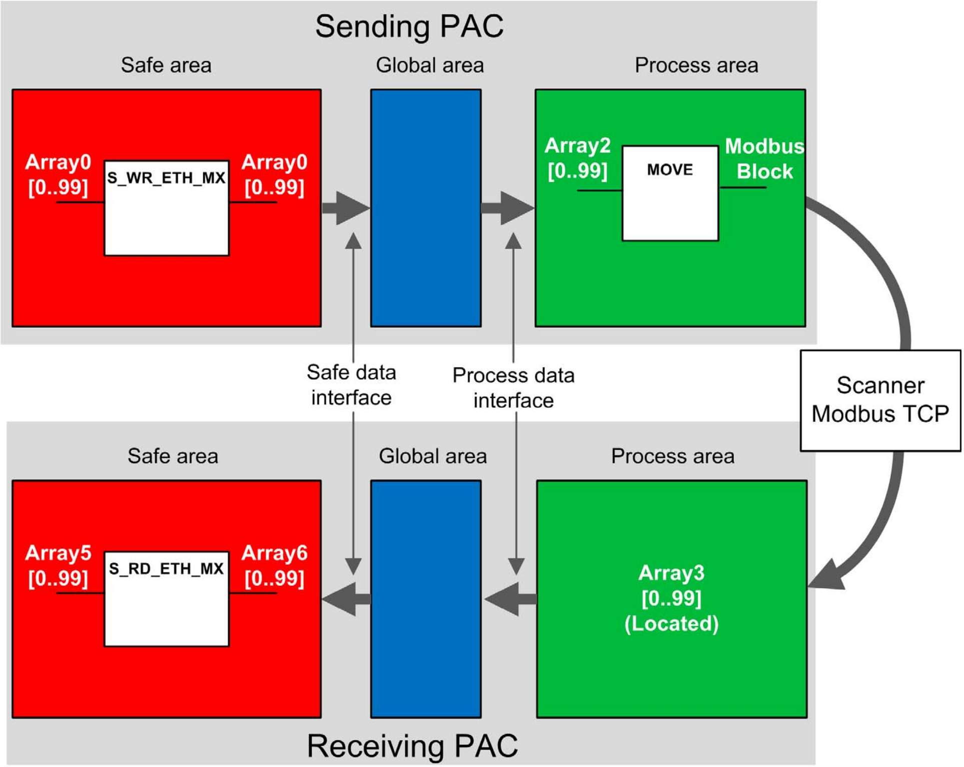
Peer-to-Peer Architecture using CPU Firmware 3.10 or Earlier

Architecture Design

Using CPU firmware 3.10 or earlier, the solution architecture is based on:

- NTP service for time base synchronization.
- Execution of 2 DFBs (`S_WR_ETH_MX` and `MOVE` in the sender PAC and 1 DFB (`S_RD_ETH_MX`) in the receiver PAC).
- Scanning via Modbus TCP, for data transportation.

The following figure shows the overview of the process required to perform the safe peer-to-peer communication:

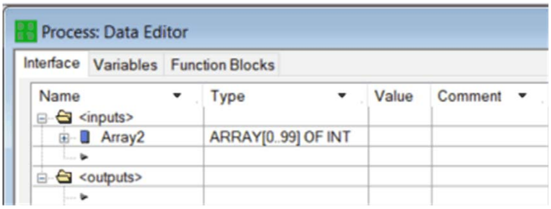
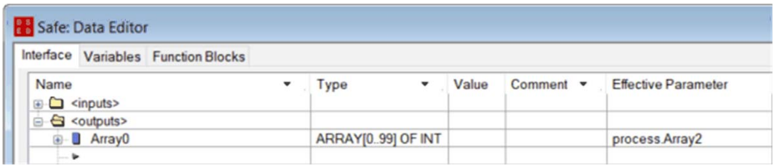
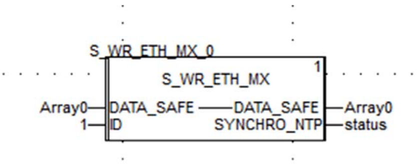


In the figure above, Control Expert automatically creates – and hides from external view – Array 1 and Array 4 in the Global areas of the peer PACs. From a user standpoint, the links are made from Array 0 to Array 2, and from Array 3 to Array 5.

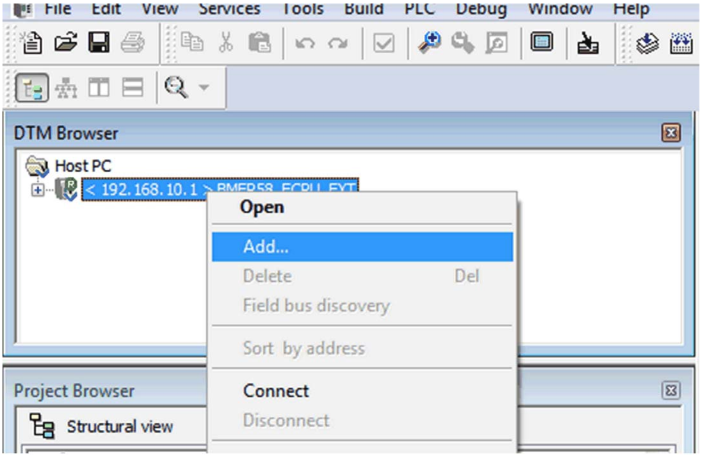
NOTE: On the Ethernet network, you are allowed to mix safety related data and non-safety related data without impact on the integrity level of the safety related data. There is no restriction on the Ethernet network when using the safe peer-to-peer communication.

Peer-to-Peer Data Transfer Configuration Details

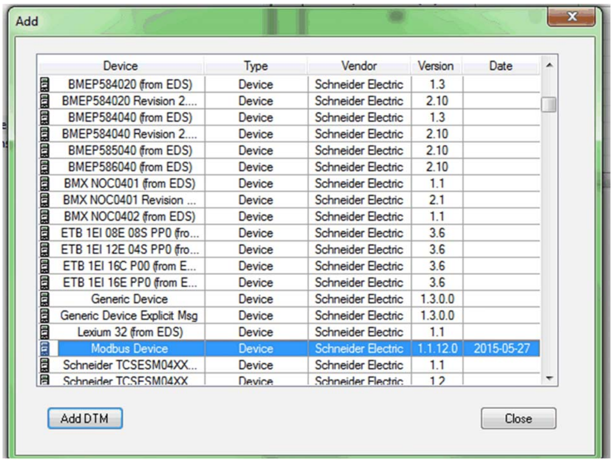
The following example shows you how to configure a peer-to-peer transfer of data between two safety PACs with CPU firmware 3.10 or earlier and Control Expert 14.1 or earlier:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>On the sending PAC, use the <b>Process Data Editor</b> to create an array of 100 integers as an input in the <b>Interface</b> area. In this example, the array name is Array2:

Process: Data Editor
Interface | Variables | Function Blocks
<table>
<tr><th>Name</th><th>Type</th><th>Value</th><th>Comment</th></tr>
<tr><td>&lt;inputs&gt;</td><td></td><td></td><td></td></tr>
<tr><td>Array2</td><td>ARRAY[0..99] OF INT</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td>&lt;outputs&gt;</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>2</td><td>On the sending PAC, create another array of 100 integers as an output in the <b>Interface</b> tab of the <b>Safety Data Editor</b> and link it to the input process area array created in step1, above, in the <b>Effective Parameter</b> column. In this example, the array name is Array0:

Safe: Data Editor
Interface | Variables | Function Blocks
<table>
<tr><th>Name</th><th>Type</th><th>Value</th><th>Comment</th><th>Effective Parameter</th></tr>
<tr><td>&lt;inputs&gt;</td><td></td><td></td><td></td><td></td></tr>
<tr><td>&lt;outputs&gt;</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Array0</td><td>ARRAY[0..99] OF INT</td><td></td><td></td><td>process.Array2</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
</table>
<b>NOTE:</b> The integer variables from index 0 to 90 of the array contain the safety variable values to exchange with the receiving PAC. The remaining area is reserved for auto-generated diagnostic data, including a CRC and time-stamp. This diagnostic data is used by the receiving PAC to determine if the transferred data is safe.</td></tr>
<tr><td>3</td><td>On the sending PAC, configure the DFB <code>S_WR_ETH_MX</code> in a section of the SAFE task. Link the DFB to Array0:

</td></tr>
</table>

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>4</td><td>In the <b>DTM Browser</b> of the sending PAC, select the CPU (in this example) or an NOC communications module (if any) then click <b>Add...</b> to create a Modbus scanner that can send data via Modbus TCP from the sending PAC to the receiving PAC:

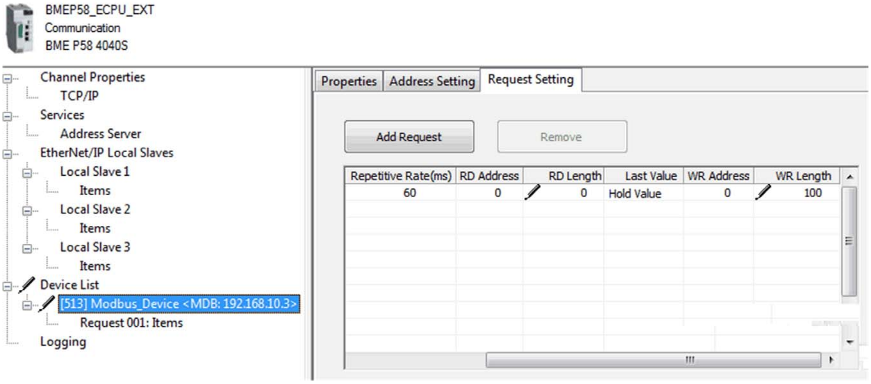


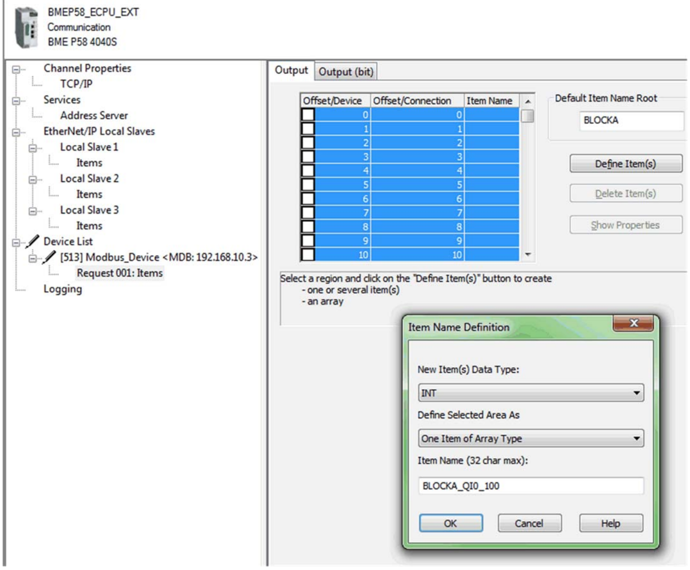
</td></tr>
<tr><td>5</td><td>Select <b>Modbus Device</b> and click <b>Add DTM</b> to add the Modbus scanner:



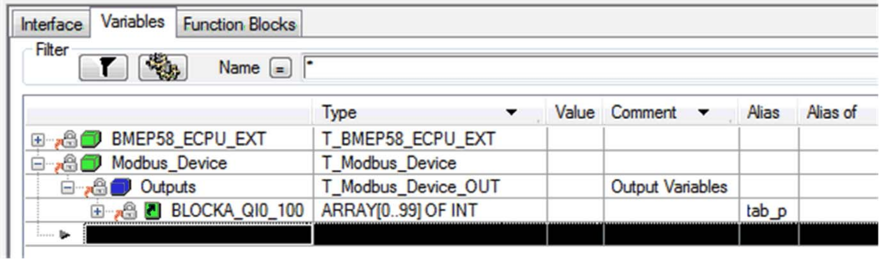
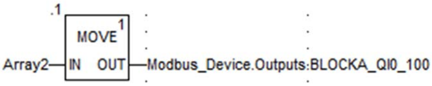
<table>
<tr><th>Device</th><th>Type</th><th>Vendor</th><th>Version</th><th>Date</th></tr>
<tr><td>BMEP584020 (from EDS)</td><td>Device</td><td>Schneider Electric</td><td>1.3</td><td></td></tr>
<tr><td>BMEP584020 Revision 2....</td><td>Device</td><td>Schneider Electric</td><td>2.10</td><td></td></tr>
<tr><td>BMEP584040 (from EDS)</td><td>Device</td><td>Schneider Electric</td><td>1.3</td><td></td></tr>
<tr><td>BMEP584040 Revision 2....</td><td>Device</td><td>Schneider Electric</td><td>2.10</td><td></td></tr>
<tr><td>BMEP585040 (from EDS)</td><td>Device</td><td>Schneider Electric</td><td>2.10</td><td></td></tr>
<tr><td>BMEP586040 (from EDS)</td><td>Device</td><td>Schneider Electric</td><td>2.10</td><td></td></tr>
<tr><td>BMX NOC0401 (from EDS)</td><td>Device</td><td>Schneider Electric</td><td>1.1</td><td></td></tr>
<tr><td>BMX NOC0401 Revision ...</td><td>Device</td><td>Schneider Electric</td><td>2.1</td><td></td></tr>
<tr><td>BMX NOC0402 (from EDS)</td><td>Device</td><td>Schneider Electric</td><td>1.1</td><td></td></tr>
<tr><td>ETB 1EI 08E 08S PP0 (fro...</td><td>Device</td><td>Schneider Electric</td><td>3.6</td><td></td></tr>
<tr><td>ETB 1EI 12E 04S PP0 (fro...</td><td>Device</td><td>Schneider Electric</td><td>3.6</td><td></td></tr>
<tr><td>ETB 1EI 16C P00 (from E...</td><td>Device</td><td>Schneider Electric</td><td>3.6</td><td></td></tr>
<tr><td>ETB 1EI 16E PP0 (from E...</td><td>Device</td><td>Schneider Electric</td><td>3.6</td><td></td></tr>
<tr><td>Generic Device</td><td>Device</td><td>Schneider Electric</td><td>1.3.0.0</td><td></td></tr>
<tr><td>Generic Device Explicit Msg</td><td>Device</td><td>Schneider Electric</td><td>1.3.0.0</td><td></td></tr>
<tr><td>Lexium 32 (from EDS)</td><td>Device</td><td>Schneider Electric</td><td>1.1</td><td></td></tr>
<tr><td>Modbus Device</td><td>Device</td><td>Schneider Electric</td><td>1.1.12.0</td><td>2015-05-27</td></tr>
<tr><td>Schneider TCSESM04XX...</td><td>Device</td><td>Schneider Electric</td><td>1.1</td><td></td></tr>
<tr><td>Schneider TCSESM04XX</td><td>Device</td><td>Schneider Electric</td><td>1.2</td><td></td></tr>
</table>

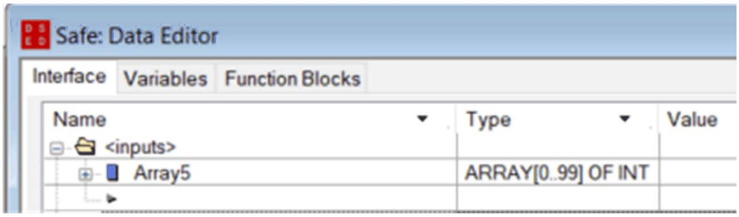
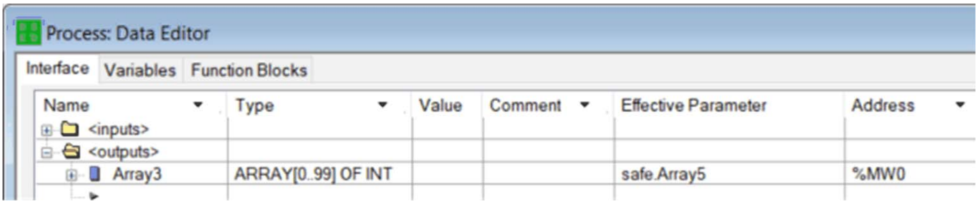
</td></tr>
</table>

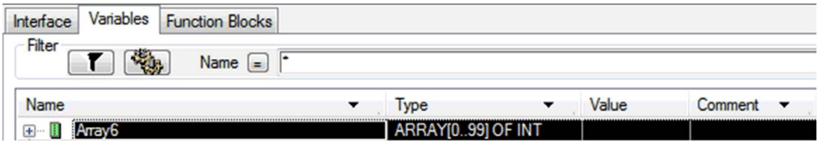
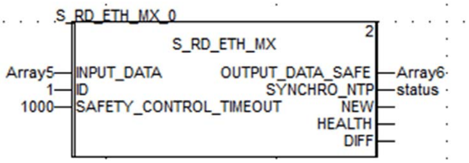
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>6</td><td>Open the newly added Modbus device, add a request, and in the <b>Request Setting</b> tab:
● Set the <b>WR Length</b> column, which is the length of the data to be written, to a value of 100, then
● Set the <b>WR Address</b> column, which is the address where the table on the receiving PAC will write the data it receives (in this example: 0, which means the sending PAC will write to the table starting to %MW0 in the receiving PAC.


<table>
<tr><th>Repetitive Rate(ms)</th><th>RD Address</th><th></th><th>RD Length</th><th>Last Value</th><th>WR Address</th><th></th><th>WR Length</th></tr>
<tr><td>60</td><td>0</td><td></td><td>0</td><td>Hold Value</td><td>0</td><td></td><td>100</td></tr>
</table>
</td></tr>
<tr><td>7</td><td>Select the <b>Request 001: Items</b> node, then in the <b>Output</b> tab define an array type of INT (that is ≥100 integers). This is the sending PAC table that will be written to the receiving PAC:


<table>
<tr><th></th><th>Offset/Device</th><th>Offset/Connection</th><th>Item Name</th></tr>
<tr><td>□</td><td>0</td><td>0</td><td></td></tr>
<tr><td>□</td><td>1</td><td>1</td><td></td></tr>
<tr><td>□</td><td>2</td><td>2</td><td></td></tr>
<tr><td>□</td><td>3</td><td>3</td><td></td></tr>
<tr><td>□</td><td>4</td><td>4</td><td></td></tr>
<tr><td>□</td><td>5</td><td>5</td><td></td></tr>
<tr><td>□</td><td>6</td><td>6</td><td></td></tr>
<tr><td>□</td><td>7</td><td>7</td><td></td></tr>
<tr><td>□</td><td>8</td><td>8</td><td></td></tr>
<tr><td>□</td><td>9</td><td>9</td><td></td></tr>
<tr><td>□</td><td>10</td><td>10</td><td></td></tr>
</table>
Default Item Name Root: BLOCKA
Select a region and click on the "Define Item(s)" button to create
- one or several item(s)
- an array
New Item(s) Data Type: INT
Define Selected Area As: One Item of Array Type
Item Name (32 char max): BLOCKA_QI0_100
</td></tr>
</table>

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>8</td><td>After saving and building the configuration, the block (BLOCKA_QI0_100 in this example) is automatically created as a process variable:

Interface | Variables | Function Blocks
Filter
Name = *
<table>
<tr><th></th><th>Type</th><th>Value</th><th>Comment</th><th>Alias</th><th>Alias of</th></tr>
<tr><td>BMEP58_ECPU_EXT</td><td>T_BMEP58_ECPU_EXT</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Modbus_Device</td><td>T_Modbus_Device</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Outputs</td><td>T_Modbus_Device_OUT</td><td></td><td>Output Variables</td><td></td><td></td></tr>
<tr><td>BLOCKA_QI0_100</td><td>ARRAY[0..99] OF INT</td><td></td><td></td><td>tab_p</td><td></td></tr>
</table></td></tr>
<tr><td>9</td><td>On the sending PAC, in a process code section, use a <code>MOVE</code> DFB to copy the contents of Array2 to the array defined above in the Modbus device structure:

.1
MOVE
1
Array2
IN
OUT
Modbus_Device.Outputs.BLOCKA_QI0_100

</td></tr>
<tr><td>10</td><td>On the receiving PAC, use the <b>Safe Data Editor</b> to create a 100 integer array (Array5) as an input in the <b>Interface</b> area:

Safe: Data Editor
Interface | Variables | Function Blocks
<table>
<tr><th>Name</th><th>Type</th><th>Value</th></tr>
<tr><td>&lt;inputs&gt;</td><td></td><td></td></tr>
<tr><td>Array5</td><td>ARRAY[0..99] OF INT</td><td></td></tr>
</table></td></tr>
<tr><td>11</td><td>On the receiving PAC, in the <b>Process Data Editor</b>, create an array (Array3) of 100 INT in the <outputs> section of the <b>Interface</b> tab. Link this array to the data area array (Array5, created in step 10) in the <b>Effective Parameter</b> column. The data sent by the sending PAC will be written into this array via the Modbus scanner, provided that this variable is located at the address defined in scanner of the sending PAC (in this example %MW0):

Process: Data Editor
Interface | Variables | Function Blocks
<table>
<tr><th>Name</th><th>Type</th><th>Value</th><th>Comment</th><th>Effective Parameter</th><th>Address</th></tr>
<tr><td>&lt;inputs&gt;</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>&lt;outputs&gt;</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Array3</td><td>ARRAY[0..99] OF INT</td><td></td><td></td><td>safe.Array5</td><td>%MW0</td></tr>
</table></td></tr>
</table>

Step	Action
12	On the receiving PAC, use the **Safety Data Editor** to create a 100 integer array (Array6):  Name: Array6 · Type: ARRAY[0..99] OF INT · Value: · Comment:
13	On the receiving PAC, in a section of code in the SAFE task, instantiate the `S_RD_ETH_MX` DFB with the array created in step 10 (Array5) as input parameter and with the array created in step 12 (Array6) as output parameter:  S_RD_ETH_MX_0 2 S_RD_ETH_MX Array5 INPUT_DATA 1 ID 1000 SAFETY_CONTROL_TIMEOUT OUTPUT_DATA_SAFE Array6 SYNCHRO_NTP status NEW HEALTH DIFF

Black Channel Peer-to-Peer

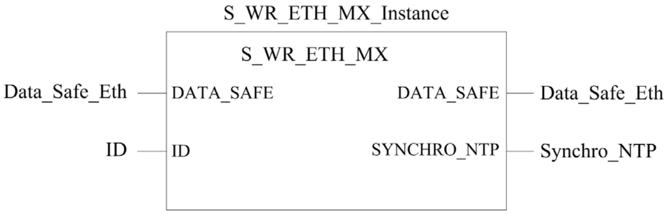
Each peer-to-peer data transmission consists of both *User Safety Data*, which contains the application-related content being transmitted, and *Reserved Data*. The *Reserved Data* is used by the safety PAC to test the reliability of the transmission so that it satisfies the requirements of SIL3. The *Reserved Data* consists of the following elements:

- A CRC calculated by the sending PAC from the data to be transmitted. The receiving PAC checks the CRC before using the transmitted data.
- A communication identifier, which is included in the CRC calculation to help prevent masquerade and insertion attacks on the transmission of safety data.
- A time stamp, containing the time of the transmission in ms. This stamped time is based on the time value provided by the NTP service and is used to synchronize both the sender PAC and receiver PAC. The data sender PAC adds a time value to the data sent to the receiver PAC. The receiver PAC compares the received time stamp with its own time value, and uses it to:
  - Check the age of the data.
  - Reject duplicate transmissions.
  - Determine the chronological order of received transmissions.
  - Determine the elapsed time between receipt of data transmissions.

Configuring the `S_WR_ETH_MX` DFB in the Program Logic of the Sender PAC

Representation

DFB representation:



For an extended description of this DFB, refer to the *EcoStruxure™ Control Expert, Safety, Block Library*.

Description

The DFB `S_WR_ETH_MX` is for PAC using CPU firmware 3.10 or earlier. It calculates data (reserved data containing a CRC and a time stamp) required by the receiver to check and manage errors detected during the safe peer-to-peer communication.

The `S_WR_ETH_MX` DFB function block has to be called at each cycle in the sender PAC. Within the cycle, it has to be executed in the logic after all required modifications have been performed on the data to be sent. This means that the data to be sent may not be modified within the cycle after the execution of the DFB, otherwise the CRC information used in the reserved data area will not be correct and the safe peer-to-peer communication will not succeed.

You have to assign the `ID` parameter a unique value that identifies the safe peer-to-peer communication between a sender and a receiver.

⚠ WARNING

LOSS OF ABILITY TO PERFORM SAFETY FUNCTIONS

The `ID` parameter value must be unique and fixed in the network for a sender/receiver pair.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

DATA_SAFE Array Description

Use the **Interface** tabs in both the **Safety Data Editor** and the **Process Data Editor** in Control Expert to make the link between the process variables and the safety variables.

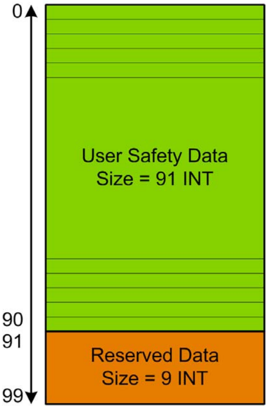
Linking process and safety variables in this manner makes it possible to:

- Transfer the value of safety variables to process variables, via linked global variables.
- Send variable values from the process area of the sending PAC to the process area of the receiving PAC, via explicit messaging over Modbus TCP.

DATA_SAFE array is composed of two zones:

- The **User Safety Data** zone contains the data from the safe area of the PAC. This zone starts at index 0 and finishes at index 90.
- The **Reserved Data** zone is reserved for auto-generated diagnostic data, including a CRC and time-stamp. This data is used by the receiving PAC to determine if the data contained in the **User Safety Data** zone is safe or not. This zone starts at index 91 and finishes at index 99.
  NOTE: Do not write in the **Reserved Data** zone.

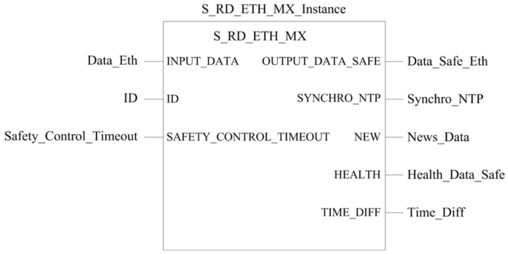
DATA_SAFE array (array[0..99] of INT) structure representation:



Configuring the S_RD_ETH_MX DFB in the Program Logic of the Receiver PAC

Representation

DFB representation:



Refer to the *EcoStruxure™ Control Expert, Safety, Block Library* for an extended description of this DFB.

Description

The DFB `S_RD_ETH_MX` is for PAC using CPU firmware 3.10 or earlier. It copies the data received in the process area to the safety area and validate the accuracy of the received data.

⚠ WARNING

LOSS OF ABILITY TO PERFORM SAFETY FUNCTIONS

- The `S_RD_ETH_MX` DFB function block must be called at each cycle in the receiver PAC program logic, and it must be executed before the data in the cycle is used.
- The `ID` parameter value must be unique and fixed in the network for a sender/receiver pair.
- You must test the `HEALTH` bit value of the `S_RD_ETH_MX` DFB at each cycle before using any safe data to manage the safety function.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

The `S_RD_ETH_MX` function block:

- Copies the data received in the `INPUT_DATA` register to the `OUTPUT_DATA_SAFE` register if it passes the following tests:
  - The function block checks the CRC of the last data packet received, via I/O scanner over Ethernet (Modbus TCP). If the CRC is not correct, the data is considered as unsafe and it is not written to the `OUTPUT_DATA_SAFE` register in the safety area.
  - The function block checks the last data received to determine if it is more recent than the data already written in the `OUTPUT_DATA_SAFE` register in the safety area (by comparing time stamps). If the last data received is not more recent, it is not copied to the `OUTPUT_DATA_SAFE` register in the safety area.

- Checks the age of the data in the safety area. If the age is higher than a configurable maximum value set in the SAFETY_CONTROL_TIMEOUT input register, the data is declared unsafe and the HEALTH bit is set to 0.
  NOTE: The data age is the time difference between the time when the data is computed in the sender PAC and the time when the data is checked in the receiver PAC. The time base reference is periodically updated with the time received from an NTP server.

  If the HEALTH bit is set to 0, the data available in the OUTPUT_DATA_SAFE array is considered as unsafe. In this case, take the appropriate reactive steps.

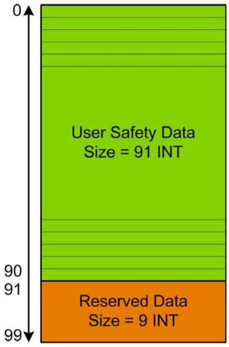
INPUT_DATA and OUTPUT_DATA_SAFE Arrays Description

The INPUT_DATA arrays consist of data coming from the process data memory area. The OUTPUT_DATA_SAFE arrays consist of safety variables. Use the **Safety Data Interface** and the **Process Data Interface** tabs in Control Expert to make the link between the process variables and the safety variables.

INPUT_DATA and OUTPUT_DATA_SAFE arrays are composed of 2 zones:

- The **User Safety Data** zone contains user data. This zone starts at index 0 and finishes at index 90.
- The **Reserved Data** zone is reserved for auto-generated diagnostic data, including a CRC and time-stamp. This data is used by the receiving PAC to determine if the data contained in the **User Safety Data** zone is safe or not. This zone starts at index 91 and finishes at index 99.
  NOTE: Writing in the **Reserved Data** zone is not recommended, as doing so will overwrite the auto-generated diagnostic data.

INPUT_DATA and OUTPUT_DATA_SAFE arrays (array[0..99] of INT) structure representation:



Calculating a SAFETY_CONTROL_TIMEOUT Value

When calculating a `SAFETY_CONTROL_TIMEOUT` value, consider the following:

- Minimum value: `SAFETY_CONTROL_TIMEOUT` > T1
- Recommended value: `SAFETY_CONTROL_TIMEOUT` > 2 * T1

T1 = $\text{CPU}_{\text{sender}}$ MAST cycle time + $\text{CPU}_{\text{sender}}$ SAFE cycle time + Repetitive_rate + Network transmission time + $\text{CPU}_{\text{receiver}}$ MAST cycle time + $\text{CPU}_{\text{receiver}}$ SAFE cycle time

Where:

- *CPU_{sender} MAST cycle time* is the MAST cycle time of the sender PAC.
- *CPU_{sender} SAFE cycle time* is the SAFE cycle time of the sender PAC.
- *Repetitive_rate* is the time rate for the I/O scanner write query from the sender PAC to the receiver PAC.
- *Network transmission time* is the time consumed on the Ethernet network for the data transmission from the sender PAC to the receiver PAC.
- *$CPU_{receiver}$ MAST cycle time* is the MAST cycle time of the receiver PAC.
- *$CPU_{receiver}$ SAFE cycle time* is the SAFE cycle time of the receiver PAC.

Note that the value defined for the `SAFETY_CONTROL_TIMEOUT` parameter has a direct effect on the robustness and availability of the safe peer-to-peer communication. If the `SAFETY_CONTROL_TIMEOUT` parameter value greatly exceeds T1, the communication will be tolerant to various delays (for example network delays) or corrupted data transmissions.

You are responsible for configuring your Ethernet network so the load that does not cause an excessive delay on the network during data transmission, which could lead to the expiration of the timeout. To help safeguard your safe peer-to-peer communications from any excessive delays due to other non-safety data transmitted on the same network, consider using a dedicated Ethernet network for the safe peer-to-peer protocol.

When commissioning your project, you have to estimate the safe peer-to-peer communication performance by checking the values provided in the output parameter `TIME_DIFF` and evaluating the margin using the value defined in the `SAFETY_CONTROL_TIMEOUT` parameter.

Understanding the **HEALTH** Bit

When the `HEALTH` bit value equals:

- 1: The integrity of the data is correct (CRC) and the age of the data is less than the value set in the `SAFTETY_CONTROL_TIMEOUT` input register.
  NOTE: The age of the data considered is the time between:
  - The beginning of the cycle where the data are computed in the sender PAC.
  - The beginning of the cycle where the data are checked in the receiver PAC.
- 0: New valid data are not received in the required time interval (the timer expires and the `HEALTH` bit is set to 0).
  NOTE: If the `HEALTH` bit is set to 0, the data in the output array `OUTPUT_DATA_SAFE` is considered to be unsafe; respond accordingly.

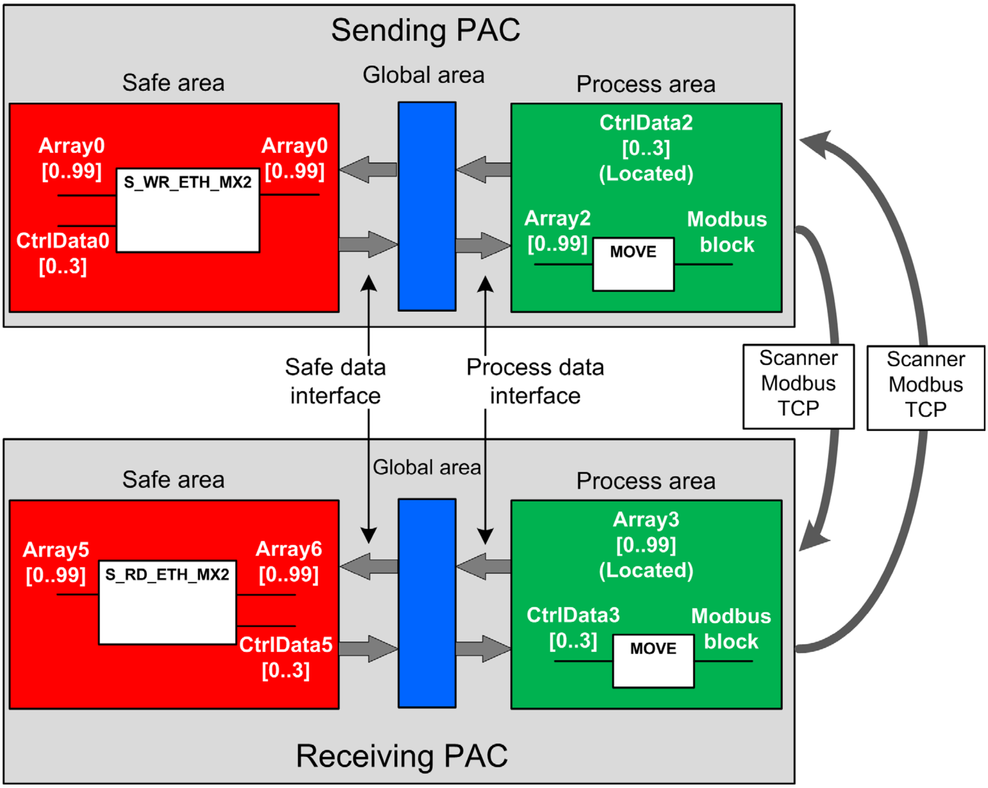
Peer-to-Peer Architecture using CPU Firmware 3.20 or Later

Architecture Design

Using CPU firmware 3.20 or later, the solution architecture is based on:

- Execution of 2 DFBs (`S_WR_ETH_MX2` and `MOVE`) in the sender PAC and 2 DFBs (`S_RD_ETH_MX2` and `MOVE`) in the receiver PAC.
- Scanning via Modbus TCP, for safe data transportation from Sender to Receiver.
- Scanning via Modbus TCP, for control data transportation from Receiver to Sender.

The following figure shows the overview of the process required to perform the safe peer-to-peer communication:

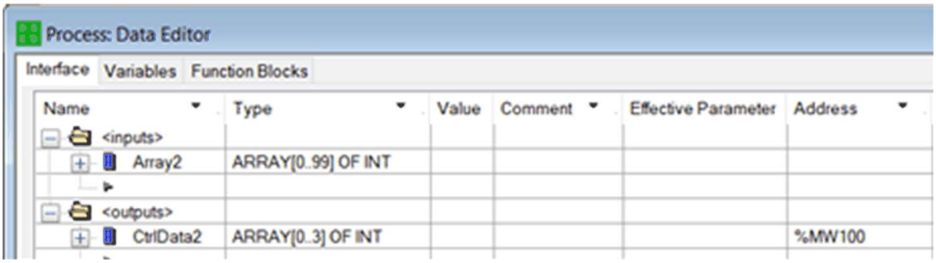
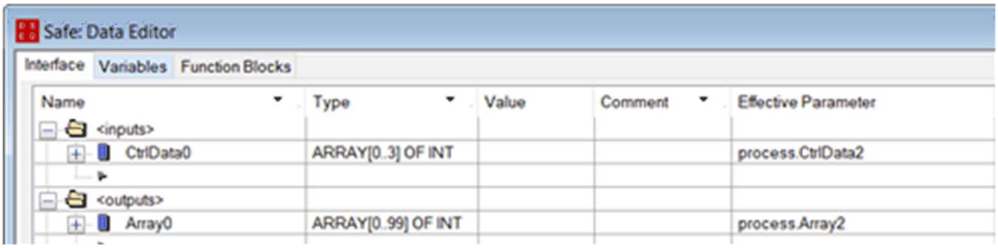


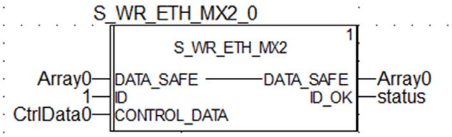
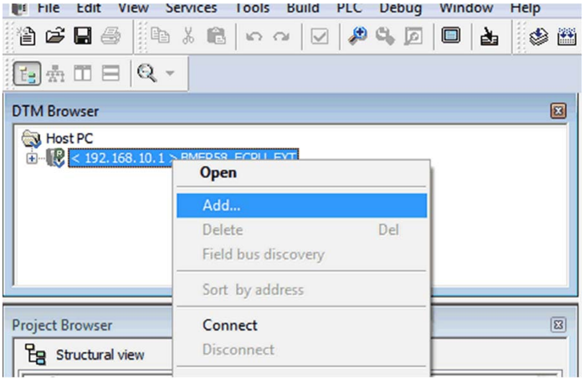
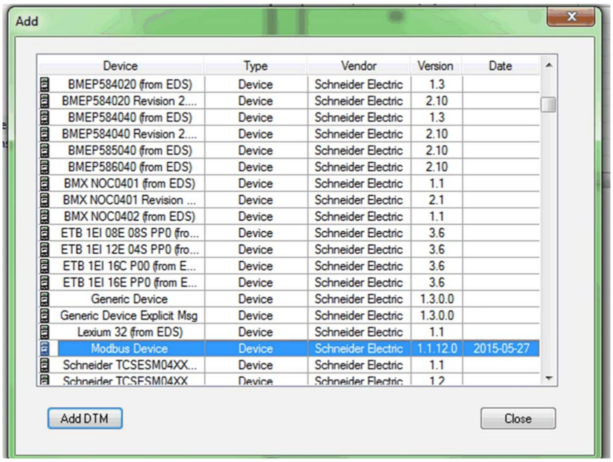
In the figure above, Control Expert automatically creates – and hides from external view – Array1 and Array4 in the Global areas of the peer PACs. From a user standpoint, the links are made from Array0 to Array2, and from Array3 to Array5.

NOTE: On the Ethernet network, you are allowed to mix safety related data and non-safety related data without impact on the integrity level of the safety related data. There is no restriction on the Ethernet network when using the safe peer-to-peer communication.

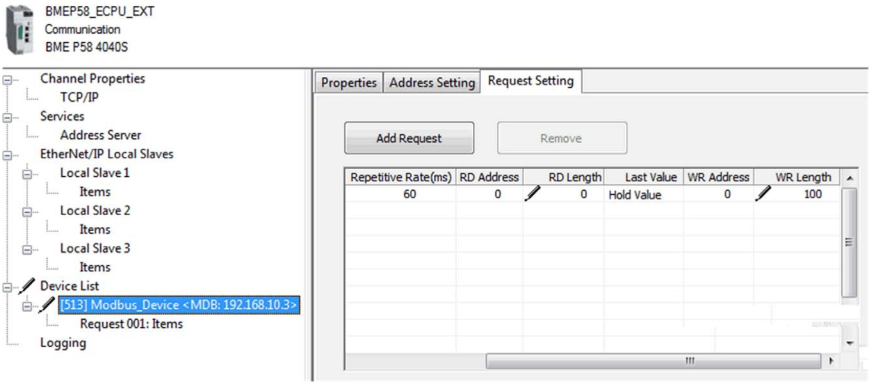
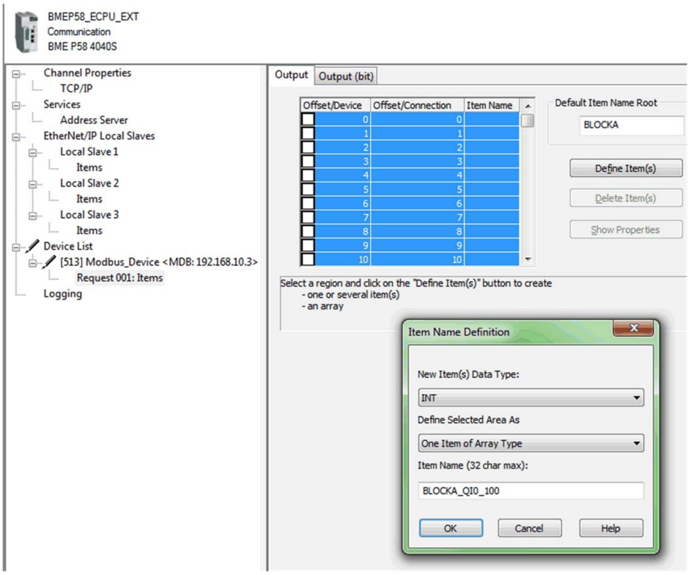
Peer-to-Peer Data Transfer Configuration Details

The following example shows you how to configure a peer-to-peer transfer of data between two safety PACs with firmware 3.20 or later and Control Expert 15.0 or later:

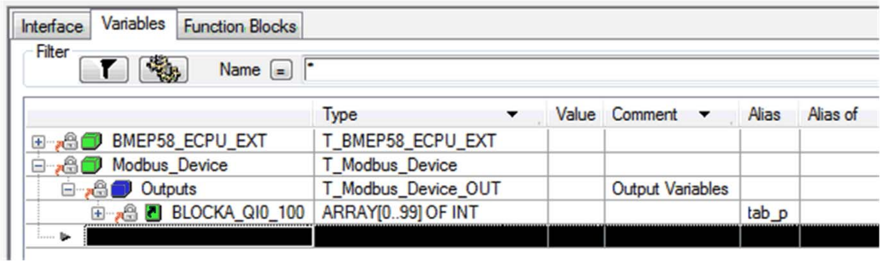
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>On the sending PAC, use the <b>Process Data Editor</b> to create a 100-integer array (Array2) as an input in the Interface area. Create in the same <b>Process Data Editor</b> a 4-Integer array (CtrlData2) as an output in the <b>Interface</b> area.
The control data from the receiving PAC will be written into this CtrlData2 via the Modbus scanner, provided this CtrlData2 is located at the address defined in scanner of the sending PAC (in this example %MW100 – see Step 14):

Process: Data Editor
Interface Variables Function Blocks
<table>
<tr><th>Name</th><th>Type</th><th>Value</th><th>Comment</th><th>Effective Parameter</th><th>Address</th></tr>
<tr><td><inputs></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Array2</td><td>ARRAY[0..99] OF INT</td><td></td><td></td><td></td><td></td></tr>
<tr><td><outputs></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CtrlData2</td><td>ARRAY[0..3] OF INT</td><td></td><td></td><td></td><td>%MW100</td></tr>
</table></td></tr>
<tr><td>2</td><td>On the sending PAC, use the <b>Safe Data Editor</b> to create another 100-integer array (Array0) as an output in the <b>Interface</b> area and link it to the process.Array2 data created in step1, above, in the <b>Effective Parameter</b> column.
Create in the same <b>Safe Data Editor</b> an array of 4-Integer array (CtrlData0) as an input in the <b>Interface</b> safety area and link it to the process.CtrlData2 data created in step1, above, in the <b>Effective Parameter</b> column.

Safe: Data Editor
Interface Variables Function Blocks
<table>
<tr><th>Name</th><th>Type</th><th>Value</th><th>Comment</th><th>Effective Parameter</th></tr>
<tr><td><inputs></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CtrlData0</td><td>ARRAY[0..3] OF INT</td><td></td><td></td><td>process.CtrlData2</td></tr>
<tr><td><outputs></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Array0</td><td>ARRAY[0..99] OF INT</td><td></td><td></td><td>process.Array2</td></tr>
</table>
<b>NOTE:</b> The integer variables from index 0 to 90 of the array contain the safety variable values you want to exchange with the receiving PAC. The remaining area is reserved for auto-generated diagnostic data, including a CRC and time-stamp. This diagnostic data is used by the receiving PAC to determine if the transferred data is safe.</td></tr>
</table>

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>3</td><td>On the sending PAC, configure the DFB <code>S_WR_ETH_MX2</code> in a section of the SAFE task. Link the DFB to Array0 and CtrlData0:

</td></tr>
<tr><td>4</td><td>In the <b>DTM Browser</b> of the sending PAC, select the CPU (in this example) or an NOC communications module (if any) then click <b>Add...</b> to create a Modbus scanner that can send data via Modbus TCP from the sending PAC to the receiving PAC:

</td></tr>
<tr><td>5</td><td>Select <b>Modbus Device</b> and click <b>Add DTM</b> to add the Modbus scanner:


<table>
<tr><th>Device</th><th>Type</th><th>Vendor</th><th>Version</th><th>Date</th></tr>
<tr><td>BMEP584020 (from EDS)</td><td>Device</td><td>Schneider Electric</td><td>1.3</td><td></td></tr>
<tr><td>BMEP584020 Revision 2....</td><td>Device</td><td>Schneider Electric</td><td>2.10</td><td></td></tr>
<tr><td>BMEP584040 (from EDS)</td><td>Device</td><td>Schneider Electric</td><td>1.3</td><td></td></tr>
<tr><td>BMEP584040 Revision 2....</td><td>Device</td><td>Schneider Electric</td><td>2.10</td><td></td></tr>
<tr><td>BMEP585040 (from EDS)</td><td>Device</td><td>Schneider Electric</td><td>2.10</td><td></td></tr>
<tr><td>BMEP586040 (from EDS)</td><td>Device</td><td>Schneider Electric</td><td>2.10</td><td></td></tr>
<tr><td>BMX NOC0401 (from EDS)</td><td>Device</td><td>Schneider Electric</td><td>1.1</td><td></td></tr>
<tr><td>BMX NOC0401 Revision ...</td><td>Device</td><td>Schneider Electric</td><td>2.1</td><td></td></tr>
<tr><td>BMX NOC0402 (from EDS)</td><td>Device</td><td>Schneider Electric</td><td>1.1</td><td></td></tr>
<tr><td>ETB 1EI 08E 08S PP0 (fro...</td><td>Device</td><td>Schneider Electric</td><td>3.6</td><td></td></tr>
<tr><td>ETB 1EI 12E 04S PP0 (fro...</td><td>Device</td><td>Schneider Electric</td><td>3.6</td><td></td></tr>
<tr><td>ETB 1EI 16C P00 (from E...</td><td>Device</td><td>Schneider Electric</td><td>3.6</td><td></td></tr>
<tr><td>ETB 1EI 16E PP0 (from E...</td><td>Device</td><td>Schneider Electric</td><td>3.6</td><td></td></tr>
<tr><td>Generic Device</td><td>Device</td><td>Schneider Electric</td><td>1.3.0.0</td><td></td></tr>
<tr><td>Generic Device Explicit Msg</td><td>Device</td><td>Schneider Electric</td><td>1.3.0.0</td><td></td></tr>
<tr><td>Lexium 32 (from EDS)</td><td>Device</td><td>Schneider Electric</td><td>1.1</td><td></td></tr>
<tr><td>Modbus Device</td><td>Device</td><td>Schneider Electric</td><td>1.1.12.0</td><td>2015-05-27</td></tr>
<tr><td>Schneider TCSESM04XX...</td><td>Device</td><td>Schneider Electric</td><td>1.1</td><td></td></tr>
<tr><td>Schneider TCSESM04XX</td><td>Device</td><td>Schneider Electric</td><td>1.2</td><td></td></tr>
</table></td></tr>
</table>

Step	Action
6	Open the newly added Modbus device, and in the **Request Setting** tab: - Set the **WR Length** column, which is the length of the data to be written, to a value of 100, then - Set the **WR Address** column, which is the address where the table on the receiving PAC will write the data it receives (in this example: 0, which means the sending PAC will write to the table starting to %MW0 in the receiving PAC.  BMEP58_ECPU_EXT Communication BME P58 4040S Channel Properties TCP/IP Services Address Server EtherNet/IP Local Slaves Local Slave 1 Items Local Slave 2 Items Local Slave 3 Items Device List [513] Modbus_Device <MDB: 192.168.10.3> Request 001: Items Logging Properties Address Setting Request Setting Add Request Remove Repetitive Rate(ms) | RD Address | RD Length | Last Value | WR Address | WR Length 60 | 0 | 0 | Hold Value | 0 | 100
7	Select the **Request 001: Items** node, then in the **Output** tab define an array type of INT (that is ≥100 integers). This is the sending PAC table that will be written to the receiving PAC:  BMEP58_ECPU_EXT Communication BME P58 4040S Channel Properties TCP/IP Services Address Server EtherNet/IP Local Slaves Local Slave 1 Items Local Slave 2 Items Local Slave 3 Items Device List [513] Modbus_Device <MDB: 192.168.10.3> Request 001: Items Logging Output Output (bit) Define Item(s) Delete Item(s) Show Properties Item Name Definition OK Cancel Help Offset/Device | Offset/Connection | Item Name 0 | 0 | 1 | 1 | 2 | 2 | 3 | 3 | 4 | 4 | 5 | 5 | 6 | 6 | 7 | 7 | 8 | 8 | 9 | 9 | 10 | 10 | Default Item Name Root BLOCKA Select a region and click on the "Define Item(s)" button to create - one or several item(s) - an array New Item(s) Data Type: INT Define Selected Area As One Item of Array Type Item Name (32 char max): BLOCKA_QI0_100

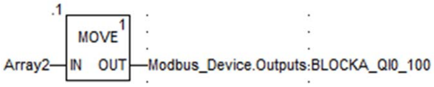
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>8</td><td>After saving and building the configuration, the block (BLOCKA_QI0_100 in his example) is automatically created as a process variable:



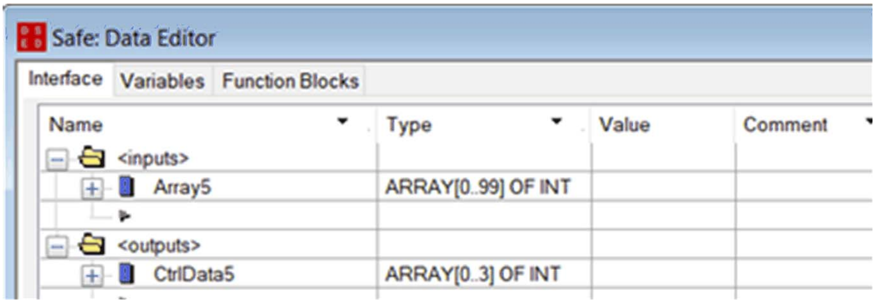
Interface | Variables | Function Blocks

Filter — Name = *

<table>
<tr><th></th><th>Type</th><th>Value</th><th>Comment</th><th>Alias</th><th>Alias of</th></tr>
<tr><td>BMEP58_ECPU_EXT</td><td>T_BMEP58_ECPU_EXT</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Modbus_Device</td><td>T_Modbus_Device</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Outputs</td><td>T_Modbus_Device_OUT</td><td></td><td>Output Variables</td><td></td><td></td></tr>
<tr><td>BLOCKA_QI0_100</td><td>ARRAY[0..99] OF INT</td><td></td><td></td><td>tab_p</td><td></td></tr>
</table>
</td></tr>
<tr><td>9</td><td>On the sending PAC, in a process code section, use a MOVE DFB to copy the contents of the "tab_p" array to the array defined above in the Modbus device structure:



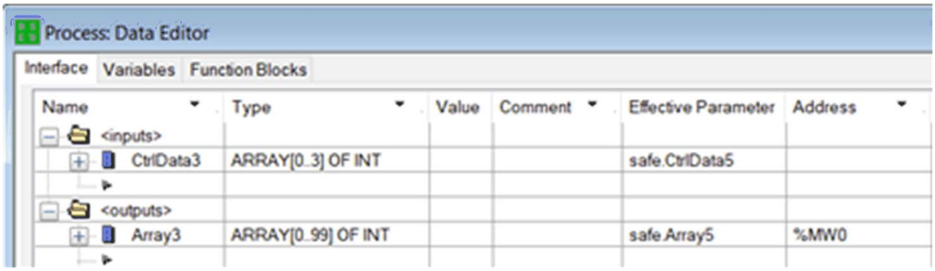
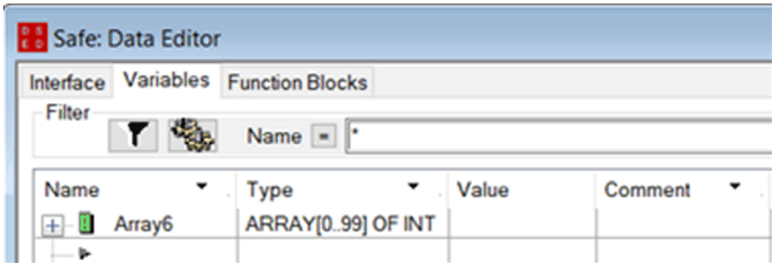
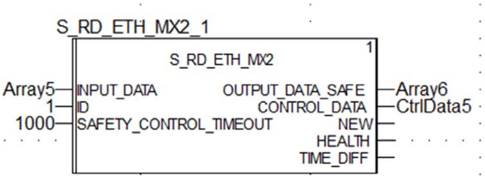
</td></tr>
<tr><td>10</td><td>On the receiving PAC, use the <b>Safe Data Editor</b> to create a 100-integer array (Array5) as an input in the <b>Interface</b> area.
Create in the same <b>Safe Data Editor</b> a 4-integer array (CtrlData5) as an output in the <b>Interface</b> area.



Safe: Data Editor

Interface | Variables | Function Blocks

<table>
<tr><th>Name</th><th>Type</th><th>Value</th><th>Comment</th></tr>
<tr><td>&lt;inputs&gt;</td><td></td><td></td><td></td></tr>
<tr><td>Array5</td><td>ARRAY[0..99] OF INT</td><td></td><td></td></tr>
<tr><td>&lt;outputs&gt;</td><td></td><td></td><td></td></tr>
<tr><td>CtrlData5</td><td>ARRAY[0..3] OF INT</td><td></td><td></td></tr>
</table>
</td></tr>
</table>

Step	Action
11	On the receiving PAC, in the **Process Data Editor**, create a 100-Integer array (Array3) as an output of the **Interface** area. Link this Array3 to the Array5 (created in step 10) in the **Effective Parameter** column. The data from the sending PAC will be written into this Array3 via the Modbus scanner, provided this Array3 is located at the address defined in scanner of the sending PAC (in this example %MW0). Create in the same **Process Data Editor** a 4-integer array (CtrlData3) as an input in the **Interface** area. Link this CtrlData3 to the CtrlData5 (created in step 10) in the **Effective Parameter** column.  Process: Data Editor Interface Variables Function Blocks Name Type Value Comment Effective Parameter Address <inputs> CtrlData3 ARRAY[0..3] OF INT safe.CtrlData5 <outputs> Array3 ARRAY[0..99] OF INT safe.Array5 %MW0
12	On the receiving PAC, use the **Safety Data Editor** to create a 100-integer array (Array6):  Safe: Data Editor Interface Variables Function Blocks Filter Name = * Name Type Value Comment Array6 ARRAY[0..99] OF INT
13	On the receiving PAC, in a section of code in the SAFE task, instantiate the `S_RD_ETH_MX2` DFB with the array created in step 10 (Array5) as input parameter and with the arrays created in step 10 (CtrlData5) and step 12 (Array6) as output parameters:  S_RD_ETH_MX2_1 S_RD_ETH_MX2 1 Array5 INPUT_DATA 1 ID 1000 SAFETY_CONTROL_TIMEOUT OUTPUT_DATA_SAFE Array6 CONTROL_DATA CtrlData5 NEW HEALTH TIME_DIFF
14	On the receiving PAC, redo Step 4 to 9 to configure a 4-Integer communication to send the CtrlData2 array from receiving PAC to sending PAC. In this example the CtrlData must be written to the sending PAC at address %MW100.

Black Channel Peer-to-Peer

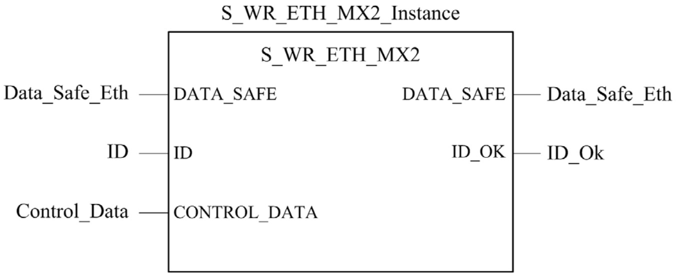
Each peer-to-peer data transmission consists of both *User Safety Data*, which contains the application-related content being transmitted, and *Reserved Data*. The *Reserved Data* is used by the safety PAC to test the reliability of the transmission so that it satisfies the requirements of SIL3. The *Reserved Data* consists of the following elements:

- A CRC calculated by the sending PAC from the data to be transmitted. The receiving PAC checks the CRC before using the transmitted data.
- A communication identifier, which is included in the CRC calculation to help prevent masquerade and insertion attacks on the transmission of safety data.
- A time stamp, containing the time of the transmission in ms. With CPU firmware 3.20 or later this stamped time is the safe time value provided by the receiver CPU. The data sender PAC adds a time value to the data sent to the receiver PAC. The receiver PAC compares the received time stamp with its own time value, and uses it to:
  - Check the age of the data.
  - Reject duplicate transmissions.
  - Determine the chronological order of received transmissions.
  - Determine the elapsed time between receipt of data transmissions.

Configuring the `S_WR_ETH_MX2` DFB in the Program Logic of the Sender PAC

Representation

DFB representation:



For an extended description of this DFB, refer to the *EcoStruxure™ Control Expert, Safety, Block Library*.

Description

The DFB S_WR_ETH_MX2 is for PAC using CPU firmware 3.20 or later. It calculates data (reserved data containing a CRC and a time stamp) required by the receiver to check and manage errors detected during the safe peer-to-peer communication.

The `S_WR_ETH_MX2` DFB function block has to be called at each cycle in the sender PAC. Within the cycle, it has to be executed in the logic after all required modifications have been performed on the data to be sent. This means that the data to be sent may not be modified within the cycle after the execution of the DFB, otherwise the CRC information used in the reserved data area will not be correct and the safe peer-to-peer communication will not succeed.

You have to assign the `ID` parameter a unique value that identifies the safe peer-to-peer communication between a sender and a receiver.

⚠ WARNING

LOSS OF ABILITY TO PERFORM SAFETY FUNCTIONS

The `ID` parameter value must be unique and fixed in the network for a sender/receiver pair.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

DATA_SAFE Array Description

Use the **Interface** tabs in both the **Safety Data Editor** and the **Process Data Editor** in Control Expert to make the link between the process variables and the safety variables.

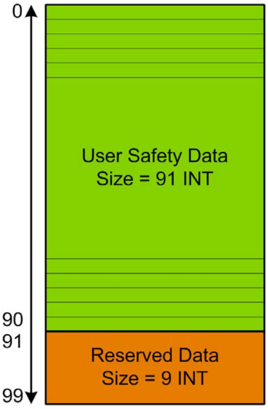
Linking process and safety variables in this manner makes it possible to:

- Transfer the value of safety variables to process variables, via linked global variables.
- Send variable values from the process area of the sending PAC to the process area of the receiving PAC, via explicit messaging over Modbus TCP.

DATA_SAFE array is composed of two zones:

- The **User Safety Data** zone contains the data from the safe area of the PAC. This zone starts at index 0 and finishes at index 90.
- The **Reserved Data** zone is reserved for auto-generated diagnostic data, including a CRC and time-stamp. This data is used by the receiving PAC to determine if the data contained in the **User Safety Data** zone is safe or not. This zone starts at index 91 and finishes at index 99.
  NOTE: Do not write in the **Reserved Data** zone.

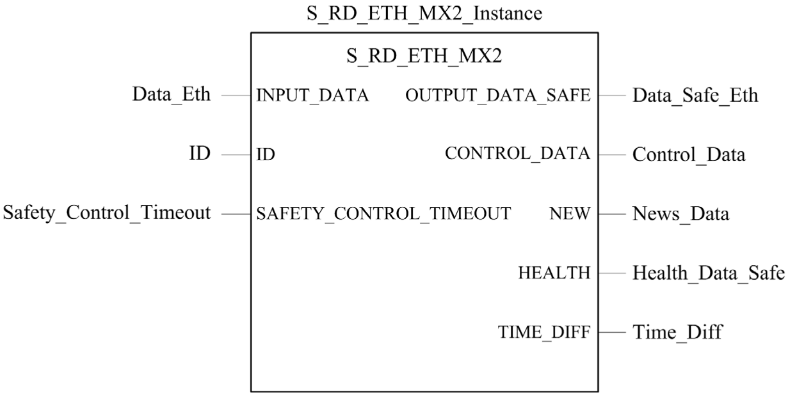
DATA_SAFE array (array[0..99] of INT) structure representation:



Configuring the `S_RD_ETH_MX2` DFB in the Program Logic of the Receiver PAC

Representation

DFB representation:



Refer to the *EcoStruxure™ Control Expert, Safety, Block Library* for an extended description of this DFB.

Description

The S_RD_ETH_MX2 DFB is for PAC using CPU firmware 3.20 or later. It copies the data received in the process area to the safety area and validate the accuracy of the received data.

⚠ **WARNING**

LOSS OF ABILITY TO PERFORM SAFETY FUNCTIONS

- The S_RD_ETH_MX2 DFB function block must be called at each cycle in the receiver PAC program logic, and it must be executed before the data in the cycle is used.
- The ID parameter value must be unique and fixed in the network for a sender/receiver pair.
- You must test the HEALTH bit value of the S_RD_ETH_MX2 DFB at each cycle before using any safe data to manage the safety function.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

The S_RD_ETH_MX2 function block:

- Copies the data received in the INPUT_DATA register to the OUTPUT_DATA_SAFE register if it passes the following tests:
  - The function block checks the CRC of the last data packet received, via I/O scanner over Ethernet (Modbus TCP). If the CRC is not correct, the data is considered as unsafe and it is not written to the OUTPUT_DATA_SAFE register in the safety area.
  - The function block checks the last data received to determine if it is more recent than the data already written in the OUTPUT_DATA_SAFE register in the safety area (by comparing time stamps). If the last data received is not more recent, it is not copied to the OUTPUT_DATA_SAFE register in the safety area.
- Checks the age of the data in the safety area. If the age is higher than a configurable maximum value set in the SAFETY_CONTROL_TIMEOUT input register, the data is declared unsafe and the HEALTH bit is set to 0.

  NOTE: The data age is the time difference between the time when the data is computed in the sender PAC and the time when the data is checked in the receiver PAC. The time base reference is periodically updated with the time received from an NTP server.

  If the HEALTH bit is set to 0, the data available in the OUTPUT_DATA_SAFE array is considered as unsafe. In this case, take the appropriate reactive steps.

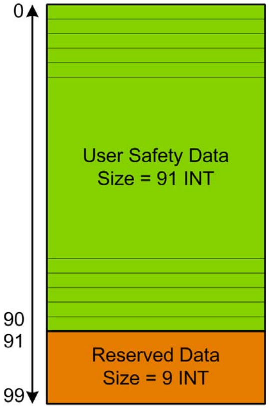
INPUT_DATA and OUTPUT_DATA_SAFE Arrays Description

The INPUT_DATA arrays consist of data coming from the process data memory area. The OUTPUT_DATA_SAFE arrays consist of safety variables. Use the **Safety Data Interface** and the **Process Data Interface** tabs in Control Expert to make the link between the process variables and the safety variables.

INPUT_DATA and OUTPUT_DATA_SAFE arrays are composed of 2 zones:

- The **User Safety Data** zone contains user data. This zone starts at index 0 and finishes at index 90.
- The **Reserved Data** zone is reserved for auto-generated diagnostic data, including a CRC and time-stamp. This data is used by the receiving PAC to determine if the data contained in the **User Safety Data** zone is safe or not. This zone starts at index 91 and finishes at index 99.
  NOTE: Writing in the **Reserved Data** zone is not recommended, as doing so will overwrite the auto-generated diagnostic data.

INPUT_DATA and OUTPUT_DATA_SAFE arrays (array[0..99] of INT) structure representation:



CONTROL_DATA Array Description

The CONTROL_DATA array has to be linked with variables in “Global” area (defined through the “Safety Data Interface”) and then, “Global” variables have to be linked to located variables in “Process” area (defined through the “Process Data Interface”) in order the data to be sent by IO Scanner to the correspondent sender.

Calculating a SAFETY_CONTROL_TIMEOUT Value

When calculating a SAFETY_CONTROL_TIMEOUT value, consider the following:

- Minimum value: SAFETY_CONTROL_TIMEOUT >2 * T1
- Recommended value: SAFETY_CONTROL_TIMEOUT > 3 * T1

T1 = CPU_{sender} MAST cycle time + CPU_{sender} SAFE cycle time + Repetitive_rate + Network transmission time + $CPU_{receiver}$ MAST cycle time + $CPU_{receiver}$ SAFE cycle time

Where:

- *CPU_{sender} MAST cycle time* is the MAST cycle time of the sender PAC.
- *CPU_{sender} SAFE cycle time* is the SAFE cycle time of the sender PAC.
- *Repetitive_rate* is the time rate for the I/O scanner write query from the sender PAC to the receiver PAC.
- *Network transmission time* is the time consumed on the Ethernet network for the data transmission from the sender PAC to the receiver PAC.
- *$CPU_{receiver}$ MAST cycle time* is the MAST cycle time of the receiver PAC.
- *$CPU_{receiver}$ SAFE cycle time* is the SAFE cycle time of the receiver PAC.

Note that the value defined for the SAFETY_CONTROL_TIMEOUT parameter has a direct effect on the robustness and availability of the safe peer-to-peer communication. If the SAFETY_CONTROL_TIMEOUT parameter value greatly exceeds T1, the communication will be tolerant to various delays (for example network delays) or corrupted data transmissions.

You are responsible for configuring your Ethernet network so the load that does not cause an excessive delay on the network during data transmission, which could lead to the expiration of the timeout. To help safeguard your safe peer-to-peer communications from any excessive delays due to other non-safety data transmitted on the same network, consider using a dedicated Ethernet network for the safe peer-to-peer protocol.

When commissioning your project, you have to estimate the safe peer-to-peer communication performance by checking the values provided in the output parameter TIME_DIFF and evaluating the margin using the value defined in the SAFETY_CONTROL_TIMEOUT parameter.

Understanding the HEALTH Bit

When the HEALTH bit value equals:

- 1: The integrity of the data is correct (CRC) and the age of the data is less than the value set in the SAFTETY_CONTROL_TIMEOUT input register.

  NOTE: The age of the data considered is the time between:

  - The beginning of the cycle where the data are computed in the sender PAC.
  - The beginning of the cycle where the data are checked in the receiver PAC.

- 0: New valid data are not received in the required time interval (the timer expires and the HEALTH bit is set to 0).

  NOTE: If the HEALTH bit is set to 0, the data in the output array OUTPUT_DATA_SAFE is considered to be unsafe; respond accordingly.

M580 Black Channel Communications

Black Channel

Black channel is the mechanism used to encrypt and validate transmitted safety data:

- Only Schneider Electric safety equipment can encrypt and decrypt the data sent via the black channel in an M580 safety system.
- The health of each safety data transmission is tested by both the transmitting and receiving safety module for each transmitted message.

The effect of using the black channel is to permit the transmission of safety data through non-safe intermediate equipment, such as backplanes, Ethernet cabling, communication adapters, and so forth. Because black channel transmissions are encrypted, the intermediate equipment cannot read or alter the content of the transmitted safety data without being detected.

Black channel transmissions operate independently of the communication protocol used for the transmission:

- X Bus is the carrier for backplane transmissions between safety devices on the same rack (e.g. from the CPU to local I/O, or from a communication remote adapter (CRA) to local I/O).
- EtherNet/IP is the carrier for data transmissions between racks (e.g. from the CPU to a CRA).

The Safety I/O modules and the CPU can send and receive black channel communications. For each transmission, the transmitting device (CPU or I/O) adds the following information to the message:

- a CRC tag to enable testing of the message content.
- a time stamp to enable testing of the timeliness of the message.
- other information– including the application version and the I/O configuration used – that identifies the I/O module in the transmission.

With CPU firmware 3.10 or earlier, when using safety I/O modules on a remote rack, configure the CPU as either an NTP client or NTP server.

If one of these designs is not implemented, the time settings of the safety I/O modules and CPU will not be synchronized and black channel communication will not operate properly. Inputs and outputs of safety I/O modules in RIO drops will enter the safe (de-energized) or the fallback state.

⚠ CAUTION

RISK OF UNINTENDED OPERATION

If you are placing safety I/O modules in an RIO drop, the current time needs to be configured for the PAC with CPU firmware 3.10 or earlier. Enable the NTP service for your M580 system and configure the safety CPU as an NTP server or an NTP client.

Failure to follow these instructions can result in injury or equipment damage.

The receiving device (I/O or CPU) decrypts the message and tests the accuracy of its content. The following conditions can be detected:

Condition	Description
Transmit errors	Error detected in the message address or routing.
Repeats	Message sent multiple times.
Deleted data	Part of the message is missing, or the message is lost.
Inserted data	Extra data is added to the message.
Out of sequence data	The message order is changed.
Corrupted data	One or more bit errors detected in the message.
Delays	The message delivery time is excessively long.
Masquerade	The source of the message is not permitted to send data.

When any of these errors are detected, the channel is determined to be unhealthy and the appropriate safety function is executed:

- If the CPU detects that a transmission from an input module is unhealthy, the CPU sets input values from that module to the safe (de-energized) or the fallback state).
- If an output module detects a transmission from the CPU is unhealthy, it places its outputs into their pre-configured fallback state.

The outputs automatically enter the state commanded by the CPU after communication between the CPU and the output module is correctly re-established.

NOTICE

UNEXPECTED OUTPUT STATE CHANGE UPON RE-ESTABLISHING COMMUNICATION

Program logic must monitor the health status of the output channels, and activate the safety function accordingly, by setting the output commands to the safe state.

Failure to follow these instructions can result in equipment damage.

Section 11.3
M580 CPU to Safety I/O Communication

M580 Safety PAC to I/O Communications

Communication Between the PAC and I/O

The M580 safety CPU and Copro together control all backplane exchanges, while the safety I/O respond to the commands of the CPU and Copro. Safety I/O modules can be installed in either a BMXXBP•••• X Bus rack or a BMEXBP•••• Ethernet rack.

Communications between the safety PAC and safety I/O modules in the local main rack are made via the backplane.

Communications between the safety PAC and safety I/O modules installed in an RIO drop are made through an adapter module installed on the RIO drop, either:

- a BMECRA31210 adapter, for an Ethernet rack, or
- a BMXCRA31210 adapter, for an X Bus rack.

NOTE: With CPU firmware 3.20 or later, the BM•CRA31210 adapter module needs a firmware 2.60 or later.

NOTE: A BMXCRA31200 adapter cannot be used to connect safety I/O modules to the M580 safety PAC.

Communications from the safety PAC and safety /O modules, both in the local main rack and in an RIO drop, are made via the black channel *(see page 205)*.

The way to synchronize the time settings of the CPU and the safety I/O modules is dependent to the CPU firmware version:

- For PAC with CPU firmware 3.10 or earlier, the NTP service configuration is required.
  NOTE: If you are installing safety I/O modules on local rack (or in an extension of the local rack), it is not required to enable the NTP service.
- For PAC with CPU firmware 3.20 or later, the safe time synchronization is based on an internal and "monotonic" time clock.

For more information refer to chapter *Time Synchronization (see page 172)*.

Optionally, you can use BMXNRP0200 or BMXNRP0201 fiber optic repeater modules to extend the physical link between the CPU and Copro in the local rack and the adapter in the RIO drop. Fiber optic repeater modules enhance RIO network noise immunity and increase cabling distance while maintaining the full dynamic range of the network and the safety integrity level.

The communication protocol between the safety I/O and PAC enables their exchanges. It permits both devices to check the accuracy of received data, detect corrupted data, and determine if the transmitting module becomes non-operational. Thus, a safety loop may include any non-interfering *(see page 27)* RIO adapters and backplane.

Supplying Power to the Safety I/O

The safety I/O is supplied 24 VDC and 3.3 VDC power over the backplane by the M580 safety power supply module *(see page 127)*. The safety power supply module monitors the power it provides so as not to exceed 36VDC.

Power for Non-Safety Functions:

5 VDC power provided by the backplane is applied by each safety I/O module to its non-safety functions.

External Power for Digital Safety I/O:

An external power supply, not greater than 60 VDC, is required for non-safety processes (sensor, actuator), and can be a protected extra-low voltage (PELV) overvoltage category II type power supply. The non-safety process power supply is supervised by the safety I/O module for overvoltage and undervoltage conditions.

Chapter 12
Diagnosing an M580 Safety System

Introduction

This chapter provides information on diagnostics that can be performed via hardware indicators (based on LED status) and system bits or words for an M580 safety system.

What Is in This Chapter?

This chapter contains the following sections:

Section 12.1
M580 Safety CPU and Coprocessor Diagnostics

Introduction

This section describes diagnostics available for both the BME•58•040S safety CPUs and the BMEP58CPROS3 safety coprocessor.

What Is in This Section?

This section contains the following topics:

Blocking Condition Diagnostics

Introduction

Blocking conditions caused during the execution of the safety or the process program result from either the detection of system errors, or of the HALT state of a task in which the error was detected.

NOTE: The M580 safety PAC presents two independent HALT states:

- Process HALT applies to the non-SAFE tasks (MAST, FAST, AUX0, and AUX1). When any process task enters the HALT state, all other process tasks also enter the HALT state.
- SAFE HALT applies only to the SAFE task.

Refer to the *M580 Safety PAC Operating States (see page 259)* topic for a description of the HALT and STOP states.

Diagnostics

When the CPU detects a blocking condition causing a system error, a description of the detected error is provided in system word %SW124.

When the CPU detects a blocking condition causing a HALT state, a description of the detected error is provided in system word %SW125.

%SW124 system word values and corresponding blocking condition description:

%SW124 Value (hex)	Blocking Condition Description
5AF2	RAM detected error in memory check
5AFB	Safety firmware code error detected
5AF6	Safety watchdog overrun detected on CPU
5AFF	Safety watchdog overrun detected on Coprocessor
5B01	Coprocessor not detected at startup

%SW125 system word values and corresponding blocking condition description:

%SW125 Value (hex)	Blocking Condition Description
0•••	execution of an unknown function
0002	SD card signature feature (used with SIG_CHECK and SIG_WRITE functions)
2258	execution of the HALT instruction
2259	execution flow different than the reference flow
23••	execution of a CALL function towards an undefined subroutine
5AF3	comparison error detected by CPU
5AF9	instruction error detected at start-up or runtime
5AFA	comparison error detected on CRC value
5AFC	comparison error detected by coprocessor

%SW125 Value (hex)	Blocking Condition Description
5AFD	internal error detected by coprocessor; sub-code in %SW126: 1 (unknown result), 2 (CRC application), 7 (incorrect activity counter)
5AFE	Copro synchronization error detected - CPU only; sub-code in %SW126: 3 (diagnostic), 4 (end UL), 5 (comparison), 6 (BC out), 8 (HALT during UL), 9 HALT during comparison), 10 (HALT during BC out).
81F4	SFC node incorrect
82F4	SFC code inaccessible
83F4	SFC work space inaccessible
84F4	too many initial SFC steps
85F4	too many active SFC steps
86F4	SFC sequence code incorrect
87F4	SFC code description incorrect
88F4	SFC reference table incorrect
89F4	SFC internal index calculation detected error
8AF4	SFC step status not available
8BF4	SFC memory too small after a change due to a download
8CF4	transition/action section inaccessible
8DF4	SFC work space too small
8EF4	version of the SFC code older than the interpreter
8FF4	version of the SFC code more recent than the interpreter
90F4	poor description of an SFC object: NULL pointer
91F4	action identifier not authorized
92F4	poor definition of the time for an action identifier
93F4	macro step cannot be found in the list of active steps for deactivation
94F4	overflow in the action table
95F4	overflow in the step activation/deactivation table
9690	error detected in the application CRC check (checksum)
DE87	calculation detected floating point error
DEB0	task watchdog overrun (%S11 and %S19 are set)
DEF0	division by 0
DEF1	character string transfer detected error
DEF2	capacity exceeded
DEF3	index overrun
DEF4	inconsistent task periods

%SW125 Value (hex)	Blocking Condition Description
DEF7	SFC execution detected error
DEFE	SFC steps undefined

Restarting the Application

After a blocking condition has occurred, the halted tasks need to be initialized. If the HALT occurred for a:

- process task (MAST, FAST, AUX0, or AUX1), initialization is performed by either the Control Expert **PLC** → **Init** command or by setting the %S0 bit to 1.
- SAFE task, initialization is performed by the Control Expert **PLC** → **Init Safety** command.

When initialized, the application behaves as follows:

- the data resume their initial value
- tasks are stopped at end of cycle
- the input image is refreshed
- outputs are controlled in fallback position

The RUN command then allows the application or tasks to be restarted.

Non-blocking Condition Diagnostics

Introduction

The system encounters a non-blocking condition when it detects an input/output error on the backplane bus (X Bus or Ethernet) or through execution of an instruction, which can be processed by the user program and does not modify the CPU operating state.

This topic describes some of the system bits and words you can use to detect the state of the safety system and its component modules.

NOTE: The available system bits and words do not include all information relating to the state of safety modules. Schneider Electric recommends that you use the DDDT structure of the safety CPU and safety IO modules to determine the state of the M580 safety system.

For information about the M580 safety CPU DDDT, refer to the topic *Standalone DDT Data Structure for M580 CPUs* in the *Modicon M580 Hardware Reference Manual*.

For information about the M580 safety I/O module DDDTs, refer to the following topics:

- BMXSAI0410 Data Structure *(see page 60)* for the safety analog input module.
- BMXSDI1602 Data Structure *(see page 90)* for the safety digital input module.
- BMXSDO0802 Data Structure *(see page 104)* for the safety digital output module.
- BMXSRA0405 Data Structure *(see page 122)* for the safety digital relay output module.

NOTE: You can also perform more advanced diagnostics of Ethernet devices by means of explicit messaging. To accomplish this, use either:

- the READ_VAR function block *(see EcoStruxure™ Control Expert, Communication, Block Library)* for Modbus TCP devices.
- the DATA_EXCH function block *(see Modicon M580, Hardware, Reference Manual)*, specifying the CIP protocol in the ADDM block, for EtherNet/IP devices.

Conditions Linked to I/O Diagnostics

A non-blocking condition linked to the I/O is diagnosed with the following indications:

- CPU **I/O** LED pattern: steady ON
- **I/O** module LED pattern: steady ON
- system bits (type of detected error):
  - `%S10` set to 0: global I/O error detected on one of the modules on the local or remote Ethernet or X Bus rack
  - `%S16` set to 0: I/O error detected in the task in progress on an X Bus rack
  - `%S40...%S47` set to 0: I/O error detected on an X Bus rack at address 0 to 7
  - `%S117` set to 0: RIO error detected on a remote X Bus rack
  - `%S119` set to 0: I/O error detected on a local X Bus rack

    NOTE: These bits (`%S10`, `%S16`, `%S40...%S47`, `%S117`, and `%S119`) report many – but not all – of the possible detected errors relating to safety I/O modules.

- system bits and words combined with the channel having an error detected (I/O channel number and type of detected error) or I/O module Device DDT information (for modules configured in Device DDT addressing mode):
  - bit `%Ir.m.c.ERR` set to 1: channel error detected (implicit exchanges)
  - word `%MWr.m.c.2`: the word value indicates the type of error detected on the specified channel and depends on the I/O module (implicit exchanges)

Conditions Linked to Execution of the Program Diagnostics

A non-blocking condition linked to execution of the program is diagnosed with the following system bits and words:

- system bits – type of error detected:
  - `%S15` set to 1: character string manipulation error detected.
  - `%S18` set to 1: capacity overrun, error detected on a floating point, or division by 0.
    (Refer to the topic *System Bits for Safe Task Execution (see page 394)* for more information.)
    When `%S18` is set to 1, `%SW17` contains a description of the causal event *(see page 396)*.
  - `%S20` set to 1: index overrun.

  NOTE: If the configurable system bit `%S78` is set in the program, the SAFE task enters the HALT state when system bit `%S18` set to 1.

- system word – nature of the error detected:
  - `%SW125` *(see Modicon M580, Hardware, Reference Manual)* (always updated)

M580 Safety CPU LED Diagnostics

CPU LEDs

Use the safety-related LEDs on the front face of the CPU *(see Modicon M580, Safety System Planning Guide)*, as described below, to diagnose the state of the PAC.

In the *Modicon M580 Hot Standby, System Planning Guide for Frequently Used Architectures*, refer to the topic LED Diagnostics for M580 Hot Standby CPUs for information on how to diagnose the redundancy-related LEDs, including **[A]**, **[B]**, **[PRIM]**, **[STBY]**, and **[REMOTE RUN]**.

NOTE: LEDs are not reliable indicators and cannot be guaranteed to provide accurate information. Use them only for general diagnostics during commissioning or troubleshooting.

⚠ WARNING

RISK OF INACCURATE SYSTEM DIAGNOSTIC

Do not use LEDs as operational indicators.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

PAC State	LED Names and Colors:							
	RUN	ERR	IO[1]	ETH MS	ETH NS	DL	SRUN	SMOD
	Green	Red	Red	Green/Red	Green/Red	Green	Green	Green
Power OFF	○	○	○	○	○	○	○	○
Power ON ● Autotest	🟩	🟥	🟥	🟩	🟩	○	🟩	🟩

1. Not all errors detected for a safety I/O module are reported via LEDs. Check the DDDTs for safety I/O modules for additional information.

PAC State	LED Names and Colors:							
	RUN	ERR	IO[1]	ETH MS	ETH NS	DL	SRUN	SMOD
	Green	Red	Red	Green/Red	Green/Red	Green	Green	Green
Not configured	○	❌	○	❎	○ No cable plugged and connected to another powered device ❎ Otherwise	○	○	○
Configured: • No external error detected	❎	○	○	🟢	❎	○	–	–
• External error detected	❎	○	○	–	–	○	–	–
• No Ethernet link, including the Ethernet backplane	❎	○	○	🟢	○	○	–	–
• Duplicate IP address	❎	○	–	🟢	🔴	○	–	–

[1] Not all errors detected for a safety I/O module are reported via LEDs. Check the DDDTs for safety I/O modules for additional information.

PAC State	LED Names and Colors:							
	RUN	ERR	IO[1]	ETH MS	ETH NS	DL	SRUN	SMOD
	Green	Red	Red	Green/Red	Green/Red	Green	Green	Green
● STOP state	❎	○	🔴 Detected error on I/O module, channel or configuration ○ No error detected on configured input/output	🟢	❎ Not connected 🟢 Connected ○ No cable	○	🟢 SAFE task running OR ❎ SAFE task stopped	🟢 Safety mode OR ❎ Maintenance mode
● RUN state	🟢	○	–	🟢	❎ Not connected 🟢 Connected ○ No cable	○	🟢 SAFE task running OR ❎ SAFE task stopped	🟢 Safety mode OR ❎ Maintenance mode
HALT state (recoverable error detected)	❎	❌	–	🟢	❌	○	🟢 SAFE task running ❎ SAFE task stopped	🟢 Safety mode ❎ Maintenance mode

[1] Not all errors detected for a safety I/O module are reported via LEDs. Check the DDDTs for safety I/O modules for additional information.

PAC State	LED Names and Colors:							
	RUN	ERR	IO[1]	ETH MS	ETH NS	DL	SRUN	SMOD
	Green	Red	Red	Green/Red	Green/Red	Green	Green	Green
SAFE state (non-recoverable error detected)	○	🔴	○	🔴	🔴	○	○	○
OS update	❎	○	○	🔴	🔴	🟢	○	○

[1] Not all errors detected for a safety I/O module are reported via LEDs. Check the DDDTs for safety I/O modules for additional information.

Legend:

Symbol	Description	Symbol	Description	Symbol	Description
🟢	Steady Green	🔴	Steady Red	○	OFF
❎	Blinking Green (500 ms ON, 500 ms OFF)	⊗	Blinking Red (500 ms ON, 500 ms OFF)	–	Not Applicable

M580 Safety Coprocessor LED Diagnostics

Coprocessor LEDs

Use the LEDs on the front face of the Coprocessor *(see Modicon M580, Safety System Planning Guide)* to diagnose the state of the PAC, as follows

Coprocessor State	LED Names and Colors:			
	SRUN	ERR	SMOD	DL
	Green	Red	Green	Green
Power OFF	○	○	○	○
WAIT state (Wait for firmware download from CPU)	○	○	○	●
Not configured (no application)	○	⊗	○	○
Configured and operating in safety mode: ● SAFE task stopped	⊗	○	●	○
● SAFE task running	●	○	●	○
Configured and operating in Maintenance mode: ● SAFE task stopped	⊗	○	⊗	○
● SAFE task running	●	○	⊗	○
SAFE task in HALT (recoverable error detected)	⊗	⊗	○	○
SAFE state (non-recoverable error detected)	○	●	○	○

Legend:

Symbol	Description		Symbol	Description		Symbol	Description
🟢	Steady Green		🔴	Steady Red		○	OFF
⊗	Blinking Green (500 ms ON, 500 ms OFF)		⊗	Blinking Red (500 ms ON, 500 ms OFF)			

Memory Card Access LED

Introduction

The green memory card access LED underneath the SD memory card door indicates the CPU access to the memory card when a card is inserted. This LED can be seen when the door is open.

Dedicated LED States

By itself, the **memory card access** LEDs indicate these states:

LED Status	Description
ON	The memory card is recognized, but the CPU is not accessing it.
flashing	The CPU is accessing the memory card.
blinking	The memory card is not recognized.
OFF	The memory card can be removed from the CPU slot or the CPU does not recognize the memory card.

NOTE: Confirm that the LED is OFF before you remove the card from the slot.

Combined LED Meanings

The access card LED operates together with the **BACKUP** LED *(see Modicon M580, Hardware, Reference Manual)*. Their combined patterns indicate the following diagnostic information:

Memory Card Status	Conditions	CPU State	Memory Card Access LED	BACKUP LED
no memory card in the slot	—	no configuration	○	○
memory card not OK	—	no configuration	⊗	○
memory card without project	—	no configuration	🟢	🔴
memory card with a non-compatible project	—	no configuration	🟢	🔴

– no specific circumstances or CPU state

Memory Card Status	Conditions	CPU State	Memory Card Access LED	BACKUP LED
memory card with a compatible project	An error is detected when the project is restored from the memory card to the CPU RAM.	no configuration	during transfer: ⊗ end of transfer: 🟢	during transfer: 🔴 end of transfer: 🔴
	No error is detected when the project is restored from the memory card to the CPU RAM.	—	during transfer: ⊗ end of transfer: 🟢	during transfer: 🔴 end of transfer: ○

– no specific circumstances or CPU state

This legend shows the different LED patterns:

Symbol	Meaning	Symbol	Meaning
○	off	🔴	steady red
🟢	steady green	⊗	blinking green

Section 12.2
M580 Safety Power Supply Diagnostics

Power Supply LED Diagnostics

Power Supply LEDs

The BMXCPS4002S, BMXCPS4022S, and BMXCPS3522S safety power supplies present a front panel that includes the following diagnostic LEDs:

- **OK**: Operating Status
- **ACT**: Activity
- **RD**: Redundancy (for redundant power supply designs)

The M580 safety power supply LEDs can present the following diagnostic information:

LED	Description
OK	● ON (green) indicates that all of the following are true: ❍ 24 Vdc backplane voltage is OK. ❍ 3.3 Vdc backplane voltage is OK. ❍ The RESET button has not been activated. ● Blinking indicates one of the following is true: ❍ 24 Vdc backplane current is not OK. ❍ 3.3 Vdc backplane current is not OK and the RESET button has not been activated. ● OFF indicates that at least one of the following is true: ❍ 24 Vdc backplane voltage is not OK. ❍ 3.3 Vdc backplane voltage is not OK. ❍ The RESET button has been activated.
ACT	● ON (green) indicates the power supply is supplying power. In a redundant power supply design, the module is the primary. ● OFF indicates the power supply is not supplying power. In a redundant power supply design, the module is the standby.
RD	● ON (green) indicates communication between the two power supply modules is OK. ● Blinking indicates one of the following is true: ❍ 24 Vdc backplane current is not OK. ❍ 3.3 Vdc backplane current is not OK. ● OFF indicates at least one of the following is true: ❍ Communication between the two power supply modules is not OK. ❍ Auto-tests are being performed.

Section 12.3
BMXSAI0410 Analog Input Diagnostics

Introduction

This section describes diagnostic tools available for the BMXSAI0410 safety analog input module.

What Is in This Section?

This section contains the following topics:

BMXSAI0410 DDDT Diagnostics

Introduction

The BMXSAI0410 safety analog input module provides the following diagnostics using its `T_U_ANA_SIS_IN_4` *(see page 60)* device DDT elements:

- input diagnostics
- internal error detection
- channel wiring diagnostics

Input Diagnostics

The sensors connected to each channel are monitored for their ability to accurately measure 10 analog input values between 4 and 20 mA. If the input measurement tests are not successful, the `CH_HEALTH` bit in the `T_U_ANA_SIS_CH_IN` *(see page 62)* DDDT structure is set to 0, indicating it is not operational.

Internal Error Detection

The module processes the input value using two separate, parallel circuits. The two values are compared to determine if an internal error is detected in the module process. If the compared values are different, the `IC` bit in the `T_U_ANA_SIS_CH_IN` DDDT structure is set to 1, indicating it is not operational.

Refer to the architecture diagram *(see page 139)* for the BMXSAI0410 safety analog input module for a visual presentation of this process.

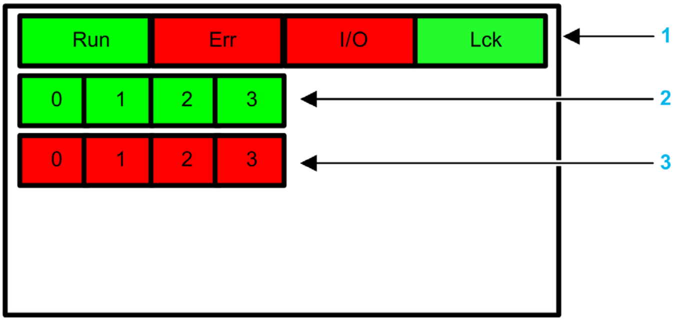
Channel Wiring Diagnostics

The wiring of the sensor to the input channel is continuously diagnosed for a cut wire condition, which is detected when the measured current is less than 3.75 mA or greater than 20.75 mA. In this case, the `OOR` bit in the `T_U_ANA_SIS_CH_IN` DDDT structure is set to 1.

BMXSAI0410 Analog Input LED Diagnostics

LED Panel

The BMXSAI0410 analog input module presents the following LED panel on its front face:



1 Module state LEDs
2 Channel state LEDs
3 Channel detected error LEDs

NOTE:

- The channel detected error LEDs are operational only after the module has been properly configured. When a channel error is detected, the corresponding LED remains ON until the underlying condition is resolved.
- Because the input module includes only four channels, LEDs in positions 4...7 are not used and are never powered on.

Module Diagnostics

Use the four LEDs at the top of the LED panel to diagnose the condition of the BMXSAI0410 analog input module:

Module LEDs				Module State	Recommended Response
Run	Err	I/O	LCK		
Blinking[1]	Blinking[1]	Blinking[1]	Blinking[1]	Auto-test at power-on.	–
Blinking[1]	ON	OFF	Blinking[1]	Auto-test at power-on has detected an internal error on input channels.	Replace the module.
OFF	ON	OFF	OFF	Internal error detected.	Replace the module if the condition persists.
OFF	Blinking[1]	OFF	X	Non-configured I/O module.	Configure the module via the CPU.
X	X	ON	X	External error detected on input channel.	Refer to *Channel Diagnostics (see page 229)* (below).
ON	Blinking[1]	X	X	No communication between CPU and I/O module.	Verify that: ● The CPU is an M580 safety CPU and that it is operational. ● The backplane is operational (if I/O module is on main rack). ● The cable between the CPU and I/O module is operational and properly connected (if the I/O module is on an extended or remote rack).
ON	Flickering[2]	X	OFF	Communication not safe and configuration unlocked.	Debug the condition using the DDDT variables *(see page 60)* for the I/O module instance.
X indicates the LED state can be either ON or OFF. 1. Blinking: 500 ms ON / 500 ms OFF. 2. Flickering: 50 ms ON / 50 ms OFF.					

Module LEDs				Module State	Recommended Response
Run	Err	I/O	LCK		
ON	Flickering[2]	X	ON	Communication not safe and configuration locked.	Verify that: ● The locked configuration in the module is equal to the module configuration stored in application in the CPU as configured using Control Expert. ● Debug the condition using the DDDT variables *(see page 60)* for the I/O module instance.
ON	ON	OFF	X	Input channel Internal error detected	Replace the module if the condition persists.
ON	OFF	OFF	OFF	Communication with CPU is OK and the configuration is unlocked.	–
ON	OFF	OFF	ON	Communication with CPU is OK and the configuration is locked.	–
X indicates the LED state can be either ON or OFF. 1. Blinking: 500 ms ON / 500 ms OFF. 2. Flickering: 50 ms ON / 50 ms OFF.					

Channel Diagnostics

Use all the LEDs on the BMXSAI0410 analog input module to diagnose channel status:

Module LEDs				Channel LEDs		Channel State	Recommended Response
Run	Err	I/O	LCK	Channel State (LED 0...3)	Detected Error (LED 0...3)		
ON	OFF	Off	X	ON	OFF	The input current is in the range 4...20 mA on the channel.	–
ON	OFF	ON	X	OFF	OFF	The input current is out of the range 4...20 mA on the channel.	Verify that the external power-supply, the external cabling, and the sensor are operational.
ON	ON	OFF	X	OFF	ON	The channel is not operational.	Replace the module if the condition persists.
X indicates the LED state can be either ON or OFF.							

Section 12.4
BMXSDI1602 Digital Input Diagnostics

Introduction

This section describes diagnostic tools available for the BMXSDI1602 safety digital input module.

What Is in This Section?

This section contains the following topics:

BMXSDI1602 DDDT Diagnostics

Introduction

The BMXSDI1602 safety digital input module provides the following diagnostics using its `T_U_DIS_SIS_IN_16` *(see page 90)* device DDT elements:

- input diagnostics
- internal error detection
- channel wiring diagnostics
- overvoltage and undervoltage diagnostics

Input Diagnostics

Each input channel is tested for operational effectiveness at the start of every cycle (or scan). Each channel is forced to the energized state and tested to check that the energized state was achieved. Then the channel is forced to the de-energized state and again is tested to check that the de-energized state was achieved.

If the channel does not successfully toggle between energized and de-energized, the `CH_HEALTH` bit in the `T_U_DIS_SIS_CH_IN` *(see page 92)* DDDT structure is set to 0 indicating it is not operational.

Internal Error Detection

Each cycle, the module performs an input diagnostic sequence. The module processes the input value using two separate, identical circuits. The two values are compared to determine if an internal error exists in the module internal process. If the compared values are different, the `IC` bit in the `T_U_DIS_SIS_CH_IN` DDDT structure is set to 1 indicating it is not operational.

Refer to the architecture diagram *(see page 140)* for the BMXSDI1602 safety digital input module for a visual presentation of this process.

Channel Wiring Diagnostics

The wiring of the sensor to the input channel can be continuously diagnosed for any of the following conditions:

- cut wire (open circuit)
- short circuit to 24 Vdc
- short circuit to 0 Vdc
- cross circuit between two parallel channels

The availability of these diagnostics depends on the power source employed by the specific wiring design *(see page 70)*, and on the diagnostic function being enabled in the configuration page of the module.

If one of these conditions is detected, the `T_U_DIS_SIS_CH_IN` DDDT structure sets the associated bit value to 1, as follows:

- the `OC` bit is set to 1 if an open (cut) wire or short circuit to 0 Vdc ground condition is detected.
- the `SC` bit is set to 1 if a short circuit to the 24 Vdc source or cross circuit between two channels is detected.

Overvoltage and Undervoltage Diagnostics

The module continuously tests for an overvoltage and an undervoltage condition. The following threshold values apply:

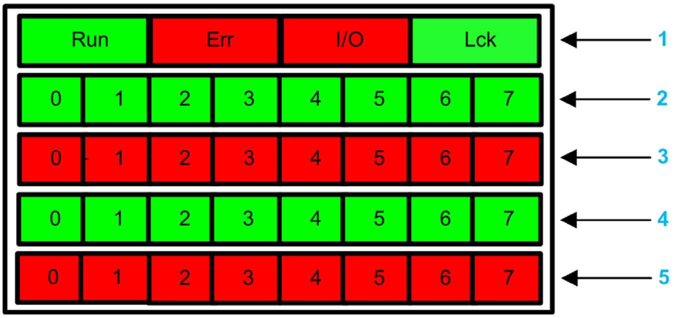
- Undervoltage threshold = 18.6 Vdc
- Overvoltage threshold = 33 Vdc

If either condition is detected, the module sets the `PP_STS` bit in the `T_U_DIS_SIS_IN_16` device DDT to 0.

BMXSDI1602 Digital Input LED Diagnostics

LED Panel

The BMXSDI1602 digital input module presents the following LED panel on its front face:



1 Module state LEDs
2 Channel state LEDs for Rank A
3 Channel detected error LEDs for Rank A
4 Channel state LEDs for Rank B
5 Channel detected error LEDs for Rank B

NOTE: When a channel error is detected, the corresponding LED remains ON until the underlying condition is resolved.

Module Diagnostics

Use the four LEDs at the top of the LED panel to diagnose the condition of the BMXSDI1602 digital input module:

Module LEDs				Module State	Recommended Response
Run	Err	I/O	LCK		
Blinking	Blinking[1]	Blinking[1]	Blinking[1]	Auto-test at power-on.	–
Blinking	ON	OFF	Blinking[1]	Auto-test at power-on has detected an internal error on input channels.	Replace the module.
Blinking	ON	ON	Blinking[1]	● Auto-test module at power-on has detected an internal error on input channels; or ● External 24VDC power supply is out of range	Verify the external pre-actuator 24 Vdc power supply is operational, and connect the 24 Vdc supply.
OFF	ON	OFF	OFF	Internal error detected.	Replace the module if the condition persists.
OFF	Blinking[1]	OFF	X	Non-configured I/O module.	Configure the module via the CPU.
X	XX	ON	X	● External 24 Vdc power supply is out of range; or ● External error detected on input channel.	● Verify the external pre-actuator 24 Vdc power supply is operational. ● Refer to *Channel Diagnostics (see page 236)*.
ON	Blinking[1]	X	X	No communication between CPU and module.	Verify that: ● The CPU is an M580 safety CPU and that it is operational. ● The backplane is operational (if I/O module is on main rack). ● The cable between the CPU and I/O module is operational and properly connected (if the I/O module is on an extended or remote rack).
ON	Flickering[2]	X	OFF	Communication not safe and configuration unlocked.	Debug the condition using the DDDT variables *(see page 90)* for the I/O module instance.
X indicates the LED state can be either ON or OFF. 1. Blinking: 500 ms ON / 500 ms OFF. 2. Flickering: 50 ms ON / 50 ms OFF.					

Module LEDs				Module State	Recommended Response
Run	Err	I/O	LCK		
ON	Flickering[2]	X	ON	Communication not safe and configuration locked.	- Verify the locked configuration in the module is equal to the module configuration stored in application in the CPU as configured using Control Expert. - Debug the condition using the DDDT variables *(see page 90)* for the I/O module instance.
ON	ON	OFF	X	Input channel Internal error detected.	Replace the module if the condition persists.
ON	OFF	OFF	OFF	Communication with CPU is OK and the configuration is unlocked.	–
ON	OFF	OFF	ON	Communication with CPU is OK and the configuration is locked.	–
X indicates the LED state can be either ON or OFF. 1. Blinking: 500 ms ON / 500 ms OFF. 2. Flickering: 50 ms ON / 50 ms OFF.					

Channel Diagnostics

Use all the LEDs on the BMXSDI1602 digital input module to diagnose channel status:

Module LEDs				Channel LEDs		Channel State	Recommended Response
Run	Err	I/O	LCK	Channel State (LED 0...7, Rank A/B)	Detected Error (LED 0...7, Rank A/B)		
ON	OFF	OFF	X	ON	OFF	Input state on.	–
ON	OFF	OFF	X	OFF	OFF	Input state off.	–
ON	ON	OFF	X	OFF	ON	Input state off. An internal error is detected on the channel.	To change the module if condition is persistent
ON	ON	ON	X	OFF	ON	External 24 Vdc power supply is out of range.	Verify the external pre-actuator 24 Vdc power supply is operational.
ON	OFF	ON	X	X	Blinking[1]	The input is in either: ● An open circuit condition; or ● A short-circuit condition with the 0 Vdc.	Verify that the cabling is operational and properly connected.
ON	OFF	ON	X	X	Flickering[2]	The input is in either: ● A short-circuit condition with the 24 Vdc; or ● A short-circuit condition with the 0 Vdc.	Verify that the cabling is operational and properly connected.
X indicates the LED state can be either ON or OFF.							

Section 12.5
BMXSDO0802 Digital Output Diagnostics

Introduction

This section describes diagnostic tools available for the BMXSDO0802 safety digital output module.

What Is in This Section?

This section contains the following topics:

BMXSDO0802 DDDT Diagnostics

Introduction

The BMXSDO0802 safety digital output module provides the following diagnostics using its T_U_DIS_SIS_OUT_8 *(see page 104)* device DDT elements:

- output diagnostics
- internal error detection
- channel wiring diagnostics
- overvoltage and undervoltage diagnostics

Output Diagnostics

Each output channel is tested for operational effectiveness at the start of each cycle (or scan). The test consists of toggling the output contact states (from ON to OFF, or OFF to ON) for a time too short to cause an actuator response (less than 1 ms). If the channel does not successfully toggle between energized and de-energized, the CH_HEALTH bit in the T_U_DIS_SIS_CH_OUT *(see page 106)* DDDT structure is set to 0, indicating it is not operational.

Internal Error Detection

The module processes the output value using two separate, identical circuits. Each circuit reads the midpoint voltage on the channel. The two values are compared, and if the values are not the expected values, an internal detected error is flagged by setting the IC bit in the T_U_DIS_SIS_CH_OUT DDDT structure to 1, indicating it is not operational.

Refer to the architecture diagram *(see page 141)* for the BMXSDO0802 safety digital output module for a visual presentation of this process.

Channel Wiring Diagnostics

The wiring of the actuator to the output channel can be continuously diagnosed for any of the following conditions:

- cut wire (open circuit)
- short circuit to 24 Vdc
- short circuit to 0 Vdc
- cross circuit between two parallel channels
- channel overload
  NOTE: Channel overload can be detected only if the output is energized.

The availability of these diagnostics depends on the diagnostic function being enabled in the configuration page of the module.

If one of these conditions is detected, the `T_U_DIS_SIS_CH_OUT` DDDT structure sets the associated bit value to 1, as follows:

- the `OC` bit is set to 1 if an open (cut) wire condition is detected.
- the `SC` bit is set to 1 if a short circuit to the 24 Vdc source or cross circuit between two channels is detected.
- the OL bit is set to 1 if a short circuit to the 0 Vdc ground or a channel overload condition is detected.

Overvoltage and Undervoltage Diagnostics

The module continuously tests for an overvoltage and an undervoltage condition. The following threshold values apply:

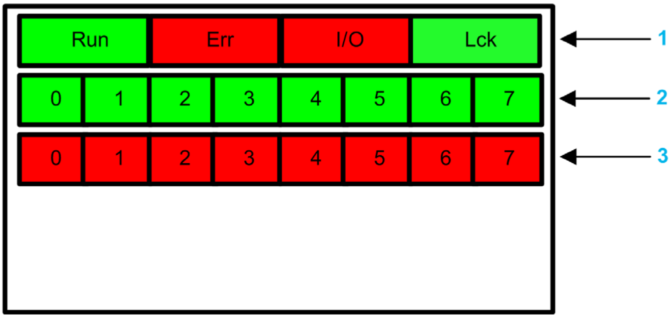
- Undervoltage threshold = 18 Vdc
- Overvoltage threshold = 31.8 Vdc

If either condition is detected, the module sets the `PP_STS` bit in the `T_U_DIS_SIS_OUT_8` device DDT to 0.

BMXSDO0802 Digital Output LED Diagnostics

LED Panel

The BMXSDO0802 digital output module presents the following LED panel on its front face:



1 Module state LEDs
2 Channel state LEDs
3 Channel detected error LEDs

NOTE: When a channel error is detected, the corresponding LED remains ON until the underlying condition is resolved.

Module Diagnostics

Use the four LEDs at the top of the LED panel to diagnose the condition of the BMXSDO0802 digital output module:

Module LEDs				Module State	Recommended Response
Run	Err	I/O	LCK		
Blinking[1]	Blinking[1]	Blinking[1]	Blinking[1]	Auto-test at power-on.	–
Blinking[1]	ON	OFF	Blinking[1]	Auto-test at power-on has detected an internal error on output channels.	Replace the module.
Blinking[1]	ON	ON	Blinking[1]	● Auto-test module at power-on has detected an internal error on output channels; or ● External 24VDC power supply is out of range	Verify the external pre-actuator 24 Vdc power supply is operational, and connect the 24 Vdc supply.
OFF	ON	OFF	OFF	Internal error detected.	Replace the module if the condition persists.
OFF	Blinking[1]	OFF	X	Non-configured I/O module.	Configure the module via the CPU.
X	X	ON	X	● External 24 Vdc power supply is out of range; or ● External error detected on output channel.	● Verify the external pre-actuator 24 Vdc power supply is operational. ● Refer to *Channel Diagnostics (see page 243)* (below).
ON	Blinking[1]	X	X	No communication between CPU and module. The module is in Fallback state (or in Reset if module has never been operating normally).	Verify that: ● The CPU is an M580 safety CPU and that it is operational. ● The backplane is operational (if I/O module is on main rack). ● The cable between the CPU and I/O module is operational and properly connected (if the I/O module is on an extended or remote rack).

X indicates the LED state can be either ON or OFF.
1. Blinking: 500 ms ON / 500 ms OFF.
2. Flickering: 50 ms ON / 50 ms OFF.

Module LEDs				Module State	Recommended Response
Run	Err	I/O	LCK		
ON	Flickering[2]	X	OFF	Communication not safe and configuration unlocked. The module is in Fallback state (or in Reset if module has never been operating normally).	To verify the variables available to debug safe communication in DDDT
ON	Flickering[2]	X	ON	Communication not safe and configuration locked. Module is in Fallback state.	● Verify the locked configuration in the module is equal to the module configuration stored in application in the CPU as configured using Control Expert. ● Debug the condition using the DDDT variables *(see page 104)* for the I/O module instance.
ON	ON	OFF	X	Internal error detected on an output channel.	Replace the module if the condition persists.
ON	OFF	OFF	OFF	Communication with CPU is safe and the configuration is unlocked	–
ON	OFF	OFF	ON	Communication with CPU is safe and the configuration is locked.	–
X indicates the LED state can be either ON or OFF. 1. Blinking: 500 ms ON / 500 ms OFF. 2. Flickering: 50 ms ON / 50 ms OFF.					

Channel Diagnostics

Use all the LEDs on the BMXSDO0802 digital output module to diagnose channel status:

Module LEDs				Channel LEDs		Channel State	Recommended Response
Run	Err	I/O	LCK	Channel State (LED 0...7)	Detected Error (LED 0...7		
ON	OFF	OFF	X	ON	OFF	Output state on.	–
ON	OFF	OFF	X	OFF	OFF	Output state off.	–
ON	ON	OFF	X	OFF	ON	Output state off. Internal error detected on output channel.	Replace the module if the condition persists.
ON	ON	ON	X	OFF	ON	External pre-actuator 24VDC power supply is out of range	Verify the 24 Vdc power supply is operational.
ON	OFF	ON	X	OFF	Blinking[1]	The output is in: ● An open circuit condition; or ● A short-circuit condition with the 0 Vdc; or ● In voltage overload.	Verify that the cabling is operational and properly connected.
ON	OFF	ON	X	ON	Flickering[2]	The output is in: ● A short-circuit condition with the 24 Vdc; or ● A short-circuit condition with another active output channel.	Verify that the cabling is operational and properly connected.
X indicates the LED state can be either ON or OFF. 1. Blinking: 500 ms ON / 500 ms OFF. 2. Flickering: 50 ms ON / 50 ms OFF.							

Section 12.6
BMXSRA0405 Digital Relay Output Diagnostics

Introduction

This section describes diagnostic tools available for the BMXSRA0405 safety digital relay output module.

What Is in This Section?

This section contains the following topics:

BMXSRA0405 DDDT Diagnostics

Introduction

The BMXSRA0405 safety digital output relay module provides the following diagnostics using its `T_U_DIS_SIS_OUT_4` *(see page 122)* device DDT elements:

- output contact diagnostics
- internal error detection

Output Contact Diagnostics

Depending on the application number that has been configured for the module, the module can automatically test its ability to toggle the output contact states (from ON to OFF, or OFF to ON) for a time too short to cause an actuator response. If the channels does not successfully toggle between energized and de-energized, the `CH_HEALTH` bit in the `T_U_DIS_SIS_CH_ROUT` *(see page 124)* DDDT structure is set to 0, indicating it is not operational.

NOTE: Application numbers 2, 4, 6, and 8 perform this automatic signal test. Application numbers 1, 3, 5, an 7 do not, and therefore require a daily manual transition of the output channel state to confirm its operability.

Output Command Diagnostics (Internal Error Detection)

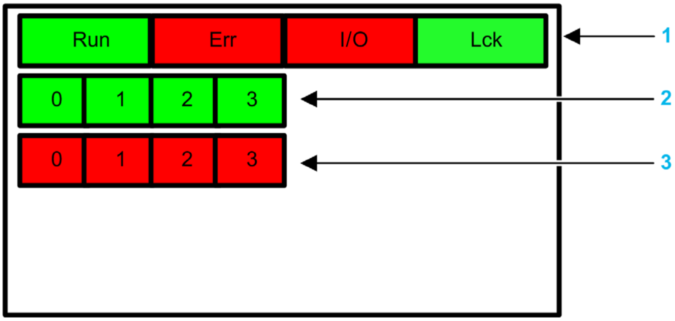
The relay command is processed using two separate, parallel circuits. The values of the circuits are compared. If the compared values are different, the channel is determined to be non-operational and the `IC` bit in the `T_U_DIS_SIS_CH_ROUT` DDDT structure is set to 1.

Refer to the architecture diagram *(see page 142)* for the BMXSRA0405 safety digital output relay module for a visual presentation of this process.

BMXSRA0405 Digital Relay Output LED Diagnostics

LED Panel

The BMXSRA0405 digital relay output module presents the following LED panel on its front face:



1 Module state LEDs
2 Channel state LEDs
3 Channel detected error LEDs

NOTE:

- When a channel error is detected, the corresponding LED remains ON until the underlying condition is resolved.
- Because the relay output module has only four channels, LEDs in positions 4...7 are not used and are never powered on.

Module Diagnostics

Use the four LEDs at the top of the LED panel to diagnose the condition of the BMXSRA0405 digital relay output module:

Module LEDs				Module State	Recommended Response
Run	Err	I/O	LCK		
Blinking[1]	Blinking[1]	Blinking[1]	Blinking[1]	Auto-test at power-on.	–
Blinking[1]	ON	Blinking[1]	Blinking[1]	Auto-test at power-on has detected an internal error on output channels.	–
OFF	ON	OFF	OFF	Internal error detected.	Replace the module if the condition persists.
OFF	Blinking[1]	OFF	X	Non-configured I/O module.	Configure the module via the CPU.
ON	Blinking[1]	OFF	X	No communication between CPU and module.The module is in Fallback state.	Verify that: ● The CPU is an M580 safety CPU and that it is operational. ● The backplane is operational (if I/O module is on main rack). ● The cable between the CPU and I/O module is operational and properly connected (if the I/O module is on an extended or remote rack).
ON	Flickering[2]	OFF	OFF	No communication between CPU and module. The module is in Fallback state (or in Reset if module has never been operating normally).	Debug the condition using the DDDT variables *(see page 122)* for the I/O module instance.
ON	Flickering[2]	OFF	ON	Communication not safe and configuration locked. The module is in Fallback state (or in Reset if module has never been operating normally).	● Verify the locked configuration in the module is equal to the module configuration stored in application in the CPU as configured using Control Expert. ● Debug the condition using the DDDT variables *(see page 122)* for the I/O module instance.

X indicates the LED state can be either ON or OFF.
1. Blinking: 500 ms ON / 500 ms OFF.
2. Flickering: 50 ms ON / 50 ms OFF.

Module LEDs				Module State	Recommended Response
Run	Err	I/O	LCK		
ON	ON	OFF	X	Internal error detected on the output channel.	Replace the module if the condition persists.
ON	OFF	OFF	OFF	Communication with CPU is safe and the configuration is unlocked.	–
ON	OFF	OFF	ON	Communication with CPU is safe and the configuration is locked.	–
X indicates the LED state can be either ON or OFF. 1. Blinking: 500 ms ON / 500 ms OFF. 2. Flickering: 50 ms ON / 50 ms OFF.					

Channel Diagnostics

Use all the LEDs on the BMXSRA0405 digital relay output module to diagnose channel status:

Module LEDs				Channel LEDs		Channel State	Recommended Response
Run	Err	I/O	LCK	Channel State (LED 0...3)	Detected Error (LED 0...3)		
ON	OFF	OFF	X	ON	OFF	Output relay is closed.	–
ON	OFF	OFF	X	OFF	OFF	Output relay is open.	–
ON	ON	OFF	X	OFF	ON	The output relay is not operational.	Replace the module if the condition persists.
X indicates the LED state can be either ON or OFF.							

Chapter 13
Operating an M580 Safety System

Introduction

This chapter provides information on how to operate an M580 safety system.

What Is in This Chapter?

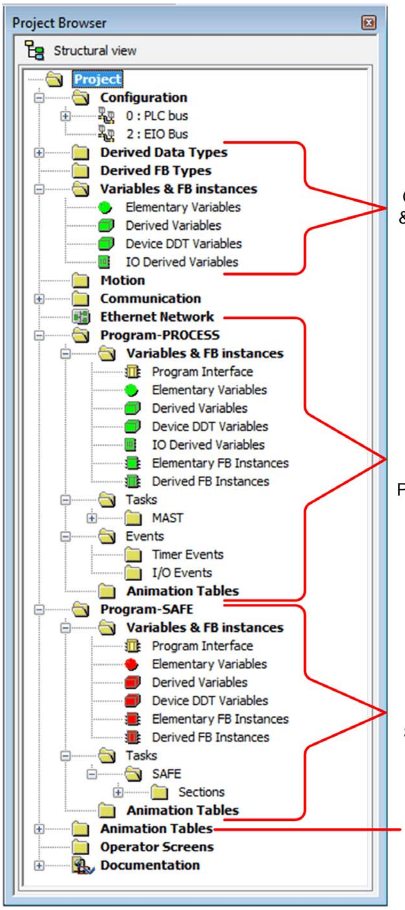
This chapter contains the following sections:

Section 13.1
Process, Safety and Global Data Areas in Control Expert

Data Separation in Control Expert

Data Areas in Control Expert

The **Structural View** of the **Project Browser** displays the separation of data in Control Expert. As shown below, each data area has its own data editor and collection of animation tables:



Global Data Editor
& Types Definitions

Process Data Editor,
MAST, FAST, AUX0,
AUX1 Tasks,
Process Animation Tables

Safety Data Editor,
Safe Task,
Safety Animation Tables

Global Animation Tables

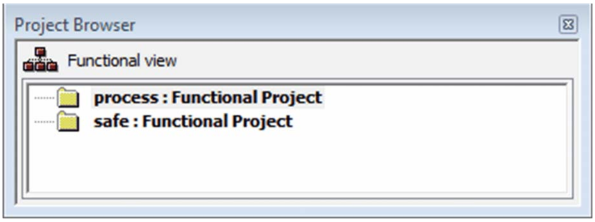
Looking at the **Project Browser** you will notice that:

- The safe area contains a Safety Data Editor, safety logic, and function block instances used by the SAFE task. However, note that:
  - I/O events, timer events, and sub-routines are not supported in a safety program.
  - IODDT variables are not supported by the SAFE task, and are not included in the safe area.
  - Red icons are used to indicate the SAFE parts of the program.
- The process area contains a Process Data Editor, process logic, and function block instances used by the non-safe tasks (i.e., MAST, FAST, AUX0 and AUX1).
- The global area contains a Global Data Editor, derived data and function block types instantiated in the process and safety programs.

NOTE: The term *Global Data* used in this topic refers to the application wide – or global – scope of data objects in a safety project. It does not refer to the Global Data service that is supported by many Schneider Electric Ethernet modules.

Project Browser in Functional View

The **Functional View** of the Control Expert. **Project Browser** for an M580 safety system presents two functional projects – one for the process namespace, one for the safe namespace:



Project Browser

Functional view

process : Functional Project

safe : Functional Project

Management of each functional project in an M580 safety system is the same as managing a project in the functional view of an M580 non-safety system, except for animation tables and code sections.

Effect on Structural View:

When you add a code section or animation table to a functional project, it becomes associated with the namespace associated with that functional project. Adding a code section or animation table to:

- the **process : Functional Project** adds it to the process namespace of the project in structural view.
- the **safe : Functional Project** adds it to the safe namespace of the project in structural view.

Availability of Language and Task Selections:

When you create a new code section for a functional project (by selecting **Create → New Section...**), the available **Language** and **Task** selections depend on the functional project:

When you create a new code section for a functional project (by selecting **Create → New Section...**), the available **Language** and **Task** selections depend on the associated functional project:

Functional Project	Available Languages and Tasks	
	Languages[1]	Tasks[2]
process : Functional Project	● IL ● FBD ● LD ● LL984 segment ● SFC ● ST	● MAST ● FAST ● AUX0 ● AUX1
safe : Functional Project	● FBD ● LD	● SAFE
1. Selected in the **General** tab of the new section dialog. 2. Selected in the **Localization** tab of the new section dialog. The MAST task is available by default. Other sections are available for selection only after they have been created in the process program.		

Color Coded Icons

To help you distinguish between the process and safe parts of the project, red colored icons are used to identify the safe parts of your application.

Section 13.2
Operating Modes, Operating States, and Tasks

Introduction

This section describes the operating modes, operating states, and tasks supported by the M580 safety PAC.

What Is in This Section?

This section contains the following topics:

M580 Safety PAC Operating Modes

Two Operating Modes

The M580 safety PAC presents two operating modes:

- Safety mode: the default operating mode used for safety operations.
- Maintenance mode: an optional operating mode that can be entered temporarily to debug and modify the application program, or change the configuration.

Control Expert Safety software is the exclusive tool you can use to manage operating mode transitions.

NOTE: The operating mode setting of a Hot Standby safety PAC – either safety mode or maintenance mode – is not included in the transfer of an application from the primary PAC to the standby PAC. On a switchover, when a safety PAC switches from standby PAC to primary PAC, the operating mode is automatically set to safety mode.

Safety Mode and its Limitations

Safety mode is the default mode of safety PAC. When the safety PAC is powered ON with a valid application present, the PAC enters safety mode. Safety mode is used to control execution of the safety function. You can upload, download, run and stop the project in safety mode.

When the M580 safety PAC is operating in safety mode, the following functions are **not** available:

- Downloading a changed configuration from Control Expert to the PAC.
- Editing and/or forcing safety variable values and safety I/O states.
- Debugging application logic, by means of breakpoints, watchpoints, and step-through code execution.
- Using animation tables or UMAS requests (for example, from an HMI) to write to safety variables and safety I/O.
- Changing the configuration settings of safety modules via CCOTF. (Note that the use of CCOTF for non-interfering modules is supported.)
- Performing online modification of the safety application.
- Using link animation.

NOTE: In safety mode, all safety variables and safety I/O states are read-only. You cannot directly edit the value of a safety variable.

You can create a global variable, and use it to pass a value between a linked process (non-safe) variable and a linked safety variable using the interface tabs of the Process Data Editor and the Safety Data Editor. After the link is made, the transfer is executed as follows:

- At the beginning of each SAFE task, the non-safe variable values are copied to the safe variables.
- At the end of the SAFE task, the safe output variable values are copied to the non-safe variables.

Maintenance Mode Functionality

Maintenance mode is comparable to the normal mode of a non-safety M580 CPU. It is used only to debug and tune the application SAFE task. Maintenance mode is temporary because the safety PAC automatically enters safety mode if communication between Control Expert and the PAC is lost, or upon the execution of a disconnect command. In maintenance mode, persons with the appropriate permissions can both read and write to safety variables and safety I/O that are configured to accept edits.

In maintenance mode, dual execution of SAFE task code is performed, but the results are not compared.

When the M580 safety PAC is operating in maintenance mode, the following functions are available:

- Downloading a changed configuration from Control Expert to the PAC.
- Editing and/or forcing safety variable values and safety I/O states.
- Debugging application logic, by means of breakpoints, watchpoints, and step-through code execution.
- Using animation tables or UMAS requests (for example, from an HMI) to write to safety variables and safety I/O.
- Changing the configuration via CCOTF.
- Performing online modification of the safety application.
- Using link animation.

In maintenance mode, the SIL level of the Safety PLC is not guaranteed.

⚠ WARNING

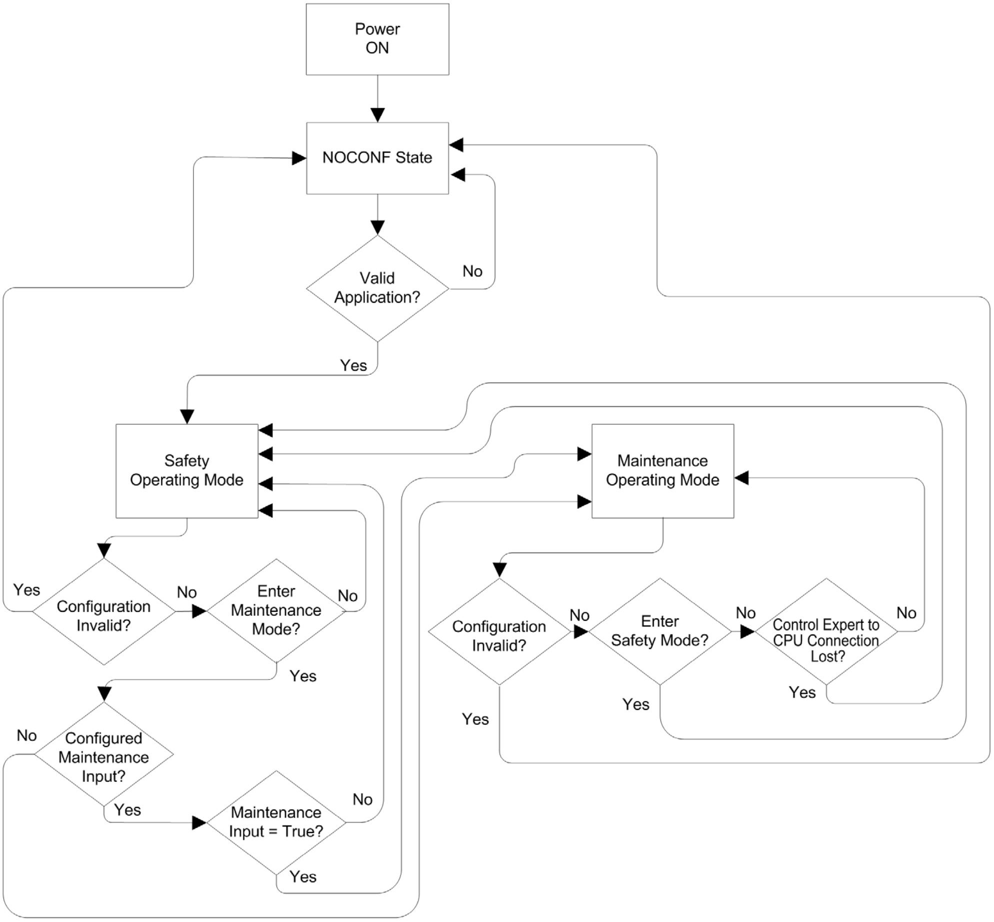
LOSS OF THE SAFETY INTEGRITY LEVEL

While the safety PAC is in maintenance mode, you need to take appropriate measures to ensure the safe state of the system.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Operating Mode Transitions

The following diagram shows how the M580 safety PAC enters, then transitions between safety mode and maintenance mode:



When switching between safety mode and maintenance mode:

- It is OK to switch from maintenance mode to safety mode with forcing ON. In this case, the forced variable value or I/O state remains forced after the transition until another transition from safety to maintenance mode occurs.
- The transition from maintenance mode to safety mode can be accomplished in the following ways:
  - Manually, by menu or toolbar command in Control Expert.
  - Automatically, by the safety PAC, when communication between Control Expert and the PAC is lost for about 50 seconds.

- The maintenance input function, when it is configured, operates as a check on the transition from safety mode to maintenance mode. The maintenance input function is configured in Control Expert in the CPU **Configuration** tab by:
  - Selecting the **Maintenance Input** setting, and
  - Entering the topological address of an input bit (%I) for a non-interfering digital input module on the local rack.

  When the maintenance input is configured, the transition from safety mode to maintenance mode takes into account the state of the designated input bit (%I). If the bit is set to 0 (false), the PAC is locked in safety mode. If the bit is set to 1 (true), a transition to maintenance mode can occur.

Switching Between Safety Mode and Maintenance Mode in Control Expert

Switching the safety PAC from maintenance mode to safety mode is not possible if:

- The PAC is in debug mode.
- A breakpoint is activated in a SAFE task section.
- A watchpoint is set in a SAFE task section.

When debug mode is not active, no SAFE task breakpoint is activated, and no SAFE task watchpoint is set, you can manually activate a transition between safety mode and maintenance mode, as follows:

- To switch from safety mode to maintenance mode, either:
  - Select **PLC** → **Maintenance**, or
  - Click the toolbar button.
- To switch from maintenance mode to safety mode, either:
  - Select **PLC** → **Safety**, or
  - Click the toolbar button.

NOTE: Entering and exiting safety mode events are logged in the SYSLOG server in the CPU.

Determining the Operating Mode

You can determine the current operating mode of an M580 safety PAC using either the **SMOD** LEDs of the CPU and coprocessor, or Control Expert.

When the **SMOD** LEDs of the CPU and coprocessor are:

- *Flashing* ON, the PAC is in maintenance mode.
- *Solid* ON, the PAC is in safety mode.

When Control Expert is connected to the PAC, it identifies the operating mode of the M580 safety PAC in several places:

- System words %SW12 (coprocessor) and %SW13 (CPU) *(see page 396)* together indicate the operating mode of the PAC, as follows:
  - if %SW12 is set to 16#A501 (hex) and %SW13 is set to 16#501A (hex), the PAC is in maintenance mode.
  - if either or both of these system words is set to 16#5AFE (hex), the PAC is in safety mode.
- Both the **Task** and **Information** sub-tabs of the CPU **Animation** tab display the operating mode of the PAC.
- The task bar, at the bottom of the Control Expert main window, indicates the operating mode as either MAINTENANCE or SAFETY.

M580 Safety PAC Operating States

Operating States

The M580 safety PAC operating states are described below.

NOTE: For a description of the relationship between M580 safety PAC operating states and M580 Hot Standby PAC operating states, refer to the document *Modicon M580 Hot Standby, System Planning Guide for Frequently Used Architectures* and the topics *Hot Standby System States* and *Hot Standby State Assignments and Transitions*.

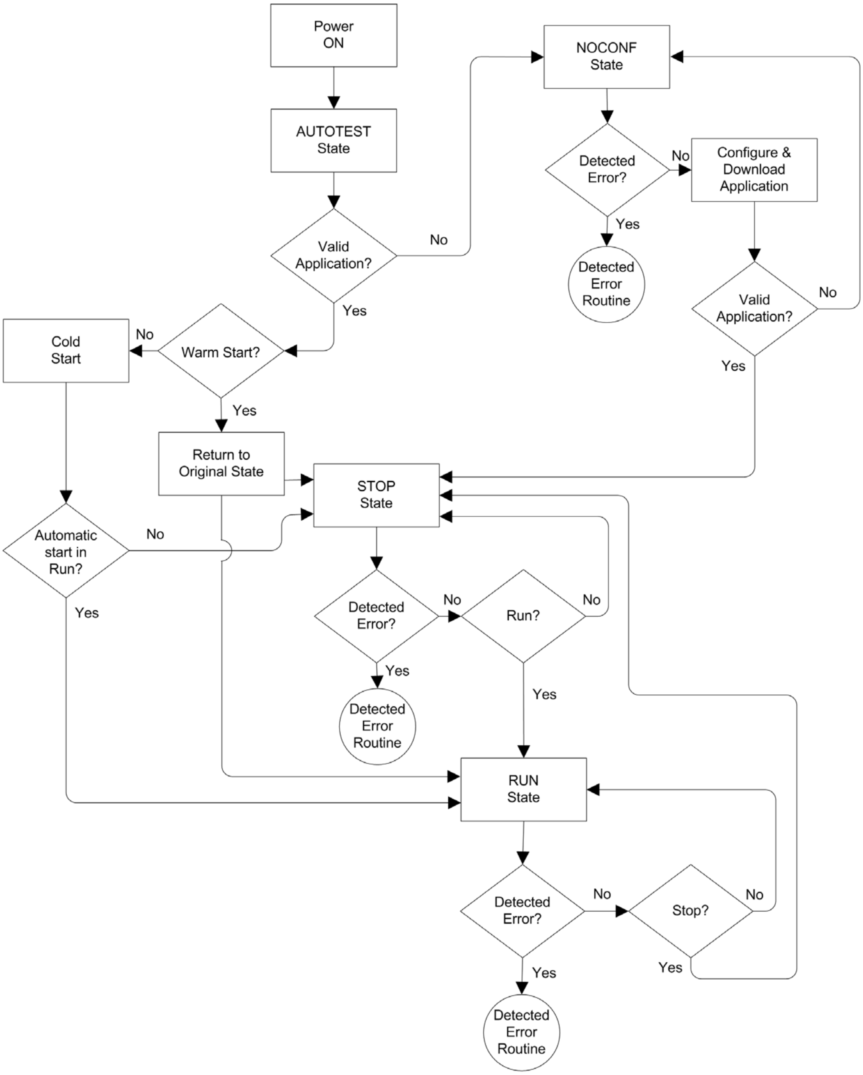
Operating State	Applies to...	Description
AUTOTEST	PAC	The CPU is executing internal self-tests. **NOTE:** If extended racks are connected to the main local rack and line terminators are not plugged into the unused connectors on the rack extender module, the CPU remains in AUTOTEST after the self-tests have completed.
NOCONF	PAC	The application program is not valid.
STOP	PAC or Task	The PAC has a valid application and no error is detected, but operation has stopped because: ● At startup **Automatic start in Run** is not set (safety mode *(see page 254)*). ● Execution stopped by execution of a STOP command (safe *(see page 254)* or maintenance *(see page 255)* mode). ● Breakpoints were set in maintenance mode, then the connection between Control Expert and the CPU was lost for more than 50 seconds. The CPU reads the inputs associated with each task, but does not refresh outputs, which enter their fallback state. The CPU can be restarted when you are ready. **NOTE:** Issuing a STOP command in Control Expert stops all tasks. The STOP event is recorded in the SYSLOG server of the CPU.
HALT	Task	The M580 safety PAC presents two independent HALT states: ● Process HALT applies to the non-SAFE tasks (MAST, FAST, AUX0, and AUX1). When any process task enters the HALT state, all other process tasks also enter the HALT state. The SAFE task is not affected by a process HALT condition. ● SAFE HALT applies only to the SAFE task. Process tasks are not affected by a SAFE HALT condition. In each case, task operations are halted because an unexpected blocking condition has been encountered, resulting in a recoverable *(see page 214)* condition. The CPU reads the inputs associated with each halted task, but does not refresh outputs, which are in fallback state.

Operating State	Applies to...	Description
RUN	PAC or Task	With a valid application and no error detected, the CPU reads the inputs associated with each task, executes the code associated with each task, and refreshes the associated outputs. ● in safety mode *(see page 254)*: the safety function is performed, and all limitations are applied. ● in maintenance mode *(see page 255)*: the PAC operates like any non-safety CPU. Dual execution of SAFE task code is performed, but the results are not compared. **NOTE:** Issuing a RUN command in Control Expert starts all tasks. The RUN event is recorded in the SYSLOG server of the CPU
WAIT	PAC	The CPU is in a transitory state while it backs up data when a power down condition is detected. The CPU starts again only when power is restored and the supply reserve is replenished. Because WAIT is a transitory state, it may not be visible. The CPU performs a warm restart *(see page 266)* to exit the WAIT state.
ERROR	PAC	The CPU is stopped because an non-recoverable *(see page 211)* hardware or system error is detected. The ERROR state triggers the safety function *(see page 17)*. When the system is ready to be restarted, perform a cold start *(see page 266)* of the CPU to exit the ERROR state, either by cycling power or performing a RESET.
OS DOWNLOAD	PAC	A CPU or COPRO firmware download is in progress.

Refer to the *M580 CPU LED Diagnostics (see page 216)* and *M580 Safety Coprocessor LED Diagnostics (see page 216)* topics for information on the operating states of the PAC.

Operating State Transitions

The transitions between the several states in an M580 safety PAC are described, below:



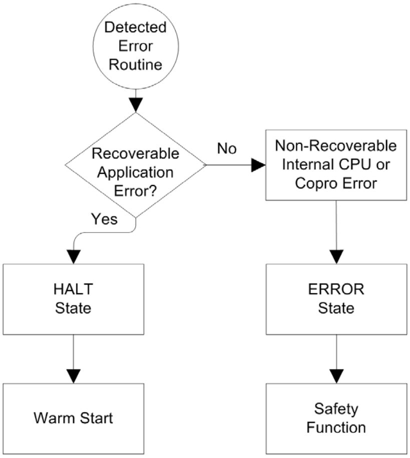
Refer to the topic *Detected Error Processing (see page 262)* for information on how the safety system handles detected errors.

Detected Error Processing

The M580 safety PAC handles the following kinds of CPU detected errors:

- Recoverable application detected errors: These events cause the related task(s) to enter the HALT state.
  NOTE: Because the MAST, FAST, and AUX tasks operate in the same memory area, an event that causes one of these tasks to enter HALT state causes the other non-safe tasks also to enter HALT state. Because the SAFE task operates in a separate memory area, the non-safe tasks are not affected if the SAFE task enters HALT state.
- Non-recoverable application detected errors: Internal CPU or coprocessor detected errors: These events cause the PAC to enter the ERROR state. The safety function is applied to the affected portion of the safety loop.

The logic of the detected error handling process is described below:


Detected Error Routine
Recoverable Application Error?
No
Yes
Non-Recoverable Internal CPU or Copro Error
HALT State
ERROR State
Warm Start
Safety Function

The impact of detected errors on individual tasks is described below:

Detected Error Type	Task State			
	FAST	SAFE	MAST	AUX
FAST task watchdog overrun	HALT	RUN[1]	HALT	HALT
SAFE task watchdog overrun	RUN	HALT[2]	RUN	RUN
MAST task watchdog overrun	HALT	RUN	HALT	HALT
AUX task watchdog overrun	HALT	RUN	HALT	HALT

1.Because FAST task has a higher priority than the SAFE task, delay of the FAST task may cause the SAFE task to enter HALT or ERROR state instead of RUN state.
2. The ERROR and HALT states on the SAFE task causes the safe outputs to be set to their user configurable state (fallback or maintain).
3. The safety watchdog is set equal to 1.5 times the SAFE task watchdog.

Detected Error Type	Task State			
	FAST	SAFE	MAST	AUX
CPU dual code execution detected error	RUN	HALT[2]	RUN	RUN
Safety watchdog overrun[3]	ERROR	ERROR[2]	ERROR	ERROR
CPU internal detected error	ERROR	ERROR[2]	ERROR	ERROR
1.Because FAST task has a higher priority than the SAFE task, delay of the FAST task may cause the SAFE task to enter HALT or ERROR state instead of RUN state. 2. The ERROR and HALT states on the SAFE task causes the safe outputs to be set to their user configurable state (fallback or maintain). 3. The safety watchdog is set equal to 1.5 times the SAFE task watchdog.				

Task Bar Safety Status Viewer

When Control Expert is connected to the M580 safety PAC, the task bar includes a field describing the combined operating states of the SAFE task and the process tasks (MAST, FAST, AUX0, AUX1), as follows:

Process task(s) state	SAFE task state	Message
STOP (all process tasks in STOP state)	STOP	STOP
STOP (all process tasks in STOP state)	RUN	RUN
STOP (all process tasks in STOP state)	HALT	SAFE HALT
RUN (at lease one process task in RUN state)	STOP	RUN
RUN (at lease one process task in RUN state)	RUN	RUN
RUN (at lease one process task in RUN state)	HALT	SAFE HALT
HALT	STOP	PROC HALT
HALT	RUN	PROC HALT
HALT	HALT	HALT

Start Up Sequences

Introduction

The M580 safety PAC can enter the start-up sequence in the following circumstances:

- At initial power-up.
- In response to a power interruption.

Depending on the type of task, and the context of the power interruption, the M580 safety PAC may perform either a cold start *(see page 266)* or a warm start *(see page 266)* when power is restored.

Initial Start-Up

At initial start-up, the M580 safety PAC performs a cold start. All tasks, including both the SAFE task and the non-safe (MAST, FAST, AUX0, AUX1) tasks, enter the STOP state unless **Automatic start in RUN** is enabled, in which case all tasks enter the RUN state.

Start-Up after a Power Interruption

The M580 safety power supply provides a power reserve that continues to supply all modules on the rack for up to 10 ms in case of a power interruption. When the power reserve is depleted, the M580 safety PAC performs a complete power cycle.

Before powering down the system, the safety CPU stores the following data that defines the operating context at power down:

- Date and time of the power down (stored in %SW54...%SW58).
- State of each task.
- State of event timers.
- Values of running counters.
- Signature of the application.
- Application data (current values of application variables)
- Application check sum.

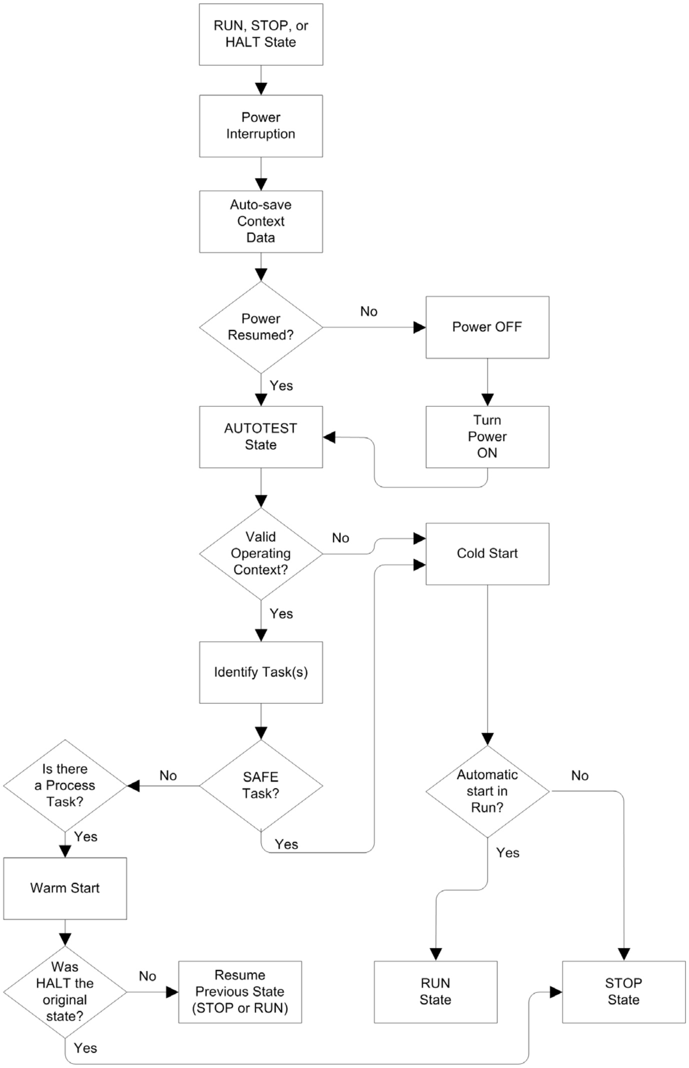
After power down, the start-up can be either automatic (if power was restored before completion of the shut-down) or manual (if not).

Next, the M580 safety PAC performs self-tests and checks the validity of the operating context data that was saved at power down, as follows:

- The application check sum is verified.
- The SD memory card is read to confirm that is contains a valid application.
- If the application in the SD memory card is valid, the signatures are checked to confirm they are identical.
- The saved application signature is verified by comparing it to the stored application signature.

If the operating context is valid, the non-safe tasks perform a warm start. If the operating context is not valid, the non-safe tasks perform a cold start. In either case, the SAFE task performs a cold start.

This start-up sequence after a power interruption is presented, below:



Cold Start

A cold start causes all tasks, including both the SAFE task and the non-safe (MAST, FAST, AUX0, AUX1) tasks, enter the STOP state, unless **Automatic start in RUN** is enabled, in which case all tasks enter the RUN state.

A cold start performs the following operations:

- Application data (including internal bits, I/O data, internal words, and so forth) are assigned the initial values defined by the application.
- Elementary functions are set to their default values.
- Elementary function blocks and their variables are set to their default values.
- System bits and words are set to their default values.
- Initializes all forced variables by applying their default (initialized) values.

A cold start can be executed for data, variables and functions in the process namespace by selecting **PLC** → **Init** in Control Expert *(see page 283)*, or by setting the system bit %S0 (COLDSTART) to 1. The %S0 system bit has no effect on the data and functions belonging to the safe namespace.

NOTE: Following a cold start, the SAFE task cannot start until after the MAST task has started.

Warm Start

A warm start causes each process task – including the (MAST, FAST, AUX0, AUX1) tasks – to re-enter its operating state as of the time of the power interruption. By contrast, a warm start causes the SAFE task to enter the STOP state, unless **Automatic start in RUN** is selected.

NOTE: If a task was in the HALT state or in breakpoint at the time of power interruption, that task enters the STOP state after the warm start.

A warm start performs the following operations:

- Restores the last held value to process namespace variables.
- Initializes safe namespace variables by applying their default (initialized) values.
- Initializes all forced variables by applying their default (initialized) values.
- Restores the last held value to application variables.
- Sets %S1 (WARMSTART) to 1.
- Connections between the PAC and CPU are reset.
- I/O modules are re-configured (if necessary) using their stored settings.
- Events, the FAST task, and the AUX tasks are disabled.
- The MAST task is re-started from the beginning of the cycle.
- %S1 is set to 0 at the conclusion of the first execution of the MAST task.
- Events, the FAST task, and the AUX tasks are enabled.

If a task was in the process of execution at the time of power interruption, after warm start the task resumes execution at the beginning of the task.

⚠ WARNING

UNEXPECTED EQUIPMENT OPERATION

You are responsible to confirm that selecting **Automatic start in RUN** is compliant with the correct behavior of your system. If it is not, de-activate this feature.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

M580 Safety PAC Tasks

Introduction

An M580 safety PAC can execute single-task and multi-task applications. Unlike a single-task application which only executes the MAST task, a multi-task application defines the priority of each task.

The M580 safety PAC supports the following tasks:

- FAST
- SAFE
- MAST
- AUX0
- AUX1

Task Characteristics

The tasks supported by the M580 safety PAC present the following task characteristics:

Task Name	Priority	Time Model	Period Range	Default Period	Watchdog Range	Default Watchdog
FAST	1	Periodic	1...255 ms	5 ms	10...500 ms[2]	100 ms[2]
SAFE	2	Periodic	10...255 ms	20 ms	10...500 ms[2]	250 ms[2]
MAST[1]	3	Cyclic[4] or Periodic	1...255 ms	20 ms	10...1500 ms[2]	250 ms[2]
AUX0[3]	4	Periodic	10...2550 ms	100 ms	100...5000 ms[2]	2000 ms[2]
AUX1[3]	5	Periodic	10...2550 ms	200 ms	100...5000 ms[2]	2000 ms[2]

1. MAST task is required and cannot be deactivated.
2. If CCOTF is enabled (by selecting **Online modification in RUN or STOP** in the **Configuration** tab of the CPU properties dialog), the minimum **Watchdog** setting is 64 ms.
3. Supported by standalone BMEP58•040S safety PACs. Not supported by BMEH58•040S safety Hot Standby PACs.
4. Standalone BMEP58•040S safety PACs support both cyclic and periodic time models. BMEH58•040S safety Hot Standby PACs support only the periodic time model.

Task Priority

M580 Safety PACs execute pending tasks according to their priority. When a task is running, it can be interrupted by another task with a higher relative priority. For example, when a periodic task is scheduled to execute its code, it would interrupt a lower priority task, but would wait until the completion of a higher priority task.

Task Configuration Considerations

All the non-safe tasks (MAST, FAST, AUX0, and AUX1) operate in the same memory area, while the SAFE task operates in its own, separate memory area. As a result:

- If one non-safe task exceeds its watchdog, all non-safe tasks enter HALT state, while the SAFE task continues to be operational.
- If the SAFE task exceeds its watchdog, only the SAFE task enters HALT state, while the non-safe tasks continue to be operational.

When creating and configuring tasks for your application, consider the following task features:

SAFE task:

Design this periodic task to execute only safety-related code sections for safety I/O modules. Because the SAFE task is assigned a lower priority than the FAST task, execution of the SAFE task may be interrupted by the FAST task.

Define the maximum execution time for the SAFE task by setting the appropriate watchdog value. Consider the time required to execute code and to read and write safe data. If the time to execute the SAFE task exceeds the watchdog setting, the SAFE task enters HALT state, and the %SW125 system word displays the detected error code 16#DEB0.

NOTE:

- Because FAST task has a higher priority than the SAFE task, you may want to include a component for FAST task delay time in the SAFE task watchdog setting.
- If the overrun of the SAFE task execution equals the “Safety watchdog” (which is a value equal to one and one-half times the SAFE task watchdog setting), the CPU and Copro will enter the ERROR state and the safety function will be applied.

MAST task:

This task can be configured as either cyclic or periodic. When operating in cyclic mode, define a maximum execution time by inputting an appropriate MAST watchdog value. Add a small time interval to this value at the end of each cycle to allow for the execution of other lower priority system tasks. Because the AUX tasks carry a lower priority than MAST, if this time slot is not provided, the AUX tasks may never be executed. Consider adding a time interval equal to 10% of cycle execution time, with a minimum of 1 ms and a maximum of 10 ms.

If the time to execute a cyclic MAST task exceeds the watchdog setting, the MAST task and all other non-SAFE tasks enter HALT state, and the %SW125 system word displays the detected error code 16#DEB0.

When operating in periodic mode, it is possible for the MAST task to exceed its period. In that case the MAST task runs in cyclic mode and the system bit %S11 is set.

FAST task:

The purpose of this periodic task is to execute a high-priority part of the application. Define a maximum execution time by setting the FAST watchdog value. Because the FAST task interrupts execution of all other tasks – including the SAFE task – it is recommended to configure the execution time of the FAST task to be as short as possible. A FAST task watchdog value not much greater than the FAST period is recommended.

If the time to execute the FAST task exceeds the watchdog setting, the FAST task and all other non-SAFE tasks enter HALT state, and the %SW125 system word displays the detected error code 16#DEB0.

AUX tasks:

AUX0 and AUX1 are optional periodic tasks. Their purpose is to execute a low priority part of the application. The AUX tasks are executed only after execution of the MAST, SAFE and FAST tasks has finished.

Define a maximum execution time for the AUX tasks by setting the appropriate watchdog value. If the time to execute an AUX task exceeds the watchdog setting, the AUX task and all other non-SAFE tasks enter HALT state, and the %SW125 system word displays the detected error code 16#DEB0.

Section 13.3
Building an M580 Safety Project

What Is in This Section?

This section contains the following topics:

Building an M580 Safety Project

Building an M580 Safety Project

The Control Expert for Safety **Build** menu presents three different build commands, and a Safe Signature command, as follows:

Command	Description
Build Changes	Compiles only the changes that have been made to the application program since the previous build command, and adds them to the previously generated application program.
Rebuild All Project	Re-compiles the entire application program, replacing the previously generated build of the application program. **NOTE:** For M580 safety I/O modules, this command does not generate a new module unique identifier (MUID) value. Instead, the previously generated MUID value is retained.
Renew Ids & Rebuild All	Re-compiles the entire application program, replacing the previously generated build of the application program. **NOTE:** ● Execute this command only when the safety I/O modules are unlocked *(see page 280)*. ● For M580 safety I/O modules, this command generates a new module unique identifier (MUID) value and replaces the existing MUID value with the new value.
Update Safe Signature	Use this to manually generate a SourceSafeSignature *(see page 273)* value for the safe application. **NOTE:** This command is enabled only when the **General** → **Build Settings** → **Safe Signature management** parameter is set to **On user request**.

Safe Signature

Introduction

M580 safety PACs - both standalone and Hot Standby - include a mechanism for producing an SHA256 algorithmic fingerprint of the safe application: the SourceSafeSignature. When transferring the application from the PC to the PAC, Control Expert compares the SourceSafeSignature in the PC with the SourceSafeSignature in the PAC to determine if the safe application in the PC is the same as, or different from the safe application in the PAC.

The safe signature feature is optional. Generating a SourceSafeSignature can be a time-consuming process, depending on the size of the safe application. Using the safe signature management options, you can generate a SourceSafeSignature value that creates an algorithmic value for your safe application

- on every build, or
- only when you want to manually generate a SourceSafeSignature and add it to the most recent build, or
- not at all

Actions that Change the SourceSafeSignature

Both configuration edits and variable value changes can cause the SourceSafeSignature to change.

Configuration changes: The following configuration actions lead to a signature change:

Device	Action
Safety CPU	Change CPU reference via **Replace Processor...**
	Change CPU version via **Replace Processor...**
	Edit any parameter on the CPU **Configuration** or **Hot Standby** configuration tabs.
	Edit any parameter on any tab of the CPU Ethernet Communicator Head (**Security**, **IP Config**, **RSTP**, **SNMP**, **NTP**, **ServicePort**, **Safety** ..).
Safety Coprocessor	Not applicable, as the coprocessor is not configurable.
Other Safety Module	**Add**/**Delete**/**Move** a module, either: ● Directly (via command) ● Indirectly (for example, by replacing an 8-slot Ethernet backplane - with a safety module in slot 7 - with a 4-slot Ethernet backplane, thereby deleting a module)
	Edit of any safety module parameter, located on the **Configuration** tab (for example **Short circuit to 24V detection**, **Open wire detection**) and on the left pane of the editor (for example **Function**, **Fallback**).
	Modification of module ID via **Renew Ids and Rebuild All** command.
	Modification of Device DDT instance name.

Device	Action
CIP Safety Module	**Add**/**Delete** a module.
	Modification of any CIP Safety module parameter in either the CIP Safety device DTM editor, or the **Device List** of the CPU master DTM editor.
	Modification of Device DDT instance name.
Safety Power Supply	**Add**/**Delete** a safety power supply.
Other Safety-Related Equipment	Modification of any topological address of equipment supporting a safety device, for example: ● Move a rack containing a safety device. ● Move a bus or drop containing a safety device.

Value Changes: Except as noted, the following items are included in the SourceSafeSignature computation. A change to their values causes a SourceSafeSignature change:

Type	Items
Program	SAFE task and related code sections.
Variables	All safe area variables and their attributes.
DDTs	Each safe DDT attribute, except date and version attributes.
	The variables inside each DDT, including their attributes.
	The safe DDTs, even if they are not used in the safe application.
DFBs	Each safe DFB attribute, except date and version attributes.
	The variables inside each DFB, including their attributes.
	The safe DFBs, even if they are not used in the safe application.
Safe Scope Settings	All **Project Settings** for **Scope** = safe.
1. These variables are not exported, but any change to their values change the configuration partial signature.	

Type	Items
Common Scope Settings	The following **Project Settings** for **Scope** = common:
	Variables ● Allow leading digits ● Character set ● Allow usage of EBOOL edge ● Allow INT/DINT in place of ANY_BIT ● Allow bit extraction of INT, WORD and BYTE ● Directly represented array variables ● Enable fast scanning for trending ● Force references initialization
	Program → **Languages** → **Common** ● Allow procedures ● Allow nested comments ● Allow multi assignment [a:=b:=c] (ST/LD) ● Allow empty parameters in non-formal call (ST/IL) ● Maintain output links on disabled EF (EN=0) ● Display complete comments of structure element
	Program → **Languages** → **LD** ● Single scan edge detection for EBOOL
	General → **Time**[1] ● Custom TimeZone ● Time Zone ● Time Offset ● Automatically adjust clock for daylight saving ❍ All START and END settings under Automatically adjust clock for daylight saving
1. These variables are not exported, but any change to their values change the configuration partial signature.	

Managing the SourceSafeSignature

The SourceSafeSignature is managed in Control Expert in the **Tools** → **Project Settings** window, by selecting **General** → **Build Settings**, then selecting one of the following **Safe Signature management** settings:

- **Automatic** (default): generates a new SourceSafeSignature every time a **Build** command is executed.
- **On user request**: generates a new SourceSafeSignature when the **Build** → **Update Safe Signature** command is executed.

NOTE: If you select **On user request**, Control Expert generates a SourceSafeSignature value of 0 on every build. If you do not execute the **Build** → **Update Safe Signature** command, you are electing not to use the Safe Signature feature.

Transferring an Application from the PC to the PLC

When you download an application from the PC to the PAC, Control Expert compares the SourceSafeSignature in the downloaded application with one in the PAC. Control Expert behaves as follows:

New Safe Signature	PAC Safe Signature	Control Expert Displays
Any	No application	Transfer confirmation
Any (except 0)	0	Transfer confirmation
0	0	Transfer confirmation
0	Any (except 0)	Transfer confirmation; Followed by a notice “This will reset the Safe Signature”; Followed by a new transfer confirmation
XXXX = YYYY[2]	YYYY	Transfer confirmation
XXXX ≠ YYYY[3]	YYYY	Transfer confirmation; Followed by a notice “This will modify the Safe Signature”; Followed by a new transfer confirmation

1. The value “0” indicates a SourceSafeSignature was not generated automatically or manually.
2. The safe application in the PC (XXXX) and the safe application in the PAC (YYYY) are EQUAL.
3. The safe application in the PC (XXXX) and the safe application in the PAC (YYYY) are DIFFERENT.

Viewing the SourceSafeSignature

When used, each SourceSafeSignature consists of a series of hexadecimal values, and can be very long, which makes direct readings and comparisons of the value very difficult for a human user. However, it is possible to copy a SourceSafeSignature value and paste it into an appropriate text tool to make comparisons. The SourceSafeSignature value can be found in the following Control Expert locations:

- **Properties of Project** → **Identification** tab *(see EcoStruxure™ Control Expert, Operating Modes)*: In the **Project Browser**, right click on **Project** and select **Properties**.
- **PLCScreen** → **Information** tab *(see EcoStruxure™ Control Expert, Operating Modes)*: In the **Project Browser**, navigate to **Project** → **Configuration** → **PLC bus** → **<CPU>**, right-click and select **Open**, then select the **Animation** tab.
- **PC < - - > PLC Comparison** dialog *(see EcoStruxure™ Control Expert, Operating Modes)*: Select this command from the **PLC** menu.
- **Transfer Project to PLC** dialog *(see EcoStruxure™ Control Expert, Operating Modes)*: Select this command from the **PLC** menu (or in the **PC < - - > PLC Comparison** dialog.

Comparing the SourceSafeSignature and the SAId

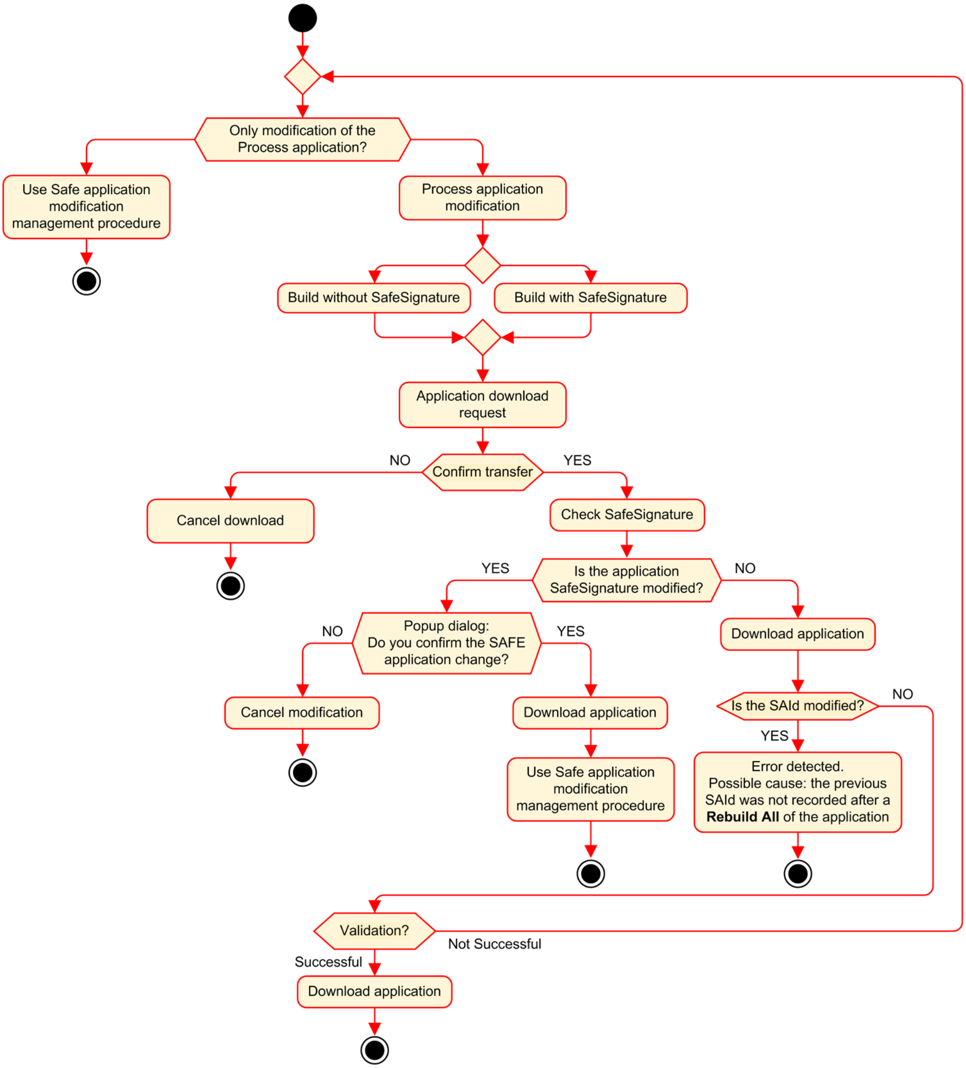
The SourceSafeSignature was introduced to provide an *a priori* verification that the safe application is unchanged. It is recommended to use this feature each time the process application is modified *(see page 278)* to avoid unintended modification of the safe application.

The SourceSafeSignature is a reliable mechanism, but is not sufficient for safety applications because the same source code may correspond to different binary (executable) codes, depending on the kind of build used after the last modification of the safe code.

The SAId *(see page 360)* can be evaluated only at run time. Its calculation is double executed and compared by both the CPU and the COPRO, based on the binary code that is executed by the safe application. Because the SAId is sensitive to all modifications, including those that may be introduced by a **Rebuild All** command after a build change, it is recommended that you use a **Rebuild All** command to generate a reference version of the safe application. This process *(see page 279)* lets you use any form of build (**Rebuild All**, **Build Changes** online or offline) for the process application changes without any change made to the SAId.

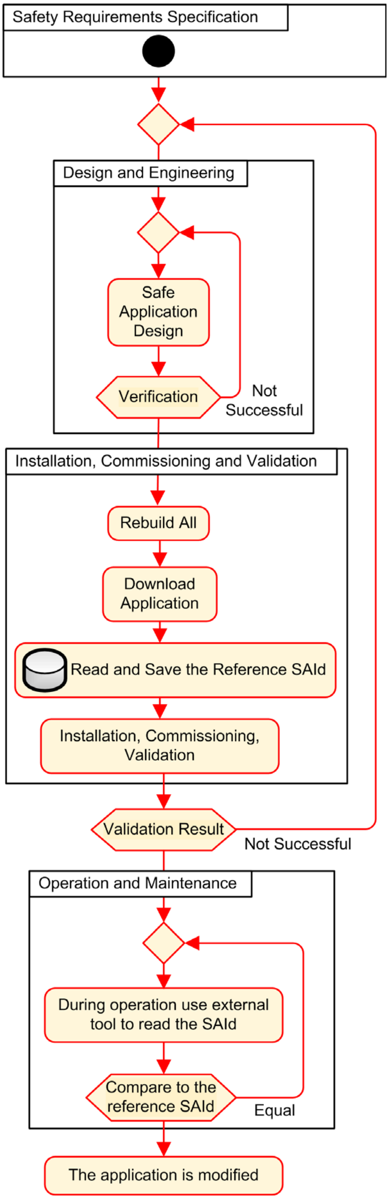
The SAId is the recommended method used to confirm that the safe application is the one that was validated. The SAId value is not automatically tested by the application. For this reason, it is recommended that you regularly verify the SAId by any convenient mean (for example, using Control Expert or an HMI) by reading the output of the S_SYST_STAT_MX function block or the content of system word %SW169 *(see page 396)*.

Modification of the Process Application Simplified Process



Only modification of the Process application?
Use Safe application modification management procedure
Process application modification
Build without SafeSignature
Build with SafeSignature
Application download request
NO
Confirm transfer
YES
Cancel download
Check SafeSignature
YES
Is the application SafeSignature modified?
NO
NO
Popup dialog: Do you confirm the SAFE application change?
YES
Download application
Cancel modification
Download application
Is the SAId modified?
NO
YES
Use Safe application modification management procedure
Error detected. Possible cause: the previous SAId was not recorded after a **Rebuild All** of the application
Validation?
Not Successful
Successful
Download application

SAId Management


Safety Requirements Specification
Design and Engineering
Safe Application Design
Verification
Not Successful
Installation, Commissioning and Validation
Rebuild All
Download Application
Read and Save the Reference SAId
Installation, Commissioning, Validation
Validation Result
Not Successful
Operation and Maintenance
During operation use external tool to read the SAId
Compare to the reference SAId
Equal
The application is modified

Section 13.4
Locking M580 Safety I/O Module Configurations

Locking M580 Safety I/O Module Configurations

Locking a Safety I/O Module Configuration

Each safety I/O module has a configuration locking button *(see Modicon M580, Safety System Planning Guide)*, which you can find at the top front of the module. The purpose of the locking function is to help prevent unintended changes to I/O module configuration. For example, locking the I/O module's current configuration can stop an attempt to assign the module a fake configuration, or merely help protect against configuration mistakes.

To achieve the intended safety integrity level (SIL), lock each safety I/O module after it has been configured, but before you begin or resume operations.

⚠ **WARNING**

RISK OF UNINTENDED DEGRADATION TO PROJECT SAFETY INTEGRITY LEVEL

You must lock each safety I/O module after it is configured but before beginning operations.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

The lock and unlock mechanisms work as follows:

- To lock a safety I/O module configuration, press and hold down the lock button for more than 3 seconds, then release the button.
- To unlock a safety I/O module configuration, press and hold down the lock button for more than 3 seconds, then release the button.

Scenarios for Locking Safety I/O Module configurations

The procedure you follow to lock the configuration of SIL3 safety I/O modules will vary, depending on the scenario, which can be:

- First configuration of the I/O modules
- Fast device replacement of I/O modules
- Perform a change configuration on the fly (CCOTF) for I/O modules

The procedure for each scenario is described below.

First configuration of SIL3 I/O safety modules:

Step	Action
1	Connect Control Expert to the M580 safety PAC.
2	Use the **Transfer project from PLC** command to load the project from the PAC into Control Expert.
3	In the **PLC bus** window in Control Expert, open each SIL3 safety I/O module and confirm that each module is accurately configured.
4	In an animation table in Control Expert, view the DDDT for each SIL3 safety I/O module and confirm that the configuration of each module is the same as in step 3, above.
5	Lock the configuration of each SIL3 I/O module by pressing and holding down the configuration locking button *(see Modicon M580, Safety System Planning Guide)* for more than 3 seconds, then release the button.
6	Check in an animation table the validity of the lock bit status (CONF_LOCKED) for each SIL3 I/O module.

Fast device replacement of a SIL3 I/O safety module:

Step	Action
1	Replace the SIL3 safety I/O module with a new one.
2	Connect Control Expert to the M580 safety PAC in maintenance operating mode *(see page 255)*.
3	In the **PLC bus** window in Control Expert, open each SIL3 safety I/O module and confirm that each module is accurately configured.
4	In an animation table in Control Expert, view the DDDT for each SIL3 safety I/O module and confirm that the configuration of each module has not changed and is the same as in step 3, above.
5	Lock the configuration of each SIL3 I/O module by pressing and holding down the configuration locking button *(see Modicon M580, Safety System Planning Guide)* for more than 3 seconds, then release the button.
6	Check in an animation table the validity of the lock bit status (CONF_LOCKED) for each SIL3 I/O module.

Performing CCOTF to add a new SIL3 I/O safety module:

Step	Action
1	Connect Control Expert to the M580 safety PAC in maintenance operating mode *(see page 255)*.
2	Add a new SIL3 safety I/O module to the configuration, and edit the module settings if necessary.
3	Execute the **Build → Build Changes** command.
4	In the **PLC bus** window in Control Expert, open each SIL3 safety I/O module and confirm that each module is accurately configured.
5	In an animation table in Control Expert, view the DDDT for each SIL3 safety I/O module and confirm that the configuration of each module has not changed and is the same as in step 3, above.
6	Lock the configuration of each SIL3 I/O module by pressing and holding down the configuration locking button *(see Modicon M580, Safety System Planning Guide)* for more than 3 seconds, then release the button.
7	Check in an animation table the validity of the lock bit status (CONF_LOCKED) for each SIL3 I/O module.
8	In the **PLC** menu of Control Expert, command the PAC to enter safety mode *(see page 254)*.

Section 13.5
Initializing Data in Control Expert

Initializing Data in Control Expert for the M580 Safety PAC

Two Init Commands

The **PLC** menu in Control Expert provides two separate commands for the initialization of data:

- The **Init** command initializes data for the process (or non-safe) namespace, which can be used by the MAST, FAST, AUX0 and AUX1 tasks. You can execute this command if the PAC is operating in either safety or maintenance mode while the PAC is in the STOP state. This command is the equivalent of setting the system bit %S0 (COLDSTART) to 1.
  NOTE: Setting the %S0 bit to 1 initializes data in the process namespace only. It does not affect data in the safe namespace.
- The **Init Safety** command initializes data only for the safe namespace, which data can be used exclusively by the SAFE task. You can execute this command only if the SAFE task is operating in maintenance mode while the SAFE task is in the STOP or HALT state. Executing this command when the SAFE task is in the HALT state causes the SAFE task to restart in the STOP state.

Both the **Init** and the **Init Safety** commands perform a cold start. *(see page 266)*

Section 13.6
Working with Animation Tables in Control Expert

Animation Tables and Operator Screens

Introduction

A M580 safety PAC supports three kinds of animation tables, each associated with one of the following data areas:

- Process area animation tables can include only data in the process namespace.
- Safety area animation tables can include only data in the safe namespace.
- Global animation tables can include data for the entire application, including data created for the safe and process namespaces, and global variables.
  NOTE: In a global animation table, data variable names include a prefix indicating the source namespace, as follows:

- A data variable from the Safe namespace is displayed as “SAFE.<varname>”.
- A data variable from the Process namespace is displayed as “PROCESS.<variable name>”.
- A data variable from the Global (or Application) namespace displays only its <variable name>, with no namespace prefix.

Both process and safety data from an M580 safety PAC are also accessible by external processes (for example, SCADA or HMI).

Your ability to create and modify an animation table, and the ability to execute animation table functions, depend on the namespace of the affected variables and the operating mode of the safety project.

Conditions for Creating and Editing Animation Tables

Creating and editing animation tables involves adding or deleting data variables. Your ability to add data variables to, or delete data variables from an animation table depends on:

- The namespace (safe or process) in which the data variable resides.
- The operating mode (safety or maintenance) of the M580 safety PAC.

When Control Expert is connected to the M580 safety PAC, you can create and edit animation tables as follows:

- Adding process namespace variables to, or deleting process namespace variables from a process or global animation table is supported while the M580 safety PAC is operating in either safety mode or maintenance mode.
- Adding safe namespace variables to, or deleting safe namespace variables from a safety animation table is supported while the M580 safety PAC is operating in maintenance mode.
- Adding safe namespace variables to, or deleting safe namespace variables from a safety animation table is supported while the M580 safety PAC is operating in safety mode only if the project settings do not include animation tables in the upload information.
  NOTE: Animation tables are included in, or excluded from, upload information in Control Expert by selecting **Tools → Project Settings...** to open the **Project Settings...** window, then navigating to **Project Settings → General → PLC embedded data → Upload information → Animation tables**.

Conditions for Operating Animation Tables

You can use animation tables to force a variable value, unforce a variable value, modify a single variable value, or modify multiple variable values. Your ability to perform these functions depends on the namespace in which a variable resides and the operating mode of the M580 safety PAC, as follows:

- Process or global variable values can be read or written in both safety operating mode and maintenance operating mode.
- Safety variable values can be read or written in maintenance operating mode.
- Safety variable values can only be read in safety operating mode.

Process for Creating Animation Tables in the Safety or Process Namespace in Control Expert

Control Expert provides two ways to create animation tables for either the safety or process namespace:

- From a safety or process code section window, right click in the code window, then select either:
  - **Initialize Animation Table** to add the data object to an existing animation table in safety or process namespace, or
  - **Initialize New Animation Table** to add the data object to a new animation table in the safety or process namespace.

  In each case, all the variables in the code section are added to the existing or new animation table.
- From the **Project Browser**, in either the process or safety data area, right click on the **Animation Tables** folder, then select **New Animation Table**. Control Expert creates a new, empty animation table. You can then add individual variables from the namespace (safety or process) related to the table.

Process for Creating Globally Scoped Animation Tables

Create a global animation table in the **Project Browser** by right clicking the global **Animation Tables** folder, then selecting **New Animation Table**. You can add variables to the new animation table in several ways:

- *Drag and drop*: You can drag a variable from a data editor and drop it into the global animation table. Because the scope of the animation table includes the entire application, you can drag the variable from the **Safety Data Editor**, the **Process Data Editor**, or the **Global Data Editor**.
- *Instance Selection dialog*: You can double-click in a row in the animation table, then click the ellipsis button to open the **Instance Selection** dialog. Use the filtering list in the top right part of the dialog to select a one of the following project areas:
  - SAFE: to display data objects associated with the safety area.
  - PROCESS: to display data object associated with the process area.
  - APPLICATION: to display higher-level application scope data objects.

  Select a data object, then click **OK** to add the item to the animation table.

NOTE: Data objects added to a global animation table from the:

- Process area have the prefix “PROCESS” affixed to the variable name (for example PROCESS.variable_01
- Safety area have the prefix “SAFE” affixed to the variable name (for example SAFE.variable_02
- Global area have no such prefix added to the variable name.

Displaying Data on Operator Screens

You can display data on an operator screen – such as an HMI, SCADA or FactoryCast application – in the same way that you link to data in an animation table. The data variables available for selection are those variables that are included in the Control Expert data dictionary.

You can enable the data dictionary by opening the **Tools → Project Settings...** window, then in the **Scope → common** area of the window, selecting **General → PLC embedded data → Data dictionary**.

The data dictionary makes data variables available to operator screens as follows:

- Safe namespace variables always include the "SAFE" prefix, and can be reached only by using the format "SAFE.<variable name>".
- Global or application namespace variables do not include a prefix, and can be reached only by using the "<variable name>" without a prefix.
- The **Usage of Process Namespace** setting determines how an operator screen can reach Process namespace variables.
  - If you select **Usage of Process Namespace**, the operator screen can read process area variables only by using the format "PROCESS.<variable name>".
  - If you de-select **Usage of Process Namespace**, the operator screen can read process area variables only by using the format "<variable name>" without the PROCESS prefix.
    NOTE: If two variables are declared with the same name, one in the Process namespace and one in the Global namespace, only the variable from the Global Namespace is accessible by an HMI, SCADA, or Factory Cast application.

You can use the **Instance Selection** dialog to access individual data objects.

⚠ CAUTION

UNEXPECTED VARIABLE VALUE

- Be sure that your application has the correct project settings.
- Verify the syntax to access the variables in the different namespaces.

Failure to follow these instructions can result in injury or equipment damage.

To prevent from accessing the incorrect variable:

- Use different names for the variables you declare in the Process namespace and in the Global namespace, or
- select **Usage of Process Namespace** and use the following syntax to access the variables with the same name:
  - "PROCESS.<variable name>" for variables declared in the Process namespace.
  - "<variable name>" without a prefix for variables declared in the Global namespace

Trending Tool

The Control Expert Trending Tool is not supported for use with an M580 safety project.

Section 13.7
Adding Code Sections

What Is in This Section?

This section contains the following topics:

Adding Code to an M580 Safety Project

Working with Tasks in Control Expert

In the process namespace, Control Expert includes the MAST task by default. The MAST task cannot be removed. However, you can add the tasks FAST, AUX0, and AUX1. Note that creating a task in the process part of a safety project is the same as creating a task in a non-safety project. Refer to the topic *Create and Configure a Task* in the *EcoStruxure™ Control Expert Operating Modes* manual for more information.

In the safe namespace, Control Expert includes the SAFE task by default. The SAFE task cannot be removed, and no other tasks can be added to the **Program Safety** section of the **Project Browser** in Control Expert. You can add multiple sections to the SAFE task.

Configuring the SAFE Task Properties

The SAFE task supports only periodic task execution (cyclic execution is not supported). Both the SAFE task **Period** and **Watch Dog** settings are input in the **Properties of SAFE** dialog and can support the following value range:

- SAFE task period: 10...255 ms with a default of 20 ms.
- SAFE task watchdog: 10...500 ms, in increments of 10 ms, with a default of 250 ms.

Set the SAFE task **Period** to a minimum value depending on the safe data size and the PLC model. The minimum SAFE task period can be calculated according to the following formulas:

- Absolute minimum necessary for safe I/O communication:
  - 10 ms
- Time (in ms) necessary to transfer and compare the safe data between the CPU and the COPRO:
  - (0.156 x Data_Safe_Size) + 2 ms (for BMEP584040S, BMEH584040S, and BMEH586040S)
  - (0.273 x Data_Safe_Size) + 2 ms (for BMEP582040S and BMEH582040S)

  Where the Data_Safe_Size is the size in Kbytes of the safe data.
- Additional time (in ms) needed by Hot Standby PACs to transfer the safe data from the primary PAC to the standby PAC:
  - (K1 x Task_{kb} + K2 x Task_{DFB}) / 500

  In this formula:
  - Task_{DFB} = the number of DFBs declared in the safe part of the application.
  - Task_{kb} = the size (in Kbytes) of the safe data exchanged by the SAFE task between the primary and standby PACs.
  - K1 and K2 are constants, with values determined by the specific CPU module used in the application:

Coefficient	BMEH582040S	BMEH584040S, and BMEH586040S
K1	32.0	10.0
K2	23.6	7.4

NOTE:

- The value produced by these formulas is an absolute minimum for the SAFE task period valuable only for a first estimation of the SAFE cycle time limit. It does not include the time necessary for user code execution or for the margin necessary for the intended operation of the PAC multi-task system. Refer to the topic System Throughput Considerations in the *Modicon M580 Standalone System Planning Guide for Frequently Used Architectures*.
- By default, Data_Safe_Size and $\text{Size}_{\text{kbytes}}$ are equal. Their values can be viewed, respectively, in the **PLC** → **Memory Consumption** menu and the **PLC** → **Hot Standby** screen.

Example Calculations

Sample results of calculating the minimum SAFE task period are set forth below

Minimum Safe Task Period (ms)					
$\text{Size}_{\text{kbytes}}$[1]	$\text{Nb}_{\text{DFB\_Inst}}$	BMEP582040S	BMEP584040S	BMEH582040S	BMEH584040S or BMEH586040S
0	0	10	10	10	10
50	10	16	10	20	11
100	10	30	18	37	20
150	10	43	25	54	29
200	10	57	33	70	37
250	10	71	41	87	46
300	20	84	49	105	55
350	20	98	57	121	64
400	20	112	64	138	73
450	20	125	72	155	81
500	20	139	80	172	90
550	30	-	88	-	99
600	30	-	96	-	108
650	30	-	103	-	117
700	30	-	111	-	126
750	30	-	119	-	134
800	40	-	127	-	143
1. $\text{Size}_{\text{kbytes}}$ and Data_Safe_Size are assumed to be equal.					

Minimum Safe Task Period (ms)					
$\text{Size}_{\text{kbytes}}$[1]	$\text{Nb}_{\text{DFB\_Inst}}$	BMEP582040S	BMEP584040S	BMEH582040S	BMEH584040S or BMEH586040S
850	40	-	135	-	152
900	40	-	142	-	161
950	40	-	150	-	170
1000	40	-	158	-	179
1. $\text{Size}_{\text{kbytes}}$ and Data_Safe_Size are assumed to be equal.					

NOTE: Configure the SAFE task watchdog with a value that is greater than the SAFE task **Period**.

Refer to the topic *Process Safety Time (see page 150)*, for information regarding how the SAFE task configuration affects process safety time.

Refer to the topic *M580 Safety PAC Tasks (see page 268)* for information describing the execution priority of the SAFE task.

Creating Code Sections

Right click on the **Section** folder for a task and select **New Section...** to open a configuration dialog. For the safety and process tasks, the following program languages are available:

Language	Safety tasks	Process tasks			
	SAFE	MAST	FAST	AUX0	AUX1
IL	–	✓	✓	✓	✓
FBD	✓	✓	✓	✓	✓
LD	✓	✓	✓	✓	✓
LL984 segment	–	✓	✓	✓	✓
SFC	–	✓	✓	✓	✓
ST	–	✓	✓	✓	✓
✓: available – : not available					

Except for these limitations on programming language availability for the SAFE task, the new section configuration dialog operates the same as for a non-safety M580 project. Refer to the topic *Properties Dialog Box for FBD, LD, IL, or ST Sections* in the *EcoStruxure™ Control Expert Operating Modes* manual for more information.

Adding Data to Code Sections

Because the SAFE task is separated from the process tasks, only data accessible in the **Safety Data Editor** is available to be added to a SAFE task code section. This data includes:

- Unlocated safety variables (i.e. with no %M or %MW address) created in the **Safety Data Editor**.
- Data objects that are part of M580 safety module device DDT structures.

Similarly, data available to non-safety task code sections includes all data within the scope of the process namespace. This includes all project data except:

- Data exclusively available to the SAFE namespace (see above).
- Data objects created in the **Global Data Editor**.

Code Analysis

When you analyze or build a project, Control Expert displays a detected error message if:

- Data belonging to the process namespace is included in the SAFE task.
- Data belonging to the safe namespace is included in a process task (MAST, FAST, AUX0, AUX1).
- Located bits (%M) or words (%MW) are included in a SAFE task section.

Diagnostic Request

Introduction

The diagnostic request is available only for M580 safety power supplies located on a main rack, using the PWS_DIAG function block. A main rack is one with an address of 0 and a CPU or communication adapter module (CRA) in slot 0 or 1. An extension rack is not a main rack.

The CPU can make a diagnostic request of redundant power supplies on the local rack and, via a communications adapter (CRA), of redundant power supplies on a remote rack. If the master and slave power supplies are operational, the master power supply enters master diagnostic mode and the slave power supply enters slave diagnostic mode. The LEDs indicate the test is ongoing.

NOTE: This request is not implemented when power cycles ON

After the diagnostic test finishes, the master returns to normal operating state and the slave transitions to either normal or error state, depending on the outcome of the tests. Test results are stored in power supply memory,

Diagnostic Request Returned Data

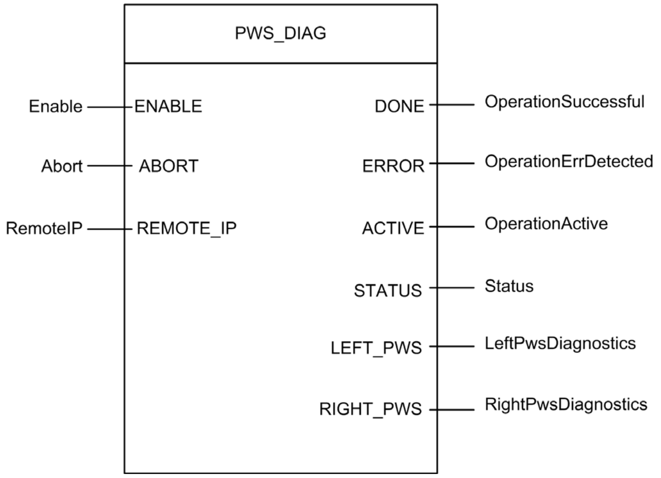
Diagnostic information sent to the CPU by the power supplies includes:

- Power supply ambient temperature.
- Voltage and current on 3.3V backplane line.
- Voltage and current on 24V backplane line.
- Power supply total cumulated energy since manufacturing on the 3.3V and 24V backplane lines.
- Operating time as master since last power-on and manufacture.
- Total operating time as slave since last power-on and manufacture.
- Remaining life time in percent (LTPC): the time before preventive maintenance from 100% to 0%.

  NOTE: No swap when at 0%.
- Number of times power supply has powered ON.

  NOTE: From the SCADA, it is possible to reset the number of power on since installation and all others diagnostics.
- Number of times BMXCPS4002S main voltage fell below under-voltage level 1 (95 Vac).
- Number of times BMXCPS4002S main voltage rises above over-voltage level 2 (195 Vac).
- Number of times BMXCPS4022S main voltage fell below under-voltage level 1 (20 Vdc).
- Number of times BMXCPS4022S main voltage rises above over-voltage level 2 (40 Vdc).
- Number of times BMXCPS3522S main voltage fell below under-voltage level 1 (110 Vdc).
- Number of times BMXCPS3522S main voltage rises above over-voltage level 2 (140 Vdc).
- Current status of the power supply (master/slave/inoperable).

Representation in FBD



Parameters

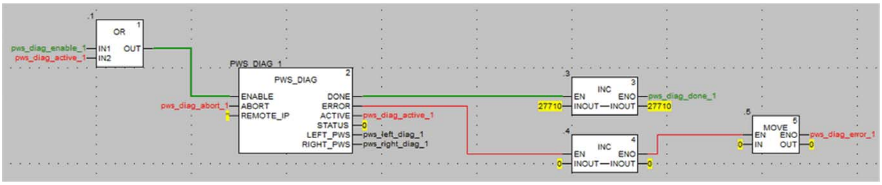
Input Parameters:

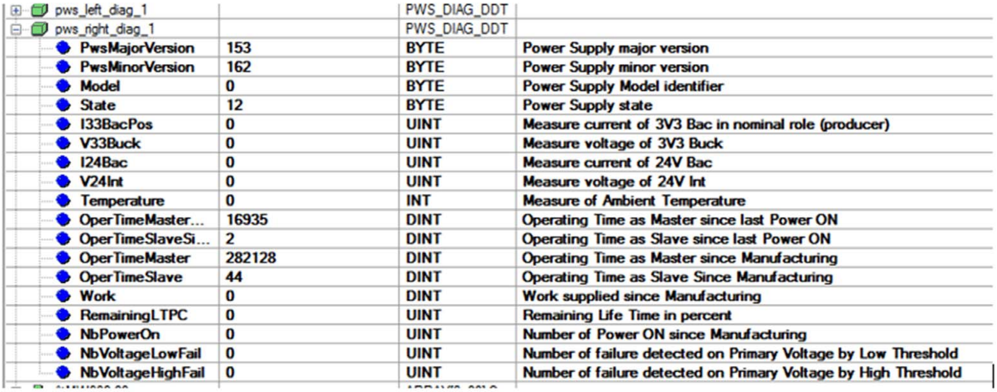
Parameter Name	Data Type	Description
ENABLE	BOOL	When ON, the operation is enabled.
ABORT	BOOL	When ON, the currently active operation is aborted.
REMOTE_IP	STRING	IP Address ("ip1.ip2.ip3.ip4") of the drop that contains the power supply module. Leave this field an empty string ("") or attach no variable to its pin to address the power supply in the local rack.

Output Parameters:

Parameter Name	Data Type	Description
DONE	BOOL	ON when the operation concludes successfully.
ERROR	BOOL	ON when the operation is aborted without success.
ACTIVE	BOOL	ON when the operation is active.
STATUS	WORD	Detected error identifier.
LEFT_PWS	ANY	Diagnostic data for left power supply. Use variable of type PWS_DIAG_DDT_V2 *(see page 131)* for correct interpretation.
RIGHT_PWS	ANY	Diagnostic data for right power supply. Use variable of type PWS_DIAG_DDT_V2 for correct interpretation.

Example





⊞ pws_left_diag_1		PWS_DIAG_DDT	
⊟ pws_right_diag_1		PWS_DIAG_DDT	
● PwsMajorVersion	153	BYTE	Power Supply major version
● PwsMinorVersion	162	BYTE	Power Supply minor version
● Model	0	BYTE	Power Supply Model identifier
● State	12	BYTE	Power Supply state
● I33BacPos	0	UINT	Measure current of 3V3 Bac in nominal role (producer)
● V33Buck	0	UINT	Measure voltage of 3V3 Buck
● I24Bac	0	UINT	Measure current of 24V Bac
● V24Int	0	UINT	Measure voltage of 24V Int
● Temperature	0	INT	Measure of Ambient Temperature
● OperTimeMaster...	16935	DINT	Operating Time as Master since last Power ON
● OperTimeSlaveSi...	2	DINT	Operating Time as Slave since last Power ON
● OperTimeMaster	282128	DINT	Operating Time as Master since Manufacturing
● OperTimeSlave	44	DINT	Operating Time as Slave Since Manufacturing
● Work	0	DINT	Work supplied since Manufacturing
● RemainingLTPC	0	UINT	Remaining Life Time in percent
● NbPowerOn	0	UINT	Number of Power ON since Manufacturing
● NbVoltageLowFail	0	UINT	Number of failure detected on Primary Voltage by Low Threshold
● NbVoltageHighFail	0	UINT	Number of failure detected on Primary Voltage by High Threshold

Swap and Clear Commands

Introduction

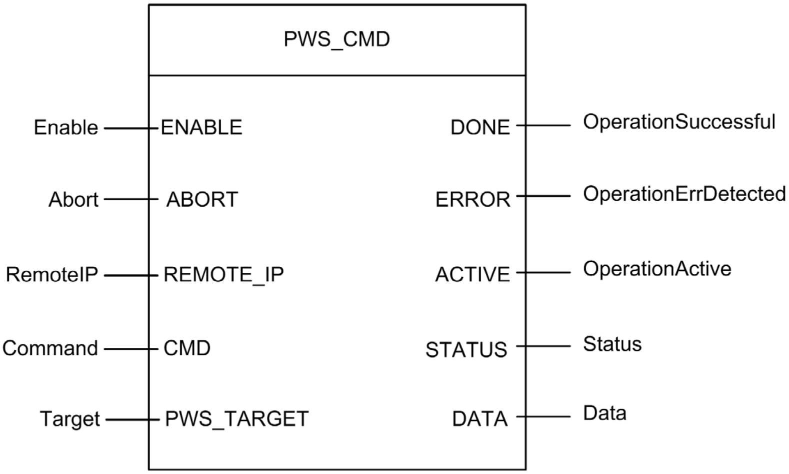
The PWS_CMD function block can be used to issue two commands:

- Swap request: This command specifies the power supply to serve as the master. If both power supplies are operational, the specified power supply becomes the master, the other becomes the slave.
- Clear request: This command resets the counters of the number of times:
  - main voltage fell below under-voltage level 1.
  - main voltage fell below under-voltage level 2.
  - power supply has powered ON.

Both requests are available only for power supplies on the main rack. A main rack is one with an address of 0 and a CPU or communication adapter module (CRA) in slot 0 or 1. An extension rack is not a main rack.

The LEDs indicate the command is ongoing. A record of the event is stored in power supply memory.First paragraph of fact block.

Representation in FBD



Parameters

Input Parameters:

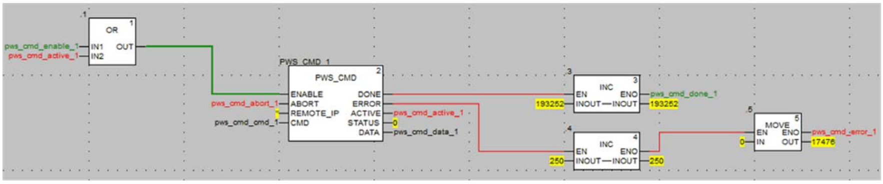
Parameter Name	Data Type	Description
ENABLE	BOOL	When ON, the operation is enabled.
ABORT	BOOL	When ON, the currently active operation is aborted.
REMOTE_IP	STRING	IP Address ("ip1.ip2.ip3.ip4") of the drop that contains the power supply module. Leave this field an empty string ("") or attach no variable to its pin to address the power supply in the local rack.
CMD	ANY	Use variable of type PWS_CMD_DDT for correct interpretation. Available command code: ● 1 = swap ● 3 = clear
PWS_TARGET	BYTE	Power supply to address: ● 1 = left ● 2 = right ● 3 = both

Output Parameters:

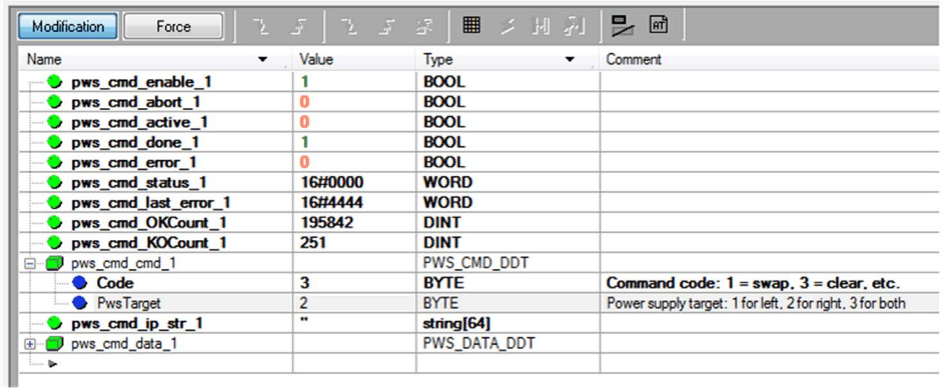
Parameter Name	Data Type	Description
DONE	BOOL	ON when the operation concludes successfully.
ERROR	BOOL	ON when the operation is aborted without success.
ACTIVE	BOOL	ON when the operation is active.
STATUS	WORD	Detected error identifier.
DATA	ANY	Response data (depending of command code). No data are reported for swap and clear commands.

Example

The following diagram demonstrated a PWS_CMD block used for a swap request:



The following data editor screen shows the variable values of a swap request:



Name	Value	Type	Comment
pws_cmd_enable_1	1	BOOL	
pws_cmd_abort_1	0	BOOL	
pws_cmd_active_1	0	BOOL	
pws_cmd_done_1	1	BOOL	
pws_cmd_error_1	0	BOOL	
pws_cmd_status_1	16#0000	WORD	
pws_cmd_last_error_1	16#4444	WORD	
pws_cmd_OKCount_1	195842	DINT	
pws_cmd_KOCount_1	251	DINT	
⊟ pws_cmd_cmd_1		PWS_CMD_DDT	
Code	3	BYTE	Command code: 1 = swap, 3 = clear, etc.
PwsTarget	2	BYTE	Power supply target: 1 for left, 2 for right, 3 for both
pws_cmd_ip_str_1	''	string[64]	
⊞ pws_cmd_data_1		PWS_DATA_DDT	

Section 13.8
Application Security Management

Introduction

Control Expert lets you restrict access to the M580 safety PAC to users with assigned passwords. This section references the password assignment processes available in Control Expert.

What Is in This Section?

This section contains the following topics:

Application Password Protection

Overview

Control Expert provides a password mechanism to help guard against unauthorized access to the application. Control Expert uses the password when you:

- Open the application in Control Expert.
- Connect to the PAC in Control Expert.

NOTE: Use of a password is optional. The application protection by password mechanism is disabled if no password is configured in the application (the default setting).

Application protection by password helps prevent unwanted application modification, download, or the opening of (.STU and .STA files). The password is stored encrypted in the application.

Application Password

An M580 safety project is not password protected by default. If you do not manually assign a password when you create the safety project, Control Expert applies an empty string as the password.

In this case, when you next open your Control Expert M580 safety project, the **Password** dialog opens. To access your project, enter **no** password text, thereby accepting the empty string, and click **OK**. Thereafter, you can follow the steps set forth below to create a new password.

It is possible to create or change a password at any time.

The password is case-sensitive and has a length from 8 to 16 characters. The password robustness is increased when it contains a mix of upper and lower case, alphabetical, numerical, and special characters.

Creating a Password

Procedure for creating the application protection password:

Step	Action
1	In the project browser right-click **Project**.
2	Select **Properties** command from the popup menu. **Result**: The **Properties of Project** window appears.
3	Select the **Project & Controller Protection** tab.
4	In the **Application** field, click **Change password ...**. **Result**: The **Modify Password** window appears.
5	Enter the new password in the **Entry** field.
6	Enter the confirmation of the new password in the **Confirmation** field.
7	Click **OK** to confirm.
8	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Changing the Password

Procedure for changing the application protection password:

Step	Action
1	In the project browser right-click **Project**.
2	Select **Properties** command from the popup menu. **Result**: The **Properties of Project** window appears.
3	Select the **Project & Controller Protection** tab.
4	In the **Application** field, click **Change password ...**. **Result**: The **Modify Password** window appears.
5	Enter previous password in the **Old password** field.
6	Enter the new password in the **Entry** field.
7	Enter the confirmation of the new password in the **Confirmation** field.
8	Click **OK** to confirm.
9	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Deleting the Password

Procedure for clearing the application protection password:

Step	Action
1	In the project browser right-click **Project**.
2	Select **Properties** command from the popup menu. **Result**: The **Properties of Project** window appears.
3	Select the **Project & Controller Protection** tab.
4	In the **Application** field, click **Clear password...**. **Result**: The **Password** window appears.
5	Enter the password in the **Password** field.
6	Click **OK** to confirm.
7	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Auto-Lock Feature

There is an optional auto-lock feature that limits access to the Control Expert software programming tool after a configured time of inactivity. You can activate the auto-lock feature with the check box **Auto-lock** and select the time-out for the time of inactivity via **Minutes before lock**.

The default values are:

- **Auto-lock** is not activated
- **Minutes before lock** is set to 10 minutes (possible values: 1...999 minutes)

If the auto-lock feature is enabled and the configured inactivity time elapses, a modal dialog box is displayed requiring the entry of the application password. Behind the modal dialog box, all opened editors remain open in the same position. As a result, anybody can read the current content of the Control Expert windows but cannot continue to work with Control Expert.

NOTE: If you have not assigned a password to the project, the modal dialog box is not displayed.

Password Request Condition

Open an existing application (project) in Control Expert:

Password Management	
When an application file is opened, an **Application Password** dialog box opens.	
Enter the password.	
Click **OK**.	If the password is correct, the application opens.
	If the password is wrong, a message box indicates an incorrect password was entered, and a new **Application Password** dialog box opens.
If you click **Cancel**, the application is not opened	

Accessing the application in Control Expert after an auto-lock, when Control Expert is not connected to the PAC or when the project in Control Expert is EQUAL to the project in the PAC:

Password Management	
When auto-lock time is elapsed, an **Application Password** dialog box opens:	
Enter the password.	
Click **OK**.	If the password is correct, Control Expert becomes active again.
	If the password is wrong, a message box indicates an incorrect password was entered, and a new **Application Password** dialog box opens.
If you click **Close**, the application is closed without being saved.	

Accessing the application in the PAC after an auto-lock, when Control Expert is connected to the PAC and the application in Control Expert is DIFFERENT from the application in the PAC:

Password Management	
On connection, if Control Expert software application and the CPU application are not equal, an **Application Password** dialog box opens:	
Enter the password.	
Click **OK**.	If the password is correct, the connection is established.
	If the password is wrong, a message box indicates an incorrect password was entered, and a new **Application Password** dialog box opens.
If you click **Cancel**, the connection is not established.	
NOTE: On connection, if Control Expert software application and the CPU applications are equal, there is no password request. If no password has been initially entered (left empty on project creation), click **OK** to establish the connection on password prompt.	

NOTE: After three attempts with a wrong password, you will have to wait an increasing amount of time between each subsequent password attempt. The wait period increases from 15 seconds to 1 hour, with the wait increment increasing by a factor of 2 after each successive attempt with a wrong password.

NOTE: In case of password loss, contact Schneider Electric support *(see page 315)*.

Safe Area Password Protection

At a Glance

Safety CPUs include a safe area password protection function, which is accessible from the **Properties** screen of the project. This function is used to help protect project elements located within the safe area of the safety project.

NOTE: When the safe area password protection function is active, the safe parts of the application cannot be modified

Modifications to the following safe area parts are not permitted when safe area password protection is enabled:

Safe Part	Forbidden action (offline AND online)
Configuration	Modify CPU characteristics
	Add, Delete, Modify a Safety module in the rack
	Modify Safety Power supply
Types	Create, Delete, Modify a Safe DDT
	Change a DDT attribute: from not safe->safe
	Change a DDT attribute: from safe->not safe
	Create, Delete, Modify a Safe DFB
	Change a DFB attribute: from not safe->safe
	Change a DFB attribute: from safe->not safe
Program-SAFE	Any Change under the **Variables an FB instances** node
	Create Task
	Import Task
	Modify Task
	Create Section
	Delete Section
	Import Section
	Modify Section
Project Settings	Modify SAFE project settings
	Modify COMMON project settings

Encryption

The safe area password uses the standard encryption SHA-256 with a salt.

Safe Area Password Function versus Safety Project User Rights

The activation of the safe area password and the implementation of user rights created in the **Security Editor** are mutually exclusive security functions, as follows:

- If the user launching Control Expert has been assigned a user profile, that user can access the safe areas of the safety application if the user knows the safe area password and has been granted access rights in the **Security Editor**.
- If user profiles have not been assigned, a user can access the safe areas of the safety application by knowing the safe area password.

Visual Indicators in Control Expert

The state of the safe area protection function can be visibly detected by viewing the **Program-SAFE** node in the **Project Browser**:

- A locked padlock indicates a safe area password has been created and activated.
- An unlocked padlock indicates a safe area password has been created but not activated.
- No padlock indicates a safe area password has not been created.

NOTE: If a safe area password has been created but not activated, and the safety application is closed then re-opened, the safe area password is automatically activated on re-opening. This behavior serves as a precaution if the safe are password was unintentionally not re-activated.

Compatibility

The safe area password function exists for Control Expert V14.0 or later, for M580 safety CPUs with firmware 2.80 or later.

NOTE:

- Application program `.STU`, `.STA`, and `.ZEF` files, which are created in Control Expert V14.0 or later, cannot be opened in Unity Pro V13.1 and earlier.
- Replacing an M580 safety CPU in a Control Expert v14.0 application has the following effect:
  - Upgrading from firmware 2.70 to 2.80 (or later) adds the safe area password functionality to the **Program & Safety Protection** tab of the **Project** → **Properties** window.
  - Downgrading from firmware 2.80 (or later) to 2.70 removes the safe area password functionality.

Activating Protection and Creating Password

Procedure for activating the protection of sections and creating the password:

Step	Action
1	In the project browser right-click **Project**.
2	Select **Properties** command from the popup menu. **Result**: The **Properties of Project** window appears.
3	Select the **Program & Safety Protection** tab.
4	In the **Safety** area, activate the protection by checking the **Protection active** box. **Result**: The **Modify Password** dialog box appears.
5	Enter a password in the **Entry** field.
6	Enter the confirmation of the password in the **Confirmation** field.
7	Click **OK** to confirm.
8	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Changing the Password

Procedure for changing the project sections protection password:

Step	Action
1	In the project browser right-click **Project**.
2	Select **Properties** command from the popup menu. **Result**: The **Properties of Project** window appears.
3	Select the **Program & Safety Protection** tab.
4	In the **Safety** area, click **Change password ...**. **Result**: The **Modify Password** dialog box appears:
5	Enter previous password in the **Old password** field.
6	Enter the new password in the **Entry** field.
7	Enter the confirmation of the new password in the **Confirmation** field.
8	Click **OK** to confirm.
9	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Deleting the Password

Procedure for deleting the project sections protection password:

Step	Action
1	In the project browser right-click **Project**.
2	Select **Properties** command from the popup menu. **Result**: The **Properties of Project** window appears.
3	Select the **Program & Safety Protection** tab.
4	In the **Safety** area, click **Clear password...**. **Result**: The **Access control** dialog box appears:
5	Enter the previous password in the **Password** field.
6	Click **OK** to confirm.
7	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Section Protection

At a Glance

The section protection function is accessible from the **Properties** screen of the project in offline mode.

This function is used to help protect individual program sections, for which a level of protection has been configured.

NOTE: The section protection is not active as long as the protection has not been activated in the project.

NOTE:
The project protection is effective to the marked sections only. This does not prevent:

- Connecting to the CPU
- Uploading application from the CPU
- Changing the configuration
- Adding new sections
- Changing the logic in a new section (without section protection)

Activating Protection and Creating Password

Procedure for activating the protection of sections and creating the password:

Step	Action
1	In the project browser right-click **Project**.
2	Select **Properties** command from the popup menu. **Result**: The **Properties of Project** window appears.
3	Select the **Program & Safety Protection** tab.
4	In the **Sections & Program Units** area, activate the protection by checking the **Protection active** box. **Result**: The **Modify Password** dialog box appears.
5	Enter a password in the **Entry** field.
6	Enter the confirmation of the password in the **Confirmation** field.
7	Select the **Crypted** check box if an enhanced password protection is required. **NOTE:** A project with a crypted password cannot be edited with a Unity Pro version lower than 4.1. **NOTE:** Unity Pro is the former name of Control Expert for version 13.1 or earlier.
8	Click **OK** to confirm.
9	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Notes

If a section *(see EcoStruxure™ Control Expert, Operating Modes)* is configured with a protection (read or read/write), when protection has been activated this will be indicated by a locked padlock at the section level.

If the section is configured with a protection but the protection is disabled, an open padlock is displayed at the section level.

Changing the Password

Procedure for changing the project sections protection password:

Step	Action
1	In the project browser right-click **Project**.
2	Select **Properties** command from the popup menu. **Result**: The **Properties of Project** window appears.
3	Select the **Program & Safety Protection** tab.
4	In the **Sections & Program Units** area, click **Change password ...**. **Result**: The **Modify Password** dialog box appears:
5	Enter previous password in the **Old password** field.
6	Enter the new password in the **Entry** field.
7	Enter the confirmation of the new password in the **Confirmation** field.
8	Select **Crypted** check box if an enhanced password protection is required. **NOTE:** A project with a crypted password cannot be edited with a Unity Pro version lower than 4.1.
9	Click **OK** to confirm.
10	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Deleting the Password

Procedure for deleting the project sections protection password:

Step	Action
1	In the project browser right-click **Project**.
2	Select **Properties** command from the popup menu. **Result**: The **Properties of Project** window appears.
3	Select the **Program & Safety Protection** tab.
4	In the **Sections** field, click **Clear password...**. **Result**: The **Access control** dialog box appears:
5	Enter the previous password in the **Password** field.
6	Click **OK** to confirm.
7	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Firmware Protection

Overview

Firmware protection by a password helps prevent unwanted access to the module firmware via FTP.

Password

The firmware is password protected by default with the following password: **`fwdownload`**.

It is possible to change a password at any time.

The password is case-sensitive and contains 8 to 16 alphanumeric characters. The password robustness is increased when it contains a mix of upper and lower case, alphabetical, numerical, and special characters.

NOTE: When importing a ZEF file, the firmware password of the module is set to its default value.

Changing the Password

NOTE: Firmware default password: **`fwdownload`**

Procedure for changing the firmware protection password:

Step	Action
1	In the project browser right-click **Project**.
2	Select **Properties** command from the popup menu. **Result**: The **Properties of Project** window appears.
3	Select **Project & Controller Protection** tab.
4	In the **Firmware** field, click **Change password ...**. **Result**: The **Modify Password** window appears.
5	Enter previous password in the **Old password** field.
6	Enter the new password in the **Entry** field.
7	Enter the confirmation of the new password in the **Confirmation** field.
8	Click **OK** to confirm.
9	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Resetting the Password

Resetting the password assigns its default value to the firmware password (`fwdownload`) once the current password is confirmed. Proceed as follows:

Step	Action
1	In the project browser right-click **Project**.
2	Select **Properties** command from the popup menu. **Result**: The **Properties of Project** window appears.
3	Select **Project & Controller Protection** tab.
4	In the **Firmware** field, click **Reset password...**. **Result**: The **Password** window appears.
5	Enter current password in the **Password** field.
6	Click **OK** to confirm.
7	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. The new password is the default password: `fwdownload`. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Data Storage Protection

Overview

Data storage protection by a password helps prevent unwanted access to the data storage zone of the SD memory card (if a valid card is inserted in the CPU).

Password

The data storage area is password protected by default with the following password: **`datadownload`**.

It is possible to change a password at any time.

The password is case-sensitive and it must have a size from 8 to 16 alphanumeric characters. The password robustness is increased when it contains a mix of upper and lower case, alphabetical, numerical, and special characters.

NOTE: When importing a ZEF file, the data storage password of the application is set to its default value.

Changing the Password

NOTE: Data storage default password: **`datadownload`**

Procedure for changing the data storage protection password:

Step	Action
1	In the project browser right-click **Project**.
2	Select **Properties** command from the popup menu. **Result**: The **Properties of Project** window appears.
3	Select **Project & Controller Protection** tab.
4	In the **Data Storage** field, click **Change password ...**. **Result**: The **Modify Password** window appears.
5	Enter previous password in the **Old password** field.
6	Enter the new password in the **Entry** field.
7	Enter the confirmation of the new password in the **Confirmation** field.
8	Click **OK** to confirm.
9	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Resetting the Password

Resetting the password assigns its default value to the data storage password (`datadownload`) once the current password is confirmed. Proceed as follows:

Step	Action
1	In the project browser right-click **Project**.
2	Select **Properties** command from the popup menu. **Result**: The **Properties of Project** window appears.
3	Select **Project & Controller Protection** tab.
4	In the **Data Storage** field, click **Reset password...**. **Result**: The **Password** window appears.
5	Enter current password in the **Password** field.
6	Click **OK** to confirm.
7	Click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. The new password is the default password: `datadownload`. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Loss of Password

Overview

If you forget a password, proceed as indicated in the following procedures and contact Schneider Electric support.

Control Expert Passwords

Schneider Electric support needs a number displayed from the **Password** dialog box reached in following conditions:

- At open time, select the application and the **Password** dialog box is displayed.
- At auto-lock time, the **Password** dialog box is displayed. If you do not remember the password, select **Close**. Open the application again and the **Password** dialog box is displayed.
  NOTE: When the application is closed without entering a password after an auto-lock, all modifications are lost.

Procedure for resetting the application password:

Step	Action
1	**Condition:** The **Password** dialog box is displayed.
2	Press `SHIFT+F2`. **Result:** A grayed number is displayed in the right side of the **Password** dialog box.
3	Give this number to Schneider Electric support.
4	Receive the generated password from Schneider Electric support. **NOTE:** The password is a temporary password, available as long as the application is not modified.
5	Enter this password.
6	Modify the password (old password = password provided by Schneider Electric support).
7	Click **Build** → **Build Changes**.
8	**Save** the application.

CPU Application Password

Procedure for resetting the CPU application password if the respective **.STU* file is available:

Step	Action
1	Open the respective **.STU* file.
2	When the password dialog box is displayed press `SHIFT+F2`. **Result:** A grayed number is displayed in the right side of the **Password** dialog box.
3	Give this number to Schneider Electric support.
4	Receive the generated password from Schneider Electric support. **Note:** The password is a temporary password, available as long the application is not modified.
5	Enter this password.
6	Modify the password (old password = password provided by Schneider Electric support).
7	**Connect** to the PLC.
8	Click **Build** → **Build Changes**.
9	**Save** the application.

Procedure for resetting the CPU application password if the respective **.STU* file is not available:

Step	Action
1	**Condition:** At connection time, the **Password** dialog box is displayed.
2	Press `SHIFT+F2`. **Result:** A grayed number is displayed in the right side of the **Password** dialog box.
3	Give this number to Schneider Electric support.
4	Receive the generated password from Schneider Electric support. **Note:** The password provided by Schneider Electric support is a temporary password, available as long as the application is not modified.
5	Enter this password.
6	Upload the application from CPU.
7	**Save** the application.
8	Modify the password (old password = the one provided by Schneider Electric support).
9	Click **Build** → **Build Changes**.
10	**Save** the application.

Safe Area Password

Schneider Electric support needs a number displayed from the **Password** dialog box reached in following condition:

- In **Properties of Project** → **Program & Safety Protection** → **Safety** field, click **Clear password...** and the **Password** dialog box is displayed.

Procedure for resetting the firmware password:

Step	Action
1	**Condition:** The **Password** dialog box is displayed.
2	Press `SHIFT+F2`. **Result:** A grayed number is displayed in the right side of the **Password** dialog box.
3	Give this number to Schneider Electric support.
4	Receive the generated password from Schneider Electric support. **Note:** The password is a temporary password, available as long as you do not modify the application.
5	Enter this password and click **OK** to close the **Password** dialog.
6	Click **Change Password** and change the password (note, the old password = password provided by Schneider Electric support).
7	Click **OK** to close the **Modify Password** dialog, then click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Firmware Password

Schneider Electric support needs a number displayed from the **Password** dialog box reached in following condition:

- In **Properties of Project** → **Project & Controller Protection** → **Firmware** field, click **Reset password...** and the **Password** dialog box is displayed.

Procedure for resetting the firmware password:

Step	Action
1	**Condition:** The **Password** dialog box is displayed.
2	Press `SHIFT+F2`. **Result:** A grayed number is displayed in the right side of the **Password** dialog box.
3	Give this number to Schneider Electric support.
4	Receive the generated password from Schneider Electric support. **Note:** The password is a temporary password, available as long as you do not modify the application.
5	Enter this password and click **OK** to close the **Password** dialog.
6	Click **Change Password** and change the password (note, the old password = password provided by Schneider Electric support).
7	Click **OK** to close the **Modify Password** dialog, then click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Data Storage Password

Schneider Electric support needs a number displayed from the **Password** dialog box reached in following condition:

- In **Properties of Project** → **Project & Controller Protection** → **Data Storage** field, click **Reset password...** and the **Password** dialog box is displayed.

Procedure for resetting the data storage password:

Step	Action
1	**Condition:** The **Password** dialog box is displayed.
2	Press `SHIFT+F2`. **Result:** A grayed number is displayed in the right side of the **Password** dialog box.
3	Give this number to Schneider Electric support.
4	Receive the generated password from Schneider Electric support. **Note:** The password is a temporary password, available as long as you do not modify the application.
5	Enter this password and click **OK** to close the **Password** dialog.
6	Click **Change Password** and change the password (note, the old password = password provided by Schneider Electric support).
7	Click **OK** to close the **Modify Password** dialog, then click **OK** or **Apply** in the **Properties of Project** window to confirm all changes. If you click **Cancel** in the **Properties of Project** window, all changes are canceled.

Section 13.9
Workstation Security Management

Introduction

Schneider Electric provides the **Security Editor** access management tool that you can use to limit and control access to the workstation on which your Control Expert software is installed. This section describes the features of this tool that uniquely relate to M580 safety projects.

What Is in This Section?

This section contains the following topics:

Managing Access to Control Expert

Introduction

Schneider Electric provides the **Security Editor** configuration tool that lets you manage access to the Control Expert software installed on a workstation. Using he *Security Editor* configuration tool to manage access to the Control Expert software is optional.

NOTE: Access management relates to the hardware – typically a workstation – on which Control Expert software is installed and not to the project, which has its own protection system.

For more information, refer to *EcoStruxure™ Control Expert, Security Editor, Operation Guide*.

NOTE: Safety user profiles also require rights to access the process part of the safety application. When you create or modify a user profile, it is your responsibility to confirm that all necessary modifications are properly made.

Categories of Users

The **Security Editor** supports two categories of users:

- **Super User (Supervisor):**
  The super user is the only person to manage access security for the software. The super user specifies who can access the software and their access rights. During installation of Control Expert on the workstation, only the super user can access the security configuration without any limitation of rights (without a password).
  NOTE: The user name reserved for the super user is Supervisor.

- **Users:**
  Software users are defined in the list of users by the super user, if Control Expert access security is active. If your name is in the user list, you can access a software instance by entering your name (exactly as it appears on the list) and your password.

User Profile

The user profile comprises all of the access rights for a user. The user profile can be custom-defined by the super user, or can be created by applying a preconfigured profile that comes with the **Security Editor** tool.

Preconfigured User Profiles

The **Security Editor** offers the following preconfigured user profiles, which apply to either the safety program or the process program:

Profile	Applicable program type		Description
	Process	Safety	
ReadOnly	✓	✓	The user can only access the project in read mode, except for the PAC address, which can be modified. The user can also copy or download the project.
Operate	✓	–	The user has the same rights as with a **ReadOnly** profile, with the added possibility of modifying process program execution parameters (constants, initial values, task cycle times, etc.).
Safety_Operate	–	✓	The user has similar rights as with the **Operate** profile, but with respect to the safety program, except that: ● Transferring data values to the PAC is not permitted. ● Commanding the safety program to enter maintenance mode is permitted.
Adjust	✓	–	The user has the same rights as with an **Operate** profile, with the added possibility of uploading a project (transfer to the PAC) and modifying the PAC operating mode (**Run**, **Stop**, ...)
Safety_Adjust	–	✓	The user has similar rights as with the **Adjust** profile, but with respect to the safety program, except that: ● Transferring data values to the PAC is not permitted. ● Commanding the safety program to enter maintenance mode is permitted.
Debug	✓	–	The user has the same rights as with an **Adjust** profile, with the added possibility of using the debugging tools.
Safety_Debug	–	✓	The user has similar rights as with the **Debug** profile, but with respect to the safety program, except that: ● Stopping or starting the program is not permitted. ● Updating initialization values is not permitted. ● Transferring data values to the PAC is not permitted. ● Forcing inputs, outputs or internal bits is not permitted. ● Commanding the safety program to enter maintenance mode is permitted.
Program	✓	–	The user has the same rights as with a **Debug** profile, with the added possibility of modifying the program.

Profile	Applicable program type		Description
	Process	Safety	
Safety_Program	–	✓	The user has similar rights as with the **Program** profile, but with respect to the safety program, except that: ● Stopping or starting the program is not permitted. ● Updating initialization values is not permitted. ● Transferring data values to the PAC is not permitted. ● Restoring the project to the PAC from a saved backup is not permitted. ● Forcing inputs, outputs or internal bits is not permitted. ● Commanding the safety program to enter maintenance mode is permitted.
Disabled	✓	✓	User cannot access the project.

Assigning a Preconfigured User

The super user can assign a preconfigured user, derived from a preconfigured profile, to a specific user in the **Users** tab of the **Security Editor**. The following preconfigured user selections are available:

- safety_user_Adjust
- safety_user_Debug
- safety_user_Operate
- safety_user_Program
- user_Adjust
- user_Debug
- user_Operate
- user_Program

Refer to the topic *User Functions (see EcoStruxure™ Control Expert, Security Editor, Operation Guide)* for more information about how a super user can assign a preconfigured profile to a user.

Access rights

Introduction

Control Expert access rights are classified in the following categories:

- project services
- adjustment/debugging
- libraries
- global modification
- elementary modification of a variable
- elementary modification of DDT compound data
- elementary modification of a DFB type
- elementary modification of a DFB instance
- bus configuration editor
- input/output configuration editor
- runtime screens
- cyber security
- safety

This topic presents the access rights available for each of the preconfigured user profiles.

Project services

The access rights for this category are as follows:

Access right	Preconfigured User Profile							
	Adjust	Safety_ Adjust	Debug	Safety_ Debug	Operate	Safety_ Operate	Program	Safety_ Program
Create a new project	–	–	–	–	–	–	✓	✓
Open an existing project	✓	✓	✓	✓	✓	✓	✓	✓
Save a project	–	–	–	–	–	–	✓	✓
SaveAs a project	✓	✓	✓	✓	✓	✓	✓	✓
Import a project	–	–	–	–	–	–	✓	✓
Build off-line	–	–	–	–	–	–	✓	✓
Build on-line STOP	–	–	–	–	–	–	✓	✓
Build on-line RUN	–	–	–	–	–	–	✓	✓
Start, stop or initialize the PAC*	✓	–	✓	–	–	–	✓	✓

* Only process tasks are started or stopped. For a non-safety PAC, this means the PAC is started or stopped. For an M580 safety PAC, this means that tasks other than the SAFE task are started or stopped.

✓ : Included

– : not included

Access right	Preconfigured User Profile							
	Adjust	Safety_ Adjust	Debug	Safety_ Debug	Operate	Safety_ Operate	Program	Safety_ Program
Update init values with current values (only non-safe data)	–	–	✓	–	–	–	✓	✓
Transfer project from PAC	✓	✓	✓	✓	✓	✓	✓	✓
Transfer project to PAC	✓	✓	✓	✓	–	–	✓	✓
Transfer data values from file to PAC (only non-safe data)	✓	–	✓	–	✓	–	✓	✓
Restore project backup in PAC	–	–	–	–	–	–	✓	✓
Save to project backup in PAC	–	–	–	–	–	–	✓	✓
Set address	✓	✓	✓	✓	✓	✓	✓	✓
Modify options	✓	✓	✓	✓	✓	✓	✓	✓
* Only process tasks are started or stopped. For a non-safety PAC, this means the PAC is started or stopped. For an M580 safety PAC, this means that tasks other than the SAFE task are started or stopped. ✓ : Included – : not included								

Adjustment/Debugging

The access rights for this category are as follows:

Access right	Preconfigured User Profile							
	Adjust	Safety_ Adjust	Debug	Safety_ Debug	Operate	Safety_ Operate	Program	Safety_ Program
Modify variable values	✓	–	✓		✓		✓	✓
Modify safety variable values	–	✓	–	✓	–	✓	–	✓
Force internal bits	–	–	✓	–	–	–	✓	✓
Force outputs	–	–	✓	–	–	–	✓	✓
Force inputs	–	–	✓	–	–	–	✓	✓
Task management	–	–	✓	–	–	–	✓	✓
SAFE Task management	–	–	–	✓	–	–	–	✓
✓ : Included – : not included								

Access right	Preconfigured User Profile							
	Adjust	Safety_ Adjust	Debug	Safety_ Debug	Operate	Safety_ Operate	Program	Safety_ Program
Task cycle time modification	✓	–	✓		✓	–	✓	✓
SAFE Task cycle time modification	–	✓	–	✓	–	✓	–	✓
Suppress message in viewer	✓	✓	✓	✓	✓	✓	✓	✓
Debug the executable	–	–	✓	✓	–	–	✓	✓
Replace a project variable	–	–	–	–	–	–	✓	✓
Replace a safety project variable	–	–	–	–	–	–	–	✓
✓ : Included – : not included								

Libraries

The access rights for this category are as follows:

Access right	Preconfigured User Profile							
	Adjust	Safety_ Adjust	Debug	Safety_ Debug	Operate	Safety_ Operate	Program	Safety_ Program
Create libraries or families	–	–	–	–	–	–	✓	✓
Create safety libraries or families	–	–	–	–	–	–	–	✓
Delete libraries or families	–	–	–	–	–	–	✓	✓
Delete safety libraries or families	–	–	–	–	–	–	–	✓
Put an object into library	–	–	–	–	–	–	✓	✓
Put an object into safety library	–	–	–	–	–	–	–	✓
Delete an object from library	–	–	–	–	–	–	✓	✓
Delete an object from safety library	–	–	–	–	–	–	–	✓
✓ : Included – : not included								

Access right	Preconfigured User Profile							
	Adjust	Safety_ Adjust	Debug	Safety_ Debug	Operate	Safety_ Operate	Program	Safety_ Program
Get an object from a library	–	–	–	–	–	–	✓	✓
Get an object from the safety library	–	–	–	–	–	–	–	✓
✓ : Included – : not included								

Global modification

The access rights for this category are as follows:

Access right	Preconfigured User Profile							
	Adjust	Safety_ Adjust	Debug	Safety_ Debug	Operate	Safety_ Operate	Program	Safety_ Program
Modify the documentation	✓	✓	✓	✓	✓	✓	✓	✓
Modify the functional view	–	–	–	–	–	–	✓	✓
Modify the animation tables	✓	✓	✓	✓	✓	✓	✓	✓
Modify constants value	✓	–	✓	–	✓	–	✓	✓
Modify safety constants value	–	✓	–	✓	–	✓	–	✓
Modify the program structure	–	–	–	–	–	–	✓	✓
Modify the safety program structure	–	–	–	–	–	–	–	✓
Modify program sections	–	–	–	–	–	–	✓	✓
Modify safety program sections	–	–	–	–	–	–	–	✓
Modify project settings	–	–	–	–	–	–	✓	✓
✓ : Included – : not included								

Elementary modification of a variable

The access rights for this category are as follows:

Access right	Preconfigured User Profile							
	Adjust	Safety_ Adjust	Debug	Safety_ Debug	Operate	Safety_ Operate	Program	Safety_ Program
Variable add/remove	–	–	–	–	–	–	✓	✓
Safety Variables add/remove	–	–	–	–	–	–	–	✓
Variable main attributes modifications	–	–	–	–	–	–	✓	✓
Safety Variables main attributes modifications	–	–	–	–	–	–	–	✓
Variable minor attributes modifications	✓	–	✓	–	✓	–	✓	✓
Safety Variables minor attributes modifications	–	✓	–	✓	–	✓	–	✓
✓ : Included – : not included								

Elementary modification of DDT compound data

The access rights for this category are as follows:

Access right	Preconfigured User Profile							
	Adjust	Safety_ Adjust	Debug	Safety_D ebug	Operate	Safety_ Operate	Program	Safety_ Program
DDT add/remove	–	–	–	–	–	–	✓	✓
DDT modifications	–	–	–	–	–	–	✓	✓
✓ : Included – : not included								

Elementary modification of a DFB type

The access rights for this category are as follows:

Access right	Preconfigured User Profile							
	Adjust	Safety_ Adjust	Debug	Safety_ Debug	Operate	Safety_ Operate	Program	Safety_ Program
DFB type add/remove	–	–	–	–	–	–	✓	✓
Safety DFB type add/remove	–	–	–	–	–	–	–	✓
DFB type structure modification	–	–	–	–	–	–	✓	✓
Safety DFB type structure modification	–	–	–	–	–	–	–	✓
DFB type sections modification	–	–	–	–	–	–	✓	✓
Safety DFB type sections modification	–	–	–	–	–	–	–	✓
✓ : Included – : not included								

Elementary modification of a DFB instance

The access rights for this category are as follows:

Access right	Preconfigured User Profile							
	Adjust	Safety_ Adjust	Debug	Safety_ Debug	Operate	Safety_ Operate	Program	Safety_ Program
DFB instance modification	–	–	–	–	–	–	✓	✓
Safety DFB instance modification	–	–	–	–	–	–	–	✓
DFB instance minor attributes modification	✓	–	✓	–	✓	–	✓	✓
Safety DFB instance minor attributes modification	–	✓	–	✓	–	✓	–	✓
✓ : Included – : not included								

Bus configuration editor

The access rights for this category are as follows:

Access right	Preconfigured User Profile							
	Adjust	Safety_ Adjust	Debug	Safety_ Debug	Operate	Safety_ Operate	Program	Safety_ Program
Modify the configuration	–	–	–	–	–	–	✓	✓
Modify the safety configuration	–	–	–	–	–	–	–	✓
I/O sniffing	–	–	–	–	–	–	✓	✓
✓ : Included – : not included								

Input/output configuration editor

The access rights for this category are as follows:

Access right	Preconfigured User Profile							
	Adjust	Safety_ Adjust	Debug	Safety_ Debug	Operate	Safety_ Operate	Program	Safety_ Program
Modify the I/O configuration	–	–	–	–	–	–	✓	✓
Modify the safety I/O configuration	–	–	–	–	–	–	–	✓
Adjust the I/O	✓	–	✓	–	✓	–	✓	✓
Adjust the safety I/O	–	✓	–	✓	–	✓	–	✓
Save_param	–	–	✓	–	–	–	✓	✓
Restore_param	–	–	✓	–	–	–	✓	✓
✓ : Included – : not included								

Runtime screens

The access rights for this category are as follows:

Access right	Preconfigured User Profile							
	Adjust	Safety_ Adjust	Debug	Safety_ Debug	Operate	Safety_ Operate	Program	Safety_ Program
Modify screens	–	–	–	–	–	–	✓	✓
Modify messages	–	–	–	–	–	–	✓	✓
Add/remove screens or families	–	–	–	–	–	–	✓	✓
✓ : Included – : not included								

Cyber Security

The access rights for this category are as follows:

Access right	Preconfigured User Profile							
	Adjust	Safety_ Adjust	Debug	Safety_ Debug	Operate	Safety_ Operate	Program	Safety_ Program
Create or modify application password	–	–	–	–	–	–	✓	✓
Enter Maintenance mode	–	✓	–	✓	–	✓	–	✓
Adapt Auto-Lock timeout	✓	✓	✓	✓	✓	✓	✓	✓
✓ : Included – : not included								

Safety

The access rights for this category are as follows:

Access right	Preconfigured User Profile							
	Adjust	Safety_ Adjust	Debug	Safety_ Debug	Operate	Safety_ Operate	Program	Safety_ Program
Enter Maintenance mode	–	✓	–	✓	–	✓	–	✓
✓ : Included – : not included								

Section 13.10
Modifications to Control Expert for the M580 Safety System

Introduction

This section describes Control Expert functionality that has been modified or limited for the M580 safety system.

What Is in This Section?

This section contains the following topics:

Transferring and Importing M580 Safety Projects and Code in Control Expert

Transferring a Safety Project from Control Expert to the Safety PAC

You can use the **PLC → Transfer Project to PLC** command to transfer the project from Control Expert to the PAC when:

- Control Expert is connected in programming mode *(see EcoStruxure™ Control Expert, Operating Modes)* to the M580 safety PAC, and
- A project is open in Control Expert, and
- All PAC tasks are in STOP state.

NOTE: You can transfer a safety application only to a safety PAC. A safety application cannot be transferred to a non-safety PAC.

Transferring a Safety Project from the Safety PAC to Control Expert

Similarly, you can use the **PLC → Transfer Project from PLC** command to transfer the project from the PAC to Control Expert when:

- Control Expert is connected in programming mode *(see EcoStruxure™ Control Expert, Operating Modes)* to the M580 safety PAC, and
- No project is open in Control Expert.

You can transfer content relating to any task (SAFE, MAST, FAST, AUX0, or AUX1) in either safety or maintenance operating mode.

Importing Projects and Code Sections in Control Expert

Control Expert Safety supports to import both entire projects (via **File → Open**) and code sections (via **Tasks → Import...** or **Sections → Import...**), subject to the following conditions:

- Only functions or function block types, which exist in either the safety library (**Data Scope Editor → <Libset> → Safety**) or the custom library (**Data Scope Editor → <Libset> → Custom Lib**, can be included in a code section handled by the SAFE task.
- Only functions or function block types, which exist in libraries other than the safety library, can be included in a non-SAFE code section handled by a process task (MAST, FAST, AUX0, or AUX1) .

Saving & Restoring Data Between a File and the PAC

Save and Restore Functions for Non-Safety Data

Control Expert supports the commands **PLC → Save Data from PLC to File** and **PLC → Restore Data from File to PLC** for process and global area data. However, the data saved and restored does not include variables and function block instances created in the safe namespace.

For information on how to use these commands for non-safe data, refer to the topic *Save/Restore Data Between a File and the PLC* in the *EcoStruxure™ Control Expert Operating Modes* document.

CCOTF for an M580 Safety PAC

Changing a Configuration on the Fly

The change configuration on the fly (CCOTF) feature makes it possible to change a Control Expert configuration while the PAC is running. Supported functions can include:

- Adding a drop.
- Adding an I/O module.
- Deleting an I/O module.
- Editing the configuration of an I/O module, including:
  - Change a parameter setting.
  - Add a channel function.
  - Delete a channel function.
  - Change a channel function.

NOTE: CCOTF functions do not apply to CIP Safety devices.

The CCOTF feature is enabled by selecting **Online modification in RUN or STOP** in the **Configuration** tab of the CPU module.

The basic functionality of CCOTF has been implemented in the M580 safety PAC, with the limitations described below.

For a full description of CCOTF, refer to the *Modicon M580 Change Configuration on the Fly User Guide*.

Limitations of CCOTF for an M580 Safety PAC

The CCOTF feature is implemented in the M580 safety PAC, with limitations that are based on the specific function and type of I/O module, as follows:

CCOTF Function	I/O Module Type & Operating Mode			
	Non-Interfering I/O		SIL3 Safety I/O	
	Maintenance mode	Safety mode	Maintenance mode	Safety mode
Add drop	✓	✓	✓[1]	✓
Add module	✓	✓	✓[1]	X
Delete module	✓	✓	✓	X
Edit I/O module configuration	✓	✓	X	X
✓: Allowed X: Not allowed 1.Adding both a drop and a safety module requires two CCOTF sessions: one CCOTF session to add the drop, the second CCOTF session to add the safety module. These actions cannot be performed in a single CCOTF session.				

NOTE: Edits made in a single CCOTF session can relate only to a single task (SAFE, MAST, FAST, AUX0, or AUX1).

Changes to M580 Safety PAC Tools

Introduction

The M580 safety PAC supports the use of several related tools. Some of these tools have been modified for use together with the M580 safety PAC. This topic addresses some of these tools.

Memory Usage

The **Memory Usage** screen presents the following information:

- the physical distribution of the PAC (internal memory and memory card)
- the space taken up in the memory by a project (data, program, configuration, system)

For the M580 safety PAC, this screen specifically provides two new parameters – **Safety Declared Data** and **Safety Executable code** – which are described below.

NOTE: You can also use the **Pack** command in this screen to reorganize the memory where possible.

For more information, refer to the topic *Memory Usage* in the *EcoStruxure™ Control Expert Operating Modes* user manual.

For the M580 safety PAC, the following parameters are displayed:

Parameter	Description
User Data	This field indicates the memory space (in words) taken up by user data (objects relating to configuration): ● **Data**: located data associated with the processor (%M, %MW, %S, %SW, etc.) or the input/output modules. ● **Declared Data**: unlocated data (declared in the process data editor) saved after power cut. ● **Unsaved Declared Data**: unlocated data (declared in the process data editor) not saved after power cut. ● **Safety Declared Data**: unlocated data (declared in the safety data editor) not saved after power cut.
User Program	This field indicates the memory space (in words) taken up by the project program: ● **Constants**: static constants associated with the processor (%KW) and the input/output modules; initial data values. ● **Executable code**: executable code of the process area part of the project program, EFs, EFBs and DFB types. ● **Upload information**: information for uploading a project (graphic code of languages, symbols, etc.). ● **Safety Executable code**: executable code of the safety area part of the project program, EFs, EFBs and DFB types.

Parameter	Description
Other	This field indicates the memory space (in words) taken up by other data relating to the configuration and the project structure: ● **Configuration**: other data relating to configuration (hardware configuration, software configuration). ● **System**: data used by the operating system (task stack, catalogs, etc.), ● **Diagnostic**: information relating to process or system diagnostics, diagnostics buffer. ● **Data Dictionary**: Dictionary of symbolized variables with their characteristic (address, type, and so forth).
Internal Memory	This field shows the organization of the PAC internal memory. It also indicates the memory space available (**Total**), the largest possible contiguous memory space (**Greatest**) and the level of fragmentation (due to online modifications).

Event Viewer

Event Viewer is an MS-Windows utility that captures events logged by Control Expert. You can use *Event Viewer* to display a history of logged events.

Access *Event Viewer* in MS-Windows in the *Administrative Tools* folder of the *Control Panel*. When you open the utility, select **Show Action Pane**, then click **Create Custom View** to open that dialog. There, you can create a custom view for Control Expert events.

NOTE: In the **Create Custom View** dialog, first select **By source**, then select **TraceServer** as the source to display Control Expert events.

Chapter 14
CIP Safety

Overview

This chapter describes IEC 61784-3 CIP Safety communications supported by the BMEP58•040S M580 standalone safety CPUs.

What Is in This Chapter?

This chapter contains the following sections:

Section 14.1
Introducing CIP Safety for M580 Safety PACs

CIP Safety Communication

Introduction

Both the BMEP582040S and the BMEP584040S standalone safety CPUs support CIP Safety (IEC 61784-3) communication, and can use this protocol to establish a connection with a CIP Safety device over EtherNet/IP.

CIP Safety uses the consumer-producer mechanism for the exchange of data between safe nodes over EtherNet/IP. (DeviceNet or Sercos III communication is not supported.) The CPU acts in the role of originator that establishes a Unicast (one-to-one) EtherNet/IP connection with each target safety device. The CPU can establish a CIP Safety connection with target devices that support the CIP Safety protocol, and a CIP (non-safety) connection with target devices that support the CIP protocol.

As is the case with all safety PACs, the CIP safety CPU and Copro double execute the CIP safety stack in parallel and compare processing results.

Supported Architectures

Standalone M580 safety CPUs support CIP Safety devices located in DIO clouds.

NOTE: At present, no CIP Safety devices exist that support RSTP and can be installed in an eX80 rack. Thus, CIP Safety devices cannot presently be connected to the dual Device Network ports of the CPU, but can be connected to the CPU Service port.

DIO clouds require only a single (non-ring) copper connection, and can be connected to:

- a BMENOS0300 network option switch module.
- the service port of the CPU.
- the service port of a BM•CRA312•0 eX80 Ethernet I/O adapter module on an RIO drop.
- a copper port of an Ethernet dual ring switch.

NOTE: When a CIP Safety device is connected to the service port of a BM•CRA312•0 eX80 Ethernet I/O adapter module on an RIO drop, the target CIP safety device may not start automatically while the CRA is loading its configuration. To cause the CIP Safety connections to open as intended, you may need to manage the control bit of the CIP Safety connection in the target DDDT (CTRL_IN or CTRL _OUT) by toggling it from False to True after the BM•CRA312•0 finishes loading its configuration.

As with all equipment located in DIO clouds, CIP Safety devices are not scanned as part of the main RIO ring, and their connection status is not reflected in the CPU LEDs.

For additional information on DIO clouds, refer to the *Modicon M580 Standalone System Planning Guide for Frequently Used Architectures* and the *Modicon M580 System Planning Guide for Complex Topologies*.

Configuration Overview

Configuring CIP Safety communications involves three separate configuration tasks:

- Configure the M580 Safety Standalone CPU with CIP Safety settings in Control Expert *(see page 341)*. This includes the creation of an Originator Unique Network Identifier (OUNID) that uniquely identifies the CPU. The OUNID is created in Control Expert as a concatenation of two components:
  - Safety Network Number (SNN): An identifier for the CPU created in Control Expert.
  - Main IP address of the CPU, entered in Control Expert as part of the CPU IP address settings.

  Schneider Electric recommends configuring the CPU OUNID setting one time only, in the initial configuration. If you subsequently change the OUNID setting, you would also need to reconfigure all CIP Safety devices that are connected to the CPU.
- Configure the CIP Safety device *(see page 345)*, using a safety network configuration tool (SNCT) provided by the device vendor. This includes two tasks:
  - Creation of a Safety Configuration Identifier (SCID): Also known as the configuration signature, the SCID is created in the SNCT and used by Control Expert when configuring the CIP Safety connection between the originator (CPU) and target (CIP Safety device).
  - Assignment of a Safety Network Number (SNN): The SNN is typically created for the CIP Safety device by Control Expert and is assigned to the device by the SNCT.
- Configure the CIP Safety connection between the CPU and the CIP Safety device *(see page 347)*. The connection is identified by a TUNID that is created using the device connection DTM in Control Expert using a CIP Safety DTM, which can be based on a manufacturer provided EDS file or used alone if no EDS file is available.

Managing CIP Safety Device Connections

The CIP Safety CPU establishes a connection to a configured CIP device, and then manages the connected device. Because Control Expert supports both the CIP protocol and the CIP Safety protocol, it can manage CIP connections to:

- CIP devices, which implement CIP over EtherNet/IP, but not CIP Safety.
- CIP safety devices, which implement CIP Safety over EtherNet/IP, but not CIP.
- CIP hybrid devices, which implement both CIP and CIP Safety over EtherNet/IP.

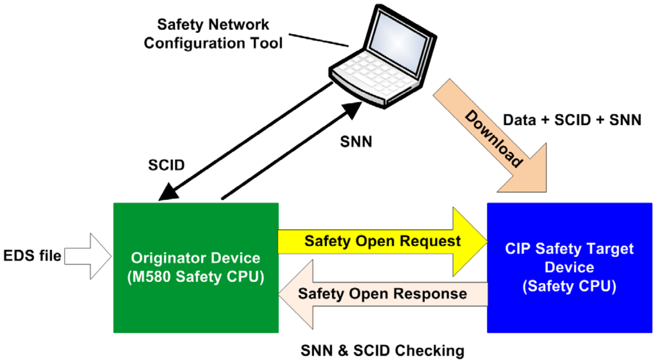
NOTE: A CIP device and a CIP Safety device each requires a single DTM for configuration. A CIP hybrid device–which incorporates both the CIP and CIP Safety protocols–requires two DTMs: one configured as a CIP device; one configured as a CIP Safety device.

Establishing an Originator -> Target Connection

The M580 standalone CPU uses only the Type 2 Safety Open request to establish a connection with a CIP Safety device. A Type 2 Safety Open connection can be made to a safety device only after the device has been configured by an SNCT. In cases where the CIP Safety device is a third-party product, Control Expert does not possess and cannot download a configuration file to a CIP Safety device and cannot be used as an SNCT.

NOTE: By contrast, a Type 1 Safety Open connection both provides the safety device its configuration settings and also establishes the connection. M580 CIP Safety CPUs do not support the Type 1 Safety Open connection request.

The following diagram presents an overview of the how a CIP Safety connection is created between the CPU as connection originator and the CIP Safety device as connection target:



In this diagram, the following events occur:

1. Control Expert uses a vendor provided EDS file as a basis for creating a DTM for the connection between the CPU and CIP Safety device.
2. The device SNN is created in Control Expert, then entered into the SNCT.
3. The SNCT creates the SCID for the device, which is entered into Control Expert as part of the connection configuration.
4. The SNCT downloads to the device its configuration settings, the SCID created by the SNCT, and the SNN created by Control Expert for the connection.
5. The CPU as originator sends the device a Type 2 Safety Open Request.
6. The CIP Safety device sends a Safety Open Response to the CPU.
7. If the checksums in both the request and response match, the connection is established.

Section 14.2
Configuring the M580 CIP Safety CPU

Configuring the CPU OUNID

CPU as Originator

Use the **Safety** tab *(see Modicon M580, Hardware, Reference Manual)* of the standalone M580 safety CPU to configure the CPU as a CIP Safety originator, by assigning it an Originator Unique Network Identifier (OUNID).

Each OUNID is a 10 byte concatenated hexadecimal value, consisting of a:

- Safety Network Number (6 bytes)
- IP Address (4 bytes)

NOTE: Changes to the OUNID can be made only offline. After the changed configuration is built, the application can be downloaded to the PAC.

Safety Network Number

The Safety Network Number component of the OUNID can be auto-generated by Control Expert, or user-generated by manual input. Create the SNN::

- Automatically, by selecting **Time-based**, then clicking the **Generate** button. The auto-generated value appears in the **Number** field.
- Manually, by selecting **Manual**, then inputting a 6 byte hexadecimal string in the **Number** field.

NOTE: The user should assign a unique SNN to each M580 CPU originator connected to the same safety network.

IP Address

This read-only setting is automatically input, based on the configured **Main IP address** CPU setting in the **IPConfig** tab *(see Modicon M580, Hardware, Reference Manual)*.

OUNID

After the OUNID is created, it is used as a parameter in the Type 2 SafetyOpen Request, *(see page 361)* establishing a connection between the CPU as originator, and the CIP Safety device as target.

Section 14.3
Configuring the CIP Safety Target Device

Overview

This section outlines the CIP Safety target device configuration process, including the configuration of the CIP Safety device using a vendor supplied configuration tool.

What Is in This Section?

This section contains the following topics:

CIP Safety Device Configuration Overview

Introduction

Configuring the CIP Safety target device, includes two tasks:

- Configure the CIP Safety device settings *(see page 345)* using a vendor supplied safety network configuration tool (SNCT).
- Configure the connection between the CIP Safety CPU originator and the CIP Safety target device, using a DTM in Control Expert. The DTM can be:
  - based on a vendor supplied EDS file.
  - a Control Expert generic DTM, if no EDS file is available.

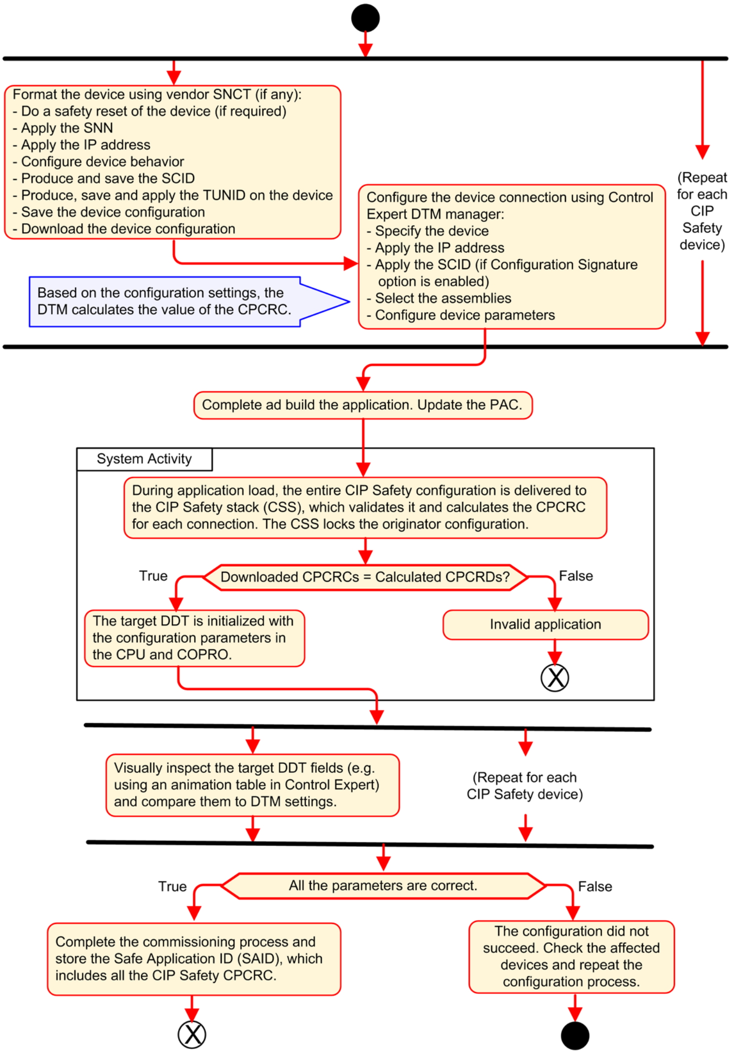
Dual Configuration Checking

The following two processes, together, can provide a high integrity confirmation that the configuration created using the Control Expert software was correctly downloaded to and saved in the M580 CIP Safety CPU as originator:

- A user-performed visual comparison (after the application download is complete) of the CIP Safety connection configuration parameters displayed in the target DDDT against the same parameters displayed in the target DTM.
- An automatic comparison, performed by the CPU and Copro, of the connection parameter CRC CPCRC calculated by the DTM against the CPCRC calculated by the CIP Safety stack (CSS) running in the CPU and Copro.

Configuration Process Overview

The CIP Safety device configuration and validation process:


Format the device using vendor SNCT (if any):
- Do a safety reset of the device (if required)
- Apply the SNN
- Apply the IP address
- Configure device behavior
- Produce and save the SCID
- Produce, save and apply the TUNID on the device
- Save the device configuration
- Download the device configuration
(Repeat for each CIP Safety device)
Configure the device connection using Control Expert DTM manager:
- Specify the device
- Apply the IP address
- Apply the SCID (if Configuration Signature option is enabled)
- Select the assemblies
- Configure device parameters
Based on the configuration settings, the DTM calculates the value of the CPCRC.
Complete ad build the application. Update the PAC.
System Activity
During application load, the entire CIP Safety configuration is delivered to the CIP Safety stack (CSS), which validates it and calculates the CPCRC for each connection. The CSS locks the originator configuration.
True
Downloaded CPCRCs = Calculated CPCRDs?
False
The target DDT is initialized with the configuration parameters in the CPU and COPRO.
Invalid application
Visually inspect the target DDT fields (e.g. using an animation table in Control Expert) and compare them to DTM settings.
(Repeat for each CIP Safety device)
True
All the parameters are correct.
False
Complete the commissioning process and store the Safe Application ID (SAID), which includes all the CIP Safety CPCRC.
The configuration did not succeed. Check the affected devices and repeat the configuration process.

Configuring the CIP Safety Device Using a Vendor Provided Tool

Introduction

The CIP Safety target device is configured using a safety network configuration tool (SNCT). It is not configured using the Control Expert software. The SNCT is provided by the CIP Safety device vendor, and thus is device dependent.

Use the SNCT to:

- Configure and download to the device the necessary settings for device operation.
- Configure for the device, then copy and transfer to the Control Expert software, a device-specific Safety Configuration Identifier (SCID). The SCID is also referred to as the device Configuration Signature. It is used in Control Expert when configuring the Originator -> Target connection. *(see page 352)*
- Assign to the device its unique TUNID, consisting of a:
  - Safety Network Number (SNN) *(see page 351)*, and
  - Unique IP address.

  NOTE: The SNN is usually generated by the Control Expert configuration software (as part of the Originator -> Target connection configuration) and applied to the device. The IP address is entered both in the SNCT and in device connection DTM in Control Expert.

Configuring the SCID

The SCID is set in the SNCT and serves as the unique hexadecimal configuration identifier for the CIP Safety target device. It is a concatenation of a:

- Safety Configuration CRC (SCCRC): a cyclic redundancy check (CRC) value of the CIP Safety device configuration settings, consisting of 4 octets.
- Safety Configuration Time Stamp (SCTS): a date and time hexadecimal timestamp value that consists of 6 octets.

NOTICE

RISK OF UNINTENDED EQUIPMENT OPERATION

if you configure an M580 CPU as a CIP Safety originator, test and verify the CIP Safety functional behavior of the system before using CIP Safety communication to control the related safety function. After testing and verification are successfully completed, enable the CIP Safety target configuration signature (if one exists) in the Control Expert CIP Safety DTMs.

Failure to follow these instructions can result in equipment damage.

After creating the SCID using the SNCT, you can enter the elements of the SCID into the device DTM **Safety** tab in Control Expert:

- **ID**: Enter the SCCRC value.
- **Date**: Enter the date the SCID was created (mm/dd/yyyy).
- **Time**: Enter the time the SCID was created (hh/mm/ss/ms).

CIP Safety Device Configuration Sequence

The following sequence describes a typical CIP Safety device configuration process:

1. Obtain the device SNN (received from Control Expert).
2. Apply the SNN inside the vendor SNCT.
3. Perform a safety reset of the device (optional: if originator OUNID has changed since the last time the device has been connected).
4. Apply the TUNID into the device.
5. Determine the configuration settings that will control device behavior.
6. Configure the device with the vendor SNCT (safety network configuration tool).
7. Lock the configuration and verify the configuration accuracy.
8. Record and save the parameters to later use in the originator configuration (SCID, Assembly numbers, IP address, and so forth).
9. Save a copy of the device configuration for further use (for example, if the device needs to be replaced).

Section 14.4
Configuring Safety Device DTMs

Overview

This section describes the configuation of target safety devices, and their connections to the originator CPU, using DTMs in Control Expert.

What Is in This Section?

This section contains the following topics:

Working with DTMs

Working with DTMs

Configuring the connection between the CPU originator and the CIP Safety device target is performed using a DTM. Control Expert supports usage of the following DTMs, depending on the device profile:

- CIP Safety DTM: To configure a connection to a CIP Safety device. This can be done with or without a vendor EDS file.
- Generic DTM: To configure a standard (i.e. non-safety) connection to a device, based on a vendor EDS file.

The settings you enter using a DTM are stored in Control Expert in the T_CIP_SAFETY_CONF DDDT *(see page 372)*, and used by the SafetyOpen Type 2 request *(see page 361)* to establish a connection between the originator CPU and the target device.

When an EDS File is Available

When a vendor EDS file is available for a device, use it to create a new DTM and add it to the **DTM Catalog** in Control Expert as follows:

Step	Action
1	In Control Expert, select **Tools → DTM Browser**.
2	In the **DTM Browser**, right click on the CPU DTM (BMEP58_ECPU_EXT) to open the context menu.
3	Navigate to and select **Device menu → Additional functions → Add EDS to library**. The **EDS Addition** wizard opens.
4	Refer to the Add an EDS File to the Hardware Catalog topic *(see EcoStruxure™ Control Expert, Operating Modes)* for step by step instructions on how to complete the process of adding an EDS file to the DTM Catalog.

After a DTM is added to the **DTM Catalog**, you can add it to your Control Expert project.

When an EDS File is Not Available

Control Expert includes a Generic Safety DTM in the **DTM Catalog**. You can use it to configure a CIP Safety device, when an EDS file is not available for that device.

Hybrid Devices

A hybrid device is a single device that supports both safety and standard connections. When you add a hybrid device to the **DTM Catalog** using the **Add EDS to library** command, two DTMs are created in the **DTM Catalog** for the device: a standard DTM and a safety DTM.

When you add a hybrid device to your project, you need to configure both the standard DTM and the safety DTM for the single device.

Adding a DTM to a Control Expert Project

To add a DTM to a Control Expert project:

Step	Action
1	In the **DTM Browser**, right click the CPU DTM (BMEP58_ECPU_EXT) and select **Add...** The **Add** dialog opens.
2	Select the DTM you want to add. It can be: ● A CIP Safety DTM created from a vendor CIP Safety device EDS file, or ● A CIP Safety DTM without a vendor EDS file.
3	Click **Add DTM**. The selected DTM appears in the **DTM Browser** below the CPU DTM.
4	Right click on the new DTM, and select **Open**. The DTM configuration window opens

Configuring the DTM

The CIP Safety DTM, created with or without a vendor EDS file present a similar series of configuration screens in Control Expert:

Navigation Tree / Configuration Tabs			DTM Type	
			With Vendor EDS	Without Vendor EDS
<Top Node>			✓	✓
	General Node			
		Device tab	✓	X
		Safety tab	✓	✓
	<Connections>			
		Connection tab	✓	✓
		Configuration Settings tab	✓	X
		Configuration verification tab	✓	✓
< > indicates user-defined name. ✓ = included X = not included				

The following topics describe the several configuration tabs presented in Control Expert for each type of DTM.

Safety Device DTM - File and Vendor Information

Introduction

The CIP Safety DTM, created from a vendor EDS file or not, presents a description of the source EDS file and the device vendor. For a:

- CIP Safety DTM created from a vendor EDS file: this information is read-only and is accessed by selecting the <Top Node> of the DTM navigation tree (left pane).
- CIP Safety DTM created without a vendor EDS file: this information appears in two separate locations:
  - <Top Node> selection displays the read-only EDS file information.

    NOTE: The EDS file reference is an internal generic safety EDS file, with Schneider Electric the vendor, which is used by Control Expert to create the CIP Safety DTM.
  - **General** → **Device** tab selection displays the editable vendor information.

EDS File Information

The EDS file information includes the following read-only data:

- Description
- File Creation Date
- File Creation Time
- Last Modification Data
- Last Modification Time
- EDS Revision

Vendor Information

The following vendor information is read-only for a CIP Safety DTM created from a vendor EDS file:

- Vendor Name
- Device Type
- Major Revision
- Minor Revision
- Product Name

The following vendor information is read-write for a CIP Safety DTM created without a vendor EDS file:

- Vendor ID
- Product Type
- Product Code
- Major Revision
- Minor Revision

NOTE: For DTM configurations made without the aid of an EDS file, enter vendor information settings with information provided by the vendor. By default, DTM vendor values are set to 0, and 0 values are not supported.

Safety Device DTM - Safety Network Number

Safety Network Number

Use the **General** → **Safety** tab of the CIP Safety device DTM to configure a Safety Network Number (SNN) for the safety device. The SNN is used to set the Target Unique Network Identifier (TUNID). TUNID identifies the CIP Safety device, and is an essential component of the Type 2 SafetyOpen Request *(see page 361)* issued by the originator CPU to initiate a CIP Safety connection.

Configuring the SNN

The SNN is a hexadecimal value that is part of both the CIP Safety connection configuration (configured using Control Expert) and the CIP Safety device configuration (configured using an SNCT). Typically the SNN is generated in Control Expert, and is copied to (or re-entered in) the SNCT. The SNCT then produces the TUNID based on SNN and IP address and transfers this value to the CIP Safety device.

It is also possible to send the SNN directly from the CIP Safety connection DTM in Control Expert to the target device *(see page 368)*.

To configure the SNN:

Step	Action
1	In the **General** → **Safety** tab, click the ellipsis (...) button. The **Safety Network Number** dialog opens.
2	In the **Safety Network Number** dialog, select one of the following: ● **Time-Based**:To generate a hex value based on the month, day, year, hour, minute, second and millisecond at the time of generation. ● **Manual**: To generate a value based on an input decimal value of 1 to 9999, which is concatenated with two hexadecimal values, as follows: ❍ word 1: 0004 (fixed) ❍ word 2: 0000 (fixed) ❍ word 3: 0001...270F (the hexadecimal value of the 1...9999 input value) ● **Vendor Specific**: A vendor specific identifier based on 3 input hexadecimal words: ❍ word 1: 05B5...2DA7 (from vendor) ❍ word 2: 0000 (fixed) ❍ word 3: 0001...270F (from vendor) ● A directly entered hex value (typed or pasted), consisting of: ❍ word 1: 2DA8...FFFE ❍ words 2 & 3: 00000000...05265BFF
3	For a Time-Based, Manual or Vendor Specific format, click **Generate**. If you directly entered a hex value, click **Set.**
4	Click **OK** to save the SNN and close the dialog. The SNN appears in the **Safety network Number** field.

Configuring the SCID

The SCID, also called the Configuration Signature, it is set in the vendor provided safety network configuration tool (SNCT) and represents the unique hexadecimal configuration identifier for the CIP safety device. It is a concatenation of:

- The Safety Configuration CRC (SCCRC): This is a cyclic redundancy check (CRC) value of the safety device configuration settings, in the form of a hex value consisting of 4 octets.
- Safety Configuration Time Stamp (SCTS): This is a date and time hexadecimal value timestamp that consists of 6 octets.

To input the SCID:

Step	Action
1	Obtain from the device configuration made using the SNCT, the following: ● The SCCRC ● The date (mm/dd/yyyy), time (hh/mm/ss/ms) the SNCT configuration was performed.
2	Select **Configuration Signature**.
3	Enter the SCCRC into the **ID** field.
4	Enter the date and time values into the **Date** and **Time** fields.

NOTE: If you configure safety connections with an SCID = 0 ("configure SCID disabled"), note that you are responsible for verifying that the M580 safety originator and the CIP Safety targets have the correct configurations.

Safety Device DTM - Verify and Validate Configuration

Visual Verification of DTM Configuration

Use the **General → Configuration verification** tab for the CIP Safety DTM, created with or without a vendor EDS file, to compare the parameters defined in this DTM (and displayed in this tab) with the parameters set in to the device target DDDT. You can do this using an animation table in Control Expert, when Control Expert is operating in connected mode and is connected to the CPU.)

NOTE: After an application download, you need to visually verify for each CIP Safety target that all CIP safety configuration parameters downloaded in the M580 originator for that target are identical to the ones configured in the target DTM. You can do this by comparing configuration parameters displayed in the CIP Safety target DDDT (using an animation table with Control Expert in connected mode) with the ones configured in the DTM and displayed in the Configuration verification tab.

Validating the Downloaded Configuration

After all CIP Safety configurations are download, user testing is the means by which all downloads are validated. One of the validation tests is to test the safety connection configurations after they are applied in an originator to confirm the target connection is operating as intended.

Safety Device DTM - I/O Connections

Introduction

The CIP Safety DTM, created with or without a vendor EDS file, presents safety connection nodes. Both safety input and safety output nodes are supported, according to the features of a specific device. The **Connection** tab presents the connection parameters for the selected input connection or output connection.

For DTMs created with a vendor EDS file, default connections are pre-selected. You can use the **Remove Connection** and **Add Connection** commands to adapt the connection settings to your application’s requirements.

Safety Input Connection Settings

Each safety input connection presents the following parameters:

- **Input Size** (Read-Write): The size of input data configured in the CIP Safety device, in bytes. Set to 0 by default.
  NOTE: You need to replace the default value with vendor provide settings. The value 0 is not supported.
- **Requested Packet Interval** (Read-Write): RPI represents the connection refresh period. Set equal to the (SAFE task period)/2 by default.
  NOTE: The SAFE task period (Tsafe) is set in the **Properties of SAFE** dialog (**Project Browser** → **Tasks** → **SAFE** → **Properties**) in Control Expert.
- **Network_Time_Expectation** (Read-Write): The time, in milliseconds, consumed by CIP safety communication *(see page 158)*. If the value is less than the *Minimum Network_Time_Expectation*, a detected error notification is displayed. By default, the value should be equal to *Minimum Network_Time_Expectation* * 1.5.
- **Timeout_Multiplier** (Read-Write): A component in producing the *Minimum Network_Time_Expectation*, the Timeout_Multiplier equals the Network_Time_Expectation /128 µSec. The *Minimum Network_Time_Expectation* = RPI * Timeout_Multiplier + Tsafe + 40.

- **Network_Transmission_max** (Read-Write): The worst case (oldest) age (in ms) of the data at the time when the packet is received by the consumer. This parameter is used only for calculating the minimum value to be entered into the Network_Time_Expectation (as described below). It can be refined by checking the value of *Max-data_age* in the consumer device after executing network CIP Safety communication for a significant period of time.
  This parameter is used in the calculation of the minimum value for parameter “Network Time Expectation” as follows:
  Min (Network Time Expectation) = RPI * Timeout_multiplier + Network_Transmission_max
  When Tsafe is modified, the value of this parameter should change and, consequently, the minimum value of *Network_Time_Expectation* also should change.
  The following attributes apply to this parameter:
  - Minimum value = 1- ms
  - Maximum value: = 5800 ms
  - Default value = 40 + Tsafe

The device DTM uses these input settings to make the following calculations:

Variable	Value		
	Default	Minimum	Maximum
Safeperiod (ms)	20	10	255
Input Repetition Packet Interval (ms)	RPI = Tsafe / 2	5	500
Timeout Multiplier	2	1	255
Network_Transmission_max (ms)	40 + 2 * Tsafe	10	5800
Network Time Expectation	Minimum Network_Time_Expectation * 1.5	RPI * Timeout_Multiplier + Network_Transmittion_max	5800

Safety Output Connection Settings

Each safety output connection presents the following parameters:

- **Output Size** (Read-Write): The size of output data configured in the CIP Safety device, in bytes. Set to 0 by default.
  NOTE: You need to replace the default value with vendor provide settings. The value 0 is not supported.

- **Requested Packet Interval** (Read Only): RPI represents the connection refresh period. Set equal to the SAFE task (Tsafe) period.
- **Network Time Expectation** (Read-Write): The time, in milliseconds, consumed by CIP safety communication *(see page 158)*. If the value is less than the *Minimum Network_Time_Expectation*, a detected error notification is displayed. By default, the value should be equal to *Minimum Network_Time_Expectation* * 1.5.
- **Timeout Multiplier** (Read-Write): A component in producing the *Minimum Network_Time_Expectation*, the Timeout Multiplier equals the Network_Time_Expectation /128 µSec. The *Minimum Network_Time_Expectation* = RPI * Timeout_Multiplier + Tsafe + 40.

- **Network_Transmission_max** (Read-Write): The worst case (oldest) age (in ms) of the data at the time when the packet is received by the consumer. This parameter is used only for calculating the minimum value to be entered into the Network_Time_Expectation (as described below). It can be refined by checking the value of *Max-data_age* in the consumer device after executing network CIP Safety communication for a significant period of time.
  This parameter is used in the calculation of the minimum value for parameter “Network Time Expectation” as follows:
  Min (Network Time Expectation) = RPI * Timeout_multiplier + Network_Transmission_max
  When Tsafe is modified, the value of this parameter should change and, consequently, the minimum value of *Network_Time_Expectation* also should change.
  The following attributes apply to this parameter:
  - Minimum value = 1- ms
  - Maximum value: = 5800 ms
  - Default value = 40 + 2*Tsafe

The device DTM uses these output settings to make the following calculations:

Variable	Value		
	Default	Minimum	Maximum
Safeperiod (ms)	20	10	255
Input Repetition Packet Interval (ms)	RPI = Tsafe	10	255
Timeout Multiplier	2	1	255
Network_Transmission_max (ms)	40 + 2 * Tsafe	10	5800
Network Time Expectation	Minimum Network_Time_Expectation * 1.5	RPI * Timeout_Multiplier + Network_Transmittion_max	5800

Safety Device DTM - I/O Connection Settings

Introduction

The CIP Safety DTM, when created without a vendor EDS file, includes the **Configuration Settings** tab of the connection node.

Use the **Configuration Settings** tab to complete the configuration of the connection between the CPU and the remote device.

Parameters

Configuration Settings tab includes the following parameters:

- **Input Instance**: The device specific assembly number associated with input (T→O) transmissions.
- **Output Instance**: The device specific assembly number associated with output (O→T) transmissions.
- **Configuration Instance**: The device specific assembly number associated with device configuration settings.

Safety Device IP Address Settings

Editing the M580 CPU Master DTM

The IP Address and the DHCP settings for a CIP Safety device are configurable in the M580 CPU Master DTM.

NOTE: Unlike other connection configuration settings for the target device, the device IP address is not set in the device connection DTM.

Accessing Safety Device IP Address Settings

Perform the following sequence of steps to edit IP Address and DHCP parameters of a CIP Safety device:

Step	Action
1	Disconnect Control Expert from the target device, and make the following edits offline.
2	In the Control Expert **DTM Browser**, double-click the M580 CPU Master DTM (BMEP58_ECPU_EXT) to open its configuration.
3	In the navigation tree, expand the Device List to see the associated local slave instances.
4	Select the device that corresponds to the CIP Safety device.
5	Select the **Address Setting** tab.

Configuring Safety Device IP Address Settings

In the **Address Setting** tab, edit these parameters for the selected safety device:

Field	Parameter	Description
IP Configuration	**IP Address**	Enter the IP address for the selected device.
	Subnet Mask	The device subnet mask. **NOTE:** Set the subnet mask so that the device IP address resides in the same subnet as the Main IP Address of the originator CPU.
	Gateway	The gateway address used to reach this device. The default of 0.0.0.0 indicates this device is located on the same subnet as the originator CPU.
Address Server	**DHCP for this Device**	● **Disabled** (default) de-activates the DHCP client in the device. ● **Enabled** activates the DHCP client in this device.
	Identified by	If DHCP service is enabled, select the device identifier type: ● **MAC Address.** ● **Device Name**.
	Identifier	If DHCP is enabled, and **Device Name** selected, enter the device name value.

For more information regarding configuring device parameters in the M580 CPU Master DTM, refer to the topic Device List Parameters *(see Modicon M580, Hardware, Reference Manual)*.

Section 14.5
CIP Safety Operations

Overview

This section describes CIP Safety operations.

What Is in This Section?

This section contains the following topics:

Transferring a CIP Safety Application from Control Expert to the PAC

Begin the Application Download

Use the **PLC → Transfer Project to PLC** command to begin the download.

If the PLC is configured with a pre-existing application (the “old application”), it is invalidated at the beginning of the new application download. If the old application includes configured devices, the PAC closes the connections to those devices.

End of Application Download

The CIP Safety configuration is written to the CPU CIP Safety Stack (CSS), which computes a Connection Parameter CRC (CPCRC) for each connection. Next, each CSS computed CPCRC is compared with the corresponding CPCRC stored in configuration and calculated by the target DTM. In the event of:

- CPCRC mismatch, the CSS rejects the application, and the PAC remains in NOCONF state.
- Equality:
  - The CPCRC and connection parameters values are copied into the corresponding target DDDT. *(see page 370)*
  - The CSIO_HEALTH parameter *(see page 378)* inside the CPU DDDT (T_BMEP58_ECPU_EXT) is set to 0.
  - The CIP Safety target device DDDT HEALTH bits *(see page 370)* are set to 0.
  - The PAC opens the connections of configured devices via Type 2 SafetyOpen Requests *(see page 361)*

In the case of a CPCRC mismatch, the CSS rejects the application, and the PAC remains in NOCONF state.

Recalculation of the Safety Application ID

The safety application ID (SAId) is a signature of the safe part of the Control Expert application. It is stored as system word %SW169 *(see page 396)*. The CSS computes a CRC on all instances of CPCRC. This CRC is added to the calculation of the SAId. Thus, a modification to the configuration of a CIP Safety target configuration changes the SAId value.

SafetyOpen Request Type 2 Structure

CIP SafetyOpen Type 2 Connection Frame Structure

The M580 standalone safety CPUs support CIP Safety connections created by SafetyOpen type 2 connection requests. The structure of the connection request frame is described below:

Parameter Name		Description
Connection Timeout Multiplier		Used by the consumer of a connection to determine if any of the three standard connections should timeout. The timeout value for the connection is defined as: Connection RPI * (CTM+1) * 4
O_to_T RPI		Originator to Target Requested Packet Interval.
T_to_O RPI		Target to Originator Requested Packet Interval.
Electronic Key.Vendor ID		Device Vendor Identifier
Electronic Key.Prod Type		Device Type
Electronic Key.Prod Code		Device Product Code
Electronic Key.Compatible/Major Rev		Major Revision
Electronic Key.Minor Rev		Minor Revision
SCID	Safety Configuration CRC	Safety Configuration Identifier: Provided by the safety network configuration tool (SNCT), it is used during commissioning, connection establishment and device replacement.
	Configuration Date	
	Configuration Time	
TUNID	TUNID Date	Target unique network Identifier: Identifies the target in the SafetyOpen request.
	TUNID Time	
	Target Node ID	
OUNID	OUNID Date	Originator unique Network Identifier: Identifies the originator in the SafetyOpen request.
	OUNID Time	
	Originator Node ID	
Ping_Interval_EPI_Multiplier		Defines the Ping_Count_Interval for the connection.
Time_Coord_Msg_Min_Multiplier		The minimum number of 128 μS increments it could take for a Time Coordination Message to travel from the consumer to the producer.
Network_Time_Expectation_Multiplier		The maximum age of safety data, measured in 128 μS increments, allowed by a consumer.
Timeout_Multiplier		The number of data production retries to include in the equation for unsuccessful connection detection.
Max_Fault_Number		The number of erroneous packets that can be dropped before the connection will be closed.
Connection Parameters CRC (CPCRC)		Connection Parameters CRC. A CRC-S32 of target connection parameters contained in the SafetyOpen type 2 request.

CIP Safety Device Operations

Introduction

This topic describes CIP Safety device operations, including system error detection and response mechanisms, and device operating state:

- Power on self check
- Non-recoverable detected error response
- Recoverable detected error
- Target connection health management
- Run / Idle state of CIP Safety device

Power on Self Check of the CIP Safety Originator and Target

At power on, and each time a new application is loaded, the CIP Safety system performs the following operations:

- The CPU transfers the configuration parameters to the CIP Safety Stack (CSS) in both the CPU and Copro.
- The CSS, in both the CPU and Copro, evaluates the CPCRC for each connection.
- For each connection, the CIP Safety system compares the downloaded CPCRC (calculated by the originator DTM) to the ones calculated by the CPU and Copro.
- The CSS locks the originator configuration.
- The application launches Type 2 SafetyOpen requests for a connection to each CIP Safety device.
- Each CIP Safety device:
  - ❍ Calculates its CPCRC and compares it to the CPCRC received from the originator.
  - ❍ Compares the received SCID to its internally stored SCID (Note: this check applies only to configurable devices).

I/O exchanges between the originator and target devices start only if all these tests succeed.

NOTE: In addition to the power on self tests described above, the system performs all the run time self tests required by the IEC 61784-3 CIP safety standard.

Non-Recoverable Detected Error Response

If CPU or I/O diagnostics detect a non-recoverable error, the safety system places the affected part of the system into a safe state. The affected part of the system is shut down and de-energized, with safety inputs set to 0. All impacted safety outputs are driven to their configured fallback state.

Recoverable Detected Error Response

Recoverable detected errors typically include events such as a loss of module connection, and so forth. These detected errors are reported in the Health bit of the device DDDT (T_CIP_SAFETY_IO *(see page 370)*), which contains the logical AND value of the Status_IN and Status_OUT Health bits. In the case of a recoverable error detected for an input, the value of that input is forced into the safe state, and set to 0.

Target Connection Health Management

The health of a connection to the CIP Safety target is reported in the Health bit of the Status_IN and Status_OUT parameters as described in T_CIP_SAFETY_STATUS data type *(see page 371)*. Target health can be either open and operational, or error detected.

For inputs, the connection state is provided by the server safety validator; for outputs, the connection state is provided by the client safety validator.

Run / Idle

The operating state of the CIP Safety device – run or idle – is reported in the Run_Idle bit of the Status_IN or Status_OUT parameter as described in the T_CIP_SAFETY_STATUS data type *(see page 371)*.

For an input device:

When a connection with an input module is established, the Run_Idle bit is set to Idle (0) by the producer (input) until the initial time coordination sequence is successfully completed. Thereafter, the value of the bit can be 1 (Run state) or 0 (Idle state). If the Run_Idle bit is set to 0 (Idle state), the input data values are forced to 0 (Safe state).

For an output device:

The Run_Idle bit for outputs is set to 1 by the originator (CPU) when the PAC is in Run state and the initial time coordination sequence is successfully completed. The run/idle state for outputs is set to 0 by originator (CPU) when the PAC is in Stop or Halt state, or when the initial time coordination sequence has not been successfully completed, or when the connection is closed. If the Run_Idle bit is set to 0 (Idle state), the output device is expected to set its outputs to their fallback state.

Interactions Between Safety PAC Operations and the Target Connection

Introduction

This topic discusses the interactions between the following safety CPU originator states/operations and the target device connection:

- System Reaction Time
- Run state
- Stop / Halt state
- Power Cycle / Restart
- Init Safety command
- Maintenance mode
- CCOTF
- Connecting / disconnecting / replacing a device

System Reaction Time

The time consumed by CIP Safety communication–called *network time expectation*–is added to and becomes part of the M580 safety *system reaction time*. Refer to the topic *Impact of CIP Safety Communications on Safety System Reaction Time* for additional information.

Run State

When the CIP Safety system is operating in Run state:

- Health bits in the CIP Safety device communication DDDT *(see page 370)* are updated at the beginning of the SAFE task cycle.
- Input values are updated at the beginning of the SAFE task cycle, based on the value most recently received.
- Output values are updated and transmitted after execution of the SAFE task program.
- The Run_Idle bit for outputs in the CIP Safety device communication DDDT is set to 1.
- Health bits in the CIP Safety device communication DDDT are updated.

Stop State

When the SAFE task enters Stop state, for example if the SAFE task is stopped or has reached a breakpoint:

- The originator to target connection remains open.
- Data exchanges between the CPU and CIP Safety device are performed.
- Health bits in the CIP Safety device communication DDDT *(see page 370)* continue to be updated.
- The Run_Idle bit for outputs in the CIP Safety device communication DDDT is set to 0, and output devices apply their configured fallback setting.

Halt State

In Halt state, output values are not sent from the CPU to the CIP Safety device, and the device CIP Safety device health bits are set to 0.

Power Cycle or Reset

On a power cycle or reset:

- The safety part of the application performs a cold start *(see page 266)*.
- The PAC executes the same sequence of operations that is performed for application download *(see page 360)*.

Init Safety Command

Executing the **PLC** → **Init Safety** command in Control Expert initializes the values of the CIP Safety device communication DDDT *(see page 370)*, by setting them to their factory default values.

Maintenance Mode

Operating the M580 safety CPU in maintenance mode *(see page 255)* does not impact CIP Safety device operations. The CPU will continue to compare calculations separately performed by the CPU and the Copro. However, there will be no additional comparison to values in the target DDDT. Hence, operating the PAC in maintenance mode is not deemed safe.

CCOTF

The change configuration on the fly (CCOTF) function is not supported for CIP Safety devices. Because a CIP Safety device gets its configuration settings from a vendor provided safety network configuration tool (SNCT) – and not the originator CPU – changes to device settings cannot be made from the CPU.

Connecting / Disconnecting / Replacing a CIP Safety Device

By default, upon application startup or execution of a **PLC → Init Safety** command, the CTRL_IN and CTRL_OUT bits in the DDDT *(see page 370)* are set to Enabled (1). When a device is connected to a PAC in Stop or Run mode and the device CTRL_IN or CTRL_OUT bit is set to Enabled (1), the device automatically initiates data exchanges.

NOTE: Because the CTRL_IN and CTRL_OUT bits are set to Enabled on a power-cycle, take appropriate measures in the SAFE task application to avoid unintended operations when a power-cycle is performed.

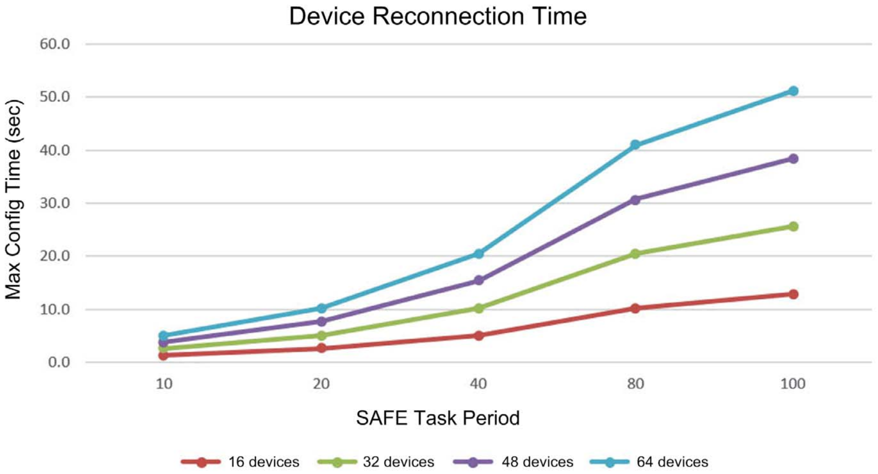
⚠ WARNING
RISK OF UNINTENDED EQUIPMENT OPERATION Do not use the CTRL_IN or CTRL_OUT bits as a safety measure to set the target data into a safe state. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

When the PAC detects an error requiring the termination of a device connection, the PAC sets the corresponding CTRL_IN or CTRL_OUT bit to Disabled (0). The device remains in the disabled state and only enters the Enabled (1) state if the transition is intended. For example, if the error is cleared and the a re-open connection request is executed.

You can execute a re-open connection request by re-setting the corresponding control bit (CTRL_IN or CTRL_OUT) from Disabled (0) to Enabled (1) in the DDDT.

When reconnecting a device, the time to connect depends on the SAFE task period and the number of devices being connected:

- For a single device with a SAFE task period less than 100 ms, the estimated reconnection time is less than 2 seconds.
- For multiple devices, refer to the following chart for estimated reconnection times.



The CIP Safety PAC treats device replacement in the same manner as a device disconnection and reconnection. The operations to reconfigure the new device with the same settings as the replaced device are local to the device and do not involve the PAC.

CIP Safety DTM Commands

Introduction

The CIP Safety DTM includes the **Safety** tab, which presents the following commands:

- **RESET Ownership**
- **SET TUNID**

These commands are accessed by first selecting a connection in the DTM navigation tree, and are enabled only when the DTM is connected to the CIP Safety device are operating online.

RESET Ownership

Use the **RESET Ownership** command to reset the CIP Safety device configuration settings to their out-of-the-box factory default values. A reset can be executed only if:

- The command is executed by the originator CPU identified by the OUNID stored in the device.
- The module configuration settings are not locked.

After the reset, the module is not owned, and can be configured by another originator.

NOTE: If a reset is performed on a module with operating connections, the reset command will not be effective.

SET TUNID

Use the **SET TUNID** command to set the Safety Network Number (SNN) in the target CIP Safety device. On execution, the Safety Network Number *(see page 351)* stored in the CIP Safety device DTM configuration is transferred to the target device and overwrites any pre-existing SNN value in the device.

NOTE: Before executing this command, confirm that you have identified the correct device to receive the SNN you intend to transfer.

Section 14.6
CIP Safety Diagnostics

Overview

This section presents diagnostic tools for the CIP Safety device, and the CIP Safety connection between the device and the M580 Safety standalone CPU.

What Is in This Section?

This section contains the following topics:

CIP Safety Device DDDT

T_CIP_SAFETY_IO DDDT

Each CIP Safety device instance is described by the T_CIP_SAFETY_IO DDDT, which consists of the following parameters:

Parameter	Data Type	Description
Health	BOOL	Global Health = the logical AND of: ● Status_IN.Health ● Status_OUT.Health Refer to the data type T_CIP_SAFETY_STATUS *(see page 371)* for a description of these health bits.
Status_IN	T_CIP_SAFETY_STATUS	Input Status.
Status_OUT	T_CIP_SAFETY_STATUS	Output Status.
CTRL_IN	BOOL	Enable/Disable Input connection.
CTRL_OUT	BOOL	Enable/Disable Output connection.
Conf_In	T_CIP_SAFETY_CONF	CIP signatures and parameters for Input connection.
Conf_Out	T_CIP_SAFETY_CONF	CIP signatures and parameters for Output connection.
Input	Array[0...n] of BYTE	Values of input, size depends on type of device. Aligned module 4 bytes with the size configured inside the DTM.
Output	Array[0...m] of BYTE	Values of output, size depends on type of device. Aligned module 4 bytes with the size configured inside the DTM.

The CIP Safety data types, referenced above, are described below.

T_CIP_SAFETY_STATUS

The T_CIP_SAFETY_SATATUS data type consists of the following parameters:

Parameter	Data Type	Description
Health	BOOL	Input or Output health: ● For input: ❍ 1: input communiction is open and operational. ❍ 0: error detected for input communication by server safety validator. ● For output: ❍ 1: output communication is open and operational. ❍ 0: error detected for output communication by client safety validator.
Run_Idle	BOOL	State of the CIP Safety device inputs or outputs: ● For inputs, set by the producer (input): ❍ 1: if the input is in Run state. ❍ 0: if the input is idle, or until the initial time coordination sequence is successfully completed. ● For outputs, set by the originator (CPU): ❍ 1: if the PAC is in Run state, after the initial time coordination sequence has been successfully completed. ❍ 0: if the PAC is in Stop or Halt state, if the connection is closed, or if the initial time coordination sequence has not successfully completed.
Error_Code	WORD	Refer to list of detected error codes *(see page 374)*.
Error_Sub_Code	WORD	Refer to list of detected error sub-codes *(see page 374)*.

T_CIP_SAFETY_CONF

The T_CIP_SAFETY_CONF data type consists of the following parameters that are transmitted in the SafetyOpen Type 2 request *(see page 361)*:

Parameter	Data Type	Description
TO_MULTIPLIER	BYTE	Timeout multiplier. Used by the consumer of a connection to determine if any of the three standard connections should timeout. The timeout value for the connection is defined as: Connection RPI * (CTM+1) * 4
Output_RPI	UDINT	Requested Packet Interval of the O→T connection.
Input_RPI	UDINT	Requested Packet Interval of the T→O connection.
Device_Vendor_ID	UINT	ODVA vendor identifier.
Device_Type	UINT	ODVA grouping to which the device belongs.
Device_Product_Code	UINT	ODVA assigned product code.
Major_Revision	BYTE	Major revision number of device firmware.
Minor_Revision	BYTE	Minor revision number of device firmware.
Configuration_Assembly_Nb	UINT	Device specific assembly number associated with the device configuration settings.
Output_Assembly_Nb	UINT	Device specific assembly number associated with output (O→T) transmissions.
Input_Assembly_Nb	UINT	Device specific assembly number associated with input (T→O) transmissions.
SC_CRC	UDINT	Safety Configuration CRC. A cyclic redundancy check (CRC) of the CIP Safety device configuration.
Configuration_Date	UINT	Month, day, and year the configuration was built.
Configuration_Time	UDINT	Hour, minute, second, and millisecond the configuration was built.
TUNID_Time	UDINT	Month, day, and year the target unique network identifier was generated.
TUNID_Date	UINT	Hour, minute, second, and millisecond the target network unique identifier was generated.
TUNID_NodeID	UDINT	A unique network identifier for the target device.
OUNID_Time	UDINT	Month, day, and year the originator unique network identifier was generated.
OUNID_Date	UINT	Hour, minute, second, and millisecond the originator unique network identifier was generated.
OUNID_NodeID	UDINT	A unique network identifier for the originator device.
Ping_Interval_EPI_Multiplier	UINT	Defines the Ping_Count_Interval for the connection.

Parameter	Data Type	Description
Time_Coordination_Msg_Min_Mult	UINT	The minimum number of 128 μS increments it could take for a Time Coordination Message to travel from the consumer to the producer.
Network_Time_Expectation_Mult	UINT	The maximum age of safety data, measured in 128 μS increments, allowed by a consumer.
Timeout_Multiplier	BYTE	The number of data production retries to include in the equation for unsuccessful connection detection.
Max_Fault_Number	UDINT	The number of erroneous packets that can be dropped before the connection will be closed.
CPCRC	UDINT	Connection Parameters CRC. A CRC-S32 of target connection parameters contained in the SafetyOpen type 2 request.

CIP Safety Device Error Codes

Detected Error Codes

The following detected error codes and sub-codes apply to the T_CIP_SAFETY_STATUS data type, and are included in the Status_IN and Status_OUT parameters of the CIP Safety device DDDT.

Detected Error Codes

Detected Error Code	Meaning
0001	Open connection: no response.
0002	Open connection: detected error response from device.
0003	Open connection: invalid response from device.
0004	Server (consumer) is non-operational.
0005	Client (producer) is non-operational.

Detected Error Sub-Codes

NOTE: Any detected error sub-codes, other than those listed below, are intended for Schneider Electric's internal use. In this case, report the detected error sub-code to Schneider Electric support.

Detected error sub-codes for Open connections:

Detected Error Sub-Code (hex)	Meaning
0100	Connection in use or duplicate Forward_Open.
0103	Transport class and trigger combination not supported.
0105	Configuration is already owned by another originator.
0106	Output is already owned by another originator.
0107	Target connection not found (Forward_Close).
0108	Invalid network connection parameter.
0109	Invalid connection size.
0110	Device not configured.
0111	O->T RPI, T->O RPI, or Time Correction RPI not supported.
0113	All Safety validator Instances are being used.
0114	Device_Vendor_ID or Device_Product_Code specified in the electronic key does not match.
0115	Device_Type specified in the electronic key does not match.
0116	Major_Revision or Minor_Revision specified in the electronic key does not match.

Detected Error Sub-Code (hex)	Meaning
0117	Invalid produced or consumed application path.
0118	Invalid or inconsistent configuration application path.
011A	Target object out of connections.
011B	RPI is smaller than the production inhibit time.
011C	Transport class not supported.
011D	Production trigger not supported.
011E	Direction not supported.
0123	Invalid Originator to Target Network Connection Type.
0124	Invalid Target to Originator Network Connection Type.
0126	Invalid Configuration Size.
0127	Invalid Originator to Target Size.
0128	Invalid Target to Originator Size.
0129	Invalid Configuration Application Path.
012A	Invalid Consuming Application Path.
012B	Invalid Producing Application Path.
012C	Configuration Symbol does not exist.
012D	Consuming Symbol does not exist.
012E	Producing Symbol does not exist.
012F	Inconsistent Application Path Combination.
0130	Inconsistent Consume Data Format.
0131	Inconsistent Produce Data Format.
0203	Connection timed out.
0204	Target did not respond on unconnected request.
0205	Parameter detected error in SafetyOpen request.
0207	Unconnected acknowledgement without reply.
0315	Invalid segment type in connection path.
031B	Module connection already established.
031C	No other extended status code applies.
031F	No more user configurable link consumer resources available in the producing module.
0801	Ping_Interval_EIP_Multiplier or Max_Consumer_Number invalid on multicast join.
0802	Invalid safety connection size.
0803	Invalid safety connection format.
0804	Invalid time correction connection parameters.
0805	Invalid Ping_interval_EIP_Multiplier.

Detected Error Sub-Code (hex)	Meaning
0806	Invalid Time_Coordination_Msg_Min Multiplier.
0807	Invalid Network_Time_Expectation_Mult.
0808	Invalid Timeout Multiplier.
0809	Invalid Max Consumer Number.
080A	Invalid CPCRC.
080B	Time Correction Connection ID invalid.
080C	SCID mismatch.
080D	TUNID not set.
080E	TUNID mismatch.
080F	Configuration operation not allowed.

Detected error sub-codes for server or client:

Detected Error Sub-Code (hex)	Meaning
271D	Time Coordination Message was received with Ping_Response bit not set.
2730	Time coordination message: Not received in allotted time.
2732	Time Coordination Message Check: message with same time stamp already received from this consumer.
2733	Time Coordination Message Check: parity check detected error.
2734	Time Coordination Message Check: Ack_Byte_2 check detected error.
2735	Time Coordination Message Check: not received within the approximatley 5 second limit.
2736	Time Coordination Message Check: not received within the same ping interval or the next ping interval.
2738	Time Coordination Message Check: CRC mismatch.
2820	Timestamp CRC mismatch.
2821	Timestamp Delta zero.
2822	Timestamp Delta greater than Network Time Expectation.
2823	Data age of a faulty message greater than Network Time Expectation.
2824	Data age of an in other respects valid message greater than Network Time Expectation.
2825	Actual Data CRC mismatch.
2826	Complemented Data CRC mismatch.
282E	Actual data CRC mismatch (no close of the connection).

Detected Error Sub-Code (hex)	Meaning
282F	Complemented data CRC mismatch (no close of the connection).
2832	Consumer activity monitor timeout.

CIP Safety Standalone CPU DDDT

CIP Safety Additions to T_BMEP58_ECPU_EXT

The M580 standalone safety CPU DDDT (T_BMEP58_ECPU_EXT) includes two CIP Safety variables:

- CSIO_SCANNER: the state of the CIP Safety I/O scanner control bit. This Boolean field can be:
  - 1: Service operating normally.
  - 0: Service not operating normally.

Refer to the list of SERVER_STATUS2 DDDT input parameters *(see Modicon M580, Hardware, Reference Manual)* for additional information.

- CSIO_HEALTH: the health of linked CIP Safety devices. This variable is an array of 64 Boolean values, each bit indicating the health of a single linked device:
  - 1: Service operating normally.
  - 0: Service not operating normally.

Refer to the topic Device Health Status *(see Modicon M580, Hardware, Reference Manual)* for additional information.

CPU DTM Diagnostics

Diagnostics via the M580 CPU DTM

The M580 CPU DTM provides the following diagnostic services:

- Device discovery
- CIP Safety I/O device health

CIP Safety Device Discovery

When Control Expert is operating online, you can use its field bus discovery service to discover first level CIP Safety devices – i.e. devices that are connected directly to the CPU – in your network. Only devices with a DTM that matches a DTM that is registered in the host PC **DTM Catalog** are discoverable.

Device discovery is performed by right-clicking the CPU DTM (BMEP58_ECPU_EXT) in the **DTM Browser**, then selecting **Field bus discovery** to open a dialog of the same name, which displays discovered devices. You can use the tools of this dialog to add device DTMs to your project. The devices you add appear under the CPU in both the **DTM Browser** and in the navigation tree of CPU DTM.

For more information on how to use this service, refer to the Field Bus Discovery Service *(see EcoStruxure™ Control Expert, Operating Modes)* topic.

CIP Safety Device Connection Health

When Control Expert is operating online, the navigation tree of the CPU DTM displays an icon indicating the health of each connection for CIP Safety I/O devices that have been added to the project:

- 🟢 indicates the connection is in RUN state.
- 🔴 indicates the connection is in STOP state, or not connected, or unknown.

For more information on how to use this feature, refer to the topic Introducing Diagnostics in the Control Expert DTM *(see Modicon M580, Hardware, Reference Manual)*.

CIP Safety Device Connection Diagnostics

Introduction

The connection nodes of a CIP Safety DTM include two tabs you can use to identify and diagnose the device connection:

- Module Info
- State Info

Module Info Tab

The CIP Safety DTM presents the **Module Info** tab, which provides static values for the following module identification parameters:

- Vendor Id
- Product Type
- Product Code
- Software Revision
- Serial Number
- Product Name
- Mac Address

State Info Tab

The CIP Safety DTM presents the **State Info** tab, which provided dynamic values for the CPU to CIP Safety device connection:

Status	Description
CIP Safety State	The current state of the device, as defined by section 5-4.2.1.5 “Device Status” of the CIP Safety standard: ● 0: Undefined ● 1: Self-testing ● 2: Idle ● 3: Self-test exception ● 4: Executing ● 5: Abort ● 6: Critical fault ● 7: Configuring ● 8: Waiting for TUNID ● 9...50: Reserved ● 51: Waiting for TUNID with torque permitted See NOTE ● 52: Executing with torque permitted See NOTE ● 53...99: Device specific ● 100...255: Vendor specific **NOTE:** Only allowed and defined in the Safety Motion Device profiles: 0x2E, 0x2F.

Status	Description
Exception Status	A single byte attribute whose value indicates the status of the alarms and warning for the device. It may be provided in either a basic or expanded method. For further details, refer to section 5-4.2.1.6 “Exception Status” of the CIP Safety standard.
Major Fault	Device-specific condition. Refer to the device manual for details.
Minor Fault	Device-specific condition. Refer to the device manual for details.
IP Address	IP address of the CIP Safety device, set in the M580 CPU DTM *(see page 358)*.
TUNID	Target Unique Network Identifier
OUNID	Originator Unique Network Identifier *(see page 341)*
Lock State	The state of the device configuration, as configured by a safety network configuration tool (SNCT): ● Locked: configuration is read-only. ● Unlocked: configuration is read-write.
Configuration Signature	The target device connection Safety Configuration Identifier (SCID *(see page 352)*).

Appendices

Introduction

The appendices contain information on the IEC 61508 and its SIL policy. Further, technical data of the Safety and non-interfering modules are provided and example calculations are carried out.

What Is in This Appendix?

The appendix contains the following chapters:

Appendix A
IEC 61508

Introduction

This chapter provides information on the Safety concepts of the IEC 61508 in general and its SIL policy in particular.

What Is in This Chapter?

This chapter contains the following topics:

General Information on the IEC 61508

Introduction

Safety-Related Systems are developed for use in processes in which risks to humans, environment, equipment and production are to be kept at an acceptable level. The risk depends on the severity and likelihood, thereby defining the necessary measures of protection.

Concerning the Safety of processes, there are 2 sides to be considered:

- the regulations and requirements defined by official authorities in order to help protect humans, environment, equipment, and production
- the measures by which these regulations and requirements are fulfilled

IEC 61508 Description

The technical standard defining the requirements for Safety-Related Systems is

- the IEC 61508.

It deals with the Functional Safety of electrical, electronic or programmable electronic Safety-Related Systems. A Safety-Related System is a system that is required to perform 1 or more specific functions to ensure risks are kept at an acceptable level. Such functions are defined as Safety Functions. A system is defined functionally Safe if random, systematic, and common cause failures do not lead to malfunctioning of the system and do not result in injury or death of humans, spills to the environment and loss of equipment and production.

The standard defines a generic approach to all lifecycle activities for systems that are used to perform Safety Functions. It constitutes procedures to be used for the design, the development, and the validation of both hardware and software applied in Safety-Related Systems. Further, it determines rules concerning both the management of Functional Safety and documentation.

IEC 61511 Description

The Functional Safety requirements defined in the IEC 61508 are refined specifically for the process industry sector in the following technical standard:

- the IEC 61511: Functional safety - safety instrumented systems for the process industry sector

This standard guides the user in the application of a Safety-Related System, starting from the earliest phase of a project, continuing through the start up, covering modifications and eventual decommissioning activities. In summary, it deals with the Safety Lifecycle of all components of a Safety-Related System used in the process industry.

Risk Description

The IEC 61508 is based on the concepts of risk analysis and Safety Function. The risk depends on severity and probability. It can be reduced to a tolerable level by applying a Safety Function that consists of an electrical, electronic or programmable electronic system. Further, it should be reduced to a level that is as low as reasonably practicable.

In summary, the IEC 61508 views risks as follows:

- Zero risk can never be reached.
- Safety is to be considered from the beginning.
- Intolerable risks are to be reduced.

SIL Policy

Introduction

The SIL value evaluates the robustness of an application against failures, thus indicating the ability of a system to perform a Safety Function within a defined probability. The IEC 61508 specifies 4 levels of Safety performance depending on the risk or impacts caused by the process for which the Safety-Related System is used. The more dangerous the possible impacts are on community and environment, the higher the Safety requirements are to lower the risk.

SIL Value Description

Discrete level (1 out of a possible 4) for specifying the Safety Integrity requirements of the Safety Functions to be allocated to the Safety-Related Systems, where Safety Integrity Level 4 has the highest level of Safety Integrity and Safety Integrity Level 1 has the lowest, see *SILs for Low Demand, page 389*.

SIL Requirements Description

To achieve Functional Safety, 2 types of requirements are necessary:

- Safety Function requirements, defining what Safety Functions have to be performed
- Safety Integrity requirements, defining what degree of certainty is necessary that the Safety Functions are performed

The Safety Function requirements are derived from hazard analysis and the Safety Integrity ones from risk assessment.

They consist of the following quantities:

- Mean time between failures
- Probabilities of failure
- Failure rates
- Diagnostic coverage
- Safe failure fraction
- Hardware fault tolerance

Depending on the level of Safety Integrity, these quantities must range between defined limits.

NOTE: Mixing different safety integrity level devices on a network or safety function requires a high degree of care with respect to the requirements of IEC 61508, and produces design and operational implications.

SIL Rating Description

As defined in the IEC 61508, the SIL value is limited by both the Safe Failure Fraction (SFF) and the hardware fault tolerance (HFT) of the subsystem that performs the Safety Function. A HFT of n means that n+1 faults could cause a loss of the Safety Function, the Safe state cannot be entered. The SFF depends on failure rates and diagnostic coverage.

The following table shows the relation between SFF, HFT, and SIL for complex Safety-Related subsystems according to IEC 61508-2, in which the failure modes of all components cannot be completely defined:

SFF	HFT=0	HFT=1	HFT=2
SFF ≤ 60%	-	SIL1	SIL2
60% < SFF ≤ 90%	SIL1	SIL2	SIL3
90% < SFF ≤ 99%	SIL2	SIL3	SIL4
SFF > 99%	SIL3	SIL4	SIL4

There are 2 ways to reach a certain Safety Integrity Level:

- via increasing the HFT by providing additional independent shutdown paths
- via increasing the SFF by additional diagnostics

SIL-Demand Relation Description

The IEC 61508 distinguishes between low demand mode and high demand (or continuous) mode of operation.

In low demand mode, the frequency of demand for operation made on a Safety-Related System is not greater than 1 per year and not greater than twice the proof test frequency. The SIL value for a low demand Safety-Related System is related directly to its average probability of failure to perform its Safety Function on demand or, simply, probability of failure on demand (PFD).

In high demand or continuous mode, the frequency of demand for operation made on a Safety-Related System is greater than 1 per year and greater than twice the proof test frequency. The SIL value for a high demand Safety-Related System is related directly to its probability of a dangerous failure occurring per hour or, simply, probability of failure per hour (PFH).

SILs for Low Demand

The following table lists the requirements for a system in low demand mode of operation:

Safety Integrity Level	Probability of Failure on Demand
4	$\geq 10^{-5}$ to $< 10^{-4}$
3	$\geq 10^{-4}$ to $< 10^{-3}$
2	$\geq 10^{-3}$ to $< 10^{-2}$
1	$\geq 10^{-2}$ to $< 10^{-1}$

SILs for High Demand

The following table lists the requirements for a system in high demand mode of operation:

Safety Integrity Level	Probability of Failure per Hour
4	$\geq 10^{-9}$ to $< 10^{-8}$
3	$\geq 10^{-8}$ to $< 10^{-7}$
2	$\geq 10^{-7}$ to $< 10^{-6}$
1	$\geq 10^{-6}$ to $< 10^{-5}$

For SIL3, the required probabilities of failure for the complete Safety integrated system are:

- PFD $\geq 10^{-4}$ to $< 10^{-3}$ for low demand
- PFH $\geq 10^{-8}$ to $< 10^{-7}$ for high demand

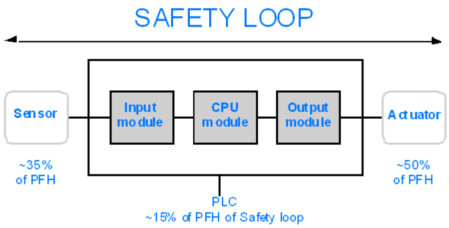
Safety Loop Description

The Safety loop to which the M580 Safety PAC consists of the following 3 parts:

- Sensors
- M580 Safety PAC with safety power supply, safety CPU, safety Coprocessor, and safety I/O modules
- Actuators

A backplane or a remote connection that includes a switch or a CRA does not destroy a Safety Loop. Backplanes, switches, and CRA modules are part of a the black channel. This means that the data exchanged by I/O and PAC cannot be corrupted without detection by the receiver.

The following figure shows a typical Safety loop:



As shown in the figure above, the contribution of the PAC is only 10-20% because the probability of failure of sensors and actuators is usually quite high.

A conservative assumption of 10% for the Safety PAC's contribution to the overall probability leaves more margin for the user and results in the following required probabilities of failure for the Safety PAC:

- PFD $\geq 10^{-5}$ to $< 10^{-4}$ for low demand
- PFH $\geq 10^{-9}$ to $< 10^{-8}$ for high demand

PFD Equation Description

The IEC 61508 assumes that half of the failures end in a Safe state. Therefore, the failure rate λ is divided into

- λ_S - the safe failure and
- λ_D - the dangerous failure, itself composed of
  - λ_{DD} - dangerous failure detected by the internal diagnostic
  - λ_{DU} - dangerous failure undetected.

The failure rate can be calculated by using the mean time between failures (MTBF), a module specific value, as follows:

$\lambda = 1/MTBF$

The equation for calculating the probability of failure on demand is:

$PFD(t) = \lambda_{DU} \times t$

t represents the time between 2 proof tests.

The probability of failure per hour implies a time interval of 1 hour. Therefore, the PFD equation is reduced to the following one:

$PFH = \lambda_{DU}$

Appendix B
System Objects

Introduction

This chapter describes the system bits and words of the M580 Safety PAC.

NOTE: The symbols associated with each bit object or system word mentioned in the descriptive tables of these objects are not implemented as standard in the software, but can be entered using the data editor.

What Is in This Chapter?

This chapter contains the following topics:

M580 Safety System Bits

System Bits for SAFE Task Execution

The following system bits apply to the M580 safety PAC. For a description of system bits that apply to both the M580 safety PAC and non-safety M580 PACs, refer to the presentation of *System Bits* in the *EcoStruxure™ Control Expert System Bits and Words Reference Manual*.

These system bits are related to the execution SAFE task, but are not directly accessible in safety program code. They can be accessed only via the `S_SYST_READ_TASK_BIT_MX` and `S_SYST_RESET_TASK_BIT_MX` blocks.

Bit Symbol	Function	Description	Initial State	Type
%S17 `CARRY`	Rotate shift output	During a rotate shift operation in the SAFE task, this bit takes the state of the outgoing bit.	0	R/W
%S18 `OVERFLOW`	Overflow or arithmetic error detected	Normally set to 0, this bit is set to 1 in the event of a capacity overflow if there is: ● A result greater than + 32 767 or less than - 32 768, in single length. ● A result greater than + 65 535, in unsigned integer. ● A result greater than + 2 147 483 647 or less than - 2 147 483 648, in double length ● A result greater than +4 294 967 296, in double length or unsigned integer. ● Division by 0. ● The root of a negative number. ● Forcing to a non-existent step on a drum. ● Stacking up of an already full register, emptying of an already empty register.	0	R/W
%S21 `1RSTTASKRUN`	First SAFE task scan in RUN	Tested in the SAFE task, this bit indicates the first cycle of this task. It is set to 1 at the start of the cycle and reset to 0 at the end of the cycle. **NOTE:** ● The first cycle of the task status can be read using the `SCOLD` output of the `S_SYST_STAT_MX` system function block. ● This bit is not effective for M580 Safety Hot Standby systems.	0	R/W

Notes Regarding Non-Safety-Specific System Bits

System Bit	Description	Notes
%S0	cold start	Can be used only in process (non-SAFE) tasks and has no influence on SAFE task.
%S9	outputs set to fallback	Has no influence on Safety output modules.
%S10	Global I/O detected error	Reports some, but not all, of the possible detected errors relating to safety I/O modules.
%S11	watchdog overflow	Takes into account an overrun on SAFE task.
%S16	task I/O detected error	Reports some, but not all, of the possible detected errors relating to safety I/O modules.
%S19	task period overrun	Information for SAFE task overrun is not available.
%S40...47	rack *n* I/O detected error	Reports some, but not all, of the possible detected errors relating to safety I/O modules.
%S78	STOP on detected error	Applies to both process tasks and the SAFE task. If the bit is set, for example if a %S18 overflow error rises, the SAFE task enters HALT state.
%S94	save adjusted values	Does not apply to SAFE variables. The SAFE initial values are not modifiable by the activation of this bit.
%S117	RIO detected error on Ethernet I/O network	Reports some, but not all, of the possible detected errors relating to safety I/O modules.
%S119	general in rack detected error	Reports some, but not all, of the possible detected errors relating to safety I/O modules.

M580 Safety System Words

System Words for M580 Safety PACs

The following system words apply to the M580 safety PAC. For a description of system words that apply to both the M580 safety PAC and non-safety M580 PACs, refer to the presentation of *System Words* in the *EcoStruxure™ Control Expert System Bits and Words Reference Manual*.

These system words and values are related to the SAFE task. They can be accessed from application program code in the non-safety sections (MAST, FAST, AUX0 or AUX1), but not from code in the SAFE task section.

Word	Function	Type
%SW4	Period of the SAFE task defined in the configuration. The period is not modifiable by the operator.	R
%SW12	Indicates the operating mode of the Copro module: ● 16#A501 = maintenance mode ● 16#5AFE = safety mode Any other value is interpreted as a detected error.	R
%SW13	Indicates the operating mode of the CPU: ● 16#501A = maintenance mode ● 16#5AFE = safety mode Any other value is interpreted as a detected error.	R
%SW42	SAFE task current time. Indicates the execution time of the last cycle of the SAFE task (in ms).	R
%SW43	SAFE task max time. Indicate the longest task execution time of the SAFE task since the last cold start (in ms).	R
%SW44	SAFE task min time. Indicate the shortest task execution time of the SAFE task since the last cold start (in ms).	R
%SW110	Percentage of system CPU load used by the system for internal services.	R
%SW111	Percentage of system CPU load used by the MAST task.	R
%SW112	Percentage of system CPU load used by the FAST task.	R
%SW113	Percentage of system CPU load used by the SAFE task.	R
%SW114	Percentage of system CPU load used by the AUX0 task.	R
%SW115	Percentage of system CPU load used by the AUX1 task.	R
%SW116	Total system CPU load.	R

Word	Function	Type
%SW124	Contains the cause of the non-recoverable detected error when the M580 Safety PAC is in Halt state: ● 0x5AF2: RAM detected error in memory check. ● 0x5AFB: Safety firmware code error detected. ● 0x5AF6: Safety watchdog overrun error detected on CPU. ● 0x5AFF: Safety watchdog overrun error detected on coprocessor. ● 0x5B01: Coprocessor not detected at start-up. ● 0x5AC03: CIP safety non-recoverable error detected by CPU. ● 0x5AC04: CIP safety non-recoverable error detected by coprocessor. **NOTE:** The above does not constitute a complete list. Refer to the *EcoStruxure™ Control Expert System Bits and Words Reference Manual* for more information.	R
%SW125	Contains the cause of the recoverable detected error in the M580 Safety PAC: ● 0x5AC0: CIP safety configuration is not correct (detected by CPU). ● 0x5AC1: CIP safety configuration is not correct (detected by coprocessor). ● 0x5AF3: Comparison error detected by main CPU. ● 0x5AFC: Comparison error detected by coprocessor. ● 0x5AFD: Internal error detected by coprocessor. ● 0x5AFE: Synchronization error detected between CPU and coprocessor. ● 0x9690: Application program checksum error detected. **NOTE:** The above does not constitute a complete list. Refer to the *EcoStruxure™ Control Expert System Bits and Words Reference Manual* for more information.	R
%SW126	These two system words contain information that is for Schneider Electric internal use to help analyze a detected error in more detail.	R
%SW127		
%SW128	With CPU firmware 3.10 or earlier, force time synchronization between NTP time and Safe time into the safe IO modules and Safe CPU task: ● Value change from 16#1AE5 to 16#E51A forces synchronization. Refer to the topic *Procedure for Synchronizing NTP Time Settings (see page 175)*. ● Other sequences and values do not force synchronization.	R/W
%SW142	Contains the safety COPRO firmware version in 4 digits BCD: for example firmware version 21.42 corresponds to %SW142 = 16#2142.	R
%SW148	Count of error correcting code (ECC) errors detected by the CPU.	R
%SW152	Status of the NTP CPU time updated by Ethernet communications module (e.g BMENOC0301/11) over the X Bus backplane via the optional forced time synchronization feature: ● 0: the CPU time is not refreshed by the Ethernet communications module. ● 1: The CPU time is refreshed by the Ethernet communications module.	R

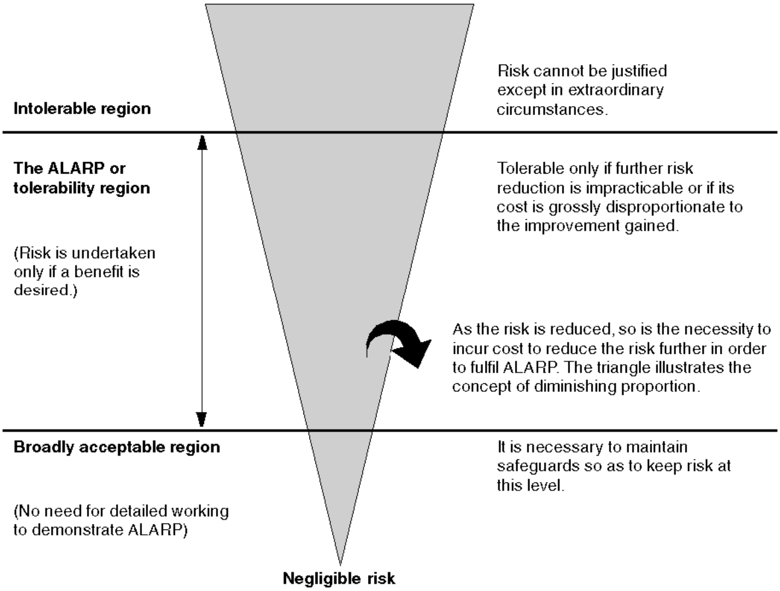
Word	Function	Type
%SW169	Safety Application ID: Contains an ID of the safety code part of the application. The ID is automatically modified when the safe application code is modified. NOTE: ● If the safe code has been changed and a **Build Changes** command has been executed since the previous **Rebuild All** command (thereby changing the Safety application ID), execution of a **Rebuild All** command may again change the Safety application ID. ● The SAFE program unique identifier can be read using the `SAID` output of the `S_SYST_STAT_MX` system function block.	R
%SW171	State of the FAST tasks: ● 0: No FAST tasks exist ● 1: Stop ● 2: Run ● 3: Breakpoint ● 4: Halt	R
%SW172	State of the SAFE task: ● 0: No SAFE task exists ● 1: Stop ● 2: Run ● 3: Breakpoint ● 4: Halt	R
%SW173	State of the MAST task: ● 0: No MAST task exists ● 1: Stop ● 2: Run ● 3: Breakpoint ● 4: Halt	R
%SW174	State of the AUX0 task: ● 0: No AUX0 task exists ● 1: Stop ● 2: Run ● 3: Breakpoint ● 4: Halt	R
%SW175	State of the AUX1 task: ● 0: No AUX1 task exists ● 1: Stop ● 2: Run ● 3: Breakpoint ● 4: Halt	R

Glossary

A

ALARP

(*as low as reasonably practicable*) (Definition IEC 61508)


Intolerable region
Risk cannot be justified except in extraordinary circumstances.
The ALARP or tolerability region
(Risk is undertaken only if a benefit is desired.)
Tolerable only if further risk reduction is impracticable or if its cost is grossly disproportionate to the improvement gained.
As the risk is reduced, so is the necessity to incur cost to reduce the risk further in order to fulfil ALARP. The triangle illustrates the concept of diminishing proportion.
Broadly acceptable region
(No need for detailed working to demonstrate ALARP)
It is necessary to maintain safeguards so as to keep risk at this level.
Negligible risk

C

CCF

(*common cause failure*) Failure, which is the result of 1 or more events, causing coincident failures of 2 or more separate channels in a multiple channel system, leading to system failure. (Definition IEC 61508) The common cause factor in a dual channel system is the crucial factor for the probability of failure on demand (PFD) for the whole system.

CPCRC

(*connection parameter cyclic redundancy check*) A CRC-S32 of the targetconnection parameters produced by the CSS for each CIP Safety connection, and contained in the SafetyOpen type 2 request.

D

DDDT

(*device derived data type*) A DDT predefined by the manufacturer and not modifiable by user. It contains the I/O language elements of an I/O module.

DIO network

A network containing distributed equipment, in which I/O scanning is performed by a CPU with DIO scanner service on the local rack. DIO network traffic is delivered after RIO traffic, which takes priority in a device network.

DRS

(*dual-ring switch*) A ConneXium extended managed switch that has been configured to operate on an Ethernet network. Predefined configuration files are provided by Schneider Electric to downloaded to a DRS to support the special features of the main ring / sub-ring architecture.

DTM

(*device type manager*) A DTM is a device driver running on the host PC. It provides a unified structure for accessing device parameters, configuring and operating the devices, and troubleshooting devices. DTMs can range from a simple graphical user interface (GUI) for setting device parameters to a highly sophisticated application capable of performing complex real-time calculations for diagnosis and maintenance purposes. In the context of a DTM, a device can be a communications module or a remote device on the network.

See FDT.

E

EDS

(*electronic data sheet*) EDS are simple text files that describe the configuration capabilities of a device. EDS files are generated and maintained by the manufacturer of the device.

EUC

(*equipment under control*) (Definition IEC 61508) This term designates equipment, machinery, apparatuses or plants used for manufacturing, process, transportation, medical or other activities.

H

HFT

(*hardware fault tolerance*) (Definition IEC 61508)

A hardware fault tolerance of N means that N + 1 faults could cause a loss of the Safety Function, for instance:

- HFT = 0: The 1st failure could cause a loss of the Safety Function
- HFT = 1: 2 faults in combination could cause a loss of the Safety Function. (There are 2 different paths to go to a Safe state. Loss of the Safety Function means that a Safe state cannot be entered.

O

OUNID

(*originator unique network identifier*) A value that uniquely identifies the connection originating device (typically a CPU) on a CIP safety network. The OUNID consists of:

- a safety network number (SNN), which can be a timestamp or other user-defined value.
- a node address (for EtherNet/IP networks, the IP address).

P

PST

(*process safety time*) The process safety time is defined as the period of time between a failure occurring in EUC or the EUC control system (with the potential to give rise to a hazardous event) and the occurrence of the hazardous event if the safety function is not performed.
(Definition IEC 61508)

R

RIO drop

A rack of Ethernet I/O modules, managed by an RIO adapter, with inputs and outputs included in the RIO scan of the CPU. A drop can be a single rack or a main rack with an extended rack.

S

SAId

(*safety application identifier*) An algorithmically calculated signature of the safe part of a Control Expert application, stored in %SW169.

SCID

(*safety configuration identifier*) See TUNID *(see page 402)*.

SFF

(*safe failure fraction*)

SNCT

(*safety network configuration tool*) A vendor-provided tool for configuring CIP safety devices. See TUNID *(see page 402)*.

SRT

(*system reaction time*) The system reaction time is the period of time between detection of a signal at the input module terminal and the reaction of setting an output at the output module terminal.

T

TUNID

(*target unique network identifier*) A value that uniquely identifies the connection target device on a CIP safety network. The TUNID consists of:

- a safety network number (SNN), which can be a timestamp or other user-defined value.
- a safety configuration identifier (SCID), also called the configuration signature, that is created in a vendor provided safety network configuration tool (SNCT) and consists of:
  - a Safety Configuration CRC (SCCRC), which is a CRC value of the safety device configuration settings, in the form of a hex value consisting of 4 octets.
  - a Safety Configuration Time Stamp (SCTS), which is a date and time hexadecimal value timestamp that consists of 6 octets.

Index