Modicon M580 Hot Standby

Frequently Used Architectures System Guide

Original instructions

NHA58880.09
05/2025

www.se.com

Legal Information

Table of Contents

Safety Information

Important Information

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 are 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:

(In case of divergence or contradiction between any translation and the English original, the original text in the English language will prevail.)

- 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 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 Document

Document Scope

EcoStruxure™ is a Schneider Electric program designed to address the key challenges of many different types of users, including plant managers, operations managers, engineers, maintenance teams, and operators, by delivering a system that is scalable, flexible, integrated, and collaborative.

This document presents one of the EcoStruxure features, using Ethernet as the backbone around the Modicon M580 offer and connecting an M580 *local rack* and M580 *RIO drops*.

This guide provides detailed information about the Modicon M580 Hot Standby system, in a daisy chain loop topology.

NOTE: The specific configuration settings contained in this guide are intended to be used for instructional purposes only. The settings required for your specific configuration may differ from the examples presented in this guide.

Validity Note

This document has been updated for the release of EcoStruxure™ Control Expert 16.2 or later.

The characteristics of the products described in this document are intended to match the characteristics that are available on *www.se.com.* As part of our corporate strategy for constant improvement, we may revise the content over time to enhance clarity and accuracy. If you see a difference between the characteristics in this document and the characteristics on *www.se.com*, consider *www.se.com* to contain the latest information.

Available Languages of this Document

Language	**Reference number**
English	NHA58880
French	NHA58881
German	NHA58882
Italian	NHA58883
Spanish	NHA58884
Chinese	NHA58885

Related Documents

Title of documentation	Reference number
Modicon M580, Hardware, Reference Manual	EIO0000001578 (English), EIO0000001579 (French), EIO0000001580 (German), EIO0000001582 (Italian), EIO0000001581 (Spanish), EIO0000001583 (Chinese)
Modicon M580 Frequently Used Architectures System Guide	HRB62666 (English), HRB65318 (French), HRB65319 (German), HRB65320 (Italian), HRB65321 (Spanish), HRB65322 (Chinese)
Modicon M580 Complex Topologies System Guide	NHA58892 (English), NHA58893 (French), NHA58894 (German), NHA58895 (Italian), NHA58896 (Spanish), NHA58897 (Chinese)
Modicon M580, Safety System Planning Guide	QGH60283 (English), QGH60284 (French), QGH60285 (German), QGH60286 (Spanish), QGH60287 (Italian), QGH60288 (Chinese)
Modicon M580, Safety Manual	QGH46982 (English), QGH46983 (French), QGH46984 (German), QGH46985 (Italian), QGH46986 (Spanish), QGH46987 (Chinese)
Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications	EIO0000002726 (English), EIO0000002727 (French), EIO0000002728 (German), EIO0000002730 (Italian), EIO0000002729 (Spanish), EIO0000002731 (Chinese)
Modicon M580, RIO Modules, Installation and Configuration Guide	EIO0000001584 (English), EIO0000001585 (French), EIO0000001586 (German), EIO0000001587 (Italian), EIO0000001588 (Spanish), EIO0000001589 (Chinese)

Title of documentation	Reference number
Modicon M580 BMENOC0301/11, Ethernet Communication Module, Installation and Configuration Guide	HRB62665 (English), HRB65311 (French), HRB65313 (German), HRB65314 (Italian), HRB65315 (Spanish), HRB65316 (Chinese)
Modicon M580 BMENOC0302 High Performance Ethernet Communication Module Installation and Configuration Guide	NNZ44174 (English)
Modicon M580, Change Configuration on the Fly, User Guide	EIO0000001590 (ENG) EIO0000001595 (CHS) EIO0000001591 (FRE) EIO0000001592 (GER) EIO0000001594 (ITA) EIO0000001593 (SPA)
Modicon X80, BMXNRP0200/0201 Fiber Converter Modules, User Guide	EIO0000001108 (ENG) EIO0000001113 (CHS) EIO0000001109 (FRE) EIO0000001110 (GER) EIO0000001112 (ITA) EIO0000001111 (SPA)
Modicon eX80, BMEAHI0812 HART Analog Input Module & BMEAHO0412 HART Analog Output Module, User Guide	EAV16400 (English), EAV28404 (French), EAV28384 (German), EAV28413 (Italian), EAV28360 (Spanish), EAV28417 (Chinese)
Modicon M580 BMENOS0300 Network Option Switch Module Installation and Configuration Guide	NHA89117 (English), NHA89119 (French), NHA89120 (German), NHA89121 (Italian), NHA89122 (Spanish), NHA89123 (Chinese)
Modicon X80, Analog Input/Output Modules, User Manual	35011978 (English), 35011979 (German), 35011980 (French), 35011981 (Spanish), 35011982 (Italian), 35011983 (Chinese)
Modicon X80, Discrete Input/Output Modules, User Manual	35012474 (English), 35012475 (German), 35012476 (French), 35012477 (Spanish), 35012478 (Italian), 35012479 (Chinese)

Title of documentation	Reference number
Modicon X80, BMXEHC0200 Counting Module, User Manual	35013355 (English), 35013356 (German), 35013357 (French), 35013358 (Spanish), 35013359 (Italian), 35013360 (Chinese)
Electrical installation guide	EIGED306001EN (English)
Control Panel Technical Guide, How to protect a machine from malfunctions due to electromagnetic disturbance	CPTG003_EN (English), CPTG003_FR (French)
EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual	35006144 (English), 35006145 (French), 35006146 (German), 35013361 (Italian), 35006147 (Spanish), 35013362 (Chinese)
EcoStruxure™ Control Expert, System Bits and Words, Reference Manual	EIO0000002135 (English), EIO0000002136 (French), EIO0000002137 (German), EIO0000002138 (Italian), EIO0000002139 (Spanish), EIO0000002140 (Chinese)
EcoStruxure™ Control Expert, Operating Modes	33003101 (English), 33003102 (French), 33003103 (German), 33003104 (Spanish), 33003696 (Italian), 33003697 (Chinese)
EcoStruxure™ Control Expert, Installation Manual	35014792 (English), 35014793 (French), 35014794 (German), 35014795 (Spanish), 35014796 (Italian), 35012191 (Chinese)
Modicon Controllers Platform Cyber Security, Reference Manual	EIO0000001999 (English), EIO0000002001 (French), EIO0000002000 (German), EIO0000002002 (Italian), EIO0000002003 (Spanish), EIO0000002004 (Chinese)

You can download these technical publications and other technical information from our website at www.se.com/ww/en/download/ .

Product Related Information

DANGER

HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH

- Disconnect all power from all equipment including connected devices prior to removing any covers or doors, or installing or removing any accessories, hardware, cables, or wires except under the specific conditions specified in the appropriate hardware guide for this equipment.
- Always use a properly rated voltage sensing device to confirm the power is off where and when indicated.
- Replace and secure all covers, accessories, hardware, cables, and wires and confirm that a proper ground connection exists before applying power to the unit.
- Use only the specified voltage when operating this equipment and any associated products.

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

WARNING

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA), or equivalent risk analysis, of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate them.
- Review the implications of communication link interruptions and take actions to mitigate them.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and error conditions) according to your risk assessment, and applicable codes and regulations.
- Apply local accident prevention and safety regulations and guidelines.[1]
- Test each implementation of a system for proper operation before placing it into service.

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

[1] For additional information, refer to NEMA ICS 1.1 (latest edition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* and to NEMA ICS 7.1

(latest edition), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* or their equivalent governing your particular location.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

- Only use software approved by Schneider Electric for use with this equipment.
- Update your application program every time you change the physical hardware configuration.

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

Terminology Derived from Standards

The technical terms, terminology, symbols and the corresponding descriptions in the information contained herein, or that appear in or on the products themselves, are generally derived from the terms or definitions of international standards.

In the area of functional safety systems, drives and general automation, this may include, but is not limited to, terms such as *safety*, *safety function*, *safe state*, *fault*, *fault reset*, *malfunction*, *failure*, *error*, *error message*, *dangerous*, etc.

Among others, these standards include:

Standard	Description
IEC 61131-2:2007	Programmable controllers, part 2: Equipment requirements and tests.
ISO 13849-1:2023	Safety of machinery: Safety related parts of control systems. General principles for design.
EN 61496-1:2013	Safety of machinery: Electro-sensitive protective equipment. Part 1: General requirements and tests.
ISO 12100:2010	Safety of machinery - General principles for design - Risk assessment and risk reduction
EN 60204-1:2006	Safety of machinery - Electrical equipment of machines - Part 1: General requirements
ISO 14119:2013	Safety of machinery - Interlocking devices associated with guards - Principles for design and selection
ISO 13850:2015	Safety of machinery - Emergency stop - Principles for design
IEC 62061:2021	Safety of machinery - Functional safety of safety-related electrical, electronic, and electronic programmable control systems
IEC 61508-1:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: General requirements.
IEC 61508-2:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Requirements for electrical/electronic/programmable electronic safety-related systems.
IEC 61508-3:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Software requirements.
IEC 61784-3:2021	Industrial communication networks - Profiles - Part 3: Functional safety fieldbuses - General rules and profile definitions.
2006/42/EC	Machinery Directive
2014/30/EU	Electromagnetic Compatibility Directive
2014/35/EU	Low Voltage Directive

In addition, terms used in the present document may tangentially be used as they are derived from other standards such as:

Standard	Description
IEC 60034 series	Rotating electrical machines
IEC 61800 series	Adjustable speed electrical power drive systems
IEC 61158 series	Digital data communications for measurement and control – Fieldbus for use in industrial control systems

Finally, the term zone of operation may be used in conjunction with the description of specific hazards, and is defined as it is for a hazard zone or danger zone in the Machinery Directive (2006/42/EC) and ISO 12100:2010.

NOTE: The aforementioned standards may or may not apply to the specific products cited in the present documentation. For more information concerning the individual standards applicable to the products described herein, see the characteristics tables for those product references.

Information on Non-Inclusive or Insensitive Terminology

As a responsible, inclusive company, Schneider Electric is constantly updating its communications and products that contain non-inclusive or insensitive terminology. However, despite these efforts, our content may still contain terms that are deemed inappropriate by some customers.

Introducing the M580 Hot Standby System

Overview

This chapter presents a brief overview of the Modicon M580 Hot Standby system, including:

- a description of primary and standby control components
- the Hot Standby controller
- LEDs and switches
- modes of operation

An M580 Hot Standby system is based on two identically configured controllers linked to each other and to the same remote I/O network. If one controller stops communications, the other assumes control of the I/O system.

Primary and Standby Controllers

Description

The M580 Hot Standby system is designed for use where downtime cannot be tolerated. The system delivers high availability through redundancy. Two backplanes are configured with identical hardware and software.

One of the controllers acts as the primary, which runs the application by executing program logic and operating RIO drops and distributed equipment.

The other controller acts as the standby controller. The primary controller updates the standby controller at the beginning of each scan. The standby is ready to assume control within one scan if the primary stops communications.

Primary and standby states are interchangeable. When the controllers are running, either controller can enter the primary state. When one running controller becomes primary, the other running controller may be in the standby or wait state.

The EIO and DIO networks are operated by the primary controller.

Controller Switchover

Introduction

The purpose of a Hot Standby system is to perform a switchover, if needed. A switchover is the immediate transfer of control of the network from the primary controller to the standby controller.

The M580 Hot Standby system continuously monitors ongoing system operations and determines if a condition requiring a switchover exists. On each scan, both the primary controller and the standby controller verify the health of the system.

The primary controller verifies the health of the following:

- the Ethernet RIO network link
- the Hot Standby link between the primary and standby controllers

The standby controller verifies the following:

- the health of the primary controller
- the identity of modules in both the primary and standby backplanes
- the application versions running in the primary and standby controllers
- the firmware versions of the primary and standby controllers
- the health of the Hot Standby link between the primary and standby controllers

Before each MAST task, the primary controller transfers to the standby controller system: status and I/O data, including date and time data. On switchover, the standby controller applies this time data and continues the same time stamping sequence. The maximum amount of transferable Hot Standby data depends on the controller.

NOTE: Both the primary controller and the standby controller maintain independent event logs. If a switchover occurs, the events recorded in the log of the former primary controller will not be included in the event log of the new primary (formerly the standby) controller.

Switchover Causes

One of the following events cause a switchover:

- The primary controller encountered a blocking condition and entered the HALT state.
- The primary controller detected an unrecoverable hardware or system error.
- The primary controller received a STOP command from Control Expert or the DDDT.
- An application transfers to the primary controller.

- The primary controller is de-energized, and a power cycle occurs.
- The following events simultaneously occur:

  The primary controller loses communication with all RIO drops.

  The Hot Standby link is healthy.

  The standby controller maintains communication with at least one RIO drop.

Similar to a switchover, a *swap* transfers control of the network from the primary controller to the standby controller. A swap can be caused by:

- Execution of the DDDT `CMD_SWAP` command by either the program logic or an animation table **Force** command.
- Manually selecting the **HSBY Swap** option in the **Task** tab of the controller's **Animation** window in Control Expert.

Non-Switchover Causes

These events **do not** cause a switchover:

- the simultaneous interruption of communication with all RIO drops from both the primary and standby controllers
- the partial interruption of communication with RIO drops from the primary controller
- a Modbus connection break
- an overload broadcast traffic generated by a peer (for example, SCADA or another controller)
- a BMENOC0301 / BMENOC0311 or BMENOC0302(H) communication module that stops operating
- the removal of an SD memory card
- the primary controller is partially in the HALT state in a safety-related Hot Standby system (either the SAFE program or the PROCESS program) and not all of the tasks in the standby controller are in RUN

Switchover Execution Time

If both the primary and standby controllers are operating normally, the Hot Standby system detects a switchover causal event within 15 ms.

For both a safety-related and non-safety-related controller system, the effect of the switchover on the application reaction time is:

- 15 ms for the inputs and outputs driven by the MAST task

- 15 ms + TTASK for the inputs and outputs driven by the FAST or SAFE task, where TTASK is the configured execution period for that task.

The application response time for a swap or a switchover can be calculated.

After the switchover, the formerly standby controller becomes the primary. In the worst case, the new primary controller operates with the data of the *N* scan cycle, while the outputs receive (from the formerly primary controller) the data of the *N+1* scan cycle. The new primary controller re-evaluates outputs beginning with the *N+1* scan.

Some FAST task program execution may stop because the Hot Standby switchover evaluation occurs during the MAST task.

Switchover Effect on the Main IP Address Assignments

Distributed equipment uses the Main IP address setting, configured in the IPconfig tab, page 125, to communicate over an Ethernet network with the primary controller. On switchover, the Main IP address setting is automatically transferred from the formerly primary controller to the formerly standby – now the new primary – controller. Similarly, on switchover the Main IP address + 1 setting is automatically transferred from the formerly standby controller to the new standby.

In this way, the configured links between the distributed equipment and the primary controller do not need to be edited in the event of a switchover.

NOTE:

- A switchover does not affect the assignment of IP address A or IP address B. These assignments are made exclusively by the **A/B/Clear** rotary switch on the back of the controller and are not affected by a change in the primary or standby Hot Standby status.
- When connecting Control Expert to the Hot Standby system, use IP address A or B to maintain the connection on switchover. Do not use the main IP address because this becomes the main IP address + 1 on switchover and disconnects Control Expert.

Switchover Effect on Remote Outputs

The switchover is transparent for RIO drops: the state of the outputs is not affected by the switchover. During Hot Standby operations, each controller maintains an independent, redundant owner connection with each RIO drop. Each controller makes this connection via IP address A or B, depending on the **A/B/Clear** rotary switch designation for its controller. When a switchover occurs, the new primary controller continues to communicate with inputs and outputs through its pre-existing redundant owner connection.

NOTE: The switchover may not be transparent with distributed equipment outputs.

Switchover Effect on Distributed Equipment Outputs

The behavior of distributed equipment outputs during a switchover depends on whether the equipment supports hold up time. If the device does not support hold up time, its outputs may go to fallback when the connection with the primary controller is interrupted, and will recover their state after reconnecting with the new primary controller.

The outputs need to support a sufficiently long hold–up time to achieve transparent behavior.

Switchover Effect on CCOTF Changes

After the standby controller becomes the new primary, it uses both the firmware and the application previously configured. If Change Configuration On The Fly (CCOTF) changes were previously made to the formerly primary controller that were not transferred to the formerly standby controller, these changes are not included in the configuration running in the new primary controller.

For example, assume that an I/O module was added to an RIO drop in the configuration running in the formerly primary controller. If the changed configuration was not transferred to the formerly standby controller, the added module will not be included in the configuration running in the formerly standby controller when it becomes the primary controller after switchover.

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION Apply any intended configuration modifications or changes to the standby controller. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

Switchover Effect on Program Logic Changes

A logic mismatch condition exists when changes have been made to the application in the primary controller, but not to the standby controller. If the LOGIC_MISMATCH_ALLOWED flag is set, the standby controller continues to operate as the standby while a logic mismatch exists. In this case, if a switchover occurs, the new primary controller executes its own, different application using data received from the formerly primary controller.

Depending on the nature of the application modification, different results occur:

Modification to the initial primary controller logic:	Effect on the new primary controller program execution:
Only the code is changed (no change to the variables).	All variable values exchanged between the controllers remain the same (EQUAL).
The new variables are added.	The new variables are not used by the new primary controller.
The existing variables are deleted.	The new primary controller still has the deleted variables in the program execution and applies the most recent values to these variables.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

Apply any intended program modifications or variable changes (adding variables / deleting variables) to the standby controller.

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

Switchover Effect on Time Management

In an M580 Hot Standby system, the primary and standby controllers operate their own system timers, which are not automatically synchronized. The primary and standby controllers can be configured to perform as an NTP client or an NTP server because they share a common configuration.

When the NTP client function is enabled in a Hot Standby system, the primary and standby controllers independently receive time settings from a designated NTP server.

When the NTP server function is enabled in a Hot Standby system, only the primary controller performs the role of server.

Before each scan, the primary controller transfers system data to the standby controller, including the following primary controller system time values:

- time of day
- application counters
- free running counter

On switchover, the formerly standby controller – now the new primary controller – applies the system time values sent by the formerly primary controller. Thereafter, the new primary controller continues to execute the application in the same time context as the formerly primary controller. If the NTP server function is enabled for the Hot Standby system, the new primary controller begins to perform the function of the NTP server.

Switchover Effect on IPsec Connections

On switchover, the BMENOC0301 / BMENOC0311 or BMENOC0302(H) communication module of the formerly primary controller closes connections that use its main IP address. These connections are re-opened in the BMENOC0301 / BMENOC0311 or BMENOC0302 (H) module of the new primary controller using the main IP address after the two controllers swap their Main IP address and Main IP address + 1.

It takes up to 5 minutes to re-establish an IPsec connection that uses the main IP address because IPsec connections take a relatively long time to establish.

Switchover Effect on Safety-related Operating Mode

When an M580 safety-related Hot Standby controller switches from standby to primary controller, the operating mode is automatically set to safety-related mode.

NOTE: The operating mode setting of a safety-related-related Hot Standby controller – either safety-related mode or maintenance mode – is not included in the transfer of an application from the primary controller to the standby controller.

Recovery of the Formerly Primary Controller

The formerly primary controller can become the standby controller depending on the cause of switchover.

If the switchover is caused by a/an:	The formerly primary controller becomes the standby controller by:
Primary halt (non-safety-related controller)	Performing an `INIT` command and RUN the controller
Primary halt (safety-related controller — Process and/ or SAFE task)	Performing an `INIT` command (Process task), and/or an `INIT_SAFETY` command (SAFE task), and the RUN the controller
Controller stop in a non-safety-related controller or in both the Process and SAFE tasks of a safety-related controller	Running the controller
An error in the primary controller is detected	Performing a controller `RESET` command
Application transfer in the primary	Completing the transfer and RUN the application
Primary power off	Powering up the controller
Loss of RIO drops while the Hot Standby link is healthy and the standby controller has access to the drops	Recovering the RIO drops

DDDT command	The formerly primary controller automatically becomes the standby if the necessary preconditions exist, for example: • Firmware mismatches are allowed. • Logic mismatches are allowed. • Online modifications are allowed.
Control Expert **HSBY Swap** button	

Physical Description of M580 Hot Standby Controllers

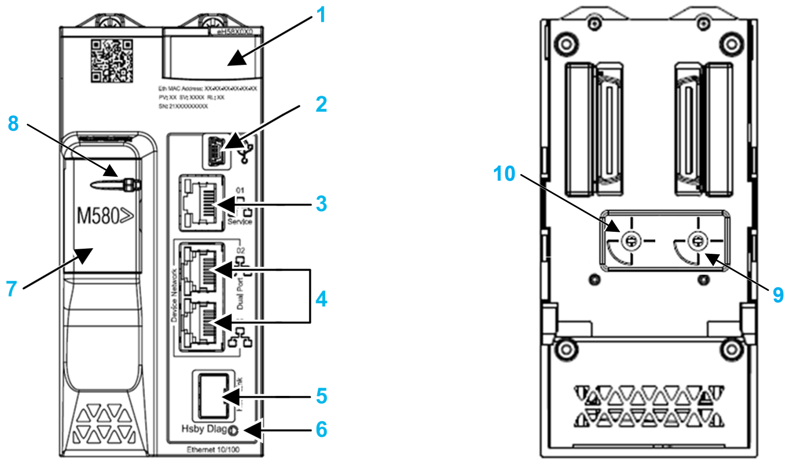
Hot Standby Controller Modules

These M580 controllers support M580 Hot Standby systems:

- BMEH582040, BMEH582040C, BMEH582040S
- BMEH584040, BMEH584040C, BMEH584040S
- BMEH586040, BMEH586040C, BMEH586040S

Controller Front and Back Views

The Hot Standby controller modules have the same external hardware features. The front of the controller is on the left. The back of the controller is on the right:



1 LED diagnostic display panel

2 Mini-B USB port for module configuration via PC running Control Expert

3 RJ45 Ethernet service port connector

4 RJ45 connectors that together serve as a dual port to the Ethernet network

5 SFP socket for copper or fiber-optic Hot Standby link connection

6 Hot Standby status link LED

7 SD memory card slot (behind door)

8 SD memory card lockable door

9 Operating mode rotary selector, with settings **Communication Security Reset**, **Secured**, **Standard**

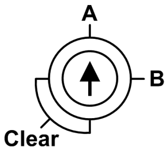
> **NOTE:** The operating mode rotary selector switch will be made operational for future product releases. For this product release, operating mode is automatically set to **Standard**, regardless of the switch position.

10 **A/B/Clear** rotary switch, used to designate the controller as either controller A or controller B, or to clear the existing Control Expert application

> **NOTE:** Safety-related controllers are red.

A/B/Clear Rotary Selector Switch

Use the **A/B/Clear** rotary switch on the back of each M580 Hot Standby controller to designate the role that the controller plays in the M580 Hot Standby configuration:



NOTE: A plastic screwdriver is provided for your convenience; use it, or an equivalent, to set the rotary switch according to its role in a Hot Standby system.

Rotary switch settings include:

Position	Result
A	• Designates the controller as controller A, page 151, as referenced in Control Expert and the `T_M_ECPU_HSBY` DDDT. • Assigns the controller **IP address A** on Ethernet RIO network.
B	• Designates the controller as controller B, page 151, as referenced in Control Expert and the `T_M_ECPU_HSBY` DDDT. • Assigns the controller **IP address B** on Ethernet RIO network.
Clear	• Clears the application in the controller, and places the controller into the `NO_CONF` operational state. • If an SD memory card is inserted in the controller, the application in the card is also cleared.

NOTE: Setting the switch for each Hot Standby controller to the same **A**/**B** position causes a conflict of controller roles, page 152.

Clearing Controller Memory

To clear the controller memory, follow these steps:

Step	Action
1	Set the rotary switch to **Clear**.
2	Power up the controller.
3	Power down the controller.
4	Set the rotary switch to **A** or **B**.

When you next power up the controller, if you have cleared the standby controller, the primary controller transfers the application to the controller.

SFP Socket

Each controller includes one SFP socket, to which you can connect either a fiber optic or a copper transceiver:

To insert a transceiver:

Step	Action
1	Verify that the controller is de-energized.
2	Position the transceiver so that its label is oriented to the left.
3	Press the SFP transceiver firmly into the socket until you feel it snap into place. **NOTE:** If the SFP transceiver resists, verify the orientation of the transceiver and repeat these steps.

To remove a transceiver:

Step	Action
1	Verify that the controller is de-energized.
2	Pull out the latch to unlock the transceiver.
3	Pull on the transceiver to remove it.

NOTICE

POTENTIAL EQUIPMENT DAMAGE

Remove power from the controller before inserting or removing the SFP transceiver.

Failure to follow these instructions can result in equipment damage.

NOTE: For part numbers and other information regarding the available transceivers, refer to the description of controller Hot Standby link transceivers, page 50.

Each SFP socket comes with a stopper. When the SFP socket is not connected to a transceiver, cover the unused socket with the cover to keep out dust.

Wiring Guidelines

These rules must be applied when wiring the system:

- Communication wiring must be kept separate from the power wiring. Route these 2 types of wiring in separate cable ducting.
- Verify that the operating conditions and environment are within the specification values.
- Use proper wire sizes to meet voltage and current requirements.
- Use copper conductors (required).
- Use twisted pair, shielded cables for networks, and fieldbus.

Use shielded, properly grounded cables for all communication connections. If you do not use shielded cable for these connections, electromagnetic interference can cause signal degradation. Degraded signals can cause the controller or attached modules and equipment to perform in an unintended manner.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Use shielded cables for all communication signals.
- Ground cable shields for all communication signals at a single point[1].
- Route communication separately from power cables.

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

[1] Multipoint grounding is permissible if connections are made to an equipotential ground plane dimensioned to help avoid cable shield damage in the event of power system short-circuit currents.

To conform to IEC 61010 standards, route primary wiring (wires connected to power mains) separately and apart from secondary wiring (extra low voltage wiring coming from intervening power sources). If that is not possible, double insulation is required such as conduit or cable gains.

The use of shielded cables requires compliance with the following wiring rules:

- For protective earth ground connections (PE), metal conduit or ducting can be used for part of the shielding length, provided there is no break in the continuity of the ground connections. For functional earth ground (FE), the shielding is intended to attenuate electromagnetic interference and the shielding must be continuous for the length of the cable. If the purpose is both functional and protective, as is often the case for communication cables, the cable must have continuous shielding.
- Wherever possible, keep cables carrying one type of signal separate from the cables carrying other types of signals or power.

Follow all local and national safety codes and standards.

SD Memory Card

BMXRMS004GPF SD Memory Card

The BMXRMS004GPF memory card is a 4 GB, Class 6 card rated for industrial use. The SD memory card slot resides behind the door on the front of the controller.

You can use a BMXRMS004GPF memory card for application and data storage.

You can use a BMXRMS004GPF memory card for storage of:

- The M580 safety-related project application.
- Data for the non-safety-related tasks (MAST, FAST, AUX0, AUX1).

  NOTE:

  - Data cannot be stored on the SD memory card for the SAFE task.
  - The SD memory card is not included in the safety-related application.

You can insert and extract the card while power is ON and the controller is in RUN mode. However, to avoid data losses, use system bit %S65 to make a system request to stop data access to the card before extracting it from the controller.

NOTE: Other memory cards, including those used in M340 controller, are not compatible with M580 controllers. If you insert an incompatible SD memory card in the controller:

- The controller remains in NOCONF state (see Modicon M580, Hardware, Reference Manual).
- The controller **BACKUP** LED turns ON.
- The memory card access LED remains flashing.

The BMXRMS004GPF memory card is formatted specifically for the M580 controllers. If you use this card with another controller or tool, the card may not be recognized.

Memory Card Characteristics

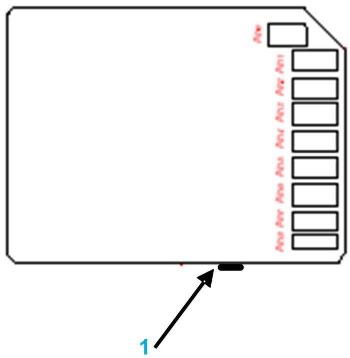
The BMXRMS004GPF memory card presents the following characteristics:

Characteristic	Value
global memory size	4 GB
application backup size	200 MB
data storage size	3.8 GB
write/erase cycles (typical)	100,000
operating temperature range	–40...+85 °C (–40...+185 °F)
file retention time	10 years
memory zone for FTP access	data storage directory only

NOTE: Due to formatting, wear-out, and other internal mechanisms, the actual available capacity of the memory card is slightly lower than its global size.

Read/Write Card Switch

The BMXRMS004GPF memory card has a read/write access switch along its non-beveled side edge, which you can use to help protect the card against non-permitted write access:



1 Read/write access switch

Formatting the Memory Card

The formatting procedure is described in *Formatting the Memory Card* topic in the *EcoStruxure™ Control Expert System Block Library*.

Hot Standby System States

Controller State Versus Hot Standby System State

The state of the Hot Standby system depends on the operating state of the controller. These Hot Standby states are supported:

Controller Operating State	Hot Standby System State
INIT	INIT
STOP	STOP

Controller Operating State	Hot Standby System State
RUN	PRIMARY with standby counterpart
	PRIMARY without standby counterpart
	STANDBY
	WAIT

This list describes the Hot Standby states:

- **Primary**: The controller controls all system processes and devices:
  - It executes program logic in a non-safety-related controller, and both process and safety-related program logic in a safety-related controller.
  - It receives input from, and controls output to, distributed equipment and RIO drops.
  - If connected to a controller in standby state, the primary controller verifies the status of, and exchanges data with, the standby controller.

  In a Hot Standby network, both controllers can be primary if both the Hot Standby and Ethernet RIO links are not functioning. When either of these two links is restored, the controller does one of the following:
  - Remains in the primary state.
  - Transitions to the standby state.
  - Transitions to the wait state.
- **Standby**: The standby controller maintains a state of readiness. It can take control of system processes and devices if the primary controller cannot continue to perform these functions:
  - It reads the data and the I/O states from the primary controller.
  - It does not scan distributed equipment, but receives this information from the primary controller.
  - It executes program logic. You can configure the standby controller to execute:

    - The first section of program logic (the default setting); or

    - Specified sections of program logic, including the MAST and FAST task sections.

      NOTE: You can specify if a section is to be executed in the **Condition** tab the **Properties** dialog box for each section.
  - On each scan, it verifies the status of the primary controller.

    NOTE: When a controller is in Standby mode, both the module health status (MOD_HEALTH) and the channels health status (CH_HEALTH) of safety-related I/O modules are set to FALSE in the Standby controller DDDT. In this case, you can diagnose the health of safety-related I/O modules by monitoring their status in the Primary controller DDDT.

- **Wait**: The controller is in RUN mode, but cannot act as either primary or standby. The controller transitions from the wait state to either the primary or standby state, when all preconditions for that state exist, including:
  - The state of the Hot Standby link.
  - The state of the Ethernet RIO link.
  - The presence of at least one connection with an Ethernet RIO drop.
  - The position of the **A/B/Clear** rotary switch on the rear of the controller.
  - The state of the configuration. For example:

    - If a firmware mismatch exists, the `FW_MISMATCH_ALLOWED` flag is set.

    - If a logic mismatch exists, the `LOGIC_MISMATCH_ALLOWED` flag is set.

  In the wait state, the controller continues to communicate with other modules on the local rack, and can execute program logic, if configured to do so. You can configure a controller in wait state to execute:
  - Specific sections of program logic in a non-safety-related controller (or process program logic in a safety-related controller), specified in the **Condition** tab of the **Properties** dialog box for each section.
  - The first section of program logic in a non-safety-related controller (or the first section of process program logic in a safety-related controller).
  - No program logic for a non-safety-related controller (or no process program logic for a safety-related controller).
- **INIT**: The controller and the Hot Standby system are initializing.
- **STOP**: The controller is in STOP mode. On the STOP to RUN transition, the controller may move to the wait, standby, or primary state. This transition depends on the state of the Ethernet RIO and Hot Standby links, and on the position of the **A/B/Clear** rotary switch on the rear of the controller.

  NOTE: In addition to the controller operating states listed here, other operating states that are not related to the Hot Standby system (see Modicon M580, Hardware, Reference Manual) exist.

Controller Functions by Hot Standby System State

A controller performs these functions, depending on its Hot Standby state:

Controller functions	Hot Standby system states		
	Primary	**Standby**	**Wait**
RIO drops	YES	NO	NO
Distributed equipment	YES	NO	NO

Controller functions	Hot Standby system states		
	Primary	Standby	Wait
Execution of program logic (non-safety-related controller) or process task logic (safety-related controller)	YES	Depending on configuration, STANDBY controller can execute: • First section (default) • Specified sections (which can include the MAST and FAST sections) • None	Depending on configuration, WAIT controller can execute: • First section (default) • Specified sections (which can include the MAST and FAST sections) • None
Execution of safety-related logic (safety-related controller)	YES	NO	NO
Program Data Exchange (non-safety-related controller) or Process Data Exchange (safety-related controller)	YES	YES	NO
Safety-related Data Exchange (safety-related controller)	YES	YES	NO
1. Data exchange is controlled by the **Exchange on STBY** attribute.			

Configuration Compatibility

Control Expert Version Requirement

An M580 non-safety-related Hot Standby system can be configured using Control Expert L or XL version 11.0 or later. By contrast, an M580 safety-related Hot Standby system can only be configured using Control Expert XL Safety version 14.0 or later.

Controller Hardware

Confirm that the primary controller and the standby controller consist of compatible hardware, including:

- Controller
- Rack
- Power supply
- Communication modules

NOTE: No I/O modules can be mounted on the local rack. Refer to the M580 Hot Standby Local Rack, page 47.

Controller Compatibility

An application created for a specific controller may not be compatible with other controllers. The M580 Hot Standby system compares the applications in the primary controller with the application in the standby controller to determine if the applications are compatible.

NOTE: An application created for a non-safety-related controller cannot be run on a safety-related controller, and an application created for a safety-related controller cannot be run on a non-safety-related controller.

For example:

- A Quantum 140CPU67•6• Hot Standby controller application is not downloadable to M580 BMEH58•040 Hot Standby controllers.
- An M580 BMEP58•0•0 controller application is not downloadable to M580 BMEH58•040 Hot Standby controllers.
- As described in the following table, an application designed for one M580 BMEH58•040 Hot Standby controller may not be downloadable to other M580 Hot Standby controllers.

The following table describes the compatibility of applications among non-safety-related M580 Hot Standby controllers:

An application built for:	Can be downloaded to the following controllers:		
	BMEH582040	**BMEH584040**	**BMEH586040**
BMEH582040	X	X	X
BMEH584040	-	X	X
BMEH586040	-	-	X
X Can receive and execute the application. **-** Cannot receive and/or execute the application.			

Standards and Certifications

Online Help

From the Control Expert online help, you can access the standards and certifications that apply to the modules in this product line by referring to the *Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications* guide.

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

Safety-Related System Certifications

For certifications relating exclusively to safety-related modules, refer to the *Modicon M580 Safety Manual* (see Modicon M580, Safety Manual).

Hardware Components in an M580 Hot Standby System

Overview

An M580 Hot Standby system requires the use of two local main racks: the primary and the standby. It can also include:

- M580 RIO drops
- Quantum RIO drops
- Distributed equipment

NOTE: An M580 Hot Standby system does not support Premium racks and I/O.

BMEH58•040 Controllers for High Availability Architectures

Introduction

The different references of redundant controllers for non-safety-related high availability architectures are:

- BMEH582040, BMEH582040C
- BMEH584040, BMEH584040C
- BMEH584040, BMEH584040C

Ruggedized Version

Some of the controllers are available in both standard and ruggedized version.

Controllers with the letter C appended to the reference are coated modules. They can be used at extended temperatures -25°C to 60°C (-13 °F to +140 °F) and in some harsh chemical environments.

The coated controllers have the same performance characteristics as the standard controllers.

M580 Controller Integrated Communication Ports

- Ethernet service port (RJ45): Engineering and Maintenance console (EcoStruxure Control Expert, Web browser, external tools), HMI, SCADA, Distributed equipment (DIO)
- Ethernet device network dual ports (RJ45) - RSTP: RIO network (RIO/DIO scanner)
- USB type mini B port: Programming port for Engineering Console (EcoStruxure Control Expert)

High Availability Architecture Capacity

The table below gives the maximum number of local racks, remote racks and distributed equipment a redundant M580 controller can manage in a high availability architecture.

M580 Architecture		**BMEH582040** **BMEH582040C**	**BMEH584040** **BMEH584040C**	**BMEH586040** **BMEH586040C**
Local racks (main + extension)		1	1	1
Remote I/O drops(1)		8	16	31
Product compatibility with Quantum (Support of Quantum Ethernet I/O and LL984 Editor)		No	Yes	Yes
Distributed equipment (DIO, CSIO)	EtherNet/IP or Modbus TCP devices (scanned by controller)	61	61	61
	EtherNet/IP or Modbus TCP devices (scanned by controller and Ethernet modules BMENOC0301 / BMENOC0311)	317	573	573
	EtherNet/IP or Modbus TCP devices (scanned by controller and Ethernet module BMENOC0302(H))	317	512	512

(1) One RIO Drop can support up to 2 racks (via X-bus extension).

NOTE: For product compatible with Quantum, a remote I/O drop can be either an X80 RIO Drop (RIO) or a Quantum Ethernet I/O drop, otherwise a remote I/O drop can only be an X80 RIO Drop (RIO).

Communication Modules

The table below gives the maximum number of Ethernet modules a redundant M580 controller can manage in a high availability architecture.

NOTE: The Ethernet modules are located on the local racks.

Module (Ethernet Network)	BMEH582040 BMEH582040C	BMEH584040 BMEH584040C	BMEH586040 BMEH586040C
BMENOC0301 (EtherNet/IP and Modbus TCP)	2	4	4
BMENOC0302(H) (Advanced EtherNet/IP)	2	4	6
BMENOC0311 (FactoryCast)	2	4	4
BMENOC0321 (IP Forwarding)	2	2	2
BMENOP0300 (IEC 61850)	2	4	4
BMENOR2200H (DNP3 NET / IEC 60870-5-104)	2	4	4
Cumulative maximum number of Ethernet networks modules (BMENOC/BMENOP/ BMENOR) listed above.	2	4	4
BMENUA0100 (OPC UA)	2	2	2
NOTE: • The redundant controllers do not support BMXNOR0200H modules, nor BMXNGD0100 (OPC UA) modules. • The BMENOC0302(H) module is not compliant with safety-related controllers.			

The table below gives the maximum number of fieldbus modules a redundant M580 controller can manage in a high availability architecture.

NOTE: The location of the fieldbus modules on the architecture (local racks and/or remote drops) is dependent on the type of module.

Module (Fieldbus)	BMEH582040 BMEH582040C	BMEH584040 BMEH584040C	BMEH586040 BMEH586040C
BMXEIA0100(1) (AS-Interface)	4	8	8
BMXNOM0200 (Modbus and Character Serial Link)	Only supported on remote drops, the number of modules is limited by the number of channels configured. A BMXNOM0200 channel counts as an application-specific channel.		
BMECXM0100 (CANopen)	Only supported on remote drops, each BMECXM0100 module counts as a distributed equipment (DIO).		
PMEPXM0100 (PROFIBUS DP)	2	6	10
(1) The BMXEIA0100 module can only be located on remote drops.			

M580 Controller I/O Channels

The table below gives the maximum number of I/O channels. The values may not be achieved depending on the I/O density and the number of available rack slots.

NOTE: Maximum number of I/O channels (Discrete, and Analog) as well as application-specific channels is not cumulative.

Maximum number of I/O channels		**BMEH582040** **BMEH582040C**	**BMEH584040** **BMEH584040C**	**BMEH586040** **BMEH586040C**
Local X80 I/O(2)		N/A	N/A	N/A
Remote X80 I/O	Discrete I/O channels	8192	16384	31744
	Analog I/O channels	2048	4096	7936
	Application specific-channels(3)	288	576	1162

(1) Maximum number of I/O channels (Discrete, and Analog) as well as application-specific channels is not cumulative.

(2) Redundant controllers do not support local X80 I/O.

(3) Application-specific channels include counters, time-stamping, SSI encoder, Motion control, Serial and Frequency input modules.

M580 Controller Internal Memory Size

This table shows the program and data memory capacity for non-safety-related redundant M580 controllers:

Internal Memory Capacity	**BMEH582040** **BMEH582040C**	**BMEH584040** **BMEH584040C**	**BMEH586040** **BMEH586040C**
Program Process	8 MB	16 MB	64 MB
Data Process	768 KB	2048 KB	Up to 64 MB
Data Storage	4 GB	4 GB	4 GB

Application Code Execution Performance

This table shows the performance of the application code for each M580 Hot Standby controller for distributed architectures. The performance is converted in number of instructions executed per ms (Kinstr/ms):

Performance (Kinstr/ms)	BMEH582040 BMEH582040C	BMEH584040 BMEH584040C	BMEH586040 BMEH586040C
100% Boolean	10	40	50
65% Boolean + 35% fixed arithmetic	7.5	30	40
Kinstr/ms: 1,024 instructions per millisecond.			

NOTE: As of firmware version 4.XX, the limitation of the BMEH582040(C) is equivalent to the BMEH584040(C).

BMEH58•040S Controllers for Safety-Related High Availability Architectures

Introduction

The different references of redundant controllers for safety-related high availability architectures are:

- BMEH582040S
- BMEH584040S
- BMEH586040S

M580 Controller Integrated Communication Ports

- Ethernet service port (RJ45): Engineering and Maintenance console (EcoStruxure Control Expert, Web browser, external tools), HMI, SCADA, Distributed equipment (DIO)
- Ethernet device network dual ports (RJ45) - RSTP: RIO network (RIO/DIO scanner)
- USB type mini B port: Programming port for Engineering Console (EcoStruxure Control Expert)

High Availability Architecture Capacity

The table below gives the maximum number of local racks, remote racks and distributed equipment a redundant M580 controller can manage in a high availability architecture.

M580 Architecture		BMEH582040S	BMEH584040S	BMEH586040S
Local racks (main + extension)		1	1	1
Remote I/O drops(1)		8	16	31
Product compatibility with Quantum (Support of Quantum Ethernet I/O and LL984 Editor)		No	Yes	Yes
Distributed equipment (DIO, CSIO)	EtherNet/IP or Modbus TCP devices (scanned by controller)	61	61	61
	EtherNet/IP or Modbus TCP devices (scanned by controller and Ethernet modules (BMENOC0301 / BMENOC0311))	317	436	436
(1) One RIO Drop can support up to 2 racks (via X-bus extension). **NOTE:** For product compatible with Quantum, a remote I/O drop can be either an X80 RIO Drop (RIO) or a Quantum Ethernet I/O drop, otherwise a remote I/O drop can only be an X80 RIO Drop (RIO).				

Communication Modules

The table below gives the maximum number of Ethernet modules a safety-related redundant M580 controller can manage in a safety-related high availability architecture.

NOTE: The Ethernet modules are located on the local racks.

Module (Ethernet Network)	BMEH582040S	BMEH584040S	BMEH586040S
BMENOC0301 (EtherNet/IP and Modbus TCP)	2	4	4
BMENOC0311 (FactoryCast)	2	4	4
BMENOC0321 (IP Forwarding)	2	2	2
BMENOP0300 (IEC 61850)	2	4	4
BMENOR2200H (DNP3 NET / IEC 60870-5-104)	2	4	4
Cumulative maximum number of Ethernet networks modules (BMENOC/BMENOP/BMENOR) listed above.	2	4	4
BMENUA0100 (OPC UA)	2	2	2
NOTE: The safety-related redundant controllers do not support BMENOC0302(H), BMXNOR0200H modules, nor BMXNGD0100 (OPC UA) modules.			

The table below gives the maximum number of fieldbus modules a safety-related redundant M580 controller can manage in a safety-related high availability architecture.

NOTE: The location of the fieldbus modules on the architecture (local racks and/or remote drops) is dependent on the type of module.

Module (Fieldbus)	BMEH582040S	BMEH584040S	BMEH586040S
BMXEIA0100(1) (AS-Interface)	4	8	8
BMXNOM0200 (Modbus and Character Serial Link)	Only supported on remote drops, the number of modules is limited by the number of channels configured. A BMXNOM0200 channel counts as an application-specific channel.		
BMECXM0100 (CANopen)	Only supported on remote drops, each BMECXM0100 module counts as a distributed equipment (DIO).		
PMEPXM0100 (PROFIBUS DP)	2	6	10
(1) The BMXEIA0100 module can only be located on remote drops.			

M580 Controller I/O Channels

The table below gives the maximum number of I/O channels. The values may not be achieved depending on the I/O density and the number of available rack slots.

NOTE: Maximum number of I/O channels as well as application-specific channels is not cumulative.

Maximum number of I/O channels(1)		BMEH582040S	BMEH584040S	BMEH586040-S
Local X80 I/O(2)		N/A	N/A	N/A
Remote X80 I/O	Non-safety-related discrete I/O channels	8192	16384	31744
	Safety-related discrete I/O channels	2944	5888	11408
	Non-safety-related analog I/O channels	2048	4096	7936
	Safety-related analog I/O channels	544	576	2108
	Application specific-channels (3)	288	576	1116

(1) Maximum number of I/O channels (non-safety-related discrete, safety-related discrete, non-safety-related analog, and safety-related analog) as well as application-specific channels is not cumulative.

(2) Redundant controllers do not support local X80 I/O.

(3) Application-specific channels include counters, time-stamping, SSI encoder, Motion control, Serial and Frequency input modules.

M580 Controller Internal Memory Size

This table shows the program and data memory capacity for safety-related redundant M580 controllers:

Internal Memory Capacity	**BMEH582040S**	**BMEH584040S**	**BMEH586040S**
Program Process	8 MB	16 MB	64 MB
Safety-related program	2 MB	4 MB	Up to 64 MB
Data Process	768 KB	2048 KB	Up to 64 MB
Safety-related data	512 KB	1024 KB	Up to 64 MB
Data Storage	4 GB	4 GB	4 GB

Application Code Execution Performance

This table shows the performance of the application code for each M580 safety-related Hot Standby controller for distributed architectures. The performance is converted in number of instructions executed per ms (kinstr/ms):

Performance (Kinstr/ms)	**BMEH582040S**	**BMEH584040S**	**BMEH586040S**
100% Boolean	10	40	50
65% Boolean + 35% fixed arithmetic	7.5	30	40
Kinstr/ms: 1,024 instructions per millisecond.			

M580 Hot Standby Local Racks

Local Rack Equipment

Each local rack, the primary and the standby, in an M580 Hot Standby system includes the following equipment:

- rack
- Hot Standby controller
- power supply
- Hot Standby SFP socket for the controller

The primary and standby local rack may also include:

- up to six communication modules, depending on the controller, page 48
- an SD memory card for each controller

NOTE: An M580 Hot Standby local rack:

- Does not support I/O modules.
- Does not support serial communication modules.
- Cannot be extended.

Controllers

The Modicon M580 Hot Standby controllers include these features:

Feature	Controller: BMEH58...					
	2040(C)	**2040S**	**4040(C)**	**4040S**	**6040(C)**	**6040S**
RIO drops (main + extended racks)	8 drops (up to 2 racks per drop)	8 drops (up to 2 racks per drop)	16 drops (up to 2 racks per drop)	16 drops (up to 2 racks per drop)	31 drops (up to 2 racks per drop)	31 drops (up to 2 racks per drop)
BMENOC0301 / BMENOC0311 Ethernet scanners	2	2	4	4	4	4
BMENOC0302(H) Ethernet scanners	2	–	4	–	6	–

Feature		Controller: BMEH58...					
		2040(C)	2040S	4040(C)	4040S	6040(C)	6040S
Memory	Non-safety-related program (MB)	8 MB	8 MB	16 MB	16 MB	64 MB[1]	64 MB[2]
	Non-safety-related data (KB)	768 KB	768 KB	2048 KB	2048 KB		up to 64 MB[2]
	Safety-related program (MB)	–	2 MB	–	4 MB	–	16 MB[2]
	Maximum retained data (KB)	768	768	2048	2048	4096	4096
	Maximum configurable redundant transfer data (KB)	768	768	2048	2048	4096	4096
	Safety-related data (non-retained data) (MB)	–	512 KB	–	1024 KB	–	1024 KB[4]
	Max configurable safety-related redundant transfer data	–	512 KB	–	1024 KB	–	1024 KB
	Shared: Global -> Safety-related	–	16 KB[3]	–	16 KB[3]	–	16 KB[3]
	Shared: Safety-related -> Global	–	16 KB[3]	–	16 KB[3]	–	16 KB[3]
	Shared: Global -> Process	–	16 KB[3]	–	16 KB[3]	–	16 KB[3]
	Shared: Process -> Global	–	16 KB[3]	–	16 KB[3]	–	16 KB[3]
	Data storage (GB)	4[4]	4[4]	4[4]	4[4]	4[4]	4[4]

1. The BMEH586040 controllers support the sum of program and data up to the stated maximum.

2. Application Program (non-safety-related) + Application Data (non-safety-related non-retain data only) + Application Program (safety-related) + Application Data (safety-related) is less than 64 Mbytes. There is a global memory pool of 64 Mbytes on BMEH586040S controller for Application Program and Application Data.

3. This data is included in both the safety-related and non-safety-related data areas.

4. 2 GB without an external memory card

For an expanded description of the performance characteristics of M580 safety-related Hot Standby controllers, refer to the *Modicon M580 Safety System Planning Guide* (see Modicon M580, Safety System Planning Guide).

Controller Hot Standby Link SFP Transceivers

When installing fiber optic transceivers, do the following to help prevent dust and pollution inside the fiber optic cable.

NOTICE

EQUIPMENT DAMAGE

- Keep caps on jumpers and transceivers when not in use.
- Insert the optical cable into the transceivers carefully, respecting the longitudinal axis of the transceiver.
- Do not use force when inserting the cable into the transceivers.

Failure to follow these instructions can result in equipment damage.

Each M580 Hot Standby controller is fitted with an SFP socket that supports a copper or fiber-optic SFP transceiver. The available SFP transceiver modules include:

SFP Transceiver	Connection	Maximum Distance	Comment
490NAC0100	RJ45 copper	100 m	–
490NAC0201	single-mode fiber-optic	15 km	This transceiver operates in these wavelengths: • *receive*: 1270...1600 nm • *transmit*:1270...1360 nm

NOTE: A BMXNRP020• cannot be used to extend the Hot Standby link.

Hot Standby Link Copper Cables

If you use a 490NAC0100 copper transceiver, you can elect to use either of the following 5 m Ethernet shielded twisted pair Cat5e straight copper cables to make the Hot Standby connection:

- 490NTW00005: CE-compliant
- 490NTW00005U: UL-compliant

Secure Digital (SD) Memory Cards

Schneider Electric offers the BMXRMS004GPF SD memory card for Hot Standby controllers. The card is rated for industrial use.

Racks

Each local rack in an M580 Hot Standby system, both the primary and the standby, consists of a single rack. Extensions to the local rack are not permitted. These racks are supported:

- BMXXBP••00 X Bus
- BMEXBP••00 Ethernet/X Bus
- BMEXBP••02 Ethernet/X Bus (supports redundant power supply)

NOTE: Your choice of rack determines the available power supply, which can be either a single or a redundant power supply.

Power Supplies

As noted above, the choice of power supply depends on the previous choice of rack. Use these power supplies in M580 Hot Standby systems:

Power Supply	Power Supply Module Name
Redundant power supplies[1]	BMXCPS3522(H)
	BMXCPS3522S
	BMXCPS4002
	BMXCPS4002S
	BMXCPS4022(H)
	BMXCPS4022S
Single power supplies[2]	BMXCPS2000
	BMXCPS2010
	BMXCPS3020
	BMXCPS3500
	BMXCPS3500
	BMXCPS3540
	BMXCPS4002
1. Require a BMEXBP••02 or BMEXBP••01H rack.	
2. Require a BMEXBP••00 or BMXXBP••00 rack.	

Exception: You can install BMXCPS4002 modules only on the following dual-bus racks (Ethernet and X Bus):

- BMEXBP0602
- BMEXBP1002

Other Local Rack Modules

The main local rack in an M580 Hot Standby system can support communication modules and embedded switching modules.

NOTE: I/O modules are not supported on the local main rack, but can be added to (e) X80 EIO drops and as distributed equipment.

You can add these modules to an M580 Hot Standby local rack:

Module Name	Description	Maximum per local rack
BMENOC0302(H)	Ethernet communication module	up to six communication modules
BMENOC0301	Ethernet communication module for distributed equipment	up to four communication modules (depends on controller, page 48)
BMENOC0311	FactoryCast Ethernet communication module for distributed equipment	
BMENOC0321[2]	Ethernet communication module that provides transparency between a control network and a device network	
BMENOS0300	network option switch module	limited only by the number of available slots
BMXNRP0200[1]	copper multi-mode fiber converter module	-
BMXNRP0201[1]	copper single mode fiber converter module	-
PMXNOW0300[1]	combination wireless / 3-port wired switch module	-

1. A BMXNRP020• can be used to extend the EIO link, but cannot be used to extend the Hot Standby link.

2. You can enable the IP forwarding service (see Modicon M580, BMENOC0321 Control Network Module, Installation and Configuration Guide) on only one BMENOC0321 module per local rack.

M580 (e)X80 RIO Drops

Introduction

All BMEH58•040(S) Hot Standby controllers support (e)X80 RIO drops. An (e)X80 RIO drop includes a main rack, and may also include an extended rack. The drop includes (e)X80 I/O modules.

An (e)X80 RIO drop can be connected via copper cable directly to the main ring, or as a sub-ring.

(e)X80 RIO drops provide deterministic communication so that (e)X80 RIO modules synchronize with controller tasks.

> **NOTE:** M580 Hot Standby controllers support MAST and FAST tasks for M580 (e)X80 RIO drops. AUX0 and AUX1 tasks are not supported.

An M580 Hot Standby system can support a maximum of 31 RIO drops. The maximum number of drops depends on the controller, page 48.

Selecting an (e)X80 EIO Adapter Module

Each remote drop contains one (e)X80 EIO adapter module. The following adapter modules are available:

- BMXCRA31200 X80 standard EIO adapter module
- BMXCRA31210 X80 performance EIO adapter module
- BMECRA31210 eX80 performance EIO adapter module

A BM•CRA312•0 adapter module in an (e)X80 EIO drop can be installed only in slot 00 (directly to the right of the power supply) in the main rack of the drop.

Remote Racks and Power Supplies

Each rack in an (e)X80 RIO drop contains its own power supply module. The choice of power supply module depends on the selected rack. Refer to the description of the Hot Standby local rack, page 47 for a presentation of available M580 racks and power supplies.

The following racks can be used in a Modicon M580 Hot Standby system:

Rack Name	Main Remote Rack	Extended Remote Rack
BMEXBP••00 Ethernet/X Bus	X	X[2]
BMEXBP••02[1] Ethernet/X Bus	X	X[2]
BMEXBP••02 H[1] Ethernet/X Bus	X	X[2]
BMXXBP••00 X Bus	X	X
X: allowed 1. Requires a four-slot, redundant power supply. 2. Only X Bus ports are supported on the extended remote rack.		

The number of racks in the (e)X80 RIO drop depends on the selected (e)X80 EIO adapter module.

- If you select a BMXCRA31200 X80 standard EIO adapter module, you cannot add an extended rack to the drop.
- If you select a BM•CRA31210 (e)X80 performance EIO adapter module, you can add one extended rack to the drop.

RIO Modules

The M580 Hot Standby system supports all M580 input and output modules. For more information, refer to the topic *Modicon X80 I/O Modules* in the *Modicon M580 Frequently Used Architectures System Guide*.

Disconnecting an RIO Drop

When an (e)X80 RIO drop is disconnected from the Ethernet RIO main ring:

- Outputs enter the fallback state.
- Inputs present a value of zero to the controller.

Quantum RIO Drops

Introduction

The BMEH584040(S) and BMEH586040(S) Hot Standby controllers support Quantum RIO drops. The Quantum RIO drops can contain only non-safety-related modules. A Quantum RIO drop consists of a main remote rack, and may contain an extended rack. For the type of modules that reside on a Quantum RIO drop, refer to the *Quantum EIO Remote I/O Modules Installation and Configuration Guide* (see Quantum EIO, System Planning Guide).

A Quantum RIO drop can be connected via copper wire directly to the main ring, or as a sub-ring.

Quantum RIO drops provide deterministic communication so that Quantum RIO modules synchronize with controller tasks.

> **NOTE:** M580 Hot Standby controllers support only MAST tasks for Quantum I/O. FAST, AUX0 and AUX1 tasks are not supported.

An M580 Hot Standby system can support a maximum of 31 RIO drops. The maximum number of drops depends on the controller, page 48.

Selecting a Quantum RIO Adapter Module

Each RIO drop contains one 140CRA31200 adapter module.

By default, the adapter module is placed in position 1 in the main remote rack. However, you can place it in another position in the rack.

Remote Racks and Power Supplies

Both the main remote rack and an optional extended remote rack can be any Quantum 140XBP0••00 rack.

Each rack has its own power supply. You can add a Quantum power supply to any available slot in a main remote and extended remote rack.

Remote I/O Modules

The M580 Hot Standby system supports all Quantum I/O modules in a Quantum RIO drop. For more information, refer to the topic *I/O Devices* in the *Quantum EIO System Planning Guide* (see Quantum EIO, System Planning Guide).

Disconnecting an RIO Drop

When a Quantum RIO drop is disconnected from the Ethernet RIO main ring:

- Outputs enter the fallback state.
- Inputs present a value of zero to the controller.

Planning a Typical M580 Hot Standby Topology

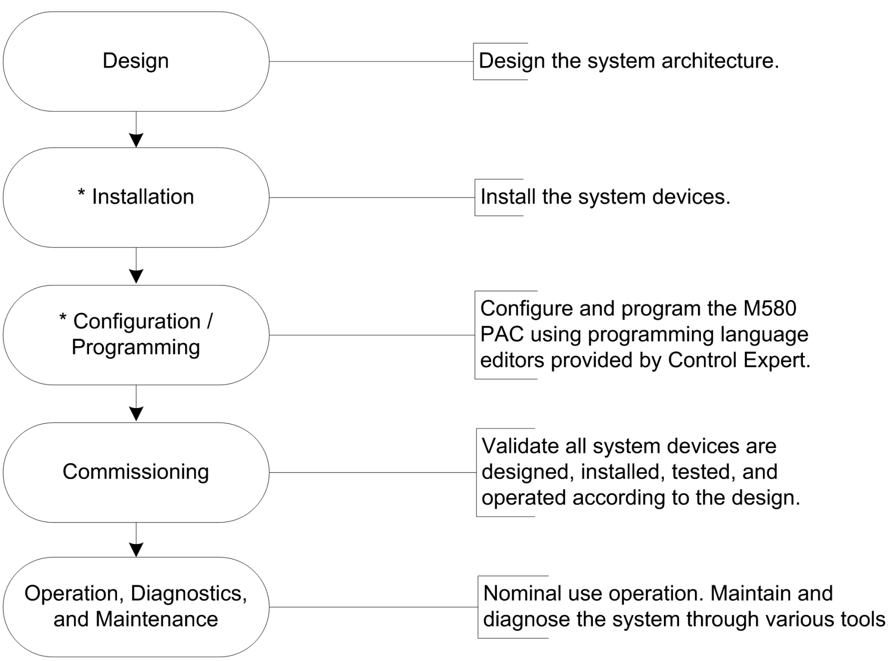
Overview

This chapter describes typical Hot Standby topologies.

Project Life Cycle

Project Life Cycle

Before planning your network topology, consult the life cycle of a project within the M580 system.



* Installation and configuration/programming instructions are explained in the *Modicon M580 Hardware Guide* (see Modicon M580, Hardware, Reference Manual) and the respective Modicon M580 communication/adapter module user guides.

Modicon M580 Typical System Introduction

Introduction

A typical Modicon M580 system is designed and tested for simultaneous use of:

- an Ethernet main local rack (see *Modicon M580 Frequently Used Architectures System Guide*) and the ability to extend to other local racks
- RIO drops (see *Modicon M580 Frequently Used Architectures System Guide*) that support Ethernet and X Bus communications across the backplane
- Ethernet distributed equipment (see *Modicon M580 Frequently Used Architectures System Guide*)
- network option switch modules that attach RIO drops and distributed equipment to the M580 system (see *Modicon M580 Frequently Used Architectures System Guide*)
- control network module that creates transparency between the device network and the control network, page 87
- RIO and distributed equipment integrated on the same physical network
- RIO and DIO sub-rings that communicate with the RIO main ring
- third-party modules and devices
- daisy-chain ring architectures provided by communication modules with dual Ethernet ports

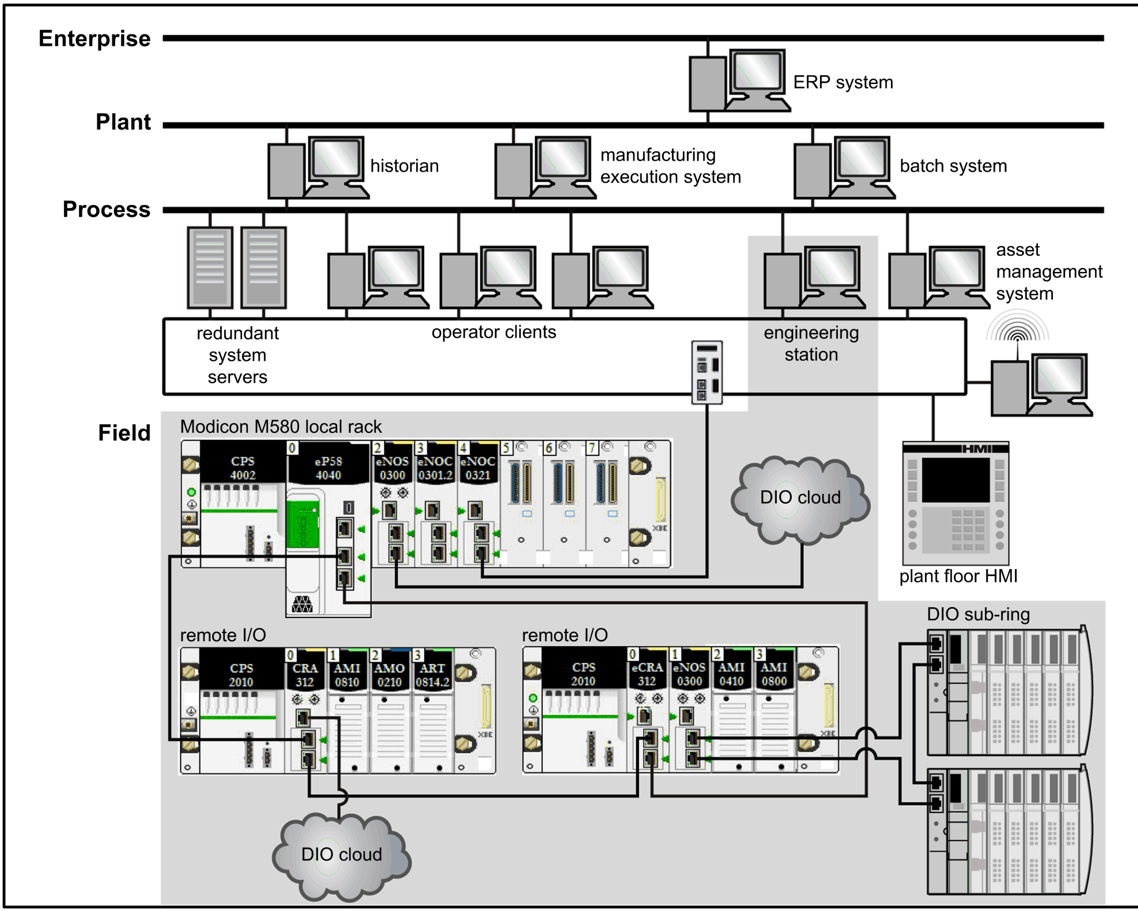
An M580 system provides automatic network recovery of less than 50 ms and *deterministic* RIO performance.

An M580 system uses Modicon X80 I/O modules, many of which can be used in an M340 system. The system also supports several Ethernet-based *eX80* I/O modules, which can be installed on both the main local rack and main remote racks. M580 also supports Premium I/O modules installed on an extended local rack.

NOTE: To use a dual-ring switch (DRS) to connect distributed equipment to the M580 network, refer to the *Modicon M580 Complex Topologies System Guide*.

Typical M580 Architecture

This is a typical M580 architecture. It includes the enterprise, plant, process, and field levels of a manufacturing plant. An M580 RIO system is shown at the field level.



⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Do not install more than one standalone controller in an M580 device network.
- Do not install more than one set of primary-standby Hot Standby controllers in an M580 Hot Standby system.

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

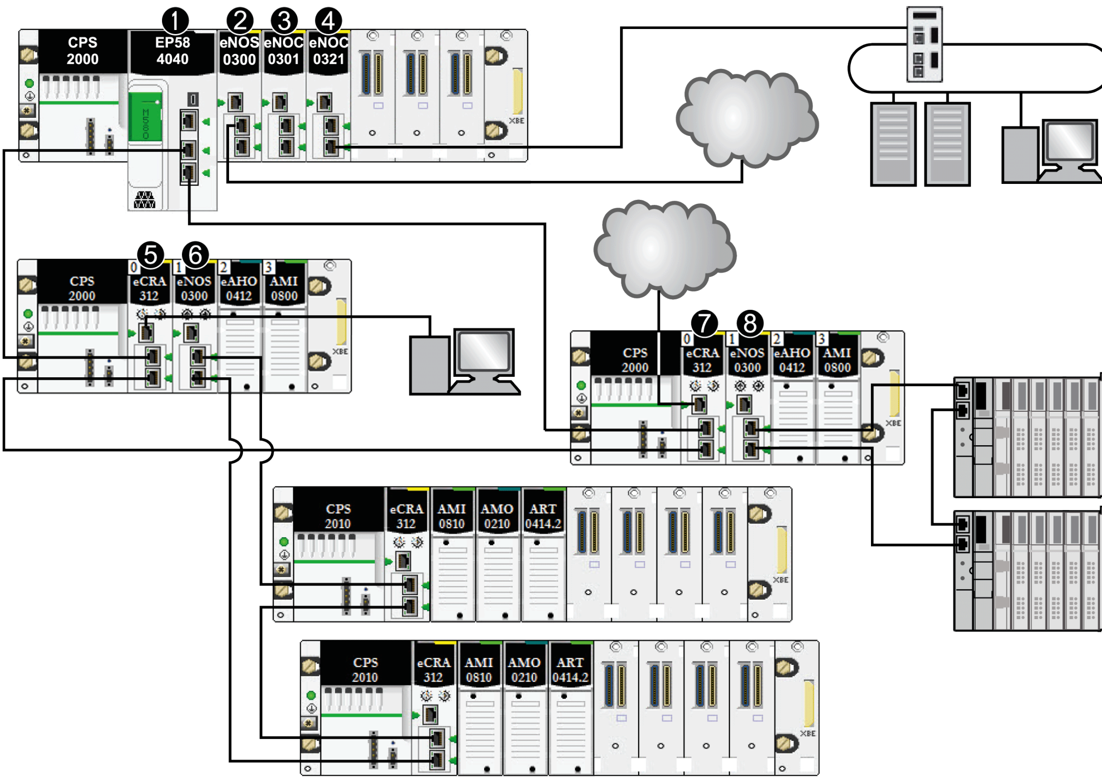
M580 Life Cycle

The life cycle of an M580 network includes these phases:

Phase	Feature	Description
design phase	standard	reduce the learning and engineering time (use standard Ethernet technology, Modicon X80 common modules, and Control Expert software for device configuration
	open	collaborate with third-party solutions
	flexible	adapt the control architecture to the plant topology
	efficient	design the solution without constraints
operation phase	transparent	provide access to I/O modules and devices from the control network
	accessible	change configuration without stopping the process, get diagnostic information from any location in the network, no switch required to create a complete M580 system
renew phase	sustainable	preserve long-term investment, allow smooth migration

M580 RIO Example

This is an example of a typical M580 system that integrates RIO modules and distributed equipment in one Ethernet I/O device network:



1 An M580 controller with Ethernet I/O scanner service on the local rack is connected to the RIO main ring. (For the Ethernet I/O scanner service, select a controller with a commercial reference that ends in *40*.)

2 A BMENOS0300 network option switch module on the local rack connects a DIO cloud to the RIO main ring.

3 A BMENOC0301 Ethernet communication module, connected to the controller through the Ethernet backplane, manages distributed equipment on the device network.

4 A BMENOC0321 control network module on the local rack creates transparency between the device network and the control network.

5 A PC for port mirroring is connected to the service port of a BMECRA312•0 (e)X80 EIO adapter module.

6 A BMENOS0300 network option switch module on an RIO drop manages an RIO sub-ring.

7 A DIO cloud is connected to the service port of a BMECRA31210 eX80 performance EIO adapter module.

8 A BMENOS0300 network option switch module on an RIO drop connects a DIO sub-ring to the RIO main ring.

NOTE: A BMENOC0301 / BMENOC0311 module supports distributed equipment through its Ethernet backplane connection to the controller **and** through its device network port(s) on the front panel, respecting the limitation of 128 devices scanned per BMENOC0301 / BMENOC0311 module.

Planning an M580 Hot Standby Topology

Linking Primary and Standby Controllers

The BMEH58•040 and BMEH58•040S primary and standby controllers are connected by a Hot Standby link, and may also be connected by an Ethernet link.

NOTE:

- Every M580 Hot Standby system design includes a Hot Standby link.
- The Ethernet link provides a redundant path on the main ring. Its presence helps prevent the occurrence of two primary controllers. Although the system can function without an Ethernet link connecting the primary and standby controllers, such a system is not optimal.

NOTE: For a description of maximum lengths between (e)X80 RIO drops, refer to the topic *Planning the Appropriate Network Topology* in the *Modicon M580 Frequently Used Architectures System Guide*.

Hot Standby Link Between Primary and Standby Controllers

Each BMEH58•040 and BMEH58•040S controller is fitted with an SFP socket for an SFP transceiver, page 50.

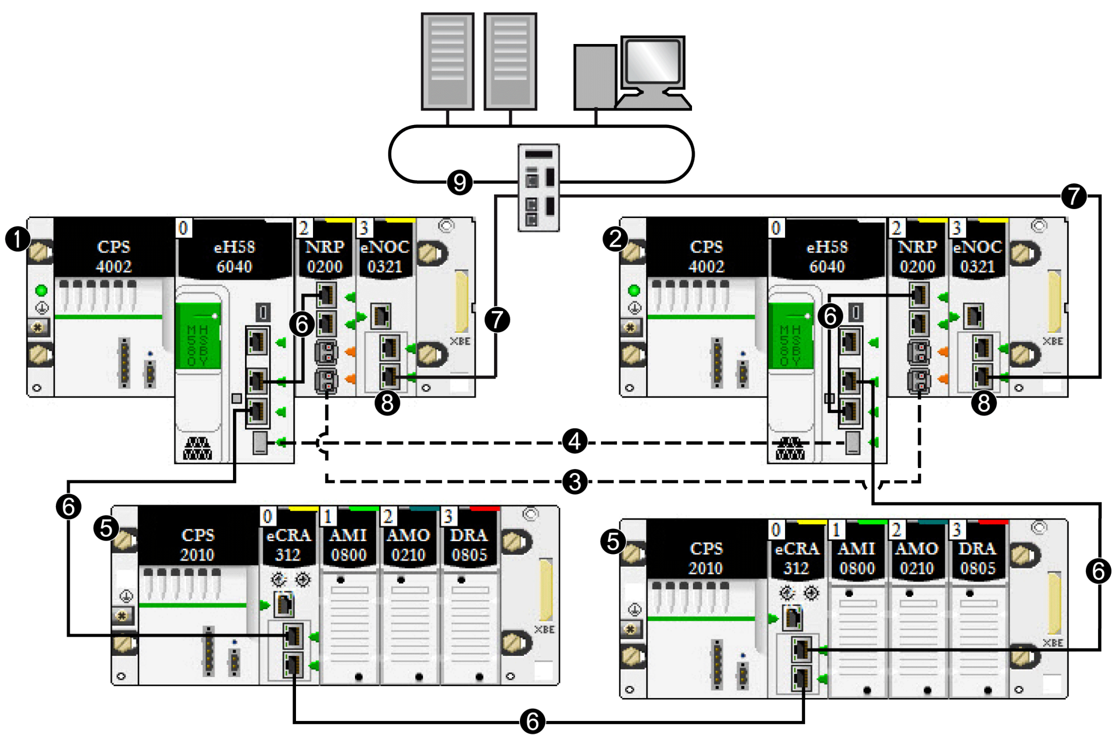
Ethernet Link Between Primary and Standby Controllers

The primary and standby BMEH58•040 or BMEH58•040S controllers can also be connected to each other by an Ethernet link. This link is typically part of an Ethernet RIO main ring, to which each Hot Standby controller is connected.

Each BMEH58•040 and BMEH58•040S controller includes two RJ45 connectors dedicated for use as a dual port to the main ring.

To create the Ethernet link between the primary and standby controllers, connect one of the dual Ethernet ports on the primary controller to a dual Ethernet port on the standby controller. You can do this in one of the following ways:

- Directly connect the primary and standby controllers using CAT-5e copper wire.
- Place a BMENOS0300 switch module, page 68 in each controller, or place a BMENOC0301 / BMENOC0311 or BMENOC0302(H) communication module, page 52 in each local rack, then connect the two modules.
- Connect each controller to a BMXNRP0201 fiber converter module, then connect the two BMXNRP0201 modules with single-mode fiber optic cable, as described below:



1 primary local rack with primary controller

2 standby local rack with standby controller

3 Ethernet RIO fiber optic link between primary and standby controllers (part of Ethernet RIO main ring)

4 Hot Standby fiber optic communication link

5 (e)X80 EIO drop

6 Ethernet RIO main ring

7 BMENOC0321 control network module redundant link

8 BMENOC0321 control network module providing transparency between the device network and the control network

9 control network

Like the Hot Standby link, your choice of connector and wiring determines the maximum distance of the Ethernet physical link:

- A copper wire link can extend a maximum distance of 100 m (328 ft).
- A fiber optic single mode link can extend a maximum distance of 15 km (9.3 mi).

Frequently Used M580 Hot Standby Topologies

Introduction

An M580 Hot Standby system includes, at minimum, two Hot Standby local racks, each containing a Hot Standby controller with Ethernet I/O scanner service. The system also may include one or more of the following:

- (e)X80 RIO drops, each containing a BM•CRA312•0 (e)X80 EIO adapter module, located on the main ring
- Quantum RIO drops, each containing a Quantum adapter module, located on the main ring

- distributed equipment, connected to the main ring via service ports of BM•CRA312•0 modules, but not residing directly on the ring

  NOTE: Distributed equipment can be connected to the main ring by either:

  - a DIO ring (if the devices support RSTP)
  - a single DIO daisy chain (non-ring)

  These connections can be made, for example, via the service port of a ••• CRA 312 •• module, or through a BMENOS0300 network option switch module.

NOTE: Premium remote I/O drops are not supported in an M580 Hot Standby system.

This topic describes the following frequently used network topologies for Hot Standby systems:

- an RIO main ring
- a DIO ring connected to:
  - a BMENOS0300 module, in a design that supports up to 64 devices
  - a BMENOC0301 / BMENOC0311 or BMENOC0302(H) communication module, in a design that supports up to 128 devices
- a DIO ring
- an RIO main ring, with a DIO daisy chain
- an RIO main ring, with a DIO ring
- an RIO main ring, with a DIO sub-ring
- an HMI connected to a Hot Standby topology
- SCADA connected to a Hot Standby topology

NOTE: In a Hot Standby system, you can install only one main ring, which connects to the controllers Ethernet backplanes. If your system contains an RIO main ring or a DIO ring that is connected to a non-isolated BMENOC0301 / BMENOC0311 or BMENOC0302(H) module on the local rack, verify that you isolate (disable the modules' Ethernet backplane ports) other BMENOC0301 / BMENOC0311 or BMENOC0302(H) modules to which you connect a DIO ring.

Connecting together the service ports of the primary and standby controllers can cause unintended system operation.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

Do not connect together the service ports of the Hot Standby controllers.

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

Connecting more than one module to both the backplane and an Ethernet network can cause a broadcast storm.

⚠ CAUTION

RISK OF BROADCAST STORM

Do not connect more than one module in a local rack to both the Ethernet backplane and an Ethernet network.

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

Use only one module in each local rack to connect an Ethernet network to the Ethernet backplane. That module can be:

- the controller, when RIO modules are used
- one BMENOS0300 module
- one BMENOC0301 / BMENOC0311 or BMENOC0302(H) module

  NOTE: You can have multiple BMENOC0301 / BMENOC0311 or BMENOC0302(H) modules in a local rack, each with its backplane port enabled, provided the Ethernet ports (the service port and the two network ports) are not used.

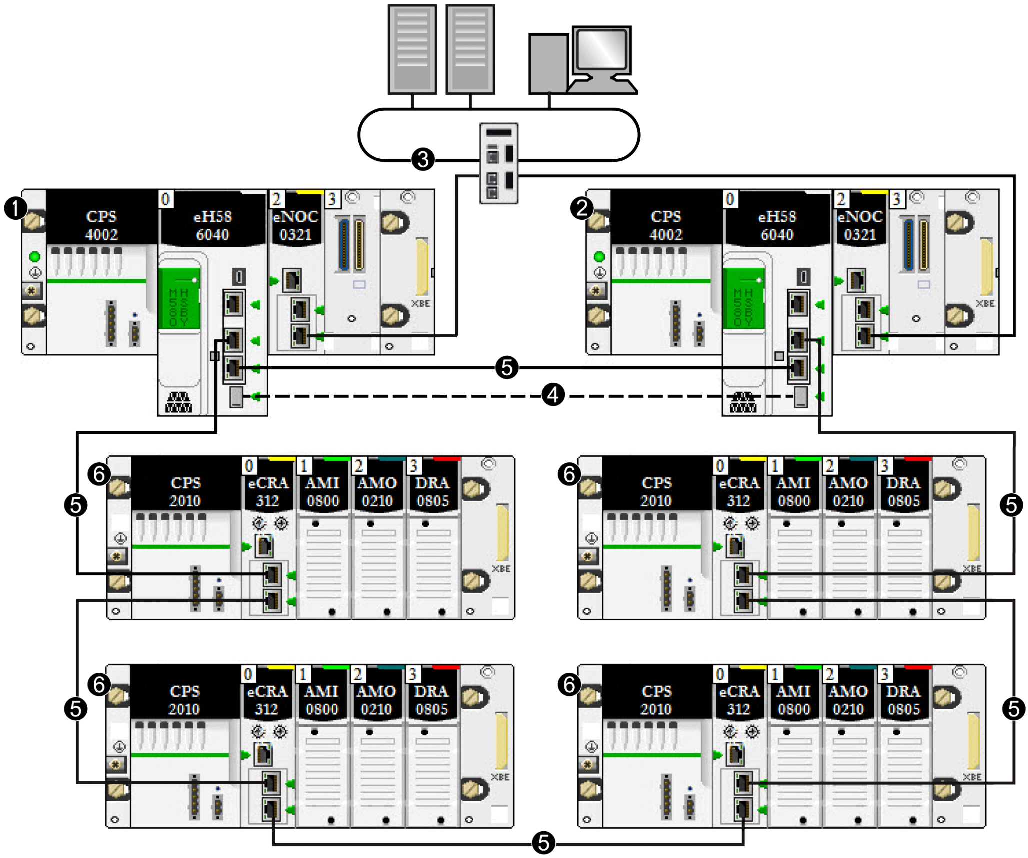
RIO Main Ring Architecture

A Hot Standby system with an RIO main ring consists of the following:

- two local Hot Standby racks, each containing a Hot Standby controller with Ethernet I/O scanner service
- one or more (e)X80 RIO drops or Quantum RIO drops, each containing an adapter module

In this topology, the (e)X80 RIO drops are connected directly to the main ring. No sub-rings, and no DRSs are used.

The following example presents a Hot Standby system with an M580 RIO main ring consisting of four (e)X80 RIO drops:



1 primary local rack with primary controller

2 standby local rack with standby controller

3 control network connected to a BMENOC0321 module on the local rack to provide transparency between the device network and the control network

4 Hot Standby communication link

5 RIO main ring

6 (e)X80 RIO drop

DIO Ring Architecture (up to 64 Devices)

A Hot Standby system with a DIO ring can be constructed using a BMENOS0300 module in place of a BMENOC0301 / BMENOC0311 or BMENOC0302(H) module. In this design, the controller scans the distributed equipment.

NOTE: Use this design to scan up to 64 distributed equipment I/O points.

The following example presents a DIO ring scanned by the controller via the BMENOS0300 modules.

Connecting more than one module to both the backplane and an Ethernet network can cause a broadcast storm.

⚠ CAUTION

RISK OF BROADCAST STORM

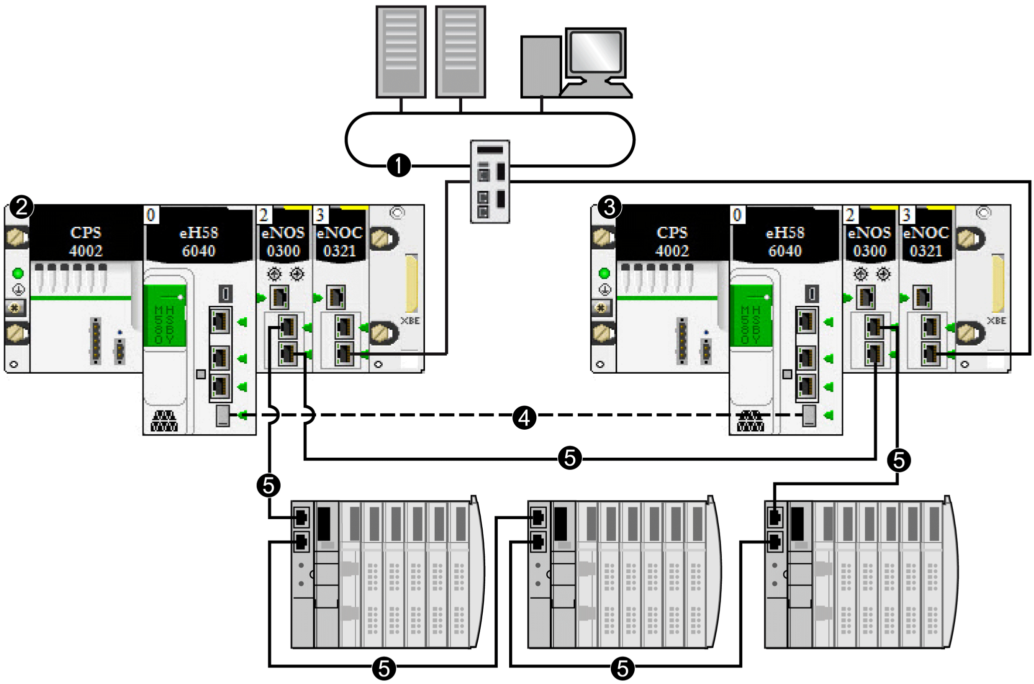
Do not connect more than one module in a local rack to both the Ethernet backplane and an Ethernet network.

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

Use only one module in each local rack to connect an Ethernet network to the Ethernet backplane. That module can be:

- the controller, when RIO modules are used
- one BMENOS0300 module
- one BMENOC0301 / BMENOC0311 or BMENOC0302(H) module

Because the Ethernet backplane port of each BMENOS0300 module is enabled, do not connect a dual Ethernet port on the primary controller to a dual Ethernet port on the standby controller. Otherwise, it can cause the Hot Standby system to cease functioning:



1 control network connected to a BMENOC0321 module on the local rack to provide transparency between the device network and the control network

2 primary local rack with primary controller

3 standby local rack with standby controller

4 Hot Standby communication link

5 DIO ring

NOTE: In this design:

- Up to 64 distributed devices can be included.
- Connect the two BMENOS0300 modules via a direct link.
- If there is an interruption in the direct connection between the two BMENOS0300 modules, the Hot Standby system may report communication issues in case the Ethernet traffic is heavily loaded (bit HSBY_SUPPLEMENTARY_LINK_ERROR in ECPU_HSBY_STS). This information has no effect on system behavior and disappears as soon as the direct connection is restored. However, you need to repair the direct connection between the two BMENOS0300 modules so that the system can remain operational in case a second interruption occurs.
- No link can be made connecting the dual Ethernet ports on the primary controller to the dual Ethernet ports on the standby controller.
- Only one DIO ring is supported.
- Set the rotary selectors on both of the BMENOS0300 modules to configure the top port as a service port, and the two lower ports as DIO ring ports.

DIO Ring Architecture (64 to 128 Devices)

A Hot Standby system with a DIO ring can be constructed using one or more BMENOC0301 / BMENOC0311 or BMENOC0302(H) modules to scan the distributed equipment.

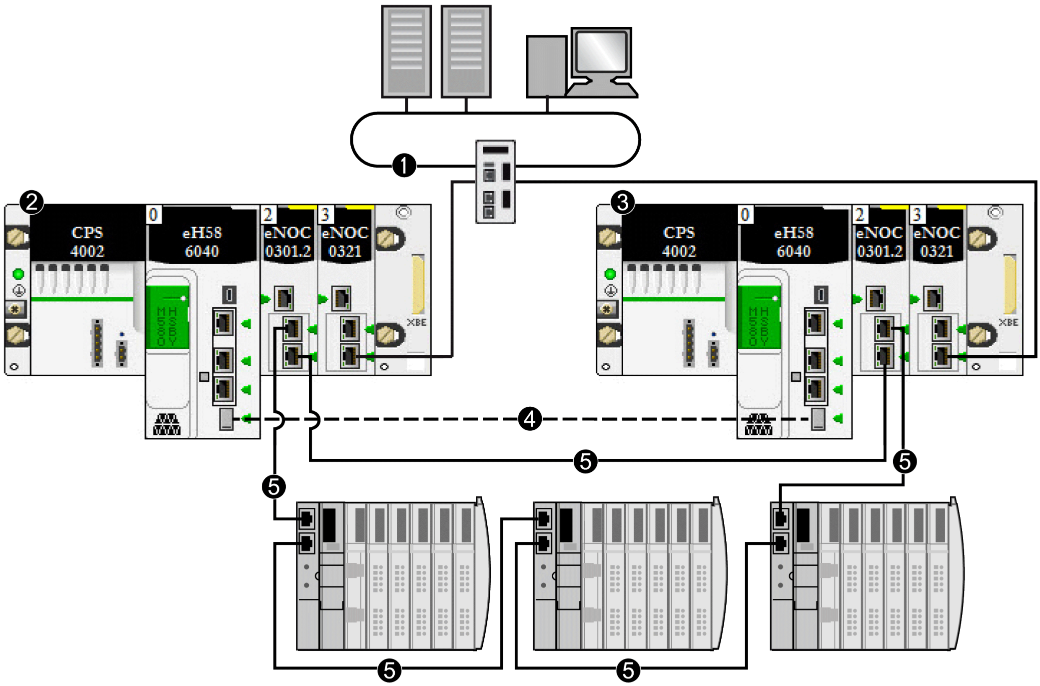
Distributed equipment in a DIO ring cannot comprise the main ring. Only (e)X80 and Quantum RIO drops are allowed on the main ring.

NOTE: Use this design to scan 64 or more distributed equipment I/O points.

Distributed equipment in a DIO ring can be connected to the service port of a BMENOC0301 / BMENOC0311 or BMENOC0302(H) module. However, if the system also supports an RIO main ring, disable the backplane port of the BMENOC0301 / BMENOC0311 or BMENOC0302(H) module.

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION Disable the Ethernet backplane port of each BMENOC0301 / BMENOC0311 or BMENOC0302(H) module in the Hot Standby configuration if the Hot Standby controllers support both an RIO main ring and distributed equipment. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

This example presents a DIO ring of distributed equipment scanned by the BMENOC0301 / BMENOC0311 or BMENOC0302(H) modules:



1 control network connected to a BMENOC0321 module on the local rack to provide transparency between the device network and the control network

2 primary local rack with primary controller

3 standby local rack with standby controller

4 Hot Standby communication link

5 DIO ring

NOTE: In this design:

- Up to 128 distributed devices can be included.

  Of these 128 connections

  - 16 are reserved for local slaves.
  - 112 are available for distributed equipment to be scanned.
- Connect the two BMENOC0301 / BMENOC0311 or BMENOC0302(H) modules via a direct link.
- In the **Services > RSTP** page of the BMENOC0301 / BMENOC0311 or BMENOC0302(H) DTM in Control Expert, set the **Bridge Priority** to **Root**.
- If there is an interruptionin the direct connection between the two BMENOC0301 / BMENOC0311 or BMENOC0302(H) modules, the Hot Standby system may report communication issues in case the Ethernet traffic is heavily loaded (bit HSBY_ SUPPLEMENTARY_LINK_ERROR in ECPU_HSBY_STS). This information has no effect on system behavior and disappears as soon as the direct connection is restored. However, you need to repair the direct connection between the two BMENOC0301 / BMENOC0311 or BMENOC0302(H) modules so that the system can remain operational in case a second interruption occurs.
- Because this design employs a non-isolated BMENOC0301 / BMENOC0311 or BMENOC0302(H) module with its Ethernet backplane port enabled no link can be made connecting the dual Ethernet ports on the primary controller to the dual Ethernet ports on the standby controller.
- Up to five additional BMENOC0301 / BMENOC0311 or BMENOC0302(H) modules can be added to both the primary and standby local racks, as part of the single DIO ring supported in this design.
- Only one DIO ring is supported.

For more details, refer to either:

- Modicon M580 BMENOC0301 / BMENOC0311 Ethernet Communications Module Installation and Configuration Guide (see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)
- Modicon M580 BMENOC0302 High Performance Ethernet Communication Module Installation and Configuration Guide

RIO Main Ring with DIO Daisy Chain Architecture

You can expand an RIO main ring by adding a DIO daisy chain (not a ring). The distributed equipment can be part of either an (e)X80 or Quantum RIO drop. In this example, an (e)X80 RIO drop is used.

Connecting the Ethernet RIO ports of the primary and standby controllers in this design can cause the Hot Standby system to cease functioning.

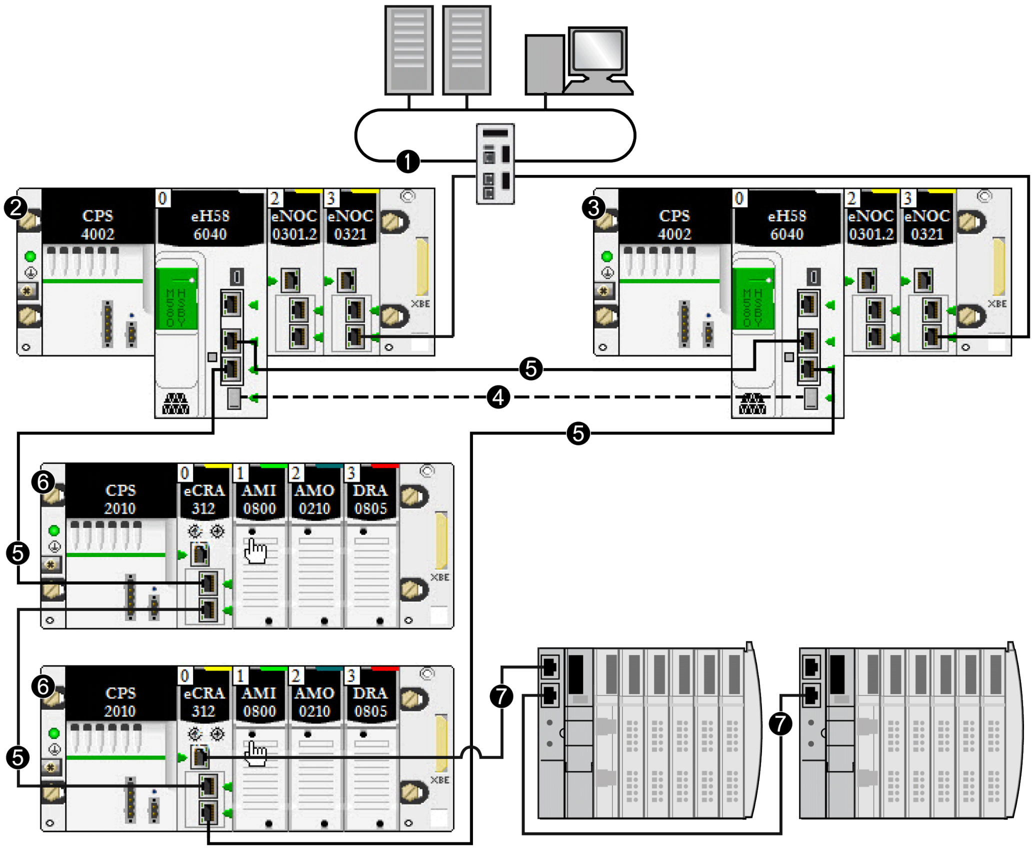
⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Do not connect the Ethernet RIO ports of the primary controller to the Ethernet RIO ports of the standby controller.

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

In this example, the distributed equipment is connected to the service port of a BMECRA31210 eX80 performance EIO adapter module:



1 control network connected to a BMENOC0321 module on the local rack to provide transparency between the device network and the control network

2 primary local rack with primary controller

3 standby local rack with standby controller

4 Hot Standby communication link

5 RIO main ring

6 (e)X80 RIO drop

7 DIO daisy chain (non-ring)

NOTE: In this design:

- Each BMENOC0301 / BMENOC0311 or BMENOC0302(H) module on the local rack is connected to the Ethernet backplane, by enabling its Ethernet backplane port.
- Up to 31 RIO drops can be included.
- Up to 64 distributed devices can be scanned by the Hot Standby controllers.
- Up to 128 distributed devices can be scanned by non-isolated BMENOC0301 / BMENOC0311 or BMENOC0302(H) modules.

  Of these 128 connections:

  - 16 are reserved for local slaves.
  - 112 are available for distributed equipment to be scanned.

  For more details, refer to either:

  - Modicon M580 BMENOC0301 / BMENOC0311 Ethernet Communications Module Installation and Configuration Guide (see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)
  - Modicon M580 BMENOC0302 High Performance Ethernet Communication Module Installation and Configuration Guide

RIO Main Ring with DIO Ring

You can design a Hot Standby system to include dual rings: an RIO main ring and a DIO ring. In this design, the controller scans the RIO main ring and a BMENOC0301 / BMENOC0311 or BMENOC0302(H) module scans the distributed equipment.

In the following example, the BMENOC0301 module scanning the distributed equipment is isolated. To isolate the communication module, disable its Ethernet backplane port. In this design, X Bus backplane communication remains enabled for the communication module.

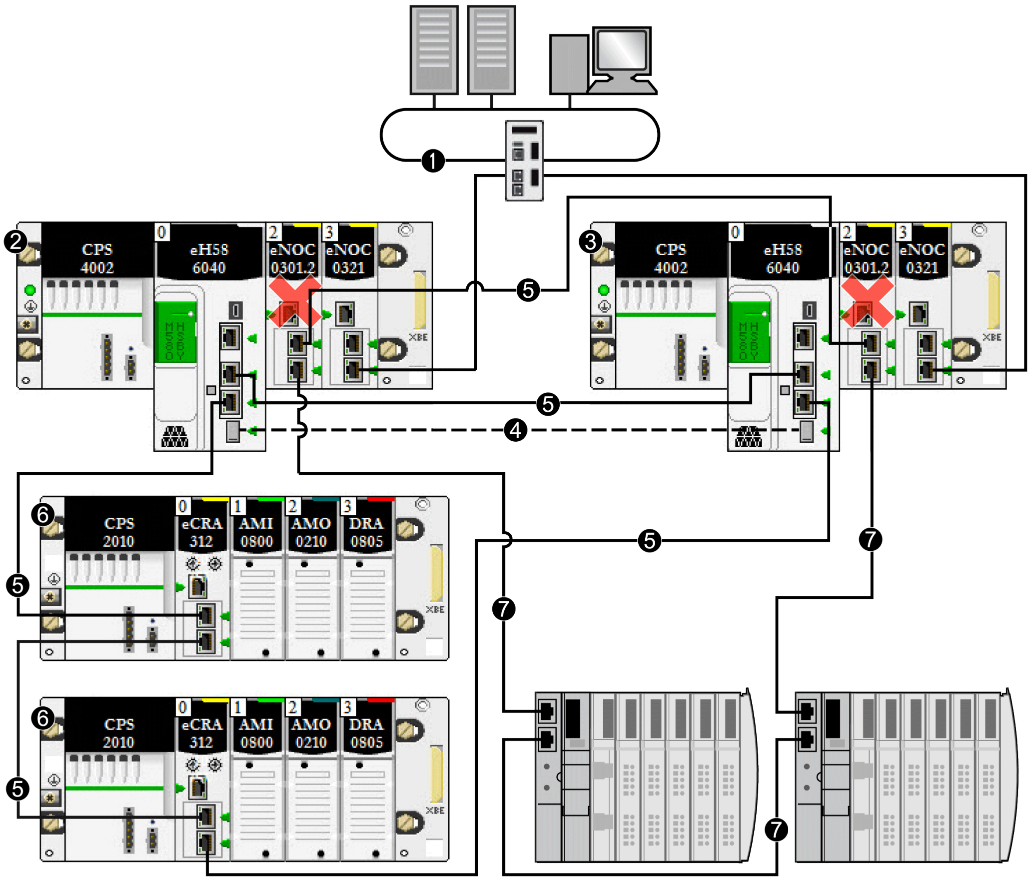
Connecting more than one module to both the backplane and an Ethernet network can cause a broadcast storm.

⚠ CAUTION
RISK OF BROADCAST STORM Do not connect more than one module in a local rack to both the Ethernet backplane and an Ethernet network. **Failure to follow these instructions can result in injury or equipment damage.**

Use only one module in each local rack to connect an Ethernet network to the Ethernet backplane. That module can be:

- the controller, when RIO modules are used
- one BMENOS0300 module
- one BMENOC0301 / BMENOC0311 or BMENOC0302(H) module

The (red) ***X*** indicates that the BMENOC0301 module is isolated from the Ethernet backplane:



1 control network connected to the BMENOC0321 module on the local rack providing transparency between the device network and the control network

2 primary local rack with primary controller

3 standby local rack with standby controller

4 Hot Standby communication link

5 RIO main ring

6 (e)X80 RIO drop

7 DIO ring

RIO Main Ring with DIO Sub-Ring

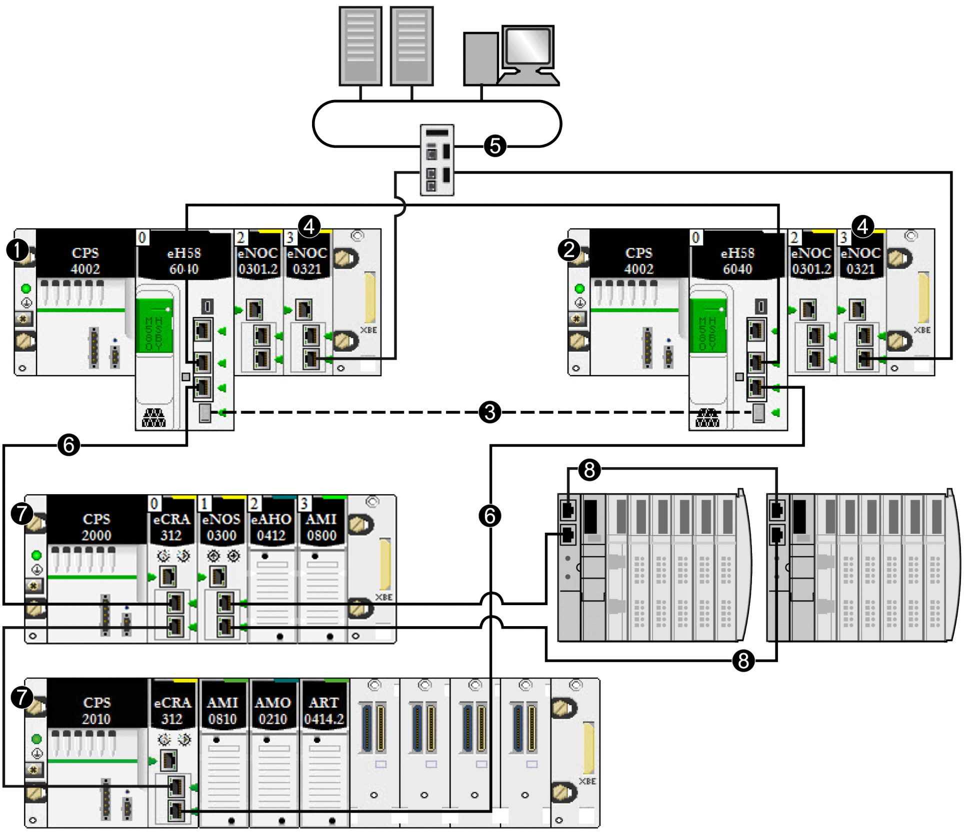
You can design a Hot Standby system to include both an RIO main ring and a DIO sub-ring. In this design, the controller scans the RIO main ring, and the BMENOC0301 module scans the distributed equipment. The DIO sub-ring is connected to the RIO main ring via a BMENOS0300 module in an (e)X80 RIO drop.

NOTE: In this design, the rotary selectors on the BMENOS0300 module (that is connected to the DIO sub-ring (7)) are set to configure the top port as a service port and the two lower ports as DIO sub-ring ports.

Connecting these ports can result in the occurrence of a broadcast storm, which can prevent the Hot Standby network from transporting intended network communications.

⚠ CAUTION
RISK OF BROADCAST STORM Do not connect the paired Ethernet ports of the primary BMENOC0301 / BMENOC0311 or BMENOC0302(H) module to the Ethernet ports of the standby BMENOC0301 / BMENOC0311 or BMENOC0302(H) module. **Failure to follow these instructions can result in injury or equipment damage.**

Unlike the prior example, the BMENOC0301 module scanning the distributed equipment is not isolated. Confirm that its Ethernet backplane port is enabled:



1 primary local rack with primary controller

2 standby local rack with standby controller

3 Hot Standby communication link

4 BMENOC0321 module on the local rack, providing transparency between the device network and the control network

5 control network

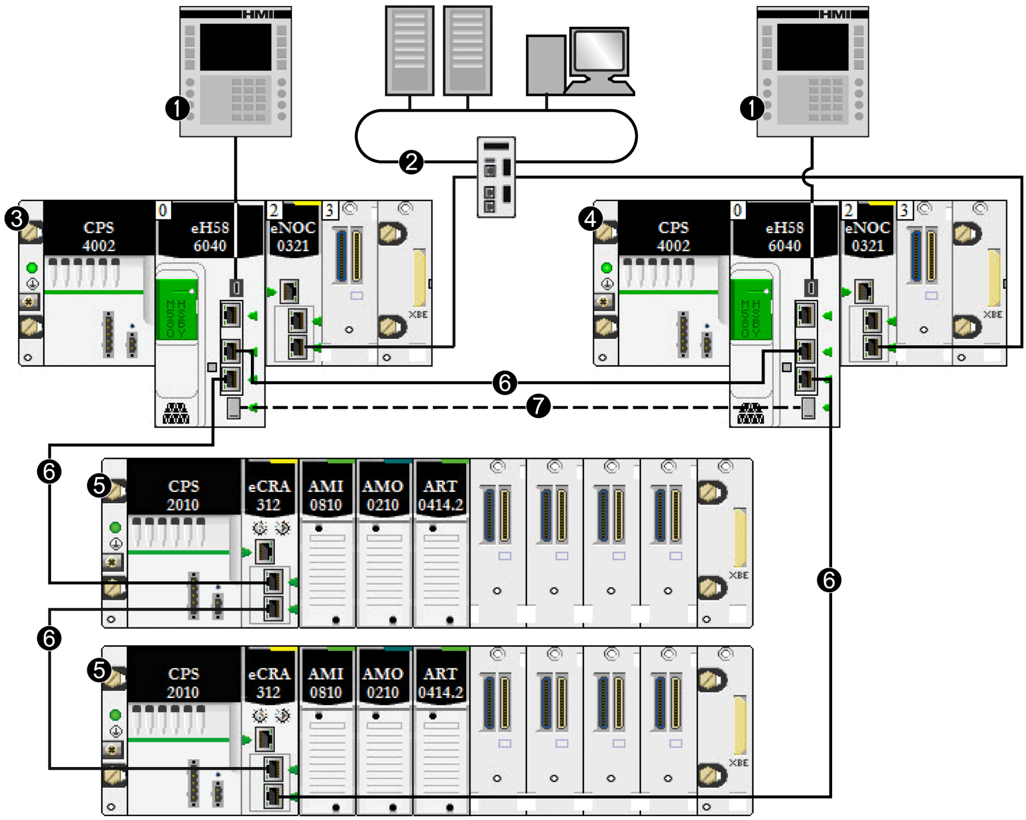
6 RIO main ring

7 (e)X80 RIO drop

8 DIO sub-ring

Connecting an HMI to a Hot Standby Topology

You can connect an HMI directly to a Hot Standby controller. To access the local controllers (for diagnostics and control), connect an HMI to both the primary and standby controllers. In the following example, the connection is made via the USB port of each controller. This design allows access for diagnostics and control to the service port of either or both controllers:



1 HMI

2 control network connected to the BMENOC0321 module on the local rack providing transparency between the device network and the control network

3 primary local rack with primary controller

4 standby local rack with standby controller

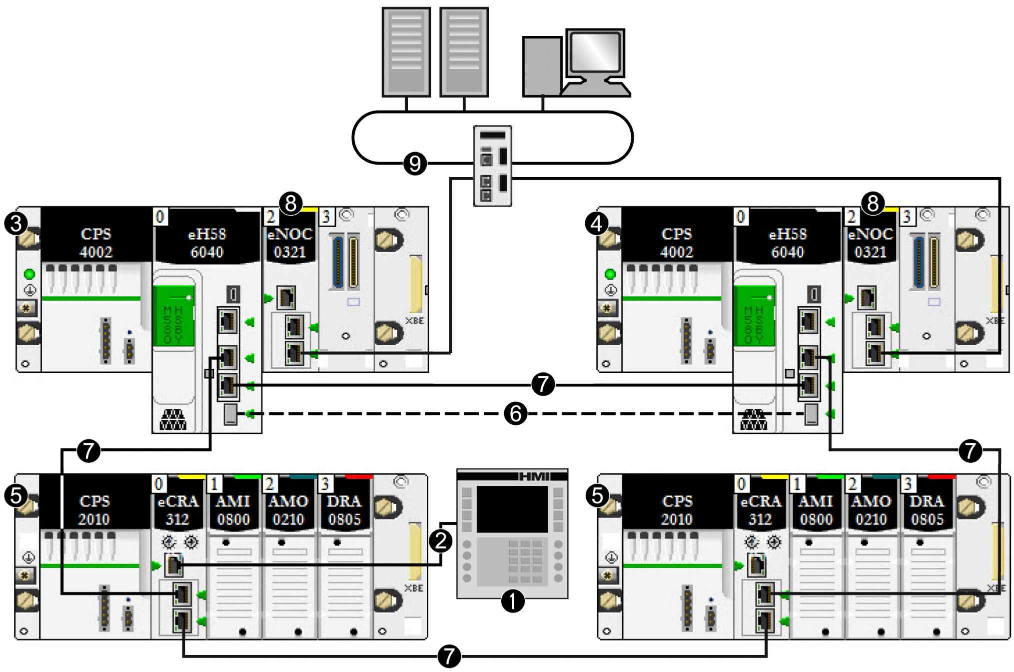
5 (e)X80 RIO drop

6 RIO main ring

7 Hot Standby communication link

You can connect an HMI to a Hot Standby system in several ways. Two examples are presented below:

Alternatively, you can indirectly connect an HMI to a Hot Standby controller. In the following example, the connection is made via the service port of a BMECRA312•0 module on the RIO main ring. In this design, only one HMI is required. The service port and USB port on both controllers can be accessed for diagnostics and control:



1 HMI

2 Ethernet connection to the service port of the BMECRA312•0 module

3 primary local rack with primary controller

4 standby local rack with standby controller

5 (e)X80 RIO drop

6 Hot Standby communication link

7 RIO main ring

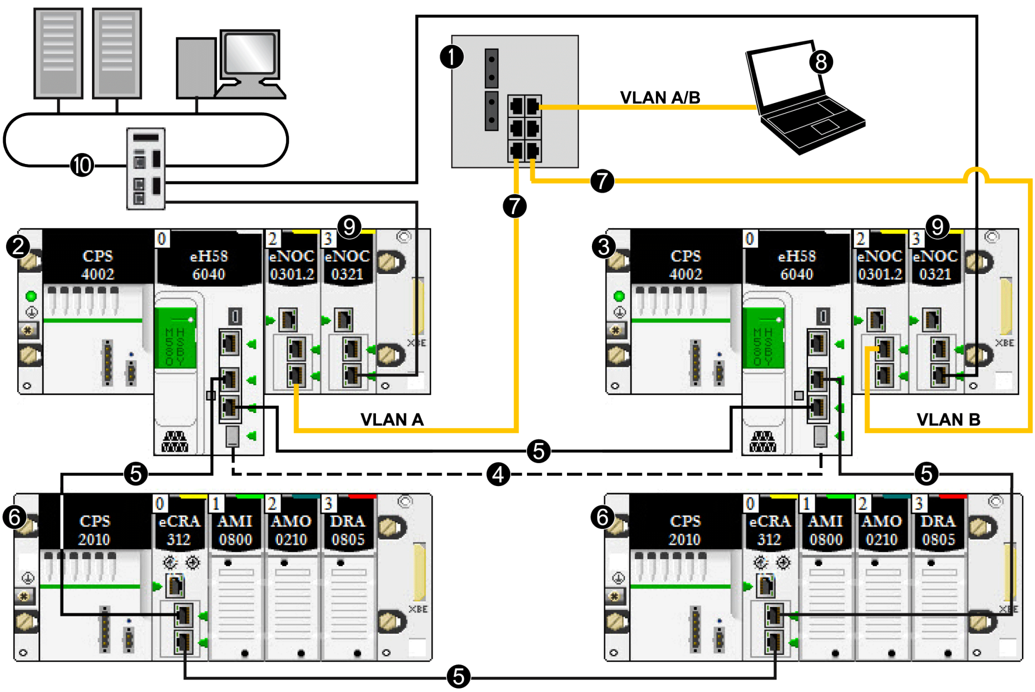
8 BMENOC0321 module on the local rack providing transparency between the device network and the control network

9 control network

Solution for Connecting an Engineering Tool to a Hot Standby Topology

The following topology presents an example of how to connect a PC running an engineering tool (for example, Control Expert) to the Hot Standby system. In this example:

- The PC is equipped with a network interface card (NIC) that supports communication on multiple VLANs, in this case VLAN A and VLAN B.
- The BMENOC0301 module in the primary controller belongs to VLAN A.
- The BMENOC0301 module in the standby controller belongs to VLAN B.



1 layer 2 switch linking control network to the RIO main ring

2 primary local rack with primary controller

3 standby local rack with standby controller

4 Hot Standby communication link

5 RIO main ring

6 (e)X80 RIO drop

7 redundant link to the Hot Standby system

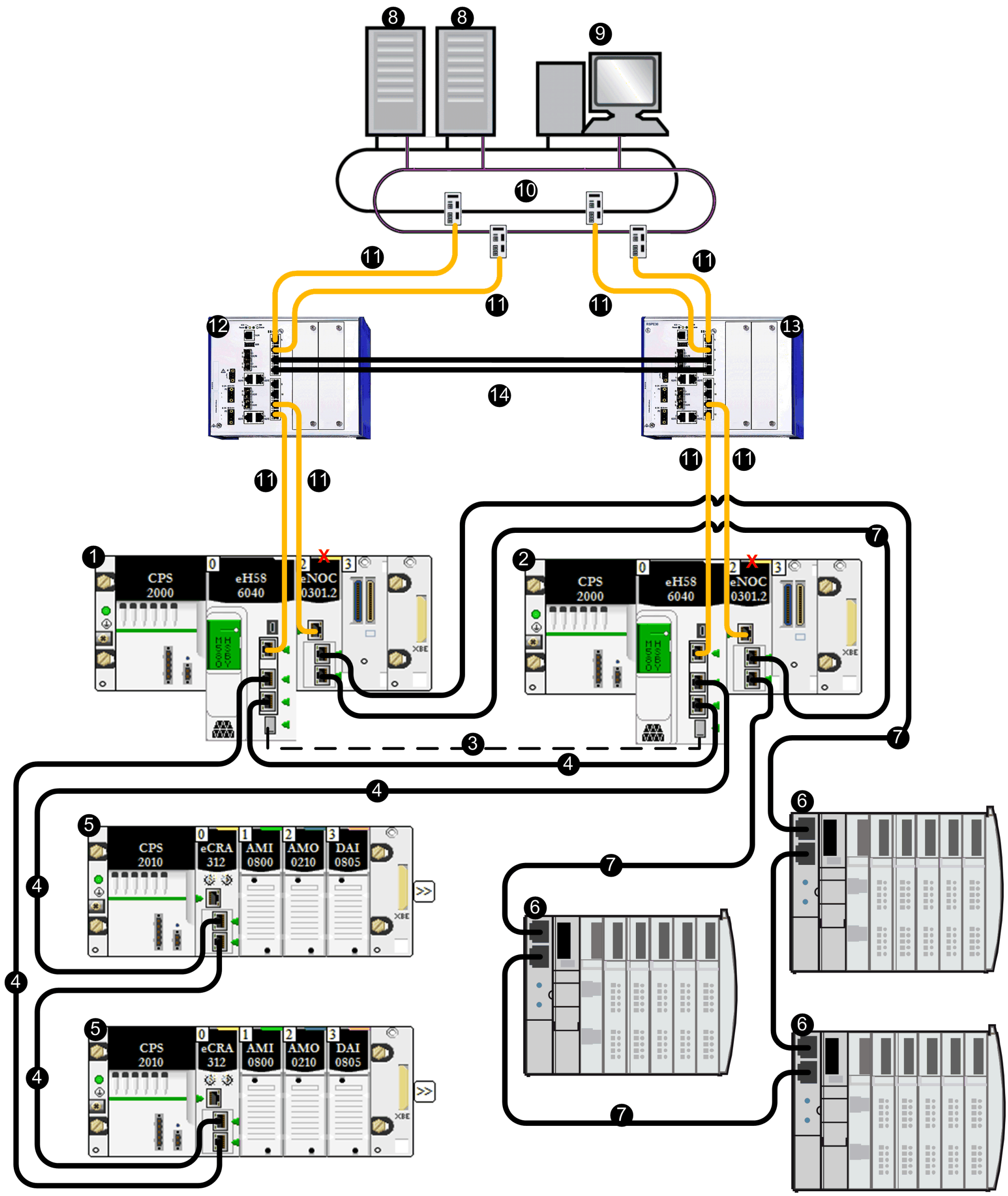
8 engineering tool resident on a PC

9 BMENOC0321 module on the local rack providing transparency between the device network and the control network

10 control network

Connecting SCADA to a Hot Standby Topology

The following network topology shows you how to connect a SCADA server located on a redundant control network to the Hot Standby process control system:



1 local rack with controller A

2 local rack with controller B

3 Hot Standby communication link

4 RIO main ring

5 (e)X80 RIO drop

6 distributed equipment

7 DIO ring

8 SCADA server

9 engineering workstations

10 control network

11 layer 3 redundant link

12 layer 3 switch configured with virtual router redundancy protocol (VRRP), configured as master

13 layer 3 switch configured with VRRP, configured as slave

14 layer 2 ring between L3 switches

X Indicates the BMENOC0301 module is isolated from the Ethernet backplane

NOTE: In the preceding example, the L3 switches are Hirschmann dual-ring switches (DRSs), part number: RSPE30-24044 O7T99-SCCZ999HHSE3S04.0.

M580 Hot Standby Topologies Using BMENOC0321 Module for Control Network Connection

Introduction

The BMENOC0321 control network module is installed on a local Ethernet backplane in an M580 system. With the Ethernet backplane enabled, the BMENOC0321 provides access to the device network (through the external ports of the controller).

Install a maximum of two BMENOC0321 modules in an M580 system to provide Ethernet transparency between a control network (for example, a SCADA system) and a device network. You can enable the IP forwarding service (see Modicon M580, BMENOC0321 Control Network Module, Installation and Configuration Guide) on only one BMENOC0321 module per local rack.

In a Hot Standby system, the BMENOC0321 module uses the same IP+1 address as the BMENOC0301 / BMENOC0311 or BMENOC0302(H) module on the local rack. Confirm that you configure the IP address used in the BMENOC0301 / BMENOC0311 or BMENOC0302 (H) module differently than the IP address of the BMENOC0321 module (for the control network and the fieldbus network when IP forwarding is enabled). Use an Ethernet manager tool to verify the system is working.

NOTE: Do not mount the BMENOC0321 module on a BMX backplane (X Bus only). The module can power up only on a BME backplane (Ethernet). Refer to the rack descriptions in the *Modicon X80 Racks and Power Supplies, Hardware, Reference Manual* (see Modicon X80, Racks and Power Supplies, Hardware Reference Manual).

Connecting together the service ports of the primary and standby controllers can cause unintended system operation.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Do not connect together the service ports of the Hot Standby controllers.

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

Connecting more than one module to both the backplane and an Ethernet network can cause a broadcast storm.

⚠ CAUTION

RISK OF BROADCAST STORM

Do not connect more than one module in a local rack to both the Ethernet backplane and an Ethernet network.

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

Use only one module in each local rack to connect an Ethernet network to the Ethernet backplane. That module can be:

- the controller, when RIO modules are used
- one BMENOS0300 module
- one BMENOC0301 / BMENOC0311 or BMENOC0302(H) module

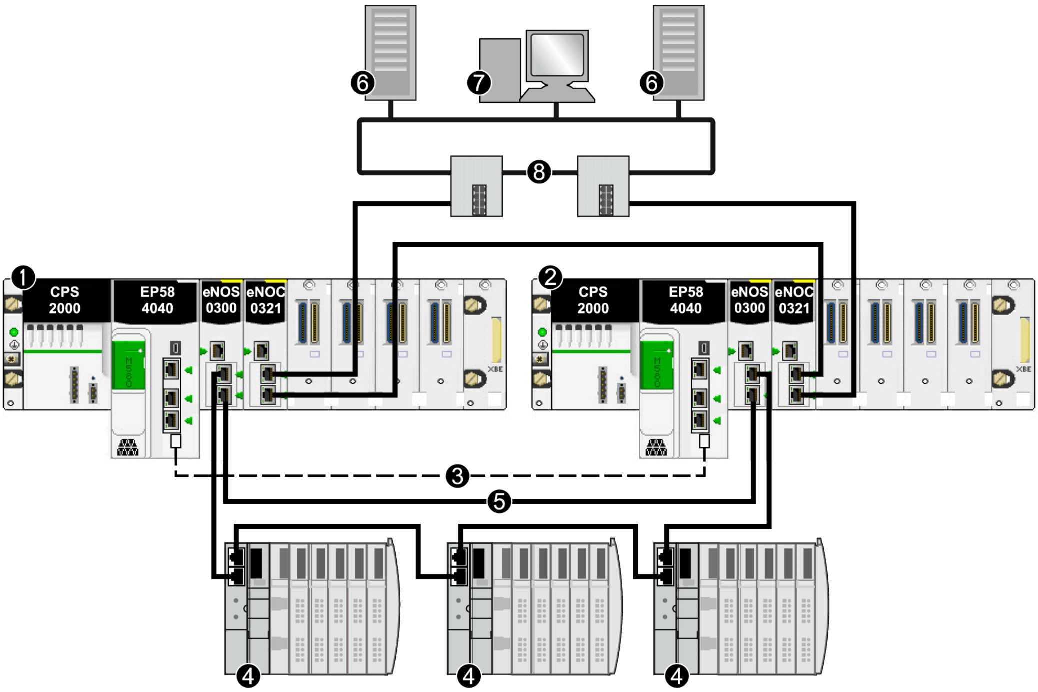
Device Network with DIO Ring and Control Network Architecture

This Hot Standby system, which connects a single DIO ring (for example, SCADA) to your system, consists of the following:

- two local Hot Standby racks, each containing a Hot Standby controller with Ethernet I/O scanner service
- BMENOS0300 module on the local rack connected to a DIO ring
- BMENOC0321 control network module on the local rack providing transparency between the DIO ring and the control network

In this topology, no (e)X80 RIO drops are used. No sub-rings, and no DRSs are used.

The following example presents a Hot Standby system with a control network communicating with a DIO ring:



1 primary local rack with primary controller

2 standby local rack with standby controller

3 Hot Standby communication link

4 distributed equipment

5 DIO ring

6 SCADA server

7 engineering workstation

8 control room Ethernet network (with Gb RSTP ring)

NOTE: In this design:

- Connect the two BMENOS0300 modules via a direct link.
- If there is an interruption in the direct connection between the two BMENOS0300 modules, the Hot Standby system may report communication issues in case the Ethernet traffic is heavily loaded (bit HSBY_SUPPLEMENTARY_LINK_ERROR in ECPU_HSBY_STS). This information has no effect on system behavior and disappears as soon as the direct connection is restored. However, you need to repair the direct connection between the two BMENOS0300 modules so that the system can remain operational in case a second interruption occurs.
- No link can be made connecting the dual Ethernet ports on the primary controller to the dual Ethernet ports on the standby controller.
- Only one DIO ring is supported.
- Set the rotary selectors on both of the BMENOS0300 modules to configure the top port as a service port, and the two lower ports as DIO ring ports.

Device Network with RIO Main Ring, DIO Sub-Ring, and Control Network Architecture

A Hot Standby system with an RIO main ring and a DIO sub-ring can be constructed using a BMENOS0300 module on an RIO drop on the main ring. In this design, the controller scans the RIO drops and the distributed equipment.

Connecting more than one module to both the backplane and an Ethernet network can cause a broadcast storm.

⚠ CAUTION

RISK OF BROADCAST STORM

Do not connect more than one module in a local rack to both the Ethernet backplane and an Ethernet network.

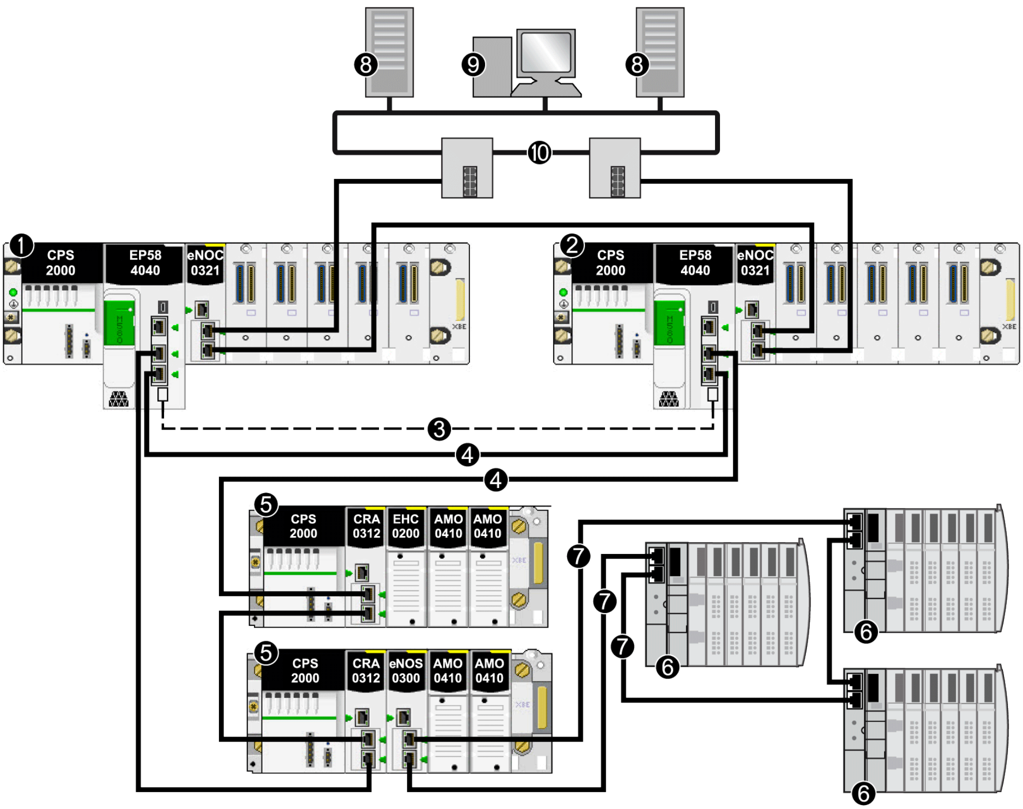
Failure to follow these instructions can result in injury or equipment damage.

Use only one module in each local rack to connect an Ethernet network to the Ethernet backplane. That module can be:

- the controller, when RIO modules are used
- one BMENOS0300 module
- one BMENOC0301 / BMENOC0311 or BMENOC0302(H) module

NOTE: Because the Ethernet backplane port of each BMENOS0300 module is enabled, do not connect a dual Ethernet port on the primary controller to a dual Ethernet port on the standby controller. Otherwise, it can cause the Hot Standby system to cease functioning.

The following example presents an RIO main ring, a DIO sub-ring (connected to the main ring via a BMENOS0300 module on an RIO drop), and a control network scanned by the controller.



1 primary local rack with primary controller

2 standby local rack with standby controller

3 Hot Standby communication link

4 RIO main ring

5 (e)X80 RIO drop

6 distributed equipment

7 DIO sub-ring

8 SCADA server

9 engineering workstation

10 control room network (with Gb RSTP ring)

NOTE: In this design:

- No link can be made connecting the dual Ethernet ports on the primary controller to the dual Ethernet ports on the standby controller.
- Set the rotary selectors on the BMENOS0300 module to configure the top port as a service port, and the two lower ports as DIO sub-ring ports.

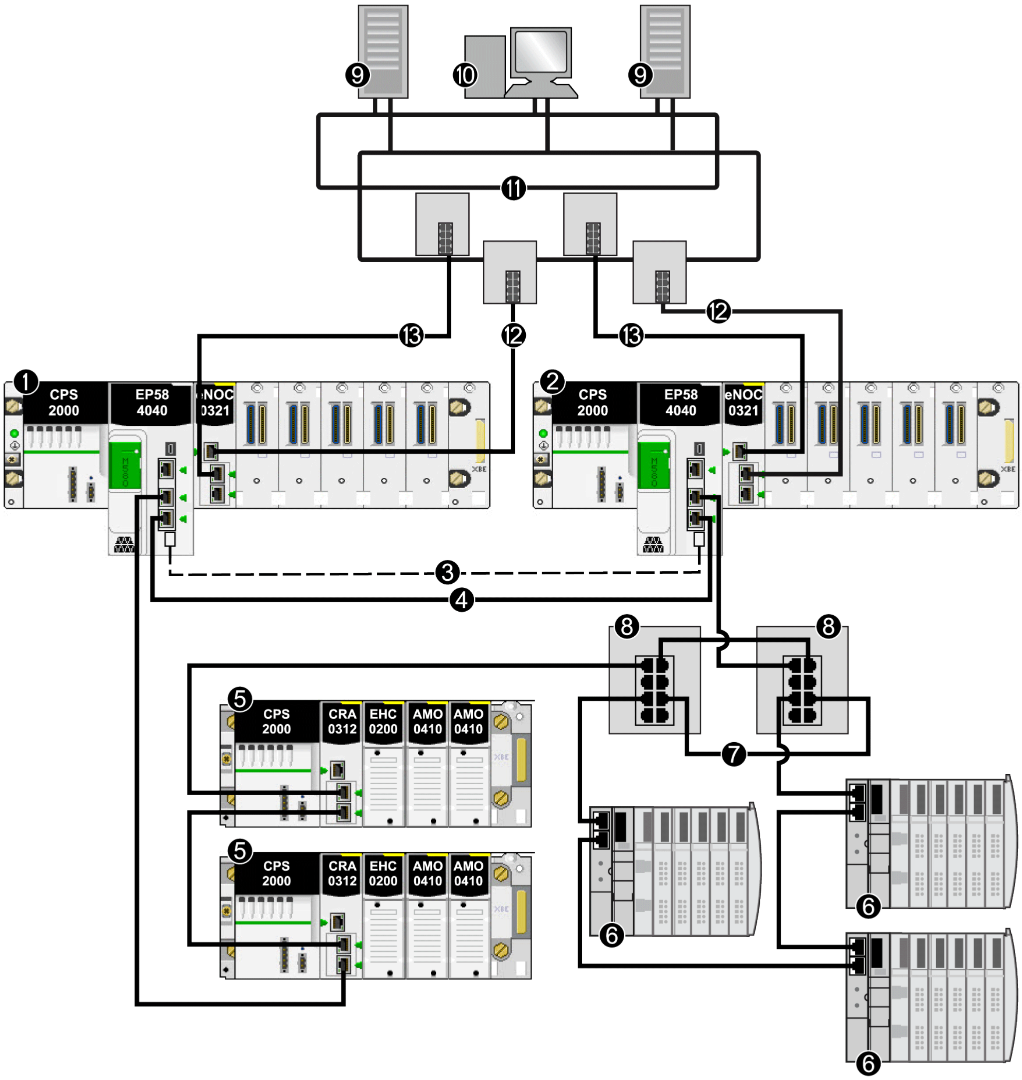
Device Network with RIO Main Ring, DIO Sub-Ring, DRSs, and Control Network Architecture

A Hot Standby system with an RIO main ring and a DIO sub-ring can be constructed using one or more dual-ring switches (DRSs) to connect the distributed equipment to the main ring. DRSs can also be used to connect the local rack to the control network.

In this design, the DRSs are configured as master and slave (C9 and C10 (see *Modicon M580 Complex Topologies System Guide*). If one DRS communication is interrupted, the DIO ring still communicates with the control network through the other DRS.

Distributed equipment in a DIO sub-ring cannot be connected directly to the main ring. Only (e)X80 and Quantum RIO drops are allowed on the main ring.

The following example presents an RIO main ring, a DIO sub-ring (connected to the main ring via master/slave DRSs), and a control network scanned by the controller:



1 primary local rack with primary controller

2 standby local rack with standby controller

3 Hot Standby communication link

4 RIO main ring

5 (e)X80 RIO drop

6 distributed equipment

7 DIO sub-ring

8 dual-ring switch (DRS) with C9/C10 downloaded pre-configuration file

9 SCADA server

10 engineering workstation

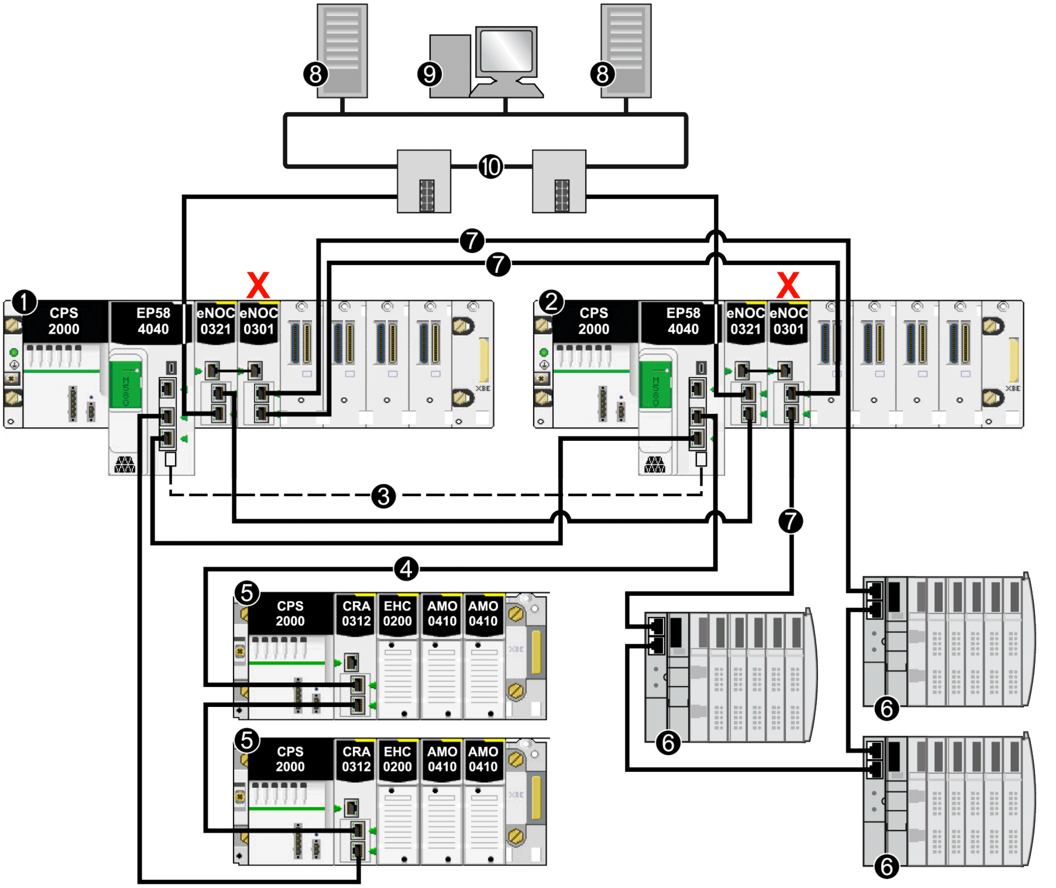
11 control room network (Gb RSTP ring)

No link can be made connecting the dual Ethernet ports on the primary controller to the dual Ethernet ports on the standby controller.

Device Network with RIO Main Ring, Isolated DIO Ring, and Control Network Architecture

You can expand an RIO main ring by adding a DIO daisy chain (not a ring). The isolated distributed equipment in the DIO chain is scanned by a BMENOC0301.2 module (its Ethernet backplane connection disabled) on the local rack. A BMENOC0321 module is connected externally with an Ethernet cable to the BMENOC0301.2 module to provide transparency between the isolated DIO network and the control network.

The following example presents an RIO main ring, an isolated DIO ring (that communicates with the control network, not the device network), and a control network scanned by the controller:



1 primary local rack with primary controller

2 standby local rack with standby controller

3 Hot Standby communication link

4 RIO main ring

5 (e)X80 RIO drop

6 distributed equipment

7 isolated DIO ring

8 SCADA server

9 engineering workstation

10 control room network (Gb RSTP ring)

X Ethernet backplane connection disabled on the BMENOC0301 module to support the isolated DIO ring (7)

NOTE: Because the Ethernet backplane port of the BMENOC0301 module on the local rack is disabled, confirm that you connect externally the service ports of the BMENOC0301 module and the BMENOC0321 module.

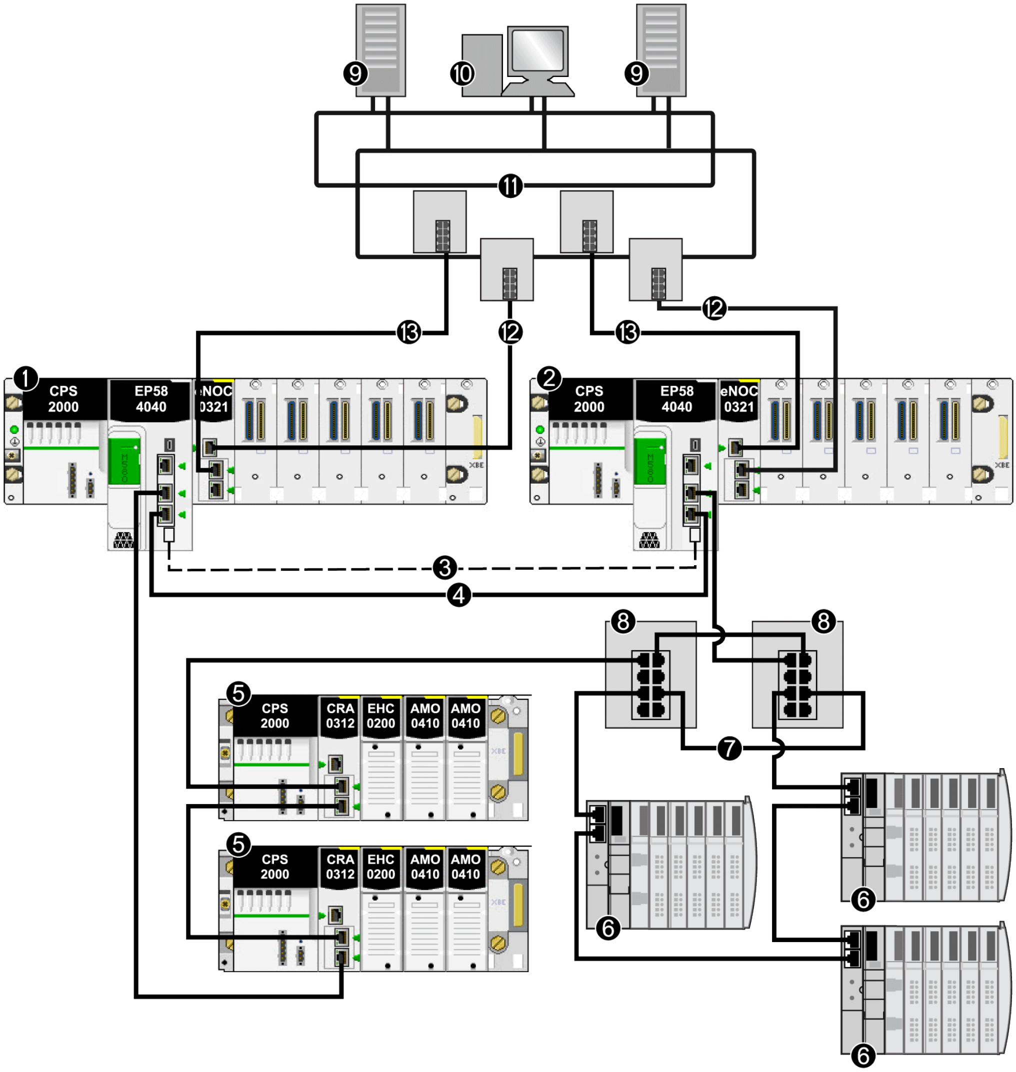
Device Network with RIO Main Ring, DIO Sub-Ring, DRSs, and Dual Control Network Architecture

This topology is similar to previous topology, page 96, except there is a dual connection in this architecture from the control network to the BMENOC0321 module on the local rack. Each physical connection from the control network to the BMENOC0321 module belongs to a unique subnet.

Follow these steps to connect the BMENOC0321 modules on the local rack to the two different control room subnets:

Step	In the Primary Local Rack...	In the Standby Local Rack...
1	Connect the Ethernet control port of the BMENOC0321 module to subnet A from the control room.	Connect the Ethernet service port of the BMENOC0321 module to subnet A from the control room.
2	Connect the Ethernet service port of the BMENOC0321 module to subnet B from the control room.	Connect the Ethernet control port of the BMENOC0321 module to subnet B from the control room.

The following example presents an RIO main ring, a DIO sub-ring (connected to the main ring via master/slave DRSs), and a control network (with 2 subnets) scanned by the controller.



1 primary local rack with primary controller

2 standby local rack with standby controller

3 Hot Standby communication link

4 RIO main ring

5 (e)X80 RIO drop

6 distributed equipment

7 DIO sub-ring

8 dual-ring switch (DRS) with C9/C10 downloaded pre-configuration file

9 SCADA server

10 engineering workstation

11 control room network

12 subnet A from control room

13 subnet B from control room

Managing Flat Ethernet Networks with M580 Hot Standby

Overview

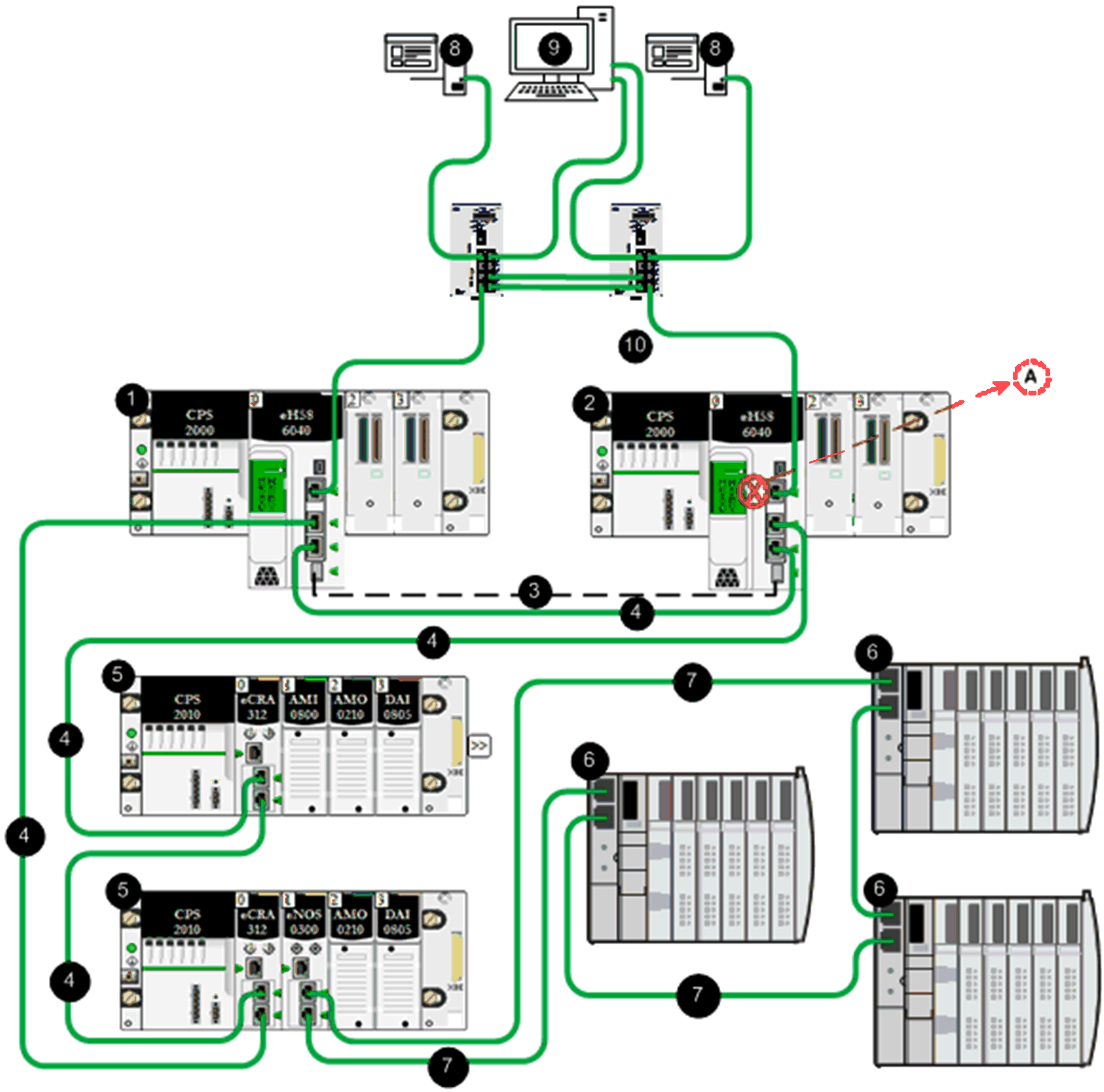
In an M580 Hot Standby configuration, some topologies may unintentionally create a loop that interferes with network communication. These topologies are essentially related to the management of flat networks, i.e., topologies in which the control network, the remote I/O network, and/or the device network belong to the same subnet.

To avoid creating an unintentional loop caused by the connection to the service port, use the automatic blocking of service port feature on the controller with firmware version 2.7 (or later) (see Modicon M580, Hardware, Reference Manual) or a BMENOC0301.4 (or later) module.

No loop condition can exist on the Ethernet backplane connection by connecting one or more BMENOC0301.4 (or later) modules to the Ethernet backplane of an M580 Hot Standby configuration. A BMENOC0301.4 (or later) module automatically blocks its backplane port on the standby local rack.

Example 1

The following topology is an M580 Hot Standby system with an Ethernet RIO main ring, distributed equipment, a BMENOS0300 module on an RIO drop, and a single SCADA connection.



A In this topology, where SCADA is connected to the Hot Standby system via the primary controller service port and the standby controller service port, confirm that you select the **Automatic**

blocking of service port on Standby controller check box in the **ServicePort** configuration tab to avoid network communication interruptions (formed by cables 4 and 10).

green cable RIO network, device network, and control network on the same subnet

1 primary local rack with primary controller

2 standby local rack with standby controller

3 Hot Standby communication link

4 Ethernet RIO main ring

5 (e)X80 RIO drop

6 distributed equipment

7 Ethernet DIO ring

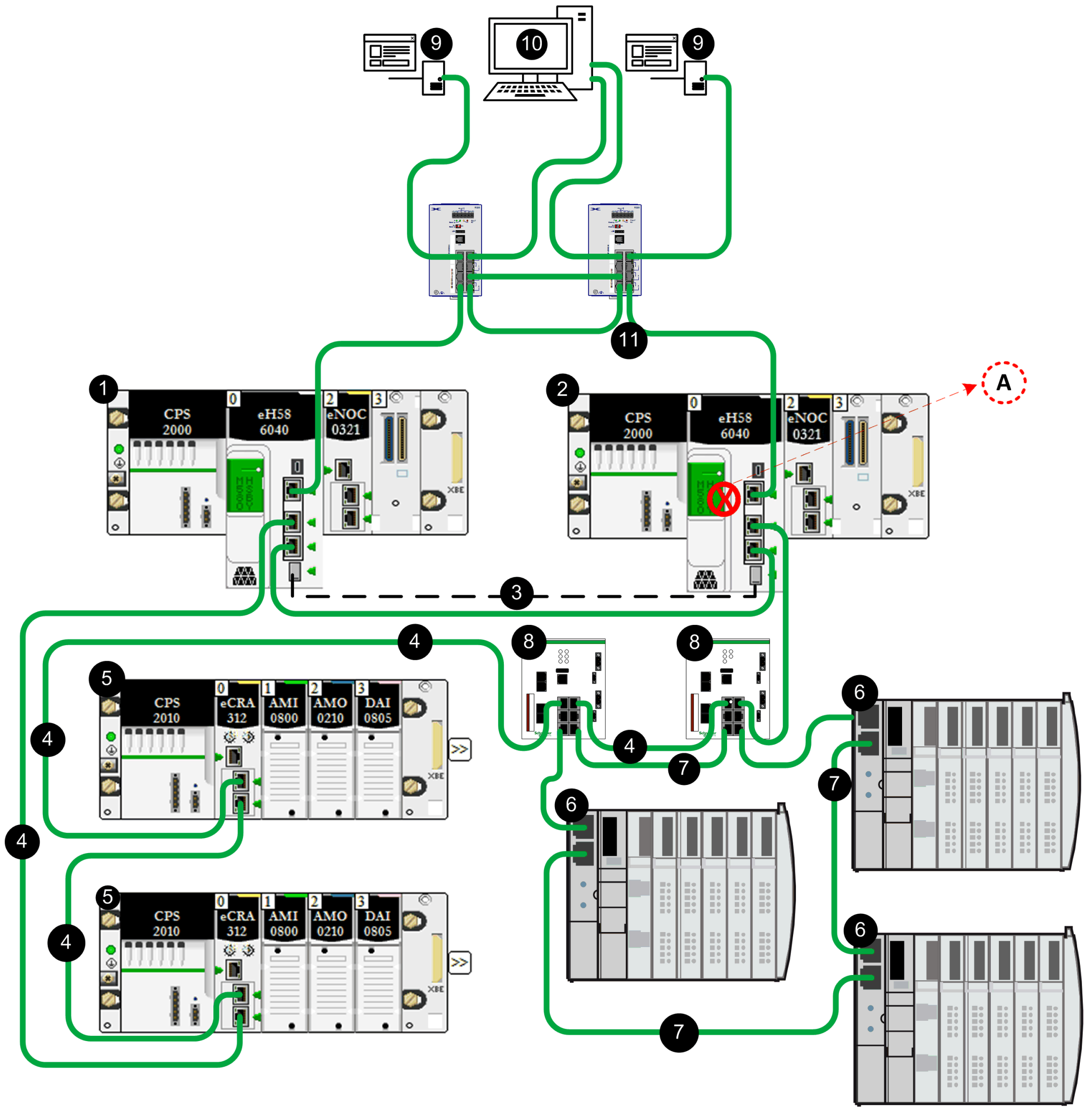
8 SCADA server

9 engineering workstation with dual Ethernet connections

10 control room network

Example 2

This topology is an M580 Hot Standby system with an Ethernet RIO main ring, distributed equipment connected to the RIO main ring via dual-ring switches (DRSs), and a single SCADA connection.



A In this topology, where SCADA is connected to the Hot Standby system via the primary and standby controller service ports, confirm that you select the **Automatic blocking of service port**

on Standby controller check box in the **ServicePort** configuration tab to avoid network communication interruptions (formed by cables 4 and 11).

green cable RIO network, device network, and control network on the same subnet

1 primary local rack with primary controller

2 standby local rack with standby controller

3 Hot Standby communication link

4 Ethernet RIO main ring

5 (e)X80 RIO drop

6 distributed equipment

7 Ethernet DIO ring

8 DRS switch

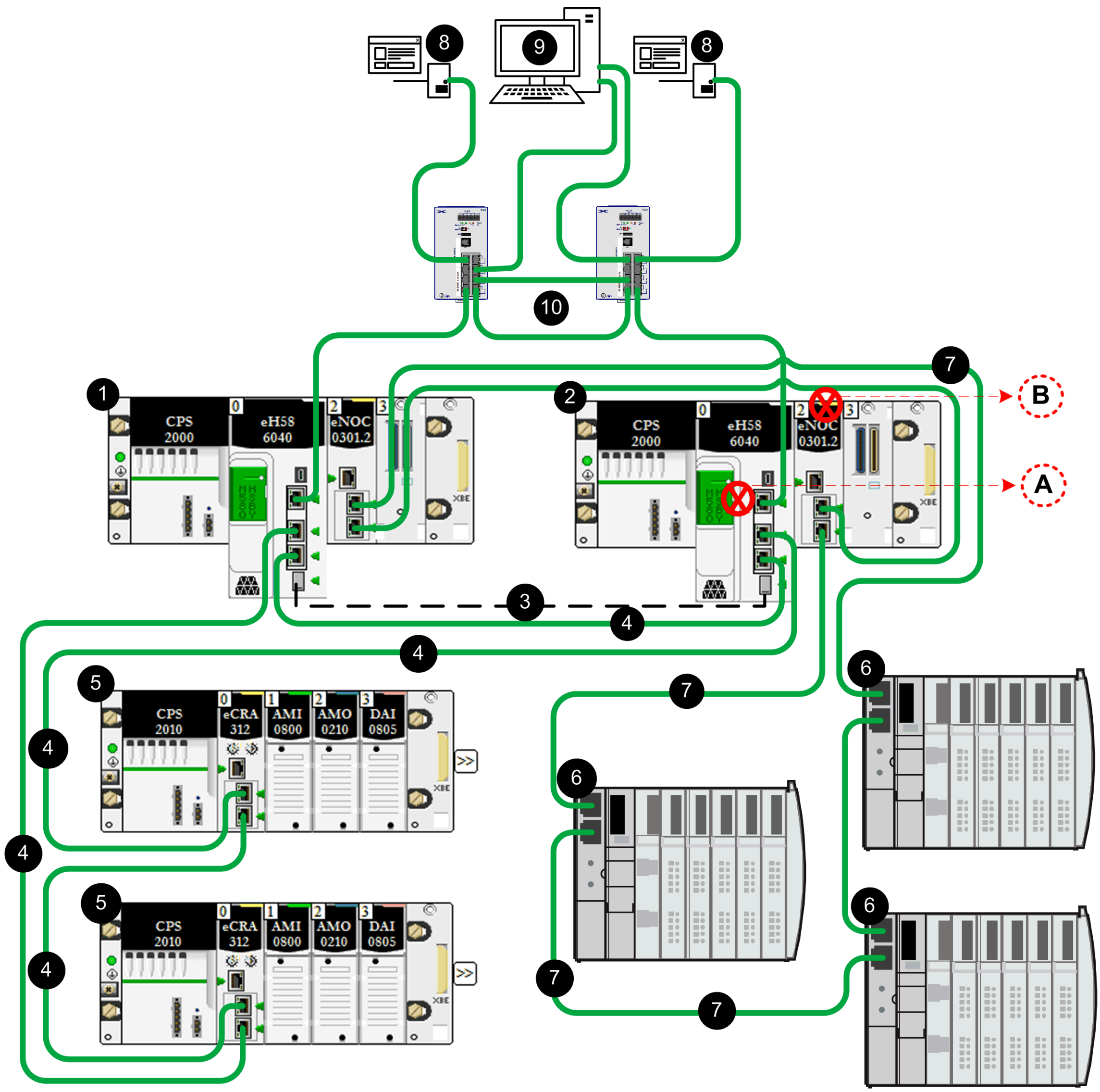
9 SCADA server

10 engineering workstation with dual Ethernet

11 control room network

Example 3

This topology is an M580 Hot Standby system with an Ethernet RIO main ring, a DIO network scanned by the BMENOC0301 module that communicates with the control network, and a SCADA connection via the service ports of the primary and standby controllers.



A In this topology, where SCADA is connected to the Hot Standby system via service ports of the controllers, confirm that you select the **Automatic blocking of service port on Standby**

controller check box in the **ServicePort** configuration tab to avoid network communication interruptions (formed by cables 4 and 10).

B In this topology, where a DIO ring/cloud network communicates with the control network via the BMENOC0301 modules, the Ethernet backplane port of the standby NOC is automatically disabled to avoid network communication interruptions (formed by cables 4 and 7).

green cable RIO network, device network, and control network on the same subnet

1 primary local rack with primary controller and BMENOC0301 module

2 standby local rack with standby controller and BMENOC0301 module

3 Hot Standby communication link

4 Ethernet RIO main ring

5 (e)X80 RIO drop

6 distributed equipment

7 Ethernet DIO ring

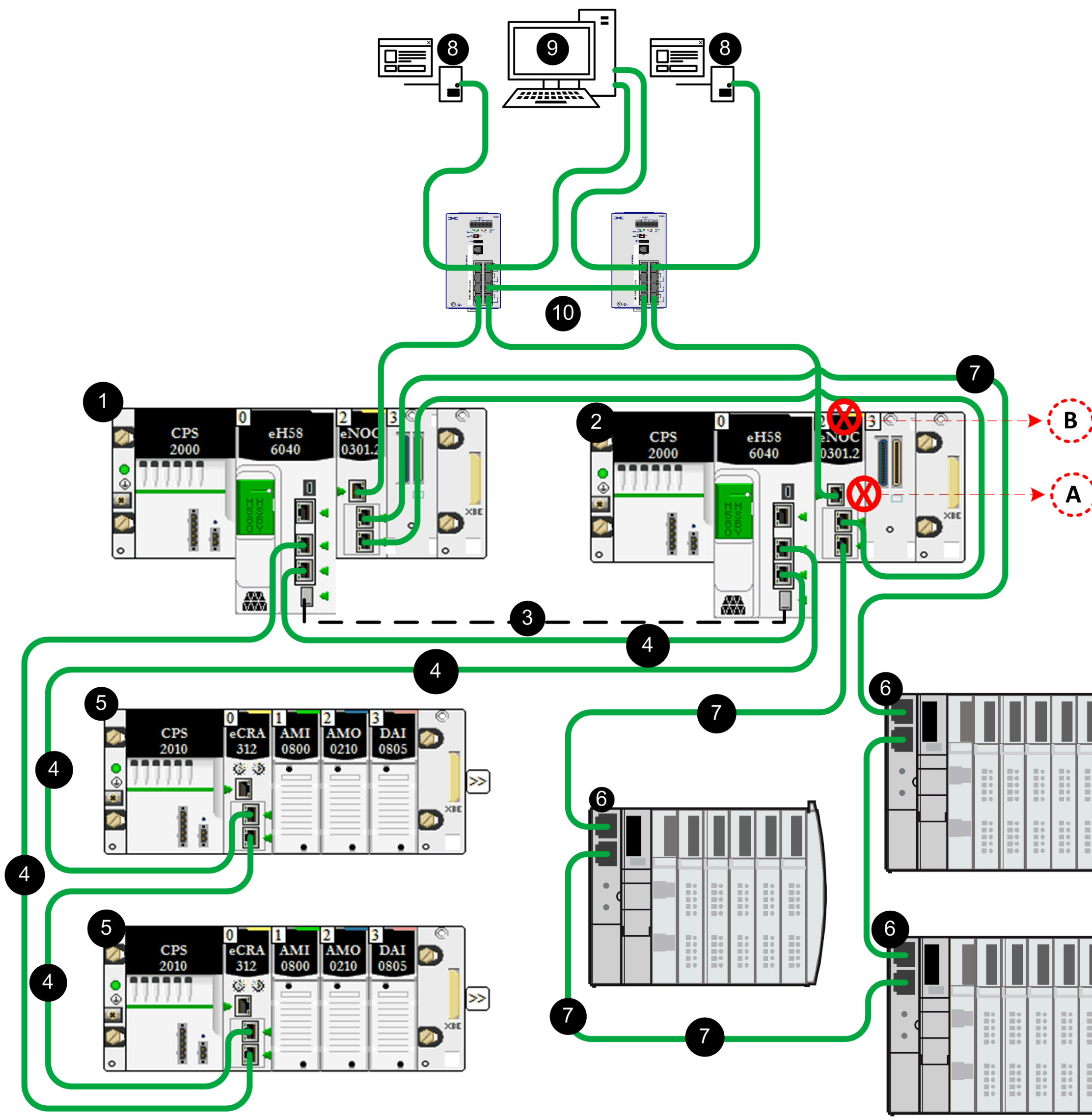
8 SCADA server

9 engineering workstation with dual Ethernet

10 control room network

Example 4

This topology is an M580 Hot Standby system with an Ethernet RIO main ring, a DIO network scanned by the BMENOC0301 module on the local rack that communicates with the control network, and a single SCADA connection via the service ports of the primary and standby BMENOC0301 modules.



A In this topology, where SCADA is connected to the Hot Standby system via service ports of the BMENOC0301 modules, confirm that you select the **Automatic blocking of service port on Standby NOC** check box in the **ServicePort** configuration tab to avoid network communication interruptions (formed by cables 7 and 10).

B In this topology, where a DIO ring/cloud network communicates with the control network via the BMENOC0301 modules, the Ethernet backplane port of the standby NOC is automatically disabled to avoid network communication interruptions (formed by cables 4 and 7).

green cable RIO network, device network, and control network on the same subnet

1 primary local rack with primary controller and BMENOC0301 module

2 standby local rack with standby controller and BMENOC0301 module

3 Hot Standby communication link

4 Ethernet RIO main ring

5 (e)X80 RIO drop

6 distributed equipment

7 Ethernet DIO ring

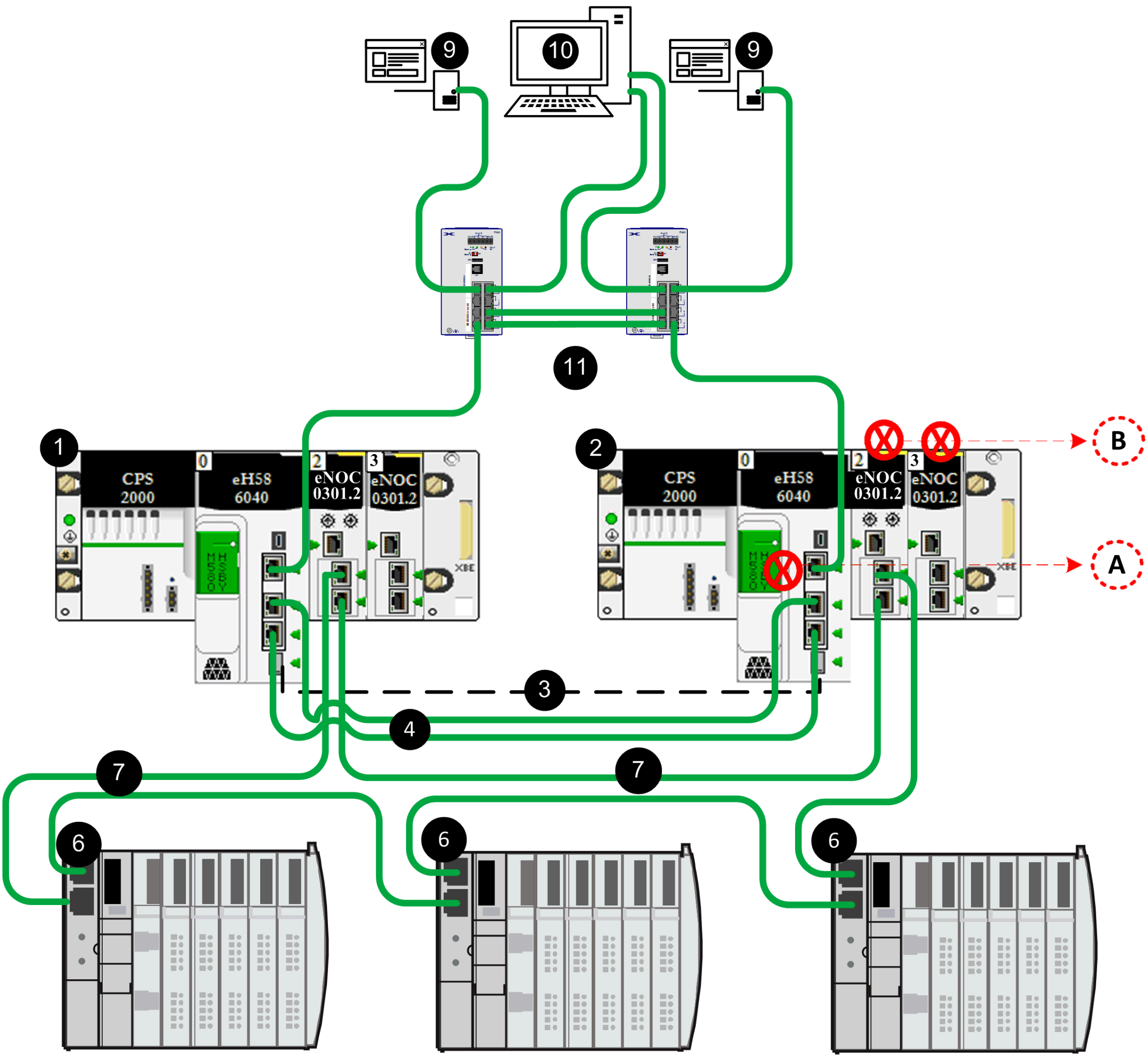
8 SCADA server

9 engineering workstation with dual Ethernet

10 control room network

Example 5

This topology is an M580 Hot Standby system with an Ethernet RIO main ring, a DIO network scanned by two BMENOC0301 modules on the local rack that communicate with the control network (one of them through the Ethernet backplane), and a single SCADA connection via the service ports of the primary and standby controllers.



A In this topology, where SCADA is connected to the Hot Standby system via service ports of the primary and standby controllers, confirm that you select the **Automatic blocking of service port**

on Standby controller check box in the **ServicePort** configuration tab avoid network communication interruptions (formed by cables 4 and 11).

B In this topology, where a DIO ring network communicates with the control network via the controller, the Ethernet backplane port of the standby NOCs are automatically disabled to avoid network communication interruptions (formed by cables 4 and 7).

green cable RIO network, device network, and control network on the same subnet

1 primary local rack with primary controller and BMENOC0301 module

2 standby local rack with standby controller and BMENOC0301 module

3 Hot Standby communication link

4 Ethernet RIO main ring

6 distributed equipment

7 Ethernet DIO ring

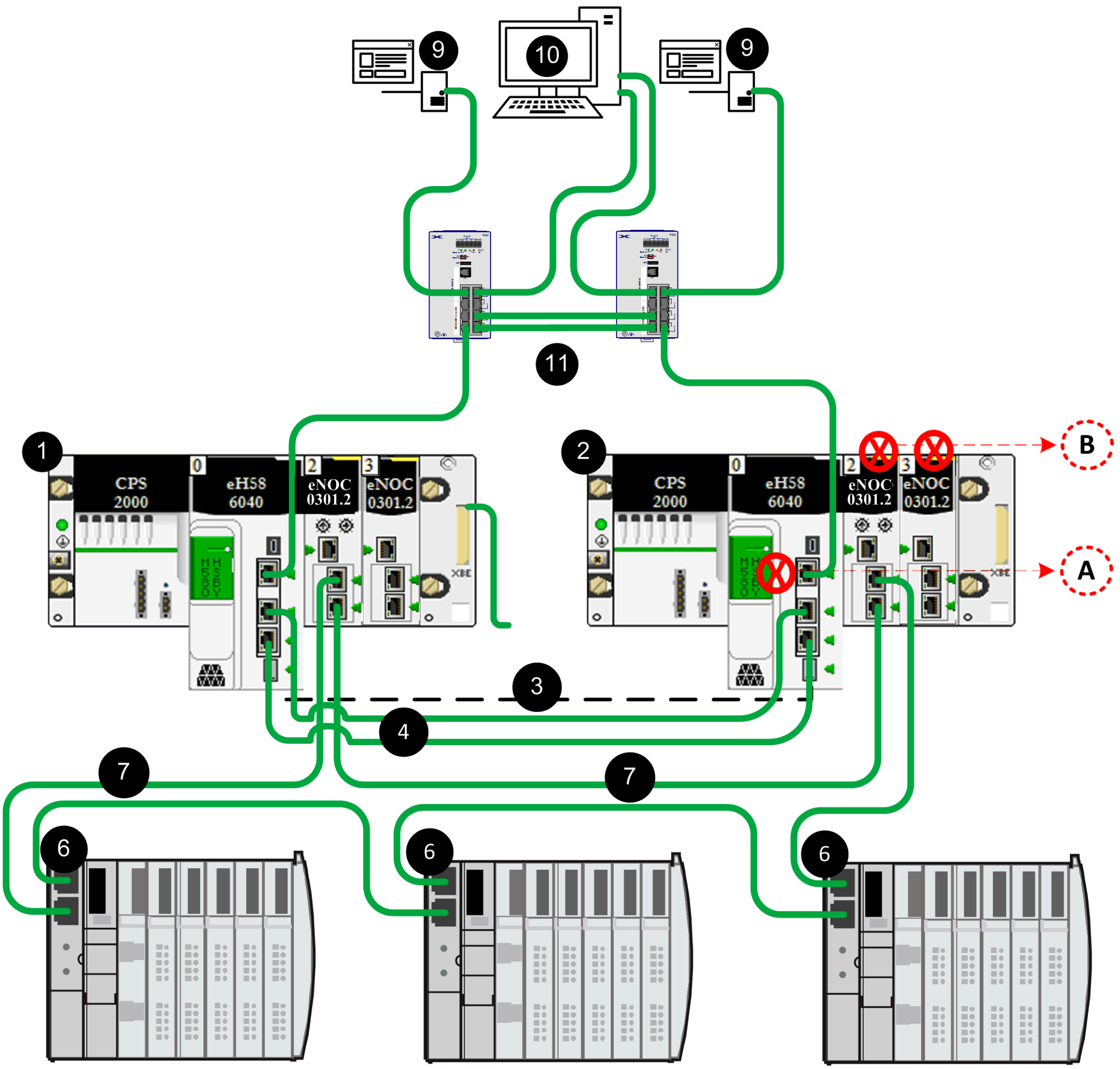
8 SCADA server

9 engineering workstation with dual Ethernet

10 control room network

Example 6

This topology is an M580 Hot Standby system with a DIO network scanned by two BMENOC0301 modules on the local rack that communicate with the control network (one of them through the Ethernet backplane), and a single SCADA connection via the service ports of the primary and standby controllers.



A In this topology, where SCADA is connected to the Hot Standby system via service ports of the primary and standby controllers, confirm that you select the **Automatic blocking of service port**

on Standby controller check box in the **ServicePort** configuration tab to avoid network communication interruptions (formed by cables 4 and 11).

B In this topology, where a DIO ring network communicates with the control network via two BMENOC0301 modules on the local rack, the Ethernet backplane port of the standby NOCs are automatically disabled to avoid network communication interruptions (formed by cables 4 and 7).

green cable RIO network, device network, and control network on same subnet

1 primary local rack with primary controller and BMENOC0301 module

2 standby local rack with standby controller and BMENOC0301 module

3 Hot Standby communication link

4 Ethernet RIO main ring

6 distributed equipment

7 Ethernet DIO ring

9 SCADA server

10 engineering workstation with dual Ethernet

11 control room network

Working with M580 Hot Standby Applications

Overview

This chapter shows you how to configure and work with M580 Hot Standby applications.

M580 Hot Standby Programming Rules

Introduction

This section summarizes some of the code features and programming rules of an M580 Hot Standby application.

Error Correcting Code (ECC)

M580 Hot Standby controllers with firmware version 2.50 and later include an error correcting code (ECC) feature. ECC enhances reliability by reducing the likelihood of memory random access errors, when an M580 Hot Standby controller accesses its internal memory, as part of a memory transfer event. The ECC function is enabled by default.

When ECC is enabled, it may impact the MAST cycle time of M580 Hot Standby controller applications. This can be the case where a relatively small amount of code is transferred, but a large amount of data is transferred. If the impact on MAST cycle time is not suitable for your application, you can:

- Reduce the amount of exchanged data from the primary to the standby controller.
- For a non-safety-related controller application, disable the ECC feature using %SW150 (see EcoStruxure™ Control Expert, System Bits and Words, Reference Manual).

Changing Declared Variables

Do not overwrite the initial values for declared variables using the save operation invoked using the system bit %S94. These changes to declared variable values are not part of the database transfer, and can lead to unintended consequences at switchover.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Do not overwrite intial values and saving them with the system bit %S94.

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

Section Executed on Standby Restrictions

The following restrictions apply to sections executed on the standby controller, first section, or all sections depending on the configuration:

- Derived Function Blocks (DFB) cannot be executed on standby controller sections.
- R_TRIG, F_TRIG, TRIGGER, TON, TOF, TP functions blocks cannot be executed on standby controller sections.
- Asynchronous communication procedures cannot be executed on standby controller sections.
- Asynchronous communication function blocks cannot be executed on standby controller sections.

Asynchronous Communication Procedures

During a switchover event, asynchronous communication procedures follow the suggested procedures below; otherwise, the new primary controller may not automatically resume operation of these communication functions. If the suggested procedures below are not conducive to your application, you must use other communication means.

- READ_VAR
- WRITE_VAR
- DATA_EXCH
- INPUT_CHAR
- INPUT_BYTE
- PRINT_CHAR

The following procedures should be used to allow asynchronous communication EFBs to automatically resume operation after a switchover.

- Program your application so that the EFBs management parameters are not exchanged with the standby controller. To do this, de-select the Exchange on STBY attribute for the management parameter.
- Initialize the Length parameter each time the function is called.
- Set the Timeout parameter accordingly to your application:

  If the communication function is sent through the controller, the typical timeout value is 500 ms.

  If the communication function is sent through a NOC module, the typical timeout value is 2 s.

NOTE: If you are unable to follow these procedures, and a switchover renders your communication function inoperative, write your application program so that it sets the function activity bit to 0 before restarting the function in the new primary controller.

Asynchronous Communication Function Blocks

During a switchover event, asynchronous communication function blocks, which use internal management parameters follow the suggested function blocks below; otherwise, the new primary controller may not automatically resume operation of these communication functions. If the suggested function blocks below are not conducive to your application, you must use other communication means.

- GET_TS_EVT_M
- READ_DDT
- READ_PARAM_MX
- READ_STS_MX
- RESTORE_PARAM_MX
- SAVE_PARAM_MX
- WRITE_CMD_MX
- WRITE_PARAM_MX
- MBP_MSTR
- READ_SDO
- WRITE_SDO
- ETH_PORT_CTRL
- PWS_DIAG
- PWS_CMD
- L9_MSTR

To allow asynchronous communication EFBs to automatically resume operation after a switchover, program your application so that the EFBs instances are not exchanged with the standby controller. To do this, de-select the Exchange on STBY attribute for the EFB instance.

While the use of the functions listed above requires special consideration in a hot standby configuration, you must use care with any function that is capable of writing to memory areas that are not part of the Hot Standby database transfer, such as Data Storage, page 147 function blocks for instance.

Debugging

Debugging your Hot Standby application program is a two-stage process:

- First, you debug the application on a single Hot Standby controller. This allows you to use debugging features available in Control Expert, such as watchpoints, and so on.
- Next, you debug your application when it has been uploaded to the two Hot Standby controllers in a working redundant system, but in a non-production environment. On this platform, you evaluate performance specific to Hot Standby redundancy. Only a subset of Control Expert debug features can be used during this stage.

NOTE: See *M580 Hot Standby Diagnostics* for further details on debugging your application.

PME UCM 0302 Universal Communication Module

Do not use a PME UCM 0302 universal communication module in a Hot Standby configuration.

M580 Hot Standby System Configuration

Control Expert Configuration Tool

The configuration tool for an M580 Hot Standby system is:

- Control Expert Version 13.1 and later for any non-safety-related Hot Standby controllers
- Control Expert XL Safety Version 14.0 and later for the BMEH582040S, BMEH584040S, and BMEH586040S controllers
- Unity Pro L Version 11.0 and later for the BMEH582040 controller

- Unity Pro XL Version 11.0 and later for the BMEH584040 and BMEH586040 controllers

Programming Application Languages and Libraries

Control Expert supports the following application languages and libraries for the M580 Hot Standby controllers:

Application language / library	**Non-safety-related controllers**		**Safety-related controllers**			
	BMEH58...		**BMEH58...**			
	2040	**4040, 6040**	**2040S**		**4040S, 6040S**	
			SAFE task	**FAST, MAST tasks**	**SAFE task**	**FAST, MAST tasks**
Function Block Diagram (FBD)	X	X	X	X	X	X
Ladder Diagram (LD)	X	X	X	X	X	X
Structured Text (ST)	X	X	–	X	–	X
Instruction List (IL)	X	X	–	X	–	X
Sequential Function Chart (SFC)	X	X	–	X	–	X
Derived Function Block (DFB)	X	X	X	X	X	X
Elementary Function (EF)	X	X	X[1]	X	X[1]	X
Elementary Function Block (EFB)	X	X	X[1]	X	X[1]	X
Ladder Logic 984 (LL984)	–	X	–	–	–	X
PL7 - Standard Function Block (SFB)	–	–	–	–	–	–
X: Supported –: Not supported 1: EF/EFB prefixed with “S_”						

Configuring an M580 Hot Standby Controller

Introduction

This topic shows you how to configure the Hot Standby functionality of a M580 BMEH58•040 controller.

NOTE: The same procedure, as described below, can also be applied to the configuration of a BMEH58•040S safety-related controller.

Accessing the M580 Controller Hot Standby Configuration Tab

Use the **Hot Standby** tab of the BMEH58•040 controller to configure its Hot Standby function. To access this tab:

Step	Action
1	Add a BMEH58•040 controller to your project.
2	In the **Project Browser**, select **Configuration** **>** **PLC Bus** **>** **<rack>** **>** **<controller>**.
3	Right-click the controller and select **Open**.
4	Click the **Hot Standby** tab.

Configuring the Hot Standby Function

The **Hot Standby** tab presents the following configurable settings:

Setting		Description
Run Mode	**Controller A Online**	Specify if controller A and controller B start in RUN at cold start: • TRUE (default): The controller attempts to start in RUN. Depending on other conditions, the controller may act as the primary or standby. • FALSE: The controller starts in either the WAIT or STOP state at cold start.
	Controller B Online	
Standby On Logic Mismatch	**Number of modifications**	The maximum number of online build changes from 1...50 that can be performed on the primary controller. When this number of online build changes has been reached, you need to transfer the application from the primary to the standby to be able to make additional online build changes. Default = 20. **NOTE:** • If this setting is set to 0, the `Logic Mismatch Allowed` flag has no effect. • This setting cannot be edited via CCOTF.
Behaviour of the CPU in Wait and Standby mode	**CPU executes**	Specify the sections of the MAST task the standby controller executes in Wait state: • All sections (default) • First section • No section at all When Control Expert is connected to the standby controller, the **Sections** in the **Project Browser** are preceded by: • a **green** light for sections without condition or with a TRUE condition even if not executed • a **red** light for sections with a FALSE condition **NOTE:** • You can also individually specify the sections of the MAST task the standby controller executes while in Wait state. Do this by adding a condition of execution in the **Condition** tab of the **Properties** window for a MAST task section. • For a safety-related controller, sections of the SAFE task are not executed when the controller is in WAIT or STANDBY state. You can also individually specify the sections of the MAST task the standby controller executes while in Wait state. Do this by adding a condition of execution in the **Condition** tab of the **Properties** window for a MAST task section.
Data Exchanged	–	A bar graph displays the percentage of controller memory used by Hot Standby data. The value depends on the M580 Hot Standby configuration. The total data exchanged is displayed in KB as well as: • data exchange by MAST • data exchange by FAST • data exchanged by SAFE (for a safety-related controller)

Configuring the Controller Online State

Controller A is the controller with the **A/B/Clear** rotary switch, page 151 (located on the back of the controller) set to **A**. Controller B is the controller with the **A/B/Clear** rotary switch set to **B**.

You can use the **Controller A Online** and **Controller B Online** settings for the following purposes:

- To specify the controller that will be primary on a cold start. For example, set **Controller A Online** to **TRUE** and **Controller B Online** to **FALSE**. Controller A powers up as primary, and controller B powers up in wait state. After power up, you can manually or programmatically set **Controller B Online** to **TRUE**.
- To avoid a switchover. For example, if controller A is primary and controller B is standby, set **Controller B Online** to **FALSE**. Controller B enters wait state, and no switchover can occur.

These settings can be modified during runtime, or when the Hot Standby system is not operating.

Settings entered when the Hot Standby system is not running take effect after the next project build, when the Hot Standby system next starts-up.

If the Change Configuration on The Fly (CCOTF) function is enabled, settings entered when the Hot Standby system is running take effect on the next project build (or re-build).

No Local I/O Configuration

Because the local rack of a Hot Standby controller cannot include I/O modules, the following settings in a BMEH58•040 or BMEH58•040S controller **Configuration** tab are disabled:

- **Run/Stop input**
- **Run/Stop by input only**
- **Memory protect**
- **Maintenance Input** (safety-related controller)

  NOTE: Instead of using the **Run/Stop input**, consider using the following approach to controlling the RUN/STOP operating state of a safety-related controller:

  - Use a BMENOC0301 / BMENOC0311 or BMENOC0302(H) communication module, along with the IPsec protocol, to help provide a secure connection to the controller.

    NOTE: The BMENOC0302(H) module is not compliant with safety-related controllers.

  - Then use the `CMD_RUN_REMOTE` or `CMD_STOP_REMOTE` commands of the `T_M_ECPU_HSBY` DDT to change a remote controller operating state.

Enabling FDR Server Synchronization in a Hot Standby System

In an M580 Hot Standby system, a BMEH58•040 controller or a BMENOC0301 / BMENOC0311 or BMENOC0302(H) communication module can perform the role of an FDR server. To permit the synchronization of the FDR server in the primary controller with the FDR server in the standby controller, you need to enable the TFTP service for the Hot Standby system.

To enable the TFTP service, follow these steps:

Step	Action
1	In the **Project Browser** double-click on the following: **Project > Configuration > 0:PLC bus > <rack> > <controller> > EIO**. The **RIO DIO Communicator Head** window opens.
2	Click the **Security** tab.
3	For the **TFTP** service, select **Enabled**.
4	If **Access Control** is enabled, create an entry for each device or subnet that you want to have TFTP access to the controller. **NOTE:** Select the **TFTP** column for each entry.
5	**Validate** and **Save** your edits.

NOTE: The FDR server cannot synchronize the primary and standby controllers when the TFTP service is disabled. The TFTP service is enabled and disabled by the execution of the `EthPort_Control_MX` function in the application.

If you want to programmatically enable or disable TFTP, include the `EthPort_Control_MX` function in a section of the application that is executed by the standby controller, so that this function is executed by both the primary and standby controllers.

Change Configuration On The Fly (CCOTF)

CCOTF Rules for Hot Standby

The M580 BMEH58•040 and BMEH58•040S controllers support CCOTF. CCOTF is enabled in the **Configuration** tab of the controller, in the Configuration Online Modification area, by selecting **Online modification in RUN or STOP**.

For information about CCOTF for M580 safety-related controllers, refer to the *Modicon M580 Safety Manual* (see Modicon M580, Safety Manual).

If a swap or switchover occurs after a CCOTF has been performed on the primary controller, and the application has not been transferred to the standby controller, then the configuration change will no longer be effective.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

- Before starting a CCOTF operation, verify that the application running in the Hot Standby system does not trigger a swap and that no condition exists that could forseeably cause a switchover.
- Always apply a CCOTF transaction on the primary controller.

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

NOTE: To download CCOTF changes to a Hot Standby system:

- Always apply a CCOTF transaction to the primary controller.
- Confirm the Hot Standby system is operational with a healthy Hot Standby link between the two controllers.
- Confirm that the impacted Ethernet RIO drop is operational, with a healthy Ethernet RIO link.

CCOTF allows modifications of a primary controller configuration in RUN mode. The changes that can be made in the primary controller are as follows:

- Add a discrete or analog module.
- Remove a discrete or analog module.
- Modify the configuration and adjustment parameters of a module.

The changes that can be made in an Ethernet RIO drop are as follows:

- Add an (e)X80 or Quantum RIO drop.
- Add a discrete or analog module.
- Remove a discrete or analog module.
- Modify the configuration and adjustment parameters of a module.

CCOTF changes made to the primary controller configuration are not automatically transmitted to the standby controller. Instead, the standby controller continues to be configured with its original application program.

CCOTF does not support all changes to the configuration. The following rules apply to CCOTF changes made to the primary controller configuration:

- A single CCOTF change can include multiple edits to multiple configuration objects.

- Only one change can be made to a single configuration object. For example, you cannot add then delete the same I/O module in a single CCOTF change.
- CCOTF edits cannot be made to distributed equipment.
- For an (e)X80 or Quantum RIO drop, the following limits apply to changes made in the same CCOTF session:
  - Up to four modifications to the same RIO drop can be included in a single CCOTF change. For example:

    - Up to four I/O modules can be added to the same RIO drop.

    - Up to four I/O modules can be removed from the same RIO drop.

    - Up to four parameters can be edited for one I/O module in the same RIO drop.
  - No edits can be made to an adapter module.
  - No edits can be made to BMXERT1604 modules (time stamp).
  - The RPI setting for the RIO drop cannot be changed.
- IP addresses cannot be changed.
- Only one CCOTF session may be made to a single RIO drop. Before an additional CCOTF session can be made to the same RIO drop, transfer the application program from the primary controller to the standby controller.

NOTE: You can set Control Expert to **Virtual connected mode** to test whether a proposed change to the configuration is a CCOTF event (see Modicon M580, Change Configuration on the Fly, User Guide).

When CCOTF changes are made to the primary controller, the `Logic_Mismatch_Allowed` flag in the `T_M_ECPU_HSBY` DDT determines if the standby controller can continue to operate. If logic mismatches are not allowed, the standby controller transitions to WAIT state.

CCOTF changes can be made to the primary controller if the **Number of modifications** setting in Control Expert is not reached. When the number of allowed modifications is reached:

- No additional CCOTF changes can be made to the primary controller. The **Build > Build Changes** command in Control Expert is disabled.
- You need to transfer the application program in the primary controller to the standby controller, page 132.

Modifying an SFC Section Online

Precautions for Modifying an SFC Section Online

When the M580 Hot Standby system executes a switchover, the new primary controller tests the `SFC_MISMATCH` bit. The `SFC_MISMATCH` bit is set when the structure of at least one SFC section in the primary controller differs from that section in the standby controller. If this bit is set, the controller re-initializes the state-machine of the modified SFC sections to help prevent any unpredictable behavior of the user application.

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION • Transfer the application from primary controller to the standby controller after each online modification of a MAST task section that is programmed using the sequential function chart (SFC) programming language. • Do not execute a switchover or trigger a swap before this transfer is successfully completed. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

To avoid the re-initialization of the SFC state-machines when you modify an SFC section, follow these steps:

Step	Action
1	Confirm that the `LOGIC_MISMATCH_ALLOWED` bit is set to 1. **NOTE:** If logic mismatch is not allowed, the standby controller enters wait after step 3.
2	Make the online edit to the SFC section in Control Expert.
3	Build the online change in Control Expert by selecting **Build > Build Changes**. The modification is made to the program running in the primary controller.

4	Transfer the application from the primary controller to the standby controller. Use a Control Expert animation table to set the `CMD_BACKUP_APPLI_TRANSFER` bit to 1.
	NOTE: Alternatively, you can automate the transfer in program logic using a code sequence like the following:
	`if (ECPU_HSBY_1->SFC_MISMATCH = 1)`
	`then ECPU_HSBY_1-->CMD_BACKUP_APPLI_TRANSFER = 1`

Configuring IP Addresses for an M580 Hot Standby System

Introduction

This topic shows you how to assign IP addresses to an M580 Hot Standby system. For information on how to configure other Ethernet communication settings for the controller, refer to the *M580 Hardware Reference Manual* (see Modicon M580, Hardware, Reference Manual).

Accessing the M580 Hot Standby Controller Animation Task Tab

Use the **IPConfig** tab of the **EIO** configuration window for an M580 BMEH58•040 or BMEH58•040S controller to assign IP addresses. To access this tab:

Step	Action
1	Add a BMEH58•040 or BMEH58•040S controller to your project.
2	In the **Project Browser**, navigate to and select **Configuration > PLC Bus > <rack> > <CPU> > EIO**.
3	Right-click and select **Open**.
4	Click the **IPConfig** tab.

Assigning IP Addresses to M580 BMEH58•040 or BMEH58•040S Controllers

An M580 Hot Standby system requires the assignment of three IP addresses. In addition, Control Expert automatically creates and assigns a fourth IP address. IP address settings include:

IP address name	Description
Main IP address	The configurable IPv4 IP address used by the primary controller for communication with distributed equipment. **NOTE:** Because this setting is always assigned to the primary controller, it can be associated with either the A or B controller. When a switchover occurs (for example, when controller B becomes primary) the main IP address assignment is transferred from controller A to controller B.
Main IP address + 1	The Control Expert auto-generated IPv4 IP address used by the standby controller for communication with distributed equipment. This auto-generated IP address equals the Main IP address plus 1 in the fourth octet. For example, if the Main IP address is 192.168.10.1, this auto-generated IP address is 192.168.10.2. **NOTE:** • This IP address is not editable in Control Expert. Its purpose is to provide communication transitions on Hot Standby controller switchovers. • Avoid assigning this IP address (the Main IP address + 1) to any device that may communicate with the Hot Standby system. If you do assign this IP address to another device, a duplicate IP assignment condition may occur.
IP address A	The configurable IPv4 IP address for the controller with its **A/B/Clear** rotary switch, page 151 set to **A**. Controller A uses this IP address for communication on the Ethernet RIO network.
IP address B	The configurable IPv4 IP address for the controller with its **A/B/Clear** rotary switch, page 151 set to **B**. Controller B uses this IP address for communication on the Ethernet RIO network.
Subnetwork mask	The configurable 32-bit value used to identify both the network address and the subnetwork portion of the IP address. **NOTE:** If you use Edge I/O NTS modules in your M580 network, modify the default M580 IP address configuration. The default M580 subnetwork mask of 255.255.0.0 is in conflict with the Edge I/O NTS default USB IP address (in the range of 192.168.200.1). Set the M580 subnetwork address to 255.255.254.0. The main IP address remains the same (in the range of 192.168.10.1 to 192.168.11.254). This action works for a maximum of 510 devices. **NOTE:** Devices in the following ranges are assigned private IP addresses • Class A: 10.0.0.0 – 10.255.255.255 • Class B: 172.16.0.0 – 172.31.255.255 • Class C: 192.168.0.0 – 192.168.255.255
Gateway address	The configurable IP address of the default gateway to which messages for other networks are transmitted.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Confirm that each module has a unique IP address.
- Do not assign an IP address equal to the Main IP Address, the Main IP Address + 1, IP Address A, or IP Address B to any Ethernet device that potentially communicates with the Hot Standby system.

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

Editing IP Address Settings for Adapter Modules

From the **IPConfig** tab, you can access IP address settings for (e)X80 EIO adapter modules. Click on the **Update CRA IP address configuration** link to open the **Ethernet Network Manager**, which lists adapter modules on connected Ethernet networks.

In the **Ethernet Network Manager**, you can edit the following settings for each adapter module:

- **IP address**: The configurable IPv4 IP address the adapter module uses for communication on the Ethernet network.
- **Identifier**: The text string used by the module to identify itself to other devices, for Ethernet services including DHCP and FDR. The value depends on the module you are using:
  - for 140CRA32100: 140CRA_*XXX*
  - for BMECRA31210: BMECRA_*XXX*
  - for BMXCRA312•0: BMXCRA_*XXX*
  - for BMECRA31310(H): PCRA_[Drop Number]_[Slot]

  Where *XXX* represents the concatenation of the two rotary switch settings on the (e) X80 EIO adapter module.

Configuring Data Variables for an M580 BMEH58•040(S) Hot Standby Application

Introduction

BMEH58•040 Hot Standby and BMEH58•040S safety-related Hot Standby controllers support the following data attributes:

Attribute	Controller	
	BMEH582040, BMEH582040S, BMEH584040, BMEH584040S	**BMEH586040, BMEH586040S**
Exchange On STBY	X	X
Retain	–	X
X: Supports the attribute. –: Does not include the attribute, because all data is retained.		

For a safety-related controller, each variable set to **Exchange On STBY** is associated with a task (MAST, FAST, or SAFE). The amount of data that can be exchanged from the primary to the standby safety-related controller depends on the task:

- MAST & FAST: up to 4 MB of data can be exchanged.
- SAFE: up to 1 MB of data can be exchanged.

For information on how to use the Control Expert **Data Editor**, and display the **Retain** and **Exchange On STBY** attributes, refer to the *EcoStruxure™Control Expert, Operating Modes* (see EcoStruxure™ Control Expert, Operating Modes) manual.

Retain

BME•586040 controllers present the **Retain** variable attribute. This attribute determines whether the variable value will persist after a warm start of the controller. If the attribute is:

- Selected: Variable data persists and is applied to the variable after a warm start.
- De-selected: Variable data is lost after a warm start; the variable value is reset.

For non-safety-related standalone Modicon M580 controllers, this attribute is read-only. It is selected by default and cannot be de-selected.

For both standalone and Hot Standby safety-related controllers, the **Retain** variable attribute is not included for variables created in the safety-related area. Safety-related data is not retained, because the SAFE task executes a cold start.

NOTE: In the event of a cold start of the controller, both retained and non-retained data is reset.

The amount of maximum configurable retained data varies, depending on the controller. Refer to Controllers, page 47.

For the BME•586040 controllers, you cannot edit the **Retain** attribute for a variable that existed at controller start-up. When a variable is created online as part of a CCOTF change, you can edit the **Retain** attribute which remains modifiable until the first build change is performed.

NOTE: The amount of retained data is presented as saved data in the **Memory Usage** window.

Exchange On STBY

Before each scan in a Hot Standby system, the primary controller exchanges data with the standby controller. It exchanges only that data with the **Exchange On STBY** attribute set to **YES**.

NOTE:

- When a reference is initialized inside the **Data Editor**, the initialization variable needs to be part of the same task as the reference. Otherwise, a detected error message is included in the **Output Window** when the project is analyzed.
- The **Exchange On STBY** attribute is not editable for all variables.
- In a Hot Standby system, if you have configured explicit messaging using a communication function, exclude the communication function block `Management_Param` from the data to be transferred from primary to standby. To do this, de-select the **Exchange on STBY** attribute for the `Management_Param` parameter in Control Expert.

You cannot edit the **Exchange On STBY** attribute for a variable that existed at controller start-up. When a variable is created online as part of a CCOTF change, you can edit the **Exchange On STBY** attribute which remains modifiable until the first build change is performed.

The amount of maximum configurable data varies, depending on the controller. Refer to Controllers, page 47

Each variable that is included in the Hot Standby exchange also presents a read-only **Task** attribute. The setting of the **Task** attribute is auto-generated by Control Expert for each variable included in the Hot Standby exchange.

Service Example

Objective: Write to a single register %MW100, Length := 5

(* REQUEST WRITE SINGLE REGISTER %MW100 Length := 5 *)

(* Data_to_send = Modbus request encoding *)

(* Byte 1 = Register Address Hi = 0 ; Byte 0 = Function code = 06 *)

Data_to_Send[0] := 6;

(* Byte 3 = Register Value Hi ; Byte 2 = Register Address Lo = 100 *)

Data_to_Send[1] := (RegisterValue & 16#FF00) + 100;

(* Byte 5 = unused; Byte 4 = Register Value Lo)

Data_to_Send[2] := RegisterValue & 16#FF;

IF ((Management_Param[ACTIVITY] & 1) = 0) THEN

Management_Param[LENGTH] := 5; (* LENGTH RQ WRITE *)

DATA_EXCH (ADDM('0.0.0.1'), 1, Data_To_Send, Management_Param, Received_Data);

END_IF;

NOTE: The MODBUS bus is BIG-ENDIAN and P-UNIT words are LITTLE-ENDIAN. For some queries, perform a conversion.

The ROL instruction can be used:

Value_read := ROL(Received_Data[1], 8); (* CONVERT BIG/LITTLE ENDIAN *)

Configuring Hold Up Time for Drops and Devices

Hold Up Time

Hold up time is part of each configuration. It represents the time (in milliseconds) that device outputs are maintained in their present states after a communication disruption before reverting to their fallback values.

Hold up time settings can range from 50...65530 ms. By default, Control Expert sets hold-up time to 4 times the MAST **Watch Dog** setting. Because the default watchdog setting is 250 ms, Control Expert applies a default drop hold up time setting of 1000 ms.

Setting Hold Up Time for RIO Drops

When configuring MAST **Hold up time**, consider both of the following:

- The maximum time between controller requests.
- MAST task watchdog time.

If **Hold up time** is not set to a sufficiently large value, the outputs of a drop may enter fallback during a switchover. This can cause a disruption in the behavior of outputs that have a fallback setting other than *hold last value*.

To accommodate both MAST and FAST tasks for (e)X80 RIO drops, set drop **Hold up time** to a value not less than 4.4 times the MAST period.

M580 Hot Standby supports the following tasks:

Task	Type	Period	Watchdog time	Remote I/O platform:	
				Quantum RIO	M580 (e)X80
MAST[1]	Periodic	1...255 ms	10...1500 ms[2]	X	X
FAST	Periodic	1...255 ms	10...500 ms[2]	–	X
SAFE	Periodic	10...255 ms	10...500 ms[2]	–	X

X: Supported

–: Not supported

1. MAST task is mandatory and cannot be deactivated for both (e)X80 and Quantum RIO drops.

2. If CCOTF is activated, the minimum watchdog value is 64 ms.

Setting Hold Up Time for Distributed Equipment

The hold up time represents the time that device outputs are maintained in their present states after a communication disruption and before taking their fallback values. Because distributed devices are not connected to the primary controller during a switchover, set the hold up time to a value greater than the expected duration of the communication interruption.

For Modbus TCP devices:

- Set the hold up time to exceed: 4.4 x (MAST period) + 600 ms.

For EtherNet/IP devices:

- Set the hold up time to exceed: 4.4 x (MAST period) + 5000 ms.

Transferring M580 Hot Standby Projects

Introduction

In an M580 Hot Standby system, both the primary controller and the standby controller begin by operating the same application. CCOTF changes that are made to the application running in the primary controller are not also made to the standby controller. This causes a logic mismatch to exist between the two controllers.

After modifications, it is necessary to transfer the application from the primary controller to the standby controller, so that both controllers are once again operating the same application. There are several ways to make this transfer.

> **NOTE:** The operating mode setting of a safety-related controller either safety mode or maintenance mode – is not included in the transfer of an application from the primary controller to the standby controller. On a switchover, when a safety-related controller switches from standby controller to primary controller, the operating mode is automatically set to safety mode.

For additional information on safety-related controller operating modes, refer to the *Modicon M580 Safety Manual* (see Modicon M580, Safety Manual).

Transferring the Application from the Primary to the Standby Controller

The Control Expert application can be transferred from the primary controller to the standby controller in many ways, including the following:

- **Automatic transfer**: If the non-primary controller is in a non-configured state, the primary controller automatically transfers the application program and data to the non-primary controller when it powers up. There are several ways a controller can be put into in a non-configured state, including:
  - It is a new device that is being deployed for the first time.
  - Its **A/B/Clear** rotary switch, page 151 was set to **Clear**, powered-up, then re-set to **A** or **B** (depending on the **A/B** designation of the primary controller).

    > **NOTE:** To place the standby controller into run mode on restart, set the `CMD_RUN_AFTER_TRANSFER` DDDT command to TRUE before power-up.

- **Transfer from PC to the standby controller**: If your PC with Control Expert has open the same application as the one running in the primary controller, you can transfer the application from your PC to the standby controller. To do this, connect your PC to either the Ethernet service port or USB port of the standby controller, then use the **PLC > Transfer Project to PLC** command to make the transfer.

  NOTE: If the standby controller is connected to a configuration tool, such as Control Expert, only the connected configuration tool can transfer an application to the standby controller. In this case, the primary controller cannot transfer an application to the standby.

- **Transfer from primary controller to standby controller**: With Control Expert connected to the primary controller, and with both the primary and standby controllers running, use one of the following methods to make the transfer:

  - Use the Control Expert **PLC > Transfer Project from Primary to StandBy PLC** GUI command.

    or

  - Use the `CMD_APP_TRANSFER` command of the `T_M_ECPU_HSBY` DDT.

  NOTE:

  - The application transferred is the backup application, stored in flash memory or on the SD card. If the application running does not match the backup application, perform an application backup (**PLC > Project Backup... > Backup Save** or set the %S66 system bit to 1) before performing the transfer.
  - If the `CMD_RUN_AFTER_TRANSFER` flag is set, the standby controller automatically starts to run after completion of the transfer, reducing down time for the standby controller.

  In each case, if both the primary and standby controllers are equipped with SD memory cards, the application is transferred to both the standby controller and its SD memory card.

- **SD memory card**: If the primary controller includes an SD memory card with the application, take the SD card from the primary controller, place it into the standby controller, then reboot the standby controller.

In each case:

- The application is transferred only if the application in the standby controller is different from the application being transferred to it.
- If the application running in the primary controller is different from the application stored in flash memory or on the SD memory card, perform a backup of the running application (**PLC > Project Backup > Backup Save**) before making the transfer.

NOTE:

- You cannot transfer the application from the standby controller to the primary controller.
- If the Logic_Mismatch_Allowed command is set, and if the **Number of modifications** has not been reached, you can connect Control Expert to the standby controller, then use the CMD_SWAP DDT command to make the standby controller the primary controller. Thereafter, you can transfer the application from new primary controller (formerly the standby) to the standby controller (formerly the primary).

Run After Transfer

If you use program logic or an animation table to set the T_M_ECPU_HSBY DDT command CMD_RUN_AFTER_TRANSFER, page 166, the primary controller automatically begins to run immediately upon completion of the transfer.

Primary Controller Online Application Modifications with Allowed Application Mismatch

Procedure

To make online modifications to an application program in the primary controller, follow these steps:

Step	Action
1	Verify that the primary and standby controllers are in Run primary controller and Run standby controller modes.
2	Connect Control Expert to the primary controller.
3	Set the Command Register system bit %SW60.3 to 1
4	Modify online the application program.
5	Perform **Build Project**. **NOTE:** If adding/removing modules using CCOTF (see Quantum using EcoStruxure™ Control Expert, Change Configuration On The Fly, User Guide), use **Build Changes**.
6	Verify that primary and standby controllers are in Run primary controller and Run standby controller modes.

Step	Action
7	Perform an application transfer to the standby controller.
8	Set the Command Register system bit %SW60.3 to 0. **NOTE:** The Command Register system bit is then returned to 0 from 1.

Managing M580 Hot Standby Data Exchanges

Overview

This chapter describes M580 Hot Standby system data management and the `T_M_ECPU_HSBY` DDT.

Exchanging M580 Hot Standby Data

Periodic Data Exchanges

The Hot Standby controllers perform two periodic data exchanges:

- Before each MAST cycle, the primary controller transmits to the standby controller application variables, system status and I/O data.
- Periodically, both controllers exchange the contents of the `T_M_ECPU_HSBY` DDT.

Data Transmitted Each MAST Cycle

Before each MAST task, the primary controller transmits data to the standby controller in two ways. The primary controller uses:

- The Hot Standby link to send application variables, system status, and I/O data.
- The Ethernet RIO link to send application variables and system status.

When the Hot Standby link is incommunicative, the standby controller does not receive updated I/O data and application variables. If it lasts for three seconds or more, the standby controller enters the wait state.

Your application needs to regularly verify the data synchronization of the MAST, FAST, and SAFE (for safety-related controllers) tasks through the Hot Standby link. You can do this using the `MAST_SYNCHRONIZED`, `FAST_SYNCHRONIZED` and `SAFE_SYNCHRONIZED` bits in the `T_M_ECPU_HSBY` DDT.

> **NOTE:** Due to I/O data size and transfer time constraints, I/O data is not exchanged by the primary controller with the standby controller over the Ethernet RIO link.

Transfer of the Hot Standby DDT

The exchange of the T_M_ECPU_HSBY DDT is a 2-way data exchange made while both controllers are running. This exchange is made over both the Hot Standby link and the Ethernet RIO link.

The exchange occurs every 5 ms over the Hot Standby link, and every 10 ms over the EIO link. The exchange occurs regardless of the Hot Standby state of the controllers (primary, standby, wait, or stop). This exchange includes up to 64 words of variable items where the **Exchange On STBY** attribute is editable and has been selected.

Identifying Exchanged Data

Only data items with the **Exchanged On STBY** attribute set to **YES** are included in the data exchange. This attribute is editable for some data variables, but is automatically set for other variables:

Variable type	Exchange On STBY default setting	Editable?
State RAM	Yes	No
Located variables	Yes	No
Unlocated variables	Yes	Yes
Device DDT (managed)	Yes	No
Device DDT (unmanaged)	Yes	Yes

You can specify which unmanaged DDDT variables are included in the data exchange by setting the **Exchange On STBY** flag to **NO**.

When you create a variable and set its **Exchange On STBY** flag to **YES**, that variable appears in the LOCAL_HSBY_STS area of the instantiated T_M_ECPU_HSBY DDDT, under the REGISTER element. The REGISTER element can contain up to 32 DWORDs (64 WORDs of data).

The amount of maximum configurable retained data varies, depending on the controller. Refer to Controllers, page 47. If the amount of data in your Hot Standby system exceeds the maximum amount the controller can transmit, you can:

- Use a controller with a higher data transfer capacity.
- De-select the **Exchange On STBY** attribute for some unmanaged DDDT variables.
- Re-design your Hot Standby network so that the amount of Hot Standby data to be exchanged does not exceed controller capacity.

Associating Variables with Tasks

Each data item is associated with a task. When you create a new data item in the **Data Editor**, you need to associate it with a task:

- A MAST task is required by the Hot Standby system, and can be assigned to data items related to the Hot Standby controller and RIO drops (both Quantum and M580).
- FAST tasks are optional for Hot Standby controllers, and can be assigned only to M580 (e)X80 drops.

  NOTE: In an M580 Hot Standby system, variables related to Quantum RIO drops cannot be assigned to a FAST task.
- Safety-related data are automatically associated only with the SAFE task.

Preconditions for Data Exchange: Primary and Standby Controllers

The Hot Standby data exchange is made while one controller remains the primary and the other is the standby. Both the primary controller and a standby controller can continue in their roles as long as the Hot Standby link remains operational.

A single break, page 161 in the Ethernet RIO main ring does not cause an interruption of Ethernet RIO communication between the primary and standby controllers. The controllers continue to function as primary and standby respectively. The primary controller continues to exchange data with the standby over both the Hot Standby and the Ethernet RIO links.

Two breaks, page 162 in the Ethernet RIO main ring (depending on their location) can cause an interruption of Ethernet RIO communication between the primary and standby controllers. However, even if the two controllers are isolated from each other on the Ethernet RIO ring, they can still communicate over the Hot Standby link. If both controllers continue to communicate with RIO drops, page 164, the controllers continue to function as primary and standby respectively. The primary controller continues to exchange data with the standby over the Hot Standby link.

Effects of Online Modifications to Hot Standby Data

When you modify the configuration of – or application in – the primary controller, those changes are not applied to the configuration of the standby controller. The exchange of Hot Standby application variables from the primary to the standby is affected, as follows:

- Data objects added to the primary controller configuration do not exist in the standby controller. In this case, the new data objects are not exchanged and:
  - The DATA_LAYOUT_MISMATCH DDT element is set.
  - The DATA_DISCARDED DDT element indicates the quantity, in kB (rounded up), of data sent by the primary controller but rejected by the standby controller.
- Data objects deleted from the primary controller configuration continue to exist in the standby controller. No updates can be exchanged for these data objects. In this case, the standby controller applies the previous value for this data and:
  - The DATA_LAYOUT_MISMATCH DDT element is set.
  - The DATA_NOT_UPDATED DDT element indicates the quantity, in kB (rounded up), of data that is retained by the standby controller but not updated.
- Unchanged data objects remain common to both the primary controller and the standby controller, and continue to be included in the data exchange.

The data structure of the primary controller and standby controller are synchronized on next application transfer.

Hot Standby DDT Data Structure

Introduction

The T_M_ECPU_HSBY DDT is the exclusive interface between the M580 Hot Standby system and the application running in a BMEH58•040 or BMEH58•040S controller. The DDT instance should appear as: ECPU_HSBY_1.

NOTE: For firmware version 2.80 and later, the T_M_ECPU_HSBY DDT is named T_ M_ECPU_HSBY_EXT.

NOTICE
UNMONITORED LOSS OF REDUNDANCY IN HOTSTANDBY SYSTEM Review and manage the T_M_ECPU_HSBY DDT for proper operation of the system. **Failure to follow these instructions can result in equipment damage.**

The T_M_ECPU_HSBY DDT presents three distinct sections:

- LOCAL_HSBY_STS: Provides information about the local controller. Data is both auto-generated by the Hot Standby system, and provided by the application. This data is exchanged with the remote controller.

- REMOTE_HSBY_STS: Provides information about the remote controller and contains the image of the last received exchange from the counterpart controller. The validity of this information is represented by the REMOTE_STS_VALID flag in the common part of this DDT . When set to 1, both controllers are communicating.

  NOTE: The structure of both the LOCAL_HSBY_STS and Remote_HSBY_STS sections are determined by the HSBY_STS_T data type, and are therefore identical. Each is used to describe data relating to one of the two Hot Standby controllers.

- A common part of the DDT: Consists of several objects, including status data, system control objects, and command objects:
  - Status data is provided by the Hot Standby system as a result of diagnostic verification.
  - System control objects enable you to define and control system behavior.
  - Command data objects include executable commands you can use to modify the system state.

Local Controller versus Remote Controller

The T_M_ECPU_HSBY DDT uses the terms *local* and *remote*:

- *Local* refers to the Hot Standby controller to which your PC is connected.
- *Remote* refers to the other Hot Standby controller.

Data Boundary Alignment

M580 BMEH58•040 and BMEH58•040S controllers feature a 32-bit data design. For this reason, stored data objects are placed on a four-byte boundary.

T_M_ECPU_HSBY DDT

You must confirm that the standby controller is ready to assume the primary role before executing a switchover command.

Verify that the value of the REMOTE_HSBY_STS.EIO_ERROR bit of the standby controller is 0 before you execute a switchover command (either by application logic or in Control Expert).

The T_M_ECPU_HSBY / T_M_ECPU_HSBY_EXT DDT consists of these objects:

Element	Type	Description	Written by
REMOTE_STS_VALID	BOOL	• TRUE: At least one of the HSBY_LINK_ERROR or HSBY_SUPPLEMENTARY_LINK_ERROR is set to 0. • FALSE (default): Both HSBY_LINK_ERROR and HSBY_SUPPLEMENTARY_LINK_ERROR are set to 1.	System
APP_MISMATCH	BOOL	The original application in the two controllers is different. (Default = FALSE)	System
LOGIC_MISMATCH_ ALLOWED	BOOL	• TRUE: The standby controller remains in a STANDBY state in case of logic mismatch. • FALSE (default): The standby controller goes into WAIT state in case of logic mismatch.	Application
LOGIC_MISMATCH	BOOL	Different revisions of the same application exist in the two controllers. (Default = FALSE)	System
SFC_MISMATCH	BOOL	• TRUE: The applications in the primary controller and the standby controller are different in at least one SFC section. In the event of a switchover, the graphs that are different are reset to their initial state. • FALSE (default): All SFC sections are identical.	System
OFFLINE_BUILD_ MISMATCH	BOOL	The two controllers are running different revisions of the same application. In this condition: • A data exchange between the two controllers may not be possible. • A switchover may not be transparent. • Neither controller can be standby. (Default = FALSE)	System
APP_BUILDCHANGE_DIFF	UINT	The number of build change differences between the applications in the primary controller and the standby controller. Evaluated by the primary controller.	System
MAX_APP_ BUILDCHANGE_DIFF	UINT	Maximum number of build change differences permitted by the Hot Standby system, from 0...50 (default = 20). Set in the **Hot Standby** tab as **Number of modifications**.	Application
FW_MISMATCH_ALLOWED	BOOL	Allows mismatched firmware between primary and standby controllers: • TRUE: the standby controller remains in a STANDBY state in case of firmware mismatch. • FALSE (default): the standby controller goes into WAIT state in case of firmware mismatch. (Default = FALSE)	Application
FW_MISMATCH	BOOL	The firmware are different in the two controllers. (Default = FALSE)	System

Element	Type	Description	Written by
DATA_LAYOUT_MISMATCH	BOOL	The data layout are different on the two controllers. The data transfer is partially performed. (Default = FALSE)	System
DATA_DISCARDED	UINT	Number of KB sent by the primary and discarded by the standby (rounded up to the next KB). Represents data for variables added to primary, but not to standby. (Default = 0)	System
DATA_NOT_UPDATED	UINT	Number of KB not updated by the standby (rounded up to the next KB). Represents variables deleted from the primary that remain in the standby. (Default = 0)	System
BACKUP_APP_MISMATCH	BOOL	• FALSE (default): The backup application in the two Hot Standby controllers are equal. **NOTE:** The backup application resides in flash memory or on the SD memory card of the controller. It is created either by the **PLC > Project Backup... > Backup Save** command, or by setting the %S66 system bit (Application Backup) to 1. • TRUE: All other cases.	System
PLCA_ONLINE	BOOL	Controller A is configured to enter the primary or standby state. (Default = TRUE) **NOTE:** Executable only on controller A.	Configuration
PLCB_ONLINE	BOOL	Controller B is configured to enter the primary or standby state. (Default = TRUE) **NOTE:** Executable only on controller B.	Configuration
CMD_SWAP	BOOL	• Set to 1 by program logic or animation table to initiate a switchover. The primary controller goes into WAIT state; then the standby controller becomes primary controller. Finally, the controller in WAIT state goes standby. The command is ignored if there is no standby. **NOTE:** Executable on both primary and standby controllers. • Reset to 0 (default) by the system on switchover completion or if there is no standby controller. **NOTE:** • This command is designed to be used by the application in response to detected errors. It is not intended to be used for periodic switchovers. • If the application has to switchover periodically, the period between switchovers must not be less than 120 seconds.	Application / System

Element	Type	Description	Written by
CMD_APP_TRANSFER	BOOL	• Set to 1 by program logic or animation table to start an application transfer from the primary controller to the standby. Executable only on the primary controller. **NOTE:** The application transferred is the backup application, stored in flash memory or on the SD card. If the application running does not match the backup application, perform an application backup (**PLC > Project Backup... > Backup Save** or set the %S66 system bit to 1) before performing the transfer. • Reset to 0 (default) by the system on transfer completion.	Application / System
CMD_RUN_AFTER_ TRANSFER	BOOL	• Set to 1 by program logic or animation table to automatically start in Run after a transfer. **NOTE:** Executable only on the primary controller. • Reset to 0 (default) by the system after transfer completion. ◦ Remote controller is in Run ◦ Controller is not primary	Application / System
CMD_RUN_REMOTE	BOOL	• Set to 1 by program logic or animation table to run the remote controller. This command is ignored if the CMD_STOP_REMOTE is TRUE. **NOTE:** Executable only on the primary. • Reset to 0 (default) by the system when the remote controller enters standby or wait state.	Application / System
CMD_STOP_REMOTE	BOOL	• Set to 1 by program logic or animation table to stop the remote controller. **NOTE:** Executable on the primary, the standby, or a stopped controller. • Reset to 0 (default) by the system at the end of the stop command	Application
CMD_COMPARE_INITIAL_ VALUE	BOOL	• Set to 1 by program logic or animation table to begin a comparison of the initial values of variables exchanged by the two Hot Standby controllers. **NOTE:** Executable on both primary and standby controllers, only in Run mode. • Reset to 0 (default) by the system when the comparison is complete, or if the comparison is not possible.	Application / System
INITIAL_VALUE_MISMATCH	BOOL	• TRUE if the initial values for exchanged variables are different or if the comparison is not possible. • FALSE if the initial values for exchanged variables are identical.	System

Element	Type	Description	Written by
MAST_SYNCHRONIZED (1)	BOOL	• TRUE if the exchanged data from the previous MAST cycle was received by the standby controller. • FALSE (default) if the exchanged data from at least the previous MAST cycle was not received by the standby controller. **NOTE:** Closely monitor the MAST_ SYNCHRONIZED and FAST_SYNCHRONIZED variables related to the MAST and FAST tasks as indicated at the end of this table.	System
FAST_SYNCHRONIZED (1)	BOOL	• TRUE if the exchanged data from the previous FAST cycle was received by the standby controller. • FALSE (default: if the exchanged data from at least the previous FAST cycle was not received by the standby controller. **NOTE:** Closely monitor the MAST_ SYNCHRONIZED and FAST_SYNCHRONIZED variables related to the MAST and FAST tasks as indicated at the end of this table.	System
SAFE_SYNCHRONIZED(1)	BOOL	• TRUE if the exchanged data from the last SAFE cycle was received by the standby controller. • FALSE (default) if, at least, the exchanged data from the last SAFE cycle was not received by the standby controller.	System
SAFETY_LOGIC_ MISMATCH	BOOL	• TRUE. The SAFE logic part of the application is different in the two controllers. • FALSE (default). The SAFE logic part of the application is identical in the two controllers. **NOTE:** • The content for this element is determined by comparing system word %SW169 for each controller. • This element is included in T_M_ECPU_ HSBY_EXT DDT version 2.80 and later.	–
LOCAL_HSBY_STS	T_M_ ECPU_ HSBY_STS	Hot Standby status for the local controller	(see T_M_ ECPU_ HSBY_STS Data Type, page 145)

Element	Type	Description	Written by
REMOTE_HSBY_STS	T_M_ECPU_HSBY_STS	Hot Standby status for the remote controller	(see T_M_ECPU_HSBY_STS Data Type, page 145)

(1):

- Closely monitor the MAST_SYNCHRONIZED, FAST_SYNCHRONIZED, and SAFE_SYNCHRONIZED variables related to the MAST, FAST and SAFE tasks. If its value is zero (FALSE), then the database exchanged between the primary and the standby controllers is not transmitted at each cycle. In this situation, change the configured period of this task with a higher value than its current execution time (for the MAST task: %SW0 > %SW30; for the FAST task %SW1 > %SW33; for the SAFE task %SW4 > %SW42. More details on %SW0, %SW1 and %SW30, %SW31 can be found in EcoStruxure™ Control Expert, System Bits and Words, Reference Manual).
- Example of consequence: upon an Application Program Transfer (APT) command, the primary controller might not be able to transfer the program to the standby controller.

T_M_ECPU_HSBY_STS Data Type

The `T_M_ECPU_HSBY_STS` / `T_M_ECPU_HSBY_STS_EXT` data type presents the following elements.

NOTE: For firmware version 2.80 and later, the T_M_ECPU_HSBY_STS DDT is named T_M_ECPU_HSBY_STS_EXT.

Element	Type	Description	Written by
HSBY_LINK_ERROR	BOOL	• TRUE: No connection on the Hot Standby link. • FALSE: The Hot Standby link is operational.	System
HSBY_SUPPLEMENTARY_LINK_ERROR	BOOL	• TRUE: No connection on the Ethernet RIO link. • FALSE: The Ethernet RIO link is operational.	System
WAIT	BOOL	• TRUE: The controller is in Run state but waiting to go primary or standby. • FALSE: The controller is in standby, primary, or stop state.	System
RUN_PRIMARY	BOOL	• TRUE: The controller is in primary state. • FALSE: The controller is in standby, wait, or stop state.	System
RUN_STANDBY	BOOL	• TRUE: The controller is in standby state. • FALSE: The controller is in primary, wait, or stop state.	System
STOP	BOOL	• TRUE: The controller is in stop state. • FALSE: The controller is in primary, standby, or wait state.	System

Element	Type	Description	Written by
PLC_A	BOOL	• TRUE: The controller **A/B/Clear** rotary switch, page 151 is in **A** position. • FALSE: The controller switch is not in **A** position.	System
PLC_B	BOOL	• TRUE: The controller **A/B/Clear** rotary switch, page 151 is in **B** position. • FALSE: The controller switch is not in **B** position.	System
EIO_ERROR	BOOL	• TRUE: The controller does not detect any of the configured Ethernet RIO drops. • FALSE: The controller detects at least one configured Ethernet RIO drop. **NOTE:** This bit is FALSE when no drop is configured.	System
SD_CARD_PRESENT	BOOL	• TRUE: A valid SD card is inserted. • FALSE: No SD card, or an invalid SD card is inserted.	System
LOCAL_RACK_STS	BOOL]	• TRUE: The local rack configuration is OK. • FALSE: The local rack configuration is not OK (for example, modules missing or in incorrect slots, etc.)	Application
MAST_TASK_STATE	BYTE	State of the MAST task: • 0: Non-existent • 1: Stop • 2: Run • 3: Breakpoint • 4: Halt	System
FAST_TASK_STATE	BYTE	State of the FAST task: • 0: Non-existent • 1: Stop • 2: Run • 3: Breakpoint • 4: Halt	System

Element	Type	Description	Written by
SAFE_TASK_STATE	BYTE	State of the SAFE task: • 0: Non-existent • 1: Stop • 2: Run • 3: Breakpoint • 4: Halt **NOTE:** This element is included in T_M_ECPU_HSBY_STS_EXT DDT version 2.80 and later.	System
REGISTER	WORD[0...63]	Unmanaged data added to the application via the **Exchange on STBY** attribute.	Application

Data Storage Elementary Functions

Data Storage Elementary Functions

The following `DataStorage_EF` elementary functions are supported in Control Expert for all tasks in the M580 BMEH58•040 non-safety-related Hot Standby controllers, and for process tasks in the M580 BMEH58•040S safety Hot Standby controllers.

EF	Hot standby controller state		
	Primary	**Standby**	**Wait**
CREATE_FILE	X	X	X
DELETE_FILE	X	X	X
GET_FILE_INFO*	X	X	X
GET_FREESIZE*	X	X	X
OPEN_FILE	X	X	X
RD_FILE_TO_DATA	X	X	X
SET_FILE_ATTRIBUTES	X	X	X
WR_DATA_TO_FILE	X	X	X
* Read-only function			

NOTE: Changes made to an SD card in either the primary or standby controller, using an elementary function, are not replicated in the SD card of the other controller in the event of a switchover.

CREATE_FILE

The CREATE_FILE (see EcoStruxure™ Control Expert, System, Block Library) function creates a file called *FILENAME*, if it does not already exist. If a file by that name already exists, the CREATE_FILE command behaves the same as the OPEN_FILE command.

DELETE_FILE

The DELETE_FILE (see EcoStruxure™ Control Expert, System, Block Library) function deletes a file identified by its *FILENAME*. Close a file, using the CLOSE_FILE function before deleting it.

GET_FILE_INFO

The GET_FILE_INFO (see EcoStruxure™ Control Expert, System, Block Library) function retrieves information about a specified target file. Execute the OPEN_FILE function for the target file before executing the GET_FILE_INFO function, because the identity of the target file comes from the output parameter of the OPEN_FILE block.

GET_FREESIZE

The GET_FREESIZE (see EcoStruxure™ Control Expert, System, Block Library) function displays the amount of available space on the SD memory card.

OPEN_FILE

The OPEN_FILE (see EcoStruxure™ Control Expert, System, Block Library) function opens a specified file, provided the file already exists.

RD_FILE_TO_DATA

The RD_FILE_TO_DATA (see EcoStruxure™ Control Expert, System, Block Library) function allows data to be read from a file, at the current position of the file, and enables it to be copied to a variable.

SET_FILE_ATTRIBUTES

The `SET_FILE_ATTRIBUTES` (see EcoStruxure™ Control Expert, System, Block Library) function enables the setting of file attributes that set or clear the read-only flag for that file.

WR_DATA_TO_FILE

The `WR_DATA_TO_FILE` (see EcoStruxure™ Control Expert, System, Block Library) function writes the value of a specified variable to the selected file. The data written is added after the current position in the file.

M580 Hot Standby System Operation

Overview

This chapter describes operation of the M580 Hot Standby system.

Starting an M580 Hot Standby System

Preconditions

During the start-up sequence, each controller is assigned a Hot Standby state (Primary, Standby, or Wait) according to the:

- State of the Ethernet remote I/O network
- State of the Hot Standby link
- **A/B/Clear** rotary switch, page 151
- Operating state (Run or Stop) of the controller

On initial start-up, confirm that the:

- Hot Standby link is connected.
- Controller you start first has been fully programmed.
- **A/B/Clear** rotary switch, page 151 on the back of the two Hot Standby controllers are set to different positions: one to **A**, the other to **B**.

  NOTE: The first controller to power up becomes the primary controller, regardless of its designation as A or B.

Starting the Hot Standby System

The following chart provides the appropriate steps for starting your Hot Standby system.

Step	Action
1	Turn on power to the first backplane. **NOTE:** In this example, this is the backplane with the controller **A/B/Clear** rotary switch, page 151 set to **A**.
2	Connect your PC with both Control Expert and the program you want to download.

Step	Action		
3	Download the program to the controller.		
4	Start the controller in that backplane. If the necessary preconditions exist, the controller becomes the primary Hot Standby controller.		
5	Turn on power to the second backplane. **NOTE:** In this example, this is the backplane with the controller **A/B/Clear** rotary switch, page 151 set to **B**.		
6	If necessary, repeat steps 2 and 3 for the second controller. **NOTE:** If the second controller is not configured, the primary controller automatically downloads the program to the second controller, which becomes the standby.		
7	Start the second controller.		
8	Verify the LED display for each controller. If both controllers are operating as intended, the LEDs will appear as follows:		
	LED	**First Controller** (A)	**Second controller** (B)
	RUN	Solid Green	Solid Green
	REMOTE RUN	Solid Green	Solid Green
	ETH MS	Solid Green	Solid Green
	ETH NS	Solid Green	Solid Green
	A	Solid Green	OFF
	B	OFF	Solid Green
	PRIM	Solid Green	OFF
	STBY	OFF	Solid Green
	SRUN (safety-related controller)	Solid Green	Solid Green
	SMOD (safety-related controller)	Solid Green	Solid Green

NOTE: For a description of:

- BMEH58•040 controller LEDs, refer to *LED Diagnostics*, page 187.
- Startup states of the BMEH58•040 controller, refer to *Hot Standby State Assignments*, page 153.

A/B/Clear Rotary Switch Role Assignment

The **A/B/Clear** rotary switch assignment does not by itself determine the Hot Standby primary or standby role of a controller. Typically, the first controller to power up becomes the primary controller, regardless of its designation as A or B; the secondary controller to power up becomes the standby.

The **A/B/Clear** rotary switch settings determine the role of a controller only in the case of a simultaneous power up. In that case:

- The controller set to **A** becomes primary.
- The controller set to **B** becomes secondary.

Conflicting A/B/Clear Rotary Switch Role Assignment

If you set the **A/B/Clear** rotary switch, page 151 to the same setting – **A** or **B** – for both Hot Standby controllers, the first controller to power up becomes the primary, and the second controller to power up enters wait state.

If you set the **A/B/Clear** rotary switch to **Clear** for both controllers, both controllers remain non-configured.

This condition can be determined by examining the following LEDs for each controller:

If both controller switches are set to:	LED	First controller to power-up	Second controller to power-up
A	A	Flashing Green	Flashing Green
	B	OFF	OFF
	PRIM	Flashing Green	OFF
	STBY	OFF	OFF
B	A	OFF	OFF
	B	Flashing Green	Flashing Green
	PRIM	Flashing Green	OFF
	STBY	OFF	OFF
Clear	A	Flashing Green	Flashing Green
	B	Flashing Green	Flashing Green
	PRIM	OFF	OFF
	STBY	OFF	OFF

NOTE: If the **A/B/Clear** rotary switch for both controllers is set to the same position (**A** or **B**), and if both controllers start-up simultaneously, both controllers enter the WAIT state.

Hot Standby State Assignments and Transitions

Hot Standby State Assignments

The purpose of assigning start-up states to Hot Standby controllers is to avoid the situation where two controllers simultaneously assume the role of primary and simultaneously attempt to drive the state of remote outputs. Assignment of the primary and secondary roles for controllers is determined by the following factors:

- The health of the Hot Standby link between the controllers.
- The health of the Ethernet link between the controllers over the Ethernet RIO main ring.
- The existence of one or more Ethernet connections between each controller and configured devices via the Ethernet RIO main ring.
- The online state, page 120 of controller A and controller B.
- The **A/B/Clear** rotary switch, page 151 position on the rear of the controller.
- The controller state (RUN or STOP).

The following matrix describes Hot Standby state assignments for paired controllers during several start-up and run-time scenarios:

Network preconditions				**Initial state**		**Final state**	
EIO link[1]	**RIO device connections[2]**		**Hot Standby link**	**Controller_A**	**Controller_B**	**Controller_A**	**Controller_B**
	Controller_A	**Controller_B**					
OK	OK	OK	OK	Starting	Starting	Run Primary[3]	Run Standby
OK	OK	Not OK	OK	Starting	Run Primary	Run Primary[4]	Wait
OK	Not OK	OK	OK	Starting	Starting	Wait	Run Primary[4]
OK	OK	OK	OK	Run Primary	Starting	Run Primary	Run Standby
OK	OK	OK	OK	Starting	Run Primary	Run Standby	Run Primary
OK	OK	OK	Not OK	Run Primary	Starting	Run Primary	Wait
OK	OK	OK	Not OK	Starting	Starting	Run Primary	Wait
OK	OK	OK	Not OK	Starting	Run Primary	Wait	Run Primary

Network preconditions				Initial state		Final state	
EIO link[1]	RIO device connections[2]		Hot Standby link	Controller_ A	Controller_ B	Controller_ A	Controller_B
	Controller_A	Controller_B					
OK	Not OK	Not OK	OK	Starting	Starting	Run Primary	Run Standby
OK	Not OK	Not OK	OK	Run Primary	Starting	Run Primary	Run Standby
OK	Not OK	Not OK	OK	Starting	Run Primary	Run Standby	Run Primary
Not OK	Not OK	Not OK	OK	Starting	Starting	Run Primary	Run Standby
Not OK	Not OK	Not OK	OK	Run Primary	Starting	Run Primary	Run Standby
Not OK	Not OK	Not OK	OK	Starting	Run Primary	Run Standby	Run Primary
Not OK	OK	OK	Not OK	Starting	Starting	Run Primary	Run Primary
Not OK	OK	OK	Not OK	Run Primary	Starting	Run Primary	Run Primary
Not OK	OK	OK	Not OK	Starting	Run Primary	Run Primary	Run Primary
Not OK	Not OK	Not OK	Not OK	Starting	Starting	Run Primary[3]	Run Primary[3]
Not OK	Not OK	Not OK	Not OK	Run Primary	Starting	Run Primary[3]	Run Primary[3]
Not OK	Not OK	Not OK	Not OK	Starting	Run Primary	Run Primary[3]	Run Primary[3]

1.The supplementary link between controller A and controller B over the RIO or DIO ring.

2. The connection between a controller and RIO drop over the ERIO network. OK indicates the controller recognizes at least one drop. Not OK indicates the controller recognizes no drops for 3 seconds.

3. Priority is given to controller designated **A** via the **A/B/Clear** rotary switch on the rear of the controller.

4. Priority is given to controller that recognizes at least one RIO drop.

Hot Standby Controller State Transitions During Operations

A controller in a Hot Standby system transitions between states in the following circumstances:

Transition	This transition occurs when...
Wait to Standby	All of the following exist: • Controller is in RUN state. • Controller is operating online, page 120. • Connected to a primary Controller via a Hot Standby link. • All other preconditions for standby state exists, for example: ◦ Firmware mismatch is allowed, if a firmware mismatch exists. ◦ Logic mismatch is allowed, if a logic mismatch exists. ◦ Online modifications are allowed, if modifications have been made.
Wait to Primary	All of the following exist: • Controller is operating online, page 120. • Controller is allowed to enter primary state (Controller transitions from STOP to RUN, or warm start in RUN). • Controller is controlling the Ethernet RIO link, or connected via the Hot Standby link to a counterpart controller that is not in RUN state.
Standby to Primary	One of the following exists: • The counterpart controller enters wait or standby state. • Communication with the counterpart controller is interrupted on both the Ethernet RIO link and the Hot Standby link. • The counterpart controller is in primary state and receives a switchover command.
Standby to Wait	The following exists: • Communication is interrupted with the counterpart controller over the Hot Standby link for more than 3 seconds. • The ERIO link between the 2 controllers remains OK. • Online modification mismatch is not allowed, if modifications have been made. • Firmware update is not allowed, if a firmware update exists. • For safety-related controllers only: Online modification mismatch is allowed, if modifications have been made in the safety-related part of the application (SAFETY_LOGIC_MISMATCH = 1) and maintenance mode has not been set on either the Primary controller or Standby controller (i.e. each controller is operating in safety mode).

Transition	This transition occurs when...
Primary to Wait	One of the following exists: • The controller has lost communication with all (e)X80 EIO adapter modules, and the secondary controller is in standby state and continues to communicate with at least one (e)X80 EIO adapter module. • The controller is designated **B** via the **A/B/Clear** rotary switch, page 151, and the secondary controller (also designated as **B**) is in primary state.
Primary to Standby[1]	One of the following exists: • During operations, all of the following occur: ◦ The primary controller is disconnected from all (e)X80 EIO adapter modules. ◦ The standby controller remains connected to at least one (e)X80 EIO adapter module. ◦ The Hot Standby link between controller A and controller B remains healthy. • The primary is in Halt (because at least one task is in Halt) and the secondary controller is in Standby state with all tasks in RUN. • The primary controller receives a switchover command, and the secondary controller is in standby state. • If other preconditions for standby state exists, for example: ◦ Firmware mismatch is allowed, if a firmware mismatch exists. ◦ Logic mismatch is allowed, if a logic mismatch exists. ◦ Online modifications are allowed, if modifications have been made.
Primary/Standby/Wait to Stop	• The controller transitions from RUN to STOP state.
[1] While the controller is switching from Primary to Standby state, the controller will pass to an intermediate Wait state for a duration of at least one cycle.	

Hot Standby System State Examples

Introduction

This topic presents visual examples of several Hot Standby system states. The focus of each example is the condition of the:

- Hot Standby link between controller A and controller B
- Ethernet RIO link between controller A and controller B
- Ethernet RIO connections between each controller and one or more (e)X80 EIO adapter modules over the RIO main ring

In each example, controller A is the module with its **A/B/Clear** rotary switch set to **A**; controller B is the module with its **A/B/Clear** rotary switch set to **B**.

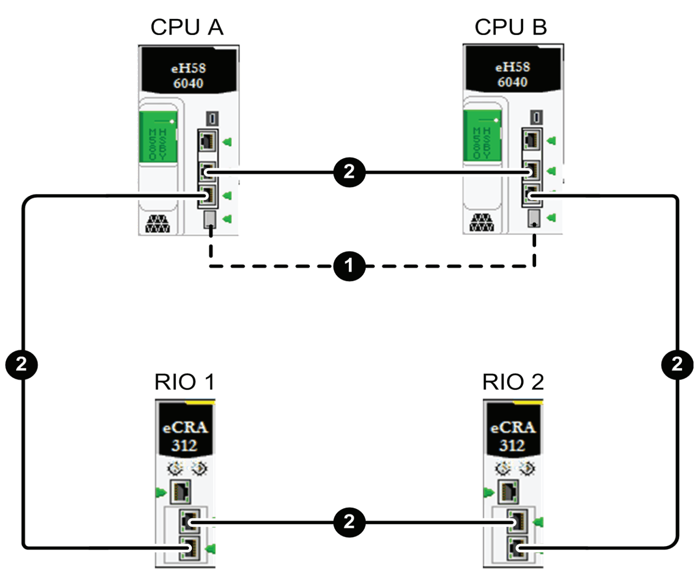
Each example presumes that every other necessary precondition exists for Hot Standby system operation. For example:

- If a firmware mismatch exists, the `FW_MISMATCH_ALLOWED` flag is set.
- If a logic mismatch exists, both the `LOGIC_MISMATCH_ALLOWED` flag and the **Online modification in RUN or STOP** parameter are set.
- For safety-related controllers only: If a logic mismatch and safety-related logic mismatch exist, the `LOGIC_MISMATCH_ALLOWED` flag, the **Online modification in RUN or STOP** parameter and the Maintenance mode are set.

Communication Links Operational for both Controllers

In this example, Hot Standby system connections are operational:

Communication link	Controller A	Controller B
Hot Standby link between controller A and controller B	OK	OK
Ethernet RIO link between controller A and controller B	OK	OK
Ethernet RIO connections between controller and one or more (e)X80 EIO adapter modules	OK	OK



1 Hot Standby fiber optic link between controller A and controller B

2 Ethernet RIO main ring

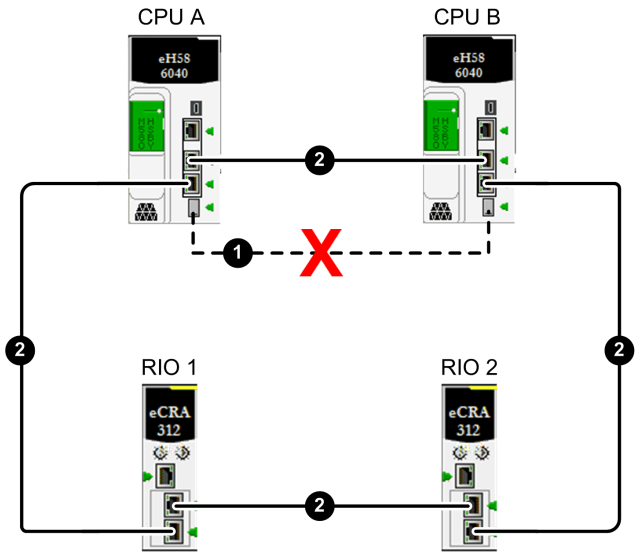
In this example, controller A and controller B enter the following Hot Standby states:

If this Hot Standby system state arises during:	Controller A and Controller B perform the following roles:
Sequential start-up of controller A and controller B	• The first controller to start up is primary. • The second controller to start up is standby.
Simultaneous start-up of controller A and controller B	• controller A is primary. • controller B is standby.
Run-time	• The primary controller remains primary. • The standby controller remains standby.

Hot Standby Link is Not Operational for both Controllers

In this example, the Hot Standby link is not operational in both directions, from controller A to controller B and from controller B to controller A. The other Hot Standby system connections are functioning:

Communication link	Controller A	Controller B
Hot Standby link between controller A and controller B	Not OK	Not OK
Ethernet RIO link between controller A and controller B	OK	OK
Ethernet RIO connections between controller and one or more (e)X80 EIO adapter modules	OK	OK



1 Hot Standby fiber optic link between controller A and controller B

2 Ethernet RIO main ring

X Indicates a broken communication link

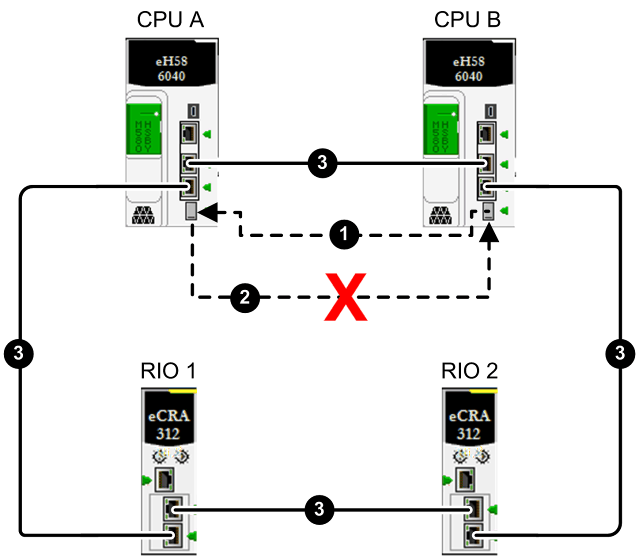
In this example, controller A and controller B enter the following Hot Standby states:

If this Hot Standby system state arises during:	Controller A and Controller B perform the following roles:
Sequential start-up of controller A and controller B	• The first controller to start up is primary. • The second controller to start up enters wait state, because there can be no standby controller if the Hot Standby link is not operational.
Simultaneous start-up of controller A and controller B	• Controller A is primary. • Controller B enters wait state.
Run-time	• The primary controller remains primary. • The standby controller enters wait state.

Hot Standby Link is Not Operational on Only one Controller

In this example, a one-directional break exists in the fiber optic cable used to implement the Hot Standby link. Controller A receives transmissions from controller B over the Hot Standby link, but controller B does not receive transmissions from controller A over the link. The Ethernet RIO connections are operational for both controllers:

Communication link	Controller A	Controller B
Hot Standby link between controller A and controller B	OK	Not OK
Ethernet RIO link between controller A and controller B	OK	OK
Ethernet RIO connections between controller and one or more (e)X80 EIO adapter modules	OK	OK



1 Operational Hot Standby fiber optic link from controller B to controller A

2 Broken Hot Standby fiber optic link from controller A to controller B

3 Ethernet RIO main ring

X Indicates a broken communication link

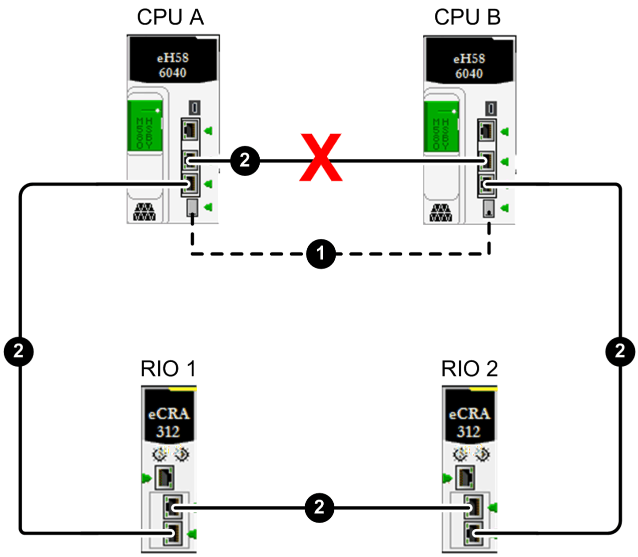
In this example, controller A and controller B enter the following Hot Standby states:

If this Hot Standby system state arises during:	Controller A and Controller B perform the following roles:
Sequential start-up of controller A and controller B	• The first controller to start up is primary. • When controller A starts up (after controller B), it is standby. • When controller B starts up (after controller A), it enters wait state.
Simultaneous start-up of controller A and controller B	• Controller A is primary. • Controller B enters wait state.
Run-time	• Controller A remains primary and controller B enters wait state. – or – • Controller B remains primary and controller A remains standby.

One Break in the Ethernet RIO Main Ring

In this example, a single break exists in the Ethernet RIO main ring. Although the break occurs in the segment between the two controllers, in this example, the break could be located at any point along the Ethernet RIO main ring (2). The other Hot Standby system connections are functioning:

Communication link	Controller A	Controller B
Hot Standby link between controller A and controller B	OK	OK
Ethernet RIO link between controller A and controller B	OK[1]	OK[1]
Ethernet RIO connections between controller and one or more (e)X80 EIO adapter modules	OK	OK
1. RSTP calculates and implements a redundant path between controller A and controller B in case of a single break in the Ethernet RIO main ring.		



1 Hot Standby fiber optic link between controller A and controller B

2 Ethernet RIO main ring

X Indicates a broken communication link

In this example, controller A and controller B enter the following Hot Standby states:

If this Hot Standby system state arises during:	Controller A and Controller B perform the following roles:
Sequential start-up of controller A and controller B	• The first controller to start up is primary. • The second controller to start up is standby.
Simultaneous start-up of controller A and controller B	• Controller A is primary. • Controller B is standby.
Run-time	• The primary controller remains primary. • The counterpart controller remains standby.

Two Breaks in the Ethernet RIO Main Ring Isolate One Controller

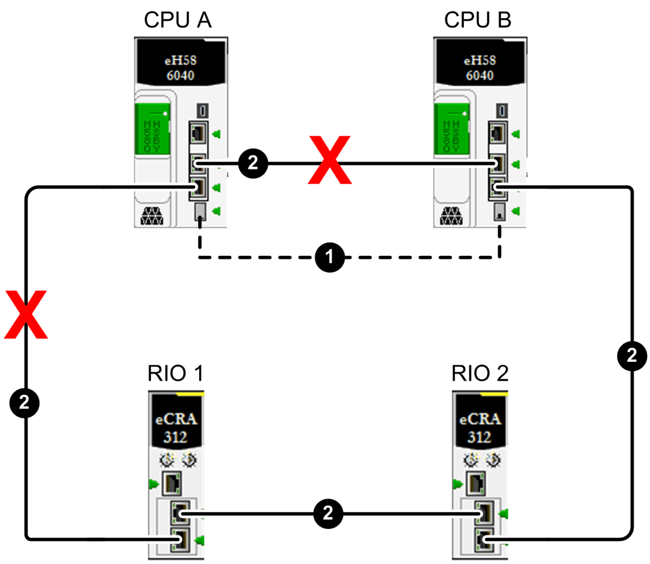
In this example, two breaks in the Ethernet RIO main ring have the following effects:

- Interruption of the Ethernet RIO link between the controllers

- Isolation of controller A from the (e)X80 EIO adapter modules on the Ethernet RIO main ring

The Hot Standby link remains operational.

Communication link	Controller A	Controller B
Hot Standby link between controller A and controller B	OK	OK
Ethernet RIO link between controller A and controller B	Not OK	Not OK
Ethernet RIO connections between controller and one or more (e) X80 EIO adapter modules	Not OK	OK



1 Hot Standby fiber optic link between controller A and controller B

2 Ethernet RIO main ring

X Indicates a broken communication link

In this example, controller A and controller B enter the following Hot Standby states:

If this Hot Standby system state arises during:	Controller A and Controller B perform the following roles:
Sequential start-up of controller A and controller B	• Controller A starts up as primary. • Controller B starts up as standby.
Simultaneous start-up of controller A and controller B	• Controller A is primary. • Controller B is standby.
Run-time	• Controller B remains or becomes primary. • Controller A enters standby state.

This example occurs due to a double RIO cable break. (The first error was not detected or not treated.) The M580 Hot Standby system is not multi-RIO cable break-tolerant. Instead, the primary controller (A) isolates from the RIO drops, and the standby controller (B) can still view the primary controller and, cannot take control. Controller A must check all drops before surrendering its primary role and during this phase, may read default input values (flagged by input or drop health diagnostics), which are transferred to the standby controller (B) and reused by controller B when it becomes primary.

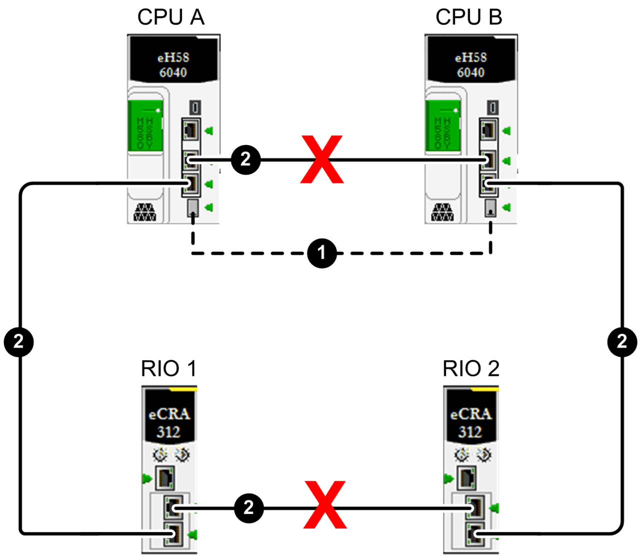
To summarize:

- Consider the health diagnostics when you design the logic.
- Perform maintenance as soon as possible when a first error is detected.
- Delay the last valid value of the inputs in the logic if this type of scenario is required.

Two Ethernet RIO Main Ring Breaks Cause Controllers to be Connected to Different Sets of Ethernet RIO Devices

In this example, two breaks exist in the Ethernet RIO main ring, causing the interruption of the Ethernet RIO link between controller A and controller B. The location of the breaks cause each controller to be connected to a different collection of (e)X80 EIO adapter modules on the Ethernet RIO main ring. The Hot Standby link remains operational:

Communication link	Controller A	Controller B
Hot Standby link between controller A and controller B	OK	OK
Ethernet RIO link between controller A and controller B	Not OK	Not OK
Ethernet RIO connections between controller and one or more (e)X80 EIO adapter modules	OK	OK



1 Hot Standby fiber optic link between controller A and controller B

2 Ethernet RIO main ring

X Indicates a broken communication link

In this example, controller A and controller B enter the following Hot Standby states:

If this Hot Standby system state arises during:	Controller A and Controller B perform the following roles:
Sequential start-up of controller A and controller B	• The first controller to start up is primary. • The second to start up is standby.
Simultaneous start-up of controller A and controller B	• Controller A is primary. • Controller B is standby.
Run-time	• The primary controller remains primary. • The standby controller remains standby.

Executing Hot Standby Commands

Introduction

This topic shows you how to execute Hot Standby commands for an M580 BMEH58•040 or BMEH58•040S controller. Hot Standby commands can be executed using:

- The Control Expert graphical user interface controller configuration screens, which include:
  - The **Task** tab of the **Animation** window.
  - The **Hot Standby** window.
- The `T_M_ECPU_HSBY` and `T_M_ECPU_HSBY_STS` DDTs, which can be called using:
  - Program logic.
  - An **Animation Table**, where you can use the **Force** and **Modification** commands.

  NOTE: The M580 Hot Standby system does not support the use of the Quantum Hot Standby elementary function blocks (EFBs), including: `HSBY_RD`, `HSBY_ST`, `HSBY_WR` and `REV_XFER`. Instead, these functions are directly managed by DDDT commands.

For information on how to operate the non-Hot Standby functions for the controller, refer to the *M580 Hardware Reference Manual* (see Modicon M580, Hardware, Reference Manual).

Hot Standby Commands

You must confirm that the standby controller is ready to assume the primary role before executing a switchover command.

Verify that the value of the REMOTE_HSBY_STS.EIO_ERROR bit of the standby controller is 0 before you execute a switchover command (either by application logic or in Control Expert).

Refer to the *EcoStruxure™ Control Expert Program Languages and Structure Reference Manual* (see EcoStruxure™ Control Expert, System Bits and Words, Reference Manual) for more details on the %SW182-%SW183 and %SW176-%SW177 system words.

The M580 BMEH58•040 and BMEH58•040S controllers support the following Hot Standby commands:

Command	Description	Executable on Primary or Standby	Supported by:	
			DDDT	GUI
CMD_APP_TRANSFER[4]	Transfers the application in the primary controller to the standby controller. **NOTE:** The backup application resides in the flash memory or on the SD memory card of the controller. It is created either by the **PLC > Project Backkup... > Backup Save** command, or by setting the %S66 system bit (Application Backup) to 1.	Both	X	X
CMD_COMPARE_INITIAL_VALUE	Compares the initial values of variables included in the Hot Standby data exchange.	Both (in RUN mode)	X	–
CMD_RUN_AFTER_TRANSFER	Places the primary controller into RUN operating mode upon completion of transfer of application to standby controller.	Primary only	X	–
CMD_RUN_REMOTE	Places the remote[1] controller into RUN operating mode.	Primary only	X	X[3]
CMD_STOP_REMOTE	Places the remote[1] controller into STOP operating mode.	Primary only	X	X[3]
CMD_SWAP	Manually performs a Hot Standby switchover. The primary goes into wait; the standby goes into primary; then the wait goes into standby. This command is designed to be used by the application in response to detected errors. If the application has to switchover periodically, the period between switchovers must not be less than 120 seconds.	Both	X	X[3]
FW_MISMATCH_ALLOWED	When changes have been made to the firmware in the primary controller, this command lets the standby controller continue to operate as standby. If this command is set to 0, the standby goes into wait state.	Primary only	X	–
LOGIC_MISMATCH_ALLOWED[4]	When changes have been made to the application in the primary controller (for example, as a a result of CCOTF changes), this command lets the standby controller continue to operate as standby. If this command is set to 0, the standby goes into wait state.	Primary only	X	–

Command	Description	Executable on Primary or Standby	Supported by:	
			DDDT	GUI
PLCA_ONLINE	Lets the controller with its **A/B/Clear** rotary switch set to **A** serve as either primary or standby, depending on other operating conditions. If set to 0, controller A goes into either wait or stop state.	controller A only	X	X[2]
PLCB_ONLINE	Lets the controller with its **A/B/Clear** rotary switch set to **B** serve as either primary or standby, depending on other operating conditions. If set to 0, controller B goes into either wait or stop state.	controller B only	X	X[2]

X: Command is supported.

–: Command is not supported.

1. *Remote* refers to the controller to which your PC and Control Expert is not connected.

2. In the controller configuration window **Hot Standby** tab.

3. In the controller configuration window **Animation > Task** tab.

4. These commands can be executed only if the remote controller is the standby controller.

Memory Usage

Introduction

The memory usage function is used to view:

- The physical distribution of the controller memory.
- The space occupied in the memory by a project (data, program, configuration, system and diagnostic).

It can also be used to reorganize the memory where possible.

NOTE: The memory usage screen is not available in simulation mode. This screen is only available in standard mode when you have built the application.

Procedure to Display and Optimize the Controller Memory

To access the memory usage details of the controller:

Step	Action
1	Select **PLC > Memory Consumption:**. The **Memory usage** window opens. The memory usage statistics of a project can only be accessed if you have generated its executable in advance.
2	To optimize memory organization, click **Pack**.

NOTE: If the application has been built and if it is in NOT BUILT state due to a program modification, the screen is accessible, but it corresponds to the application built previously. Modifications will be taken into account at the next build.

Description of the parameters

The following information fields are available:

Parameter	Description
User Data	This field indicates the memory space (in words) occupied by user data (objects relating to configuration): • **saved Data**: located data associated with the controller (%M, %MW, %S, %SW, etc.) or the input/output modules. This data is retained by the controller in the event of a controller warm start. • **saved Declared Data**: unlocated data (declared in the data editor) that is retained by the controller in the event of a controller warm start. • **unsaved Declared Data** unlocated data (declared in the data editor) that is not retained by the controller in the event of a controller warm start.
User program	This field indicates the memory space (in words) occupied by the project program: • **Constants**: static constants associated with the controller (%KW) and the input/output modules; initial data values, • **Executable code**: executable code of the project program, EFs, EFBs and DFB types, • **Upload information**: information for uploading a project (graphic code of languages, symbols, etc.).
Other	This field indicates the memory space (in words) occupied by other data relating to the configuration and the project structure: • **Configuration**: other data relating to configuration (Page0 for a Quantum controller, 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....)

Parameter	Description
Internal memory	This field shows the organization of the controller internal memory, for both program and data storage. It indicates the memory space available (**Total**), the largest possible contiguous memory space (**Greatest**) and the level of **Fragmentation** (due to online modifications).
Pack	This command is used to reorganize the memory structure.

Memory re-organization

Memory re-organization is activated using the **Pack** command.

Memory re-organization can be performed in online or offline mode (regardless whether the controller is in Run or in Stop).

NOTE: Certain blocks cannot be moved in online mode. There is less fragmentation by re-organizing the memory in offline mode.

M580 Hot Standby System Performance

Overview

Creating a deterministic Hot Standby system requires the use of network components and designs that support Ethernet communication including:

- Full duplex transmissions
- 100 Mbps transmission speeds
- QoS prioritization of Ethernet packets transmitted over the RIO network

This chapter describes methods to help improve system performance.

TASK_SYNCHRONIZED

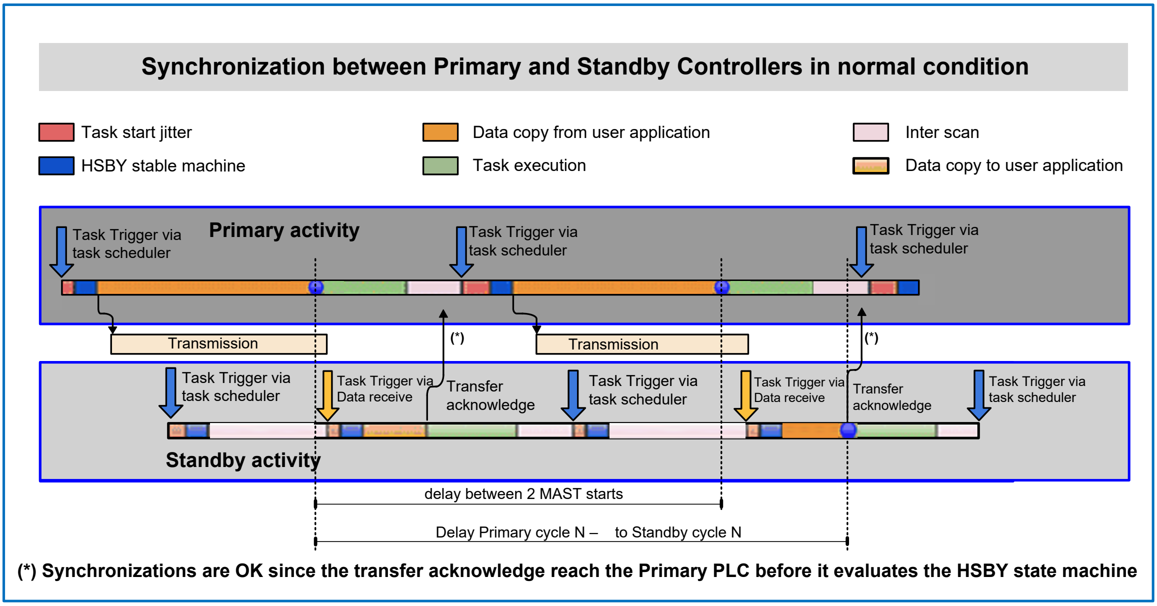
To evaluate system performance, you need to verify that the tasks are synchronized.

Synchronized Task Example

A task is synchronized when the data associated with this task are exchanged at each cycle. The system provides one bit per task:

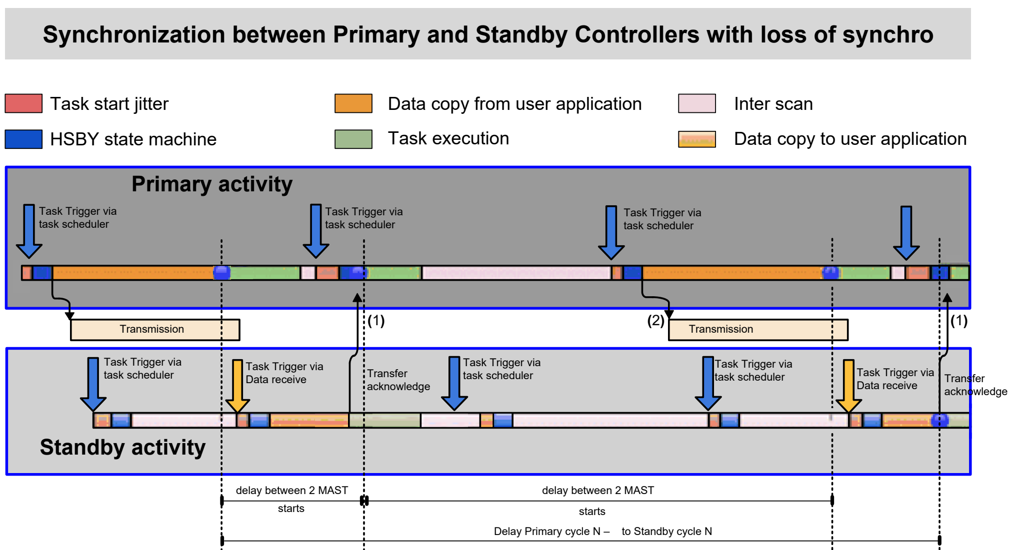
- MAST_SYNCHRONIZED
- SAFE_SYNCHRONIZED
- FAST_SYNCHRONIZED

The following diagram depicts a case of normal task synchronization (limited to one task for simplification):



Non-Synchronized Task Example

A task is not synchronized if the standby controller did not acknowledge receipt of the previous data-set before the next task execution on the primary. The design choice in this case is not to wait for an acknowledgement or a timeout but instead to immediately begin the task execution as required by the configuration. In most cases, the primary will receive the acknowledgement later and will become synchronized with the standby after one missed cycle. The following diagram depicts this non-synchronization case:



(1) Synchronizations are not OK since the transfer acknowledge reach the Primary PLC after it evaluates the HSBY state machine. The next cycle, the primary will not send the data, and reset the Mast_synchronized bit

(2) The next cycle, as the acknowledge was receieved, the primary will send again the data to the standby

When a de-synchronization exists, the primary does not send any data to the standby. The task duration is thus shorter than it should be and impacts the different measures performed by the system (present and minimum task duration, processor load).

Task synchronization is a transient bit of information. It cannot be reliably tested directly with an HMI, a Control Expert animation table or the Control Expert controller status bar. These methods rely on controller communication and sample only a few points in time.

As for the overrun, use a counter or any appropriate program to capture de-synchronizations.

> **NOTE:** Task de-synchronizations are reported in the HSBY DDT and managed in the MAST task. Thus the counters must be implemented in the MAST task.

Task Synchronization Reporting Variations

Task synchronization reporting has changed over time:

- Firmware version 2.7 and earlier:
  - Some incorrect de-synchronizations were reported in rare cases.
  - The FAST_SYNCHRONIZED information was set and reset by the FAST task but captured only in the MAST task. Thus, it was theoretically possible to miss the de-synchronizations that occurred only seldom, even using a counter in MAST task
- Firmware versions 2.8 through 3.1:
  - Incorrect de-synchronizations were no longer reported.
  - Because of this modification, in extreme overrun condition (for example, when the task period was configured at less than half of the data transfer time), the system did not detect part or all of the de-synchronizations. This could happen if the task was in overrun.
  - FAST_SYNCHRONIZED and SAFE_SYNCHRONIZED are reset only when the MAST task acknowledged the capture of the information in the HSBY DDT. It was not possible to produce an exact count of FAST and SAFE de-synchronizations (inherent in the multitasking execution) but using a counter in MAST, it became possible to detect them.
- Firmware version 3.2 and later:
  - Extreme overrun conditions are detected, and related de-synchronizations are correctly reported.

System Performance

Memory Consumption

Input and output memory specification for I/O data:

Scope	Type	Maximum Value per Scan(1)		
		BMEH582040(S)	**BMEH584040(S)**	**BMEH586040(S)**
M580 Hot Standby controller	input bytes per network	16384 bytes(2)	24576 bytes(3)	24576 bytes (3)
	output bytes per network	16384 bytes(2)	24576 bytes(3)	24576 bytes (3)
Ethernet RIO	input words per drop	1400		
	output words per drop	1400		

Scope	Type	Maximum Value per Scan[1]		
		BMEH582040(S)	BMEH584040(S)	BMEH586040(S)
Ethernet DIO	input bytes per device	up to 1400, depending on EtherNet/IP or Modbus/TCP function code		
	output bytes per device	1400		
Total DIO scanning capability	input (KB)	up to 2		
	output (KB)	up to 2		

(1) Hot Standby supports the MAST, FAST, and SAFE tasks. AUX0 and AUX1 are not supported.

(2) Of the total amount, up to 3072 bytes can be FAST task data.

(3) Of the total amount, up to 5120 bytes can be FAST task data.

Displaying I/O Memory Consumption

You can monitor the I/O memory consumption in Control Expert. Use one of these methods:

- In the **Project Browser**, expand **Project > Configuration > EIO Bus**. Right-click **Properties**.

  — or —
- In the background of the **EIO Bus** window, right-click **Bus properties**.

  — or —
- In the **Edit** menu, select **Bus properties**.

Exceeding RIO Drop Limitations

Control Expert detects an **error** and displays it in the log if one of these events occurs:

- The size of the **RIO drop** memory for the MAST task exceeds 1,400 input bytes or 1,400 output bytes.
- The size of the **RIO drop** memory for the FAST task exceeds 1,400 input bytes or 1,400 output bytes.
- The size of the M580 network exceeds 80% of the maximum drop limit, page 48 for the selected controller.

Minimum / Maximum System Channels

The minimum and maximum number of channels that a global M580 configuration can manage is a function of the M580 controller that you are using. For detailed information on configuring channels, refer to the *Modicon M580 Hardware Reference Manual* (see Modicon M580, Hardware, Reference Manual).

OVERRUN

The system provides one OVERRUN status per task, page 185. Verify the OVERRUN status (system bit %S19) at each step of the application life cycle. An OVERRUN condition is not automatically managed by the system. Instead, design the application to verify and reset it for each task, and to execute the desired logic.

A permanent or frequent OVERRUN event is a non-recoverable detected error. It indicates that some tasks (auxiliary tasks and the background task) cannot execute. On the other hand, an infrequent and temporary OVERRUN condition can be acceptable

Calculating the Minimum Task Cycle Time

Introduction

By configuring sufficiently large task cycle times, the controller in your M580 Hot Standby system can process the data produced by the system in a single scan. If the configured task cycle time is smaller than the required processing time, the controller will make the task overrun.

There are two main conditions to be fulfilled, to help avoid over-run events. Use the formulas (set forth below) to compute the minimum task time periods and processor load for your system.

Condition 1: the I/O Configuration

The general formula for a multi task application is:

(# of RIO drops using MAST task/MAST cycle time) +

(# of RIO drops using SAFE task/SAFE cycle time) +

(# of RIO drops using FAST task/FAST cycle time) < 1.5

Where all cycle times are measured in ms.

NOTE: M580 Hot Standby controllers support only the MAST, FAST and SAFE tasks. AUX0 and AUX1 are not supported.

Distributed Equipment:

If DIO devices are configured, increase the minimum cycle time.

Example:

In this example, the configuration consists of:

- a local rack with a controller with Ethernet I/O scanner service, using only the MAST task
- 10 RIO drops

The minimum MAST cycle time equals:

$10 / T_{MAST} < 1.5$

or

$T_{MAST} > 6.7$ ms

Condition 2: The Processor Load

Each application task uses a percentage of the processor resources, referred to here as the task bandwidth ($Task_{BW}$).

A way to verify the controller processor bandwidth consumption is to add together the load contributed by each task. Limit total consumption to 80%. Above this percentage, task jitter risks creating an OVERRUN condition which can:

- De-synchronize the standby controller, and
- Interrupt communication to both the standby controller and the I/O.

For each task, the load contribution is the sum of:

- The execution.
- The communication (MAST only).
- The transfers (SAFE task and/or HSBY) .

The bandwidth for user-configured tasks depends on:

- The user code executed at each cycle.
- The task itself.
- The size of the data associated with the task.
- The configured task period.

The controller manages a set of user and system tasks and schedules them using a fixed priority strategy. It is necessary to reserve a minimum of processor bandwidth to permit the controller to behave as expected and to be able to handle other seldom-performed intermittent operations, for example, online modifications. This condition is described by the expression:

$Mast_{BW} + Safe_{BW} + Fast_{BW} < 0.8$

Task Bandwidth Evaluation:

The task bandwidth evaluation ($Task_{BW}$) is the ratio of the processor resources that are used by a task per time unit. To be evaluated, it is necessary to know or evaluate the following information:

1. The task execution time (T_{EXE}) of each task on the target. It can be measured with Control Expert under the following conditions:
   - The controller is not connected to a standby controller. (See Note 1, below.)
   - The task periods are tuned to their highest possible setting to limit the task interference.

   T_{EXE} = minimum execution time. (See Note 2, below.)
2. The task period (T_{PER}), which is a user-configured setting. Use the configured task period, and do not adjust the value in the application program, because some communication time-outs can be automatically set by the Control Expert configuration software, based upon the configured task period values.

   T_{PER} = the period planned to be used during normal operation.
3. For a Hot Standby controller, the time needed to transfer (T_{TFR}) the data with the remote controller, for each task, needs to be added. This is given as:
   - For the MAST and FAST tasks:

     $T_{TFR} = (K1 \times Task_{KB} + K2 \times Task_{DFB}) / 1000$
   - For the SAFE task:

     $T_{TFR} = (K1 \times Task_{KB} + K2 \times Task_{DFB}) / 500$

   In each case, T_{TFR} is measured in milliseconds.

In the preceding formulas, K1 and K2 are constants, with values determined by the specific controller used in the application:

Coefficient	BMEH582040S	BMEH584040S or BMEH586040S
K1	32.0	10.0
K2	23.6	7.4

The task bandwidth is described using the following formula:

$Task_{BW} = (T_{EXE} + T_{TFR}) / T_{PER}$

NOTE:

1. If the controller is connected to the standby, then the execution time provided by Control Expert includes the transfer time and is equal to $T_{EXE} + T_{TFR}$. Nevertheless, the measure may be more unstable, and may be affected by the lack of task synchronization. (Check the MAST, FAST, and SAFE synchronization bits in the T_M_ECPU_HSBY DDT).
2. Some applications have an unstable execution time because the code executed is slightly different from one task cycle to the next. There is no general rule to determine which value for minimum task bandwidth should be considered in this case.

Application Response Time for M580 Hot Standby Controllers

Introducing Application Response Time

Each Ethernet RIO input signal packet travels from an RIO drop to the controller, and the controller sends an output signal back to the RIO drop. The time it takes for the controller to receive the input signal and effect a change in the output module based on the input is called Application Response Time (ART).

In an M580 system, ART is deterministic, which means you can calculate the maximum time the controller uses to resolve an RIO logic scan.

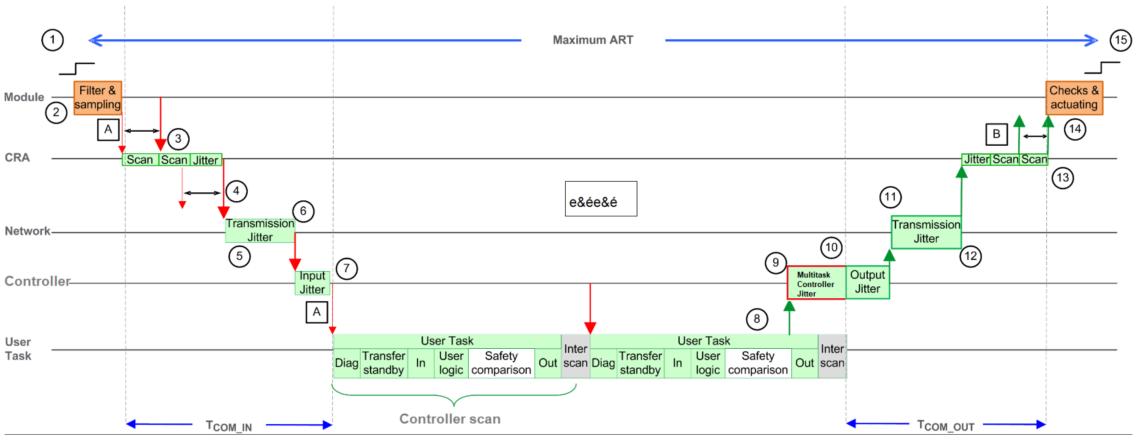
Calculating ART for M580 Hot Standby Controllers

The *Modicon M580 Frequently Used Architectures System Guide* describes both a simplified method and a more complex method of calculating ART for a standalone controller.

To calculate a maximum ART for an M580 Hot Standby controller, it is necessary to add to the standalone controller ART calculation estimates of the maximum time for:

- A switchover event, during which the standby controller takes over the role of primary controller, after the original primary controller ceases to be operational or loses communication.
- A switchover event, during which the standby controller takes over the role of primary controller in response to a user or application command.

ART: General Calculation for an M580 Safety-related Hot Standby Controller in a Multi-task Configuration



A: missed input scan	8: operation of application logic (1 scan)
B: missed output scan	9: additional jitter due to the controller multitasking
1: input turns ON	10: controller output jitter
2: input filtering and sampling (safety-related modules)	11: network delay
3: CRA drop processing time	12: network jitter
4: CRA input request packet interval (RPI) rate	13: CRA drop processing time
5: network delay	14: output check and actuating (safety-related modules)
6: network jitter	15: output applied
7: controller input jitter	–

The more complex method of ART calculation, in the *Modicon M580 Frequently Used Architectures System Guide* remains valid. Refer to that topic for an evaluation of the T_{COM_IN} and T_{COM_OUT} components.

NOTE: The following evaluations assume that the controller is not overloaded, and that total bandwidth of all tasks is less than 80%, page 177.

- Hot Standby effect: An additional time (Transfer Standby) is necessary to exchange the user data from the primary controller to the standby controller. This time is included in the execution time displayed in the **PLC > Animation** screen if the primary controller is connected to the standby.

- Functional safety effect:
  - An additional time (Safety Comparison) is necessary to exchange and compare the safety-related data between the controller and the Safety CoPro. This time is always included in the execution time displayed in the **PLC > Animation** screen.
  - The safety-related I/O modules have an internal cycle used for filtering and diagnostics. They introduce a delay on the sampling and the activation of the external signals.
- Multi-task effect: This introduces a jitter before the emission of the output message.

The following elements are necessary for the ART evaluation:

Name	Description		Value
T_{INPUT}	Time used by the safety input modules for the filtering and the sampling of the external signal.		6 ms
T_{COM_IN}	Sum of all the time used by the communication with input modules.		See[1]
Lost_scan	Lost scan time because the input message arrived after the begin of the scan.		T_{PER}[2]
Eff_scan	Effective scan which computes the outputs depending on the latest sampled input value.		T_{PER}[2]
Multitask_jitter	Jitter introduced by the multi-task system on the output message emission. The real value is based on the task execution time, for simplification they are capped by the task period.	MAST task	T_{SAFE}[3] + T_{FAST}[3]
		SAFE task	T_{FAST}[3]
		FAST task	0
T_{COM_OUT}	Sum of all the time used by the communication with output modules[1].		-
T_{OUTPUT}	Time used by the safety output modules for the diagnostics and for actuating the external signal.		6 ms

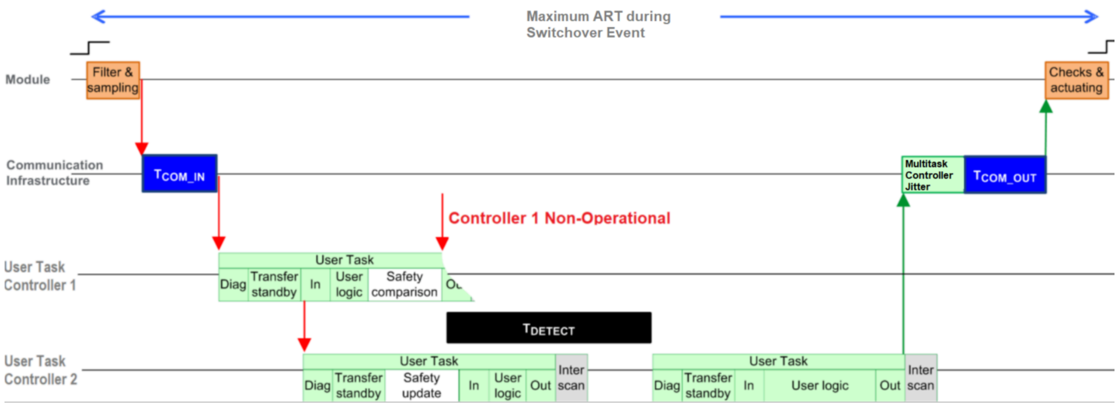
1. For a description of these elements, refer to the more complex ART calculation method in the *Modicon M580 Frequently Used Architectures System Guide*.

2. Refer to the **Processor Load** topic for a discussion of these elements.

3. The configured cycle period for the respective task (SAFE, FAST).

ART: M580 Safety-related Hot Standby Controller in a Multi-task Configuration During a Switchover Event

A switchover event occurs in a Hot Standby system when the primary controller ceases to be operational or loses communication. The standby controller, after a detection time, will restart the MAST task in its role as the new primary controller. Thereafter, the SAFE and FAST tasks will be able to start in the new primary controller. The worst-case switchover scenario from an ART standpoint, i.e., the one that takes the longest to complete, is shown in the following diagram:



The following elements are necessary for the ART evaluation in the case of a switchover:

Name	Description		Value
T_{INPUT}	Time used by the safety input modules for the filtering and the sampling of the external signal.		6 ms
T_{COM_IN}	Sum of all the time used by the communication with input modules.		See[1]
Lost_scan	Lost scan time because the input message arrived after the begin of the scan.		T_{PER}[2]
T_{DETECT}	Time used by the standby controller to detect and confirm the primary controller has become non-operational.		15 ms
Eff_scan	Effective scan which computes the outputs depending on the latest sampled input value.		T_{PER}[2]
Multitask_jitter	Jitter introduced by the multi-task system on the output message emission. The real value is based on the task execution time, for simplification they are capped by the task period.	MAST task	T_{SAFE}[3] + T_{FAST}[3]
		SAFE task	T_{FAST}[3]
		FAST task	0
Additional_jitter	Jitter introduced by the multi-task system to restart the task on the new controller.	MAST task	0
		SAFE task	T_{SAFE}[3]

Name	Description		Value
		FAST task	T_{FAST}[3]
T_{COM_OUT}	Sum of all the time used by the communication with output modules[1].		-
T_{OUTPUT}	Time used by the safety output modules for the diagnostics and for actuating the external signal.		6 ms

1. For a description of these elements, refer to the more complex ART calculation method in the *Modicon M580 Frequently Used Architectures System Guide*.

2. Refer to the **Processor Load** topic for a discussion of these elements.

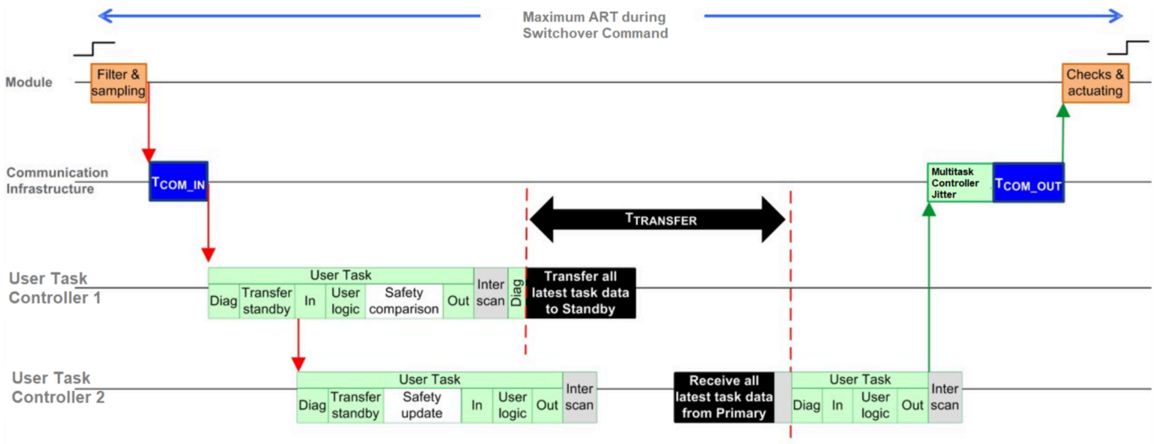
3. The configured cycle period for the respective task (SAFE, FAST).

ART: M580 Safety-related Hot Standby Controller in a Multi-task Configuration During a Switchover Command

A switchover command occurs in a Hot Standby system when the user requests it, either by program logic or through a communication request (for example, from the Hot Standby screen, an animation table, the HMI, and so forth).

On request, the primary controller verifies that all the conditions necessary to authorize a switchover are met, then confirms that all the tasks have updated the standby controller with the latest data. The primary controller then goes into wait mode. The remote controller switches to the primary mode, starting the MAST task first and then the other tasks SAFE and FAST). In the meantime, the other controller (i.e., the original primary controller) goes to standby mode.

The worst-case switchover scenario from an ART standpoint, i.e., the one that takes the longest to complete, is shown in the following diagram:



The following elements are necessary for the ART evaluation in the case of a switchover:

Name	Description		Value
T_{INPUT}	Time used by the safety input modules for the filtering and the sampling of the external signal.		6 ms
T_{COM_IN}	Sum of all the time used by the communication with input modules.		See[1]
Lost_scan	Lost scan time because the input message arrived after the begin of the scan.		T_{PER}[2]
$T_{TRANSFER}$	During the diagnostics of the MAST task, the controller accepts the switchover command and begins to perform the transfer of all the latest data for each task.		Refer to the formula, below.
Eff_scan	Effective scan which computes the outputs depending on the latest sampled input value.		T_{PER}[2]
Multitask_jitter	Jitter introduced by the multi-task system on the output message emission. The real value is based on the task execution time, for simplification they are capped by the task period.	MAST task	T_{SAFE}[3] + T_{FAST}[3]
		SAFE task	T_{FAST}[3]
		FAST task	0
Additional_jitter	Jitter introduced by the multi-task system to restart the task on the new controller.	MAST task	0
		SAFE task	T_{SAFE}[3]
		FAST task	Min(T_{FAST}, 5 ms)[3]
T_{COM_OUT}	Sum of all the time used by the communication with output modules[1].		-

Name	Description	Value
T_{OUTPUT}	Time used by the safety output modules for the diagnostics and for actuating the external signal.	6 ms
1. For a description of these elements, refer to the more complex ART calculation method in the *Modicon M580 Frequently Used Architectures System Guide*. 2. Refer to the **Processor Load** topic for a discussion of these elements. 3. The configured cycle period for the respective task (SAFE, FAST).		

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

max((K3 x ($MAST_{KB}$ + 2 x $SAFE_{KB}$ + $FAST_{KB}$) + K4 x ($MAST_{DFB}$ + 2 x $SAFE_{DFB}$ + $FAST_{DFB}$)) / 1000, T_{SAFE})

Where:

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

Coefficient	BMEH582040S	BMEH584040S or BMEH586040S
K3	46.4	14.8
K4	34.5	11.0

Task OVERRUN

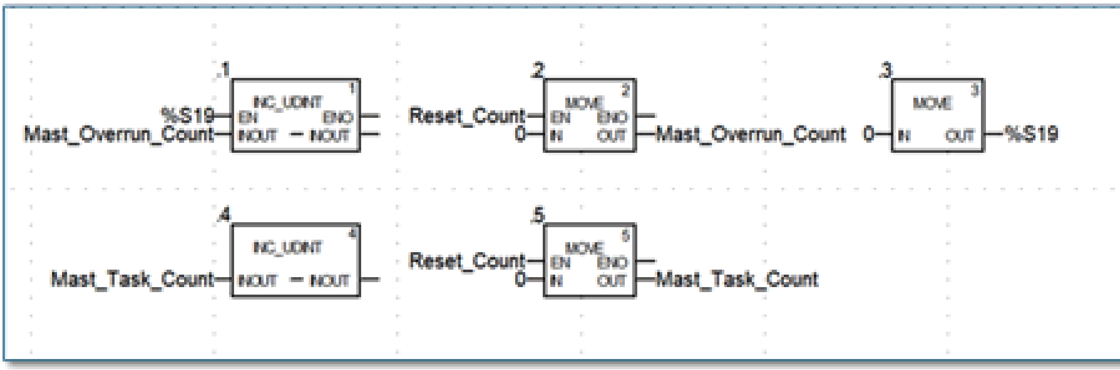
Testing for Task OVERRUN

The OVERRUN system bit is the first indicator to verify when testing and commissioning an application. If a task is in permanent OVERRUN, it helps prevent lower priority tasks from running, including the background system task.

To help prevent OVERRUN conditions from occurring, the system applies a short idle state at the end of each task execution. This time window is inserted to give testing and commissioning personnel the opportunity to analyze and, if necessary, modify the application and thereby avoid unrecoverable detected errors.

The task OVERRUN condition is signaled by the %S19 system bit. Normally set to 0, this bit is set to 1 by the system in the event of a time period OVERRUN (the task execution time is greater than the user-defined period in the configuration or programmed into the %SW word associated with the task).

NOTE: On M580 Safety, information for SAFE task overrun is not available. Each task manages its own %S19 bit. You can use Control Expert or other communication means to verify the MAST task overrun only if the program does not reset it. Thus, use a counter to monitor this error condition:



M580 Hot Standby Diagnostics

Overview

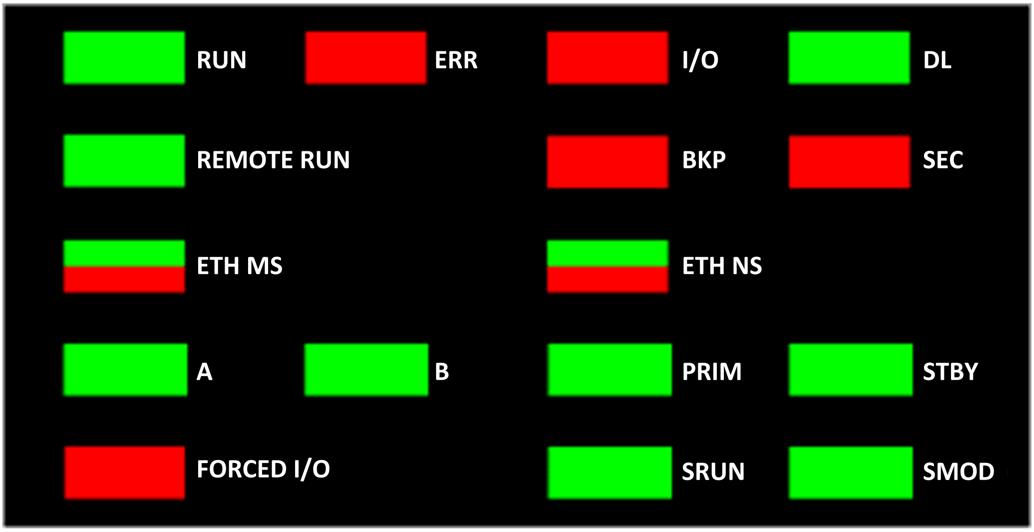
This chapter describes M580 Hot Standby diagnostic tools provided by the:

- BMEH58•040 controller Hot Standby LEDs
- Control Expert graphical user interface

LED Diagnostics for Hot Standby Controllers

LED Panel

The front face of a BMEH58•040 Hot Standby controller presents the following LED panel, which you can use to diagnose the state of the M580 Hot Standby system:



NOTE: The **SRUN** and **SMOD** LEDs apply only to safety-related controllers. The **SEC** LED is not used.

- For a description of the safety-related controller LEDs **SRUN** and **SMOD**, refer to the topic *LED Displays for the M580 Safety Controller and CoPro* in the *Modicon M580, Safety System Planning Guide*.
- For a presentation of other LED diagnostics for safety-related controllers, refer to the topic *M580 Safety Controller LED Diagnostics* in the *Modicon M580, Safety Manual*.

Hot Standby Panel LEDs

Use the BMEH58•040 Hot Standby controller **A** and **B** LEDs to identify the controller configurations, as set by the rotary switch on each controller:

A/B/Clear rotary switch position	**LED**	
	A	**B**
Local controller is A, remote controller is B	ON	OFF
Local controller is B, remote controller is A	OFF	ON
Both controllers configured as A	Flashing	OFF
Both controllers configured as B	OFF	Flashing
Local rotary switch on **Clear**	Flashing	Flashing

In the Hot Standby Panel LED diagnostic presentation, above:

- The local controller is the controller whose LEDs you are observing, which could be either **A** or **B**.
- The remote controller is the controller whose LEDs you are not observing, typically located in a remote location.

For example, consider the design where the two controllers are physically distant but communicate via a tunnel, with a controller located at each tunnel terminus. In this case, the local controller is the one in front of you; the remote controller is the one at the distant end of the tunnel. But, if you move to the other end of the tunnel, the formerly remote controller becomes the local controller and the original local controller becomes the remote controller. By contrast, the designations of controller A and controller B do not change.

Use the BMEH58•040 **REMOTE RUN** LED on the local controller to identify the operational status of the remote controller:

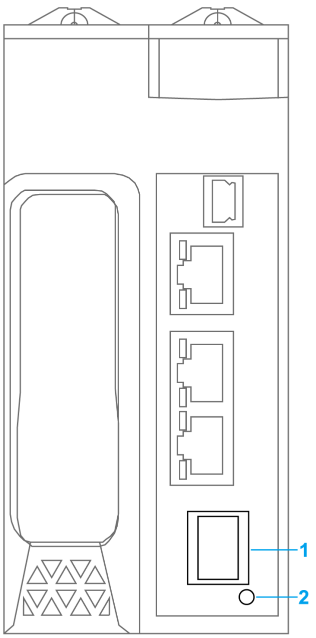
REMOTE RUN LED	Remote controller state
ON	RUN
Flashing	STOP
OFF	Indeterminate

Use the BMEH58•040 **PRIM**, and **STBY** LEDs to identify the operational status of the local and remote controller:

LED		Controller state	
PRIM	**STBY**	**Local controller**	**Remote controller**
ON	OFF	Primary	Standby
ON	Flashing	Primary	Wait
Flashing	Flashing	Wait	Indeterminate
OFF	OFF	Wait	Indeterminate
OFF	ON	Standby	Primary

Hot Standby Link LED

A Hot Standby link LED is located on the front of the BMEH58•040 controller:



1 SFP socket for copper or fiber-optic Hot Standby link connection

2 Hot Standby link LED

Use this LED to diagnose the state of the Hot Standby link:

Status	Color	Description
on	green	The port is communicating with the remote controller.
flashing	green	The port is configured and operational, but a Hot Standby link is not made.
off	—	The Hot Standby link is not configured or is not operational.

Ethernet Port Connector LEDs

Each Ethernet RJ45 connector presents a pair of LED indicators:

The Ethernet connector LEDs indicate the following states:

LED	Color	State	Description
ACT	Green	Flashing	Data is being transmitted over the link.
		Off	No transmission activity is occurring.
LNK	Green	On	Link speed = 100 Mbit/s.
	Yellow	On	Link speed = 10 Mbit/s.
	Green / Yellow	Off	No link is established.

Non-Hot Standby Panel LEDs

Refer to the following topics for additional information regarding non-Hot Standby LEDs:

- *LED Diagnostics for M580 Standalone Controllers* in the Modicon M580 Hardware Reference Manual for standalone, non-safety-related LEDs.
- *M580 Safety Controllers LED Diagnostics* in the M580 Safety Manual (see Modicon M580, Safety Manual), for safety-related LEDs.

Modicon M580-specific System Words %SW132 to %SW167

Diagnostic System Words

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Do not use undocumented system objects (%Si, %SWi) in your program, nor write to them in data tables.

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

Control Expert presents the following M580-specific system words you can use when diagnosing the state of your M580 Hot Standby system:

- **%SW132** to **%SW134**: controller MAC Address.
- **%SW135** to **%SW137**: controller serial number
- **%SW146** and **%SW147**: SD card serial number
- **%SW160** to **%SW167**: Detected errors for racks 0...7

For a more detailed description of these system words, refer to the M580 section (see EcoStruxure™ Control Expert, System Bits and Words, Reference Manual) of the *EcoStruxure™ Control Expert System Bits and Words Reference Manual*.

Embedded Web Pages

Introducing the M580 Hot Standby Controller Web Pages

Introduction

The M580 BMEH58•040(S) Hot Standby controllers includes an embedded web server that provide monitoring, diagnostic and file transfer functions.

These following web pages are common to both standalone and Hot Standby controllers:

- **Module**:
  - **Status Summary (Hot Standby)**, page 194
  - **HSBY Status**, page 197
  - **Performance**
  - **Port Statistics**
- **Connected Devices**:
  - **I/O Scanner**
  - Messaging
- **Services**:
  - **QoS**
  - **NTP**
  - **Redundancy**
- **System**:
  - **Alarm Viewer**
  - **Rack Viewer**, page 199
- **File Manager**:
  - **Data Storage**
  - **Event Log**

Status Summary (Hot Standby Controllers)

Introduction

The **Status Summary** web page provides this information about the controller:

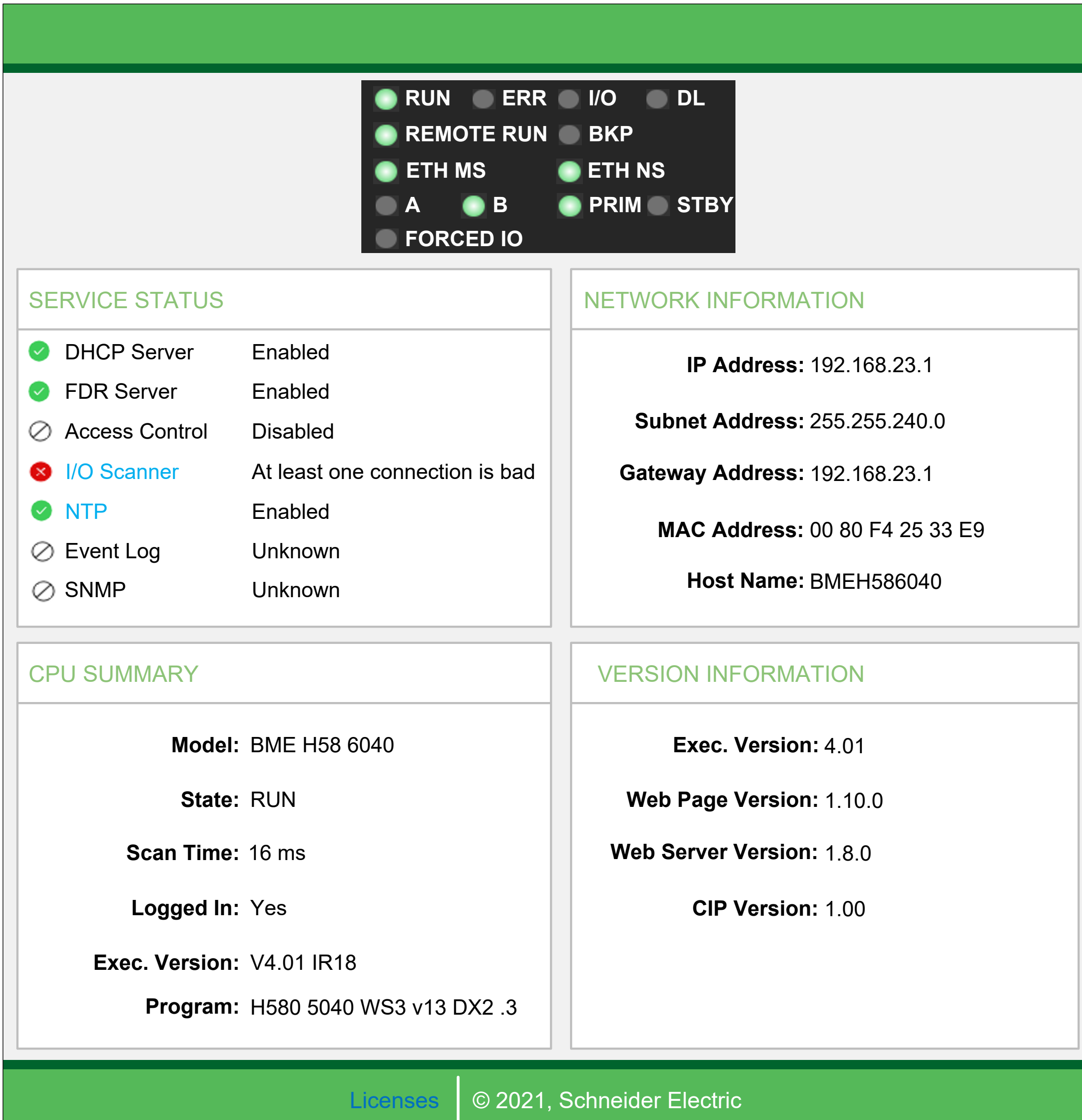
- Ethernet service diagnostic information
- Version descriptions for installed firmware and software
- Controller description and operating state
- IP addressing settings

NOTE: The **Status Summary** web page is refreshed every 5 seconds.

Open the Page

Access the **Status Summary** page on the **Diagnostics** tab (**Module > Status Summary**):

Status Summary page (Hot Standby controller):



Diagnostic and Status Information

The **Status Summary** web page provides this information:

<table>
<tr><th>Parameters</th><th colspan="2">Description</th></tr>
<tr><td rowspan="3">LEDs</td><td colspan="2">The web page displays the state of these LEDs:</td></tr>
<tr><td>• RUN
• ERR
• I/O
• DL
• REMOTE RUN
• BKP
• ETH MS
• ETH NS</td><td>• A
• B
• PRIM
• STBY
• FORCED_IO
• SRUN (safety-related controller)
• SMOD (safety-related controller)</td></tr>
<tr><td colspan="2">NOTE: The LEDs on the web page reflect the LEDs on the controller, page 188.</td></tr>
<tr><td rowspan="6">Service Status</td><td colspan="2">This area presents information describing the status of controller Ethernet services. The colored icons appearing to the left of some items indicate the following status:</td></tr>
<tr><td>green</td><td>The available service is operational and running.</td></tr>
<tr><td>red</td><td>An error is detected in an available service.</td></tr>
<tr><td>black</td><td>The available service is not present or not configured.</td></tr>
<tr><td colspan="2">The status of these Ethernet services is included:</td></tr>
<tr><td>• DHCP Server
• FDR Server
• Access Control</td><td>• Scanner Status
• NTP Status
• FDR Usage</td></tr>
<tr><td rowspan="2">Version Info.</td><td colspan="2">This area describes the software versions that are running on the controller, including:</td></tr>
<tr><td>• Executable Version
• Web Server Version</td><td>• Web Site Version
• CIP Version</td></tr>
<tr><td>Controller Summary</td><td colspan="2">This area describes the controller hardware and the applications that are running on the controller, including:
• Model
• State
• Scan Time</td></tr>
<tr><td>Network Info.</td><td colspan="2">This field contains IP addressing settings for the controller, including:
• IP Address
• Subnet Address
• Gateway Address</td></tr>
</table>

HSBY Status

Introduction

The **HSBY Status** web page provides this information about the Hot Standby system:

- Hot Standby role and status of the **Local** controller
- Hot Standby role and status of the **Remote** controller
- General errors detected for the Hot Standby system

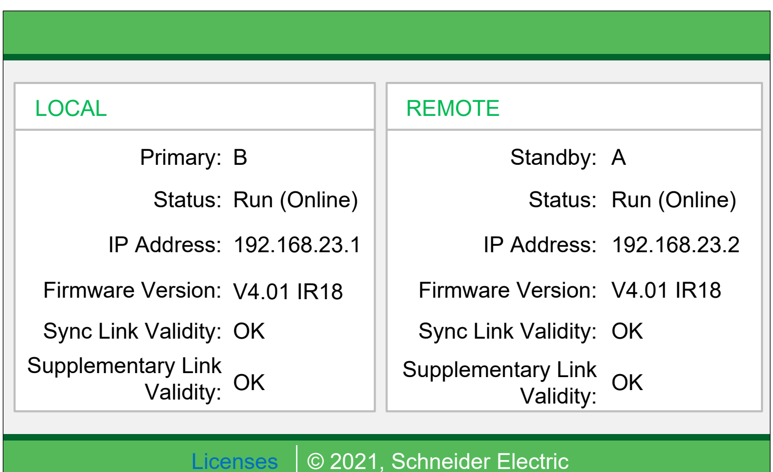
NOTE:

- The local controller is the controller configured with the **Main IP Address** (primary) or **Main IP Address + 1** (standby) used to access this web page.
- The **HSBY Status** web page is refreshed every 5 seconds.

Open the Page

Access the **HSBY Status** page from the **Diagnostics** tab (**Module > HSBY Status**):

HSBY Status page:



Diagnostic and Status Information

The **HSBY Status** web page provides this information:

<table>
<tr><th>Area</th><th colspan="2">Description</th></tr>
<tr><td rowspan="8">Local/Remote</td><td colspan="2">This area displays the state of Hot Standby settings for the local and remote controllers:</td></tr>
<tr><td><Hot Standby Role></td><td>The Hot Standby system role of the controller. Valid values include:
• Primary
• Standby
• Wait</td></tr>
<tr><td><A/B switch setting></td><td>The designation of the controller, defined by the rotary switch on the back of the controller. Valid values include:
• A
• B</td></tr>
<tr><td>Status</td><td>The operating state of the controller. Valid values include:
• RUN
• STOP
• NoConf
• HALT</td></tr>
<tr><td>IP Address</td><td>The IP address used to communicate with the controller for web page access:
• For the primary Hot Standby controller, this is the Main IP Address setting.
• For the standby Hot Standby controller, this is the Main IP Address+ 1 setting.</td></tr>
<tr><td>Firmware Version</td><td>Firmware version of the controller operating system.</td></tr>
<tr><td>Sync Link Validity</td><td>The status of the Hot Standby link :
• OK: the link is operational.
• NOK: the link is not operational.</td></tr>
<tr><td>Supplementary Link Validity</td><td>The status of the Ethernet RIO link:
• OK: the link is operational.
• NOK: the link is not operational.</td></tr>
</table>

Rack Viewer

Introducing the Controller Status Page

The BMEH584040(S) and BMEH586040(S) Hot Standby controllers include a **Rack Viewer** web page. Use this page to view controller information, including:

- LEDs status
- controller identification
- application signature identification
- select application configuration settings

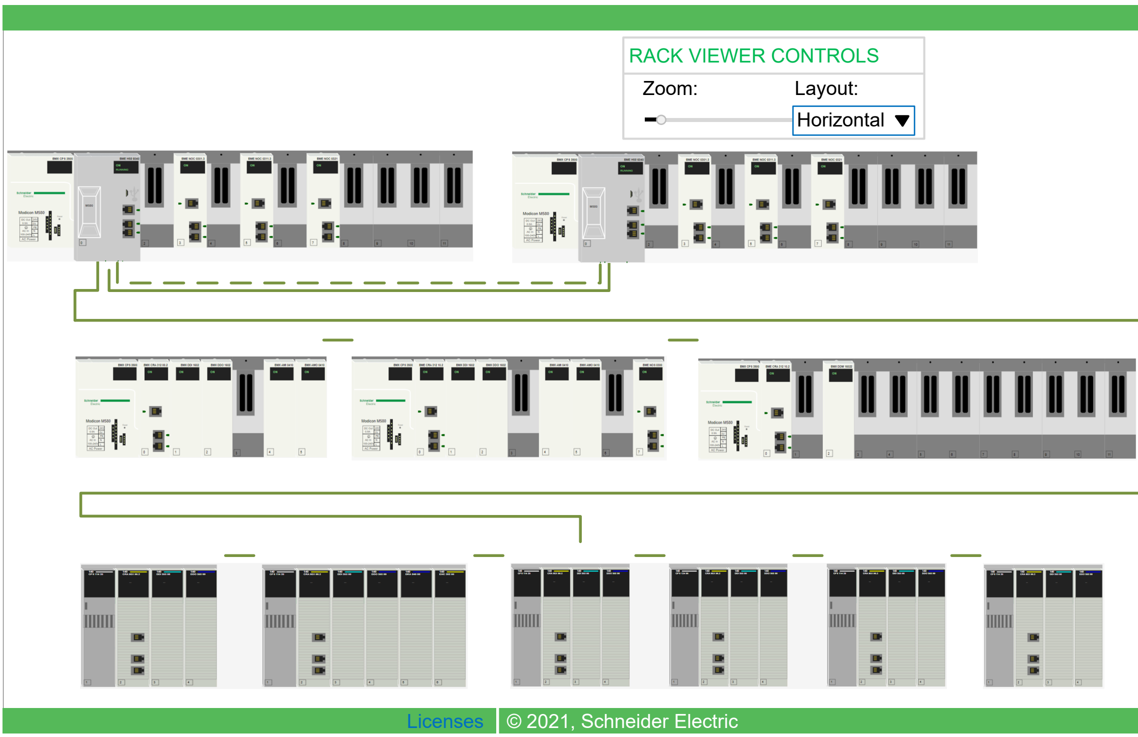
Access this page from the **Diagnostics** tab (**System > Rack Viewer**).

This example of a **Rack Viewer** page shows a Hot Standby controller on its rack with a power supply:

Accessing the Rack Viewer Page

Access the **Rack Viewer** page from the **Diagnostics** menu. In the navigation menu at the left side of the page, select **Menu > System > Rack Viewer**:

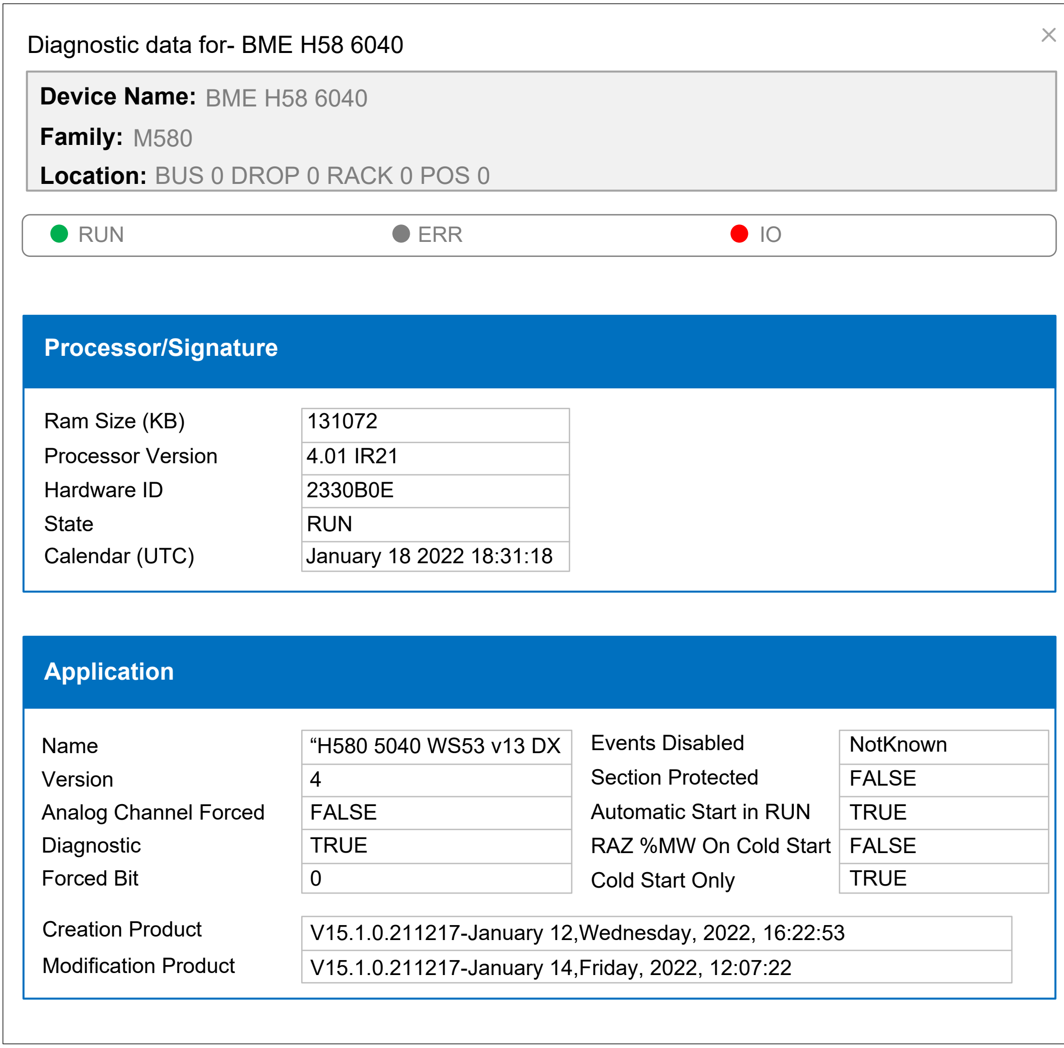
Rack Viewer page (HSBY Controller):



This example of a Rack Viewer page shows the Hot Standby connection between a primary controller rack and a standby controller rack. The Hot Standby connection (dashed line) is green when the Hot Standby link is healthy. If the Hot Standby link is not healthy, the dashed line is red.

Rack Viewer Data

Double-click on the **Rack Viewer** page to display Hot Standby controller data.



Data Field	Description
Processor/Signature	
RAM size (kb)	The size of controller RAM in KB
Processor Version	Firmware version
Hardware ID	An identifier for the module hardware.
Calendar (UTC)	Date and time of the last detected error

Data Field	Description
State	The operating state of the controller: • NO CONFIGURATION • IDLE • STOP • RUN • HALT • INITIALIZING • ERROR
Application	
Name	Name of the Control Expert project
Version	Project version
Analog channel forced:	Indicates if one or more inputs or outputs for an analog channel have been forced: • **TRUE** indicates the an analog input or output has been forced. • **FALSE** indicates no analog input or output has been forced.
Diagnostic	Indicates if the diagnostic buffer has been activated for the project: • **TRUE** indicates that **Application diagnostics** and/or **System diagnostics** has been selected in the **General > PAC Diagnostics** tab of the **Project Settings** dialog for the application. • **FALSE** indicates **Application diagnostics** and **System diagnostics** have not been selected.
Forced bit	The number of forced bits in the application.
Events Disabled	Indicates if all event processing has been disabled: • **TRUE** indicates that all event processing has been disabled. • **FALSE** indicates that event processing has not been disabled. **NOTE:** Events can be enabled/disabled by using: • The **Enable or Disable All** command in the **Task** tab of the controller (refer to *EcoStruxure™Control Expert, Operating Modes*) • The *MASKEVT* and *UNMASKEVT* functions. • The system bit %S38.
Section protected	Indicates if password access is required to edit one or more sections of the application: • **TRUE** indicates that a password is required to edit specified sections of the application. • **FALSE** indicates that no password is required for application editing.
Automatic Start in RUN	Indicates if the application is automatically set to start when the controller goes into RUN operational mode: • **TRUE** indicates the application automatically starts. • **FALSE** indicates the application does not automatically start.

Data Field	Description
RAZ %MW on cold start	Indicates if %MW registers are reset to their initial values on a cold start: • **TRUE** indicates that values are reset. • **FALSE** indicates that values are not reset.
Cold Start only	Indicates if a cold start is forced on a system re-start: • **TRUE** indicates that a reset forces a cold start of the application. • **FALSE** indicates that a warm start will occur on application reset.
Creation Product	Includes both: • Version and build of Control Expert used to edit the project. • Date and time the project was created.
Modification Product	Includes both: • Version and build of Control Expert used to edit the project. • Date and time the project was last edited.

Hot Standby Elements Firmware Update

Firmware Update

Introduction

You can update firmware for modules in an M580 Hot Standby system by installing a new firmware version using EcoStruxure Automation Device Maintenance (EADM).

NOTE: Update the firmware in system communication modules (for example, BMENOC0301 / BMENOC0311 or BMENOC0302(H)) before updating the firmware in the controllers.

Refer to the *EcoStruxure Automation Device Maintenance, Firmware Upgrade Tool, Online Help* for a description of the download procedure. In addition, refer to the *Modicon M580 Controller Firmware Installation Guide* for more information.

Updating Controller Firmware Without Stopping the Hot Standby System Process

You can update firmware in the two Hot Standby controllers without interrupting the system process if the following pre-conditions exist:

- One controller is running as the primary.
- The other controller is running as the standby.
- The **FTP** setting is enabled in the **Security** tab of the **EIO** network.
- Logic mismatch is allowed for the Hot Standby system, by setting the `LOGIC_MISMATCH_ALLOWED` property.
- Firmware mismatch is allowed for the Hot Standby system, by setting the `FW_MISMATCH_ALLOWED` property
- The new firmware to be installed is not fundamentally different from the previous firmware, and therefore will not trigger an `APP_MISMATCH` condition.

When updating the controller firmware, install the standby controller firmware first, then install the primary controller firmware.

Interrupting the update procedure before it has completed can cause irreparable damage to the controller.

NOTICE

EQUIPMENT DAMAGE

During the transfer of the firmware file:

- Do not power OFF the controller.
- Do not power OFF the PC.
- Do not shut down the firmware loader.
- Do not disconnect the communication cable.
- Do not remove or insert the optional SD memory card.

Failure to follow these instructions can result in equipment damage.

To update the Hot Standby controller firmware, refer to the *Modicon M580 Controller Firmware Installation Guide*.

Updating Other Firmware

You can update firmware to other modules in the Hot Standby network, without interrupting the system process. This is accomplished by connecting your PC directly to the Ethernet service port of the (e)X80 EIO adapter module, Ethernet communication module, or Ethernet network option switch module.

> **NOTE:** If you instead connect your PC to the controller to perform this update, the process executing in the controller stops until the update is complete.

Refer to the *Modicon M580 Controller Firmware Installation Guide* for more information.

Verifying the Network Configuration

Using the Ethernet Network Manager

Introduction

In Control Expert, click **Tools > Ethernet Network Manager** to visualize and verify a complex network configuration. The tool can:

- provide a global view of your network
- edit IP addresses and device identifiers for (e)X80 EIO adapter modules

Use either method to access the **Ethernet Network Manager**:

- Select **Tools > Ethernet Network Manager**.
- Select **Ethernet Network Manager** in the **Project Browser**.

  NOTE: The **Ethernet Network Manager** tool is available on all M580 controllers. Only devices enabled in the address server (DHCP) are available.

Network Topology Configuration

The **Ethernet Network Manager** tool provides a snapshot of IP address settings for devices included in network topologies that are part of your application. If the tool detects an addressing error, it displays the detected error with a red background. If the tool detects an error, you can re-configure the affected setting in Control Expert.

Parameters in the **Ethernet Network Manager**:

Parameter	Description
Name	Ethernet communication device name
Type	The device type: • Scanner • Module
Subtype	The device sub-type: • RIO/DIO • CRA
Profiles	The kind of control network communications: • Remote (RIO) • Distributed (DIO)

Parameter	Description
Topo address	The topological address of the device, in the sequence: bus, drop, rack, slot.
DHCP Enable	Indicates if the device is a DHCP client and receives its IP address(es) from a DHCP server (yes/no).
IP Address	The IP address, or addresses, assigned to the device. **NOTE:** Editable for scanned modules.
Subnet Mask	The subnet mask related to each assigned IP address.
Gateway Address	The IP address of the default gateway, to which messages for other networks are transmitted.
Identified By	For scanned devices, the type of network identifier.
Identifier	The string used to identify a scanned device. The default value is the device Name. **NOTE:** Editable for scanned modules.
SNMP	For scanning devices, the IP address of up to two SNMP network manager devices.
NTP State	The state of the NTP client: • Enabled • Disabled
NTP Configuration	The IP addresses of up to two NTP servers that send updates to the NTP client resident in the device.

NOTE:

- The red cells indicate detected errors (defined by network management rules).
- After editing a scanned module **IP Address** or **Identifier** setting, click the **Validate** button to save your edits.

Verifying a Hot Standby Network

Follow these steps to use the **Ethernet Network Manager** tool while building your network in Control Expert:

Step	Action
1	In Control Expert, click **Tools > Ethernet Network Manager**. **Result:** A preliminary, read-only global view of your network is displayed.
2	Verify the settings with a red background, indicating the tool has detected a configuration error.
3	Click **OK** to close the **Network Inspector** tool.

Step	Action
4	If the tool displayed a detected error: • in a scanning device, go to the specific device editor and change the IP configuration settings. • in a scanned device, you can edit the **IP address** and **Identifier** settings in the **Ethernet Network Manager**, or go to the specific device editor and change the IP configuration settings. When you finish your edits, run the **Ethernet Network Manager** again.
5	Add distributed equipment and/or RIO modules to the **EIO Bus**. **NOTE:** Only devices enabled in the address server (DHCP) are available.
6	Configure the scanners.
7	Repeat steps 1, 2, 3, and 4 until the **Ethernet Network Manager** no longer detects any errors.

Network Manager Services

The network manager starts automatically when you open the **Network Inspector** tool. The global network management system (GNMS) is responsible for global network consistency. The following verifications are performed:

- GNMS verifies that all IP addresses are unique for the modules in the application.
- Each gateway that exists on your network is displayed in the network manager. By default, Control Expert notifies you if one of the gateways is missing an IP address. You can change this notification by clicking **Tools > Project Settings > General > Management of build messages > Missing gateway IP @ generates**. The options are a `detected warning` (default value) or nothing.
- Only a single RSTP switch can be configured as a root for a given network.
- The range of IP addresses is 1.0.0.0 … 126.255.255.255 or 128.0.0.0 … 223.255.255.255. Otherwise, an error is detected. Addresses 224.0.0.0 and greater are multicast or experimental addresses. Addresses starting at 127 are loopback addresses. Addresses 169.254/16 are reserved for automatic private IP addressing (APIPA).
- The tool verifies that the network address of the IP address is valid.
- The tool verifies that the host address of the IP address is valid, including that broadcast IP addresses are blocked.
- While an M580 controller uses *classless inter-domain routing* (CIDR), some IP addresses are not allowed to maintain compatibility:
  - in a class A network, IP addresses that end in 255.255.255
  - in a class B network, IP addresses that end in 255.255
  - in a class C network, IP addresses that end in 255

- The IP address is configured to access the gateway address. Therefore, the gateway address is within the subnetwork defined by the mask. The gateway is not accessible when it is not on the same subnetwork as the IP address.

Network Bandwidth Considerations

Control Expert alerts you when there are possible bandwidth considerations.

Ethernet RIO bandwidth:

- Control Expert displays a detected error message in the log window if the RIO bandwidth (originator -> target) or (target->originator) is greater than 8%.
- Control Expert displays a **warning** in the log window if the RIO bandwidth (originator -> target) or (target->originator) is greater than 6%.

Device network bandwidth (DIO and RIO combined):

- Control Expert displays a detected **error** in the log window if total Modbus and EIP bandwidth (originator -> target) or (target->originator) is greater than 40%.
- Control Expert displays a **warning** in the log window if total Modbus and EIP bandwidth (originator -> target) or (target->originator) is greater than 30%.

Troubleshooting

Best Practices

Task Processor Load

Standalone and Hot Standby

Because the multi-task system impacts task measurements, perform measurements per task. Depending on your program, you can limit the multi-task impact using one of these two methods:

- Disable or Stop all the tasks except the one to measure.
- If the previous method is not possible because it would adversely impact program execution, in the **PLCScreen** adjust the task periods to their maximum value.

Initialize the measure with the corresponding **Init Duration** button and consider the minimum or present duration. The processor load due to the task is given by the formula:

$$L_{Task} = \frac{T_{Task}}{P_{Task}}$$

Where L is the load, T the task duration and P the configured period.

The total processor load is the sum of all the tasks contributions plus the system load (around 4% for standalone and 7% for Hot Standby):

$$L_{CPU} = L_{Sys} + L_{Fast} + L_{Safe} + L_{Mast} + L_{Aux0} + L_{Aux1}$$

Hot Standby

In a Hot Standby system, evaluate the margin regarding the task synchronization between the primary and the standby controller.

This is significant for an application where data transfer from primary to standby is large compared to the task period (large data size, numerous DFB instances).

The principle is to evaluate the minimum task period, PTask_sync, necessary to achieve the task synchronization in a single task context. A good approximation is to sum the task duration on the primary with the task duration on the standby.

NOTE: This is true if the standby executes only a few sections of program. Select no execution or first section only in the Hot Standby configuration screen or use condition execution in each section property.

$$P_{Task\_sync} = T_{Task\_Primary} + T_{Task\_Standby}$$

Next, configure the task period to a value greater than T_{task_min} divided by the available bandwidth. For example, for the MAST task:

$$P_{Mast\_min} = \frac{P_{Mast\_sync}}{1 - \left(L_{Sys} + L_{Fast} + L_{Mast}\right)}$$

NOTE: In most applications, both FAST and SAFE tasks are far from the de-synchronization condition when the 80% processor load is met. In this case, the only condition to verify is for the MAST task, and it can be "measured" directly if you:

- Start from the configured condition, and
- Step down the MAST task period (using the **Adjusted Period** entry in the **PLCScreen**) until the first MAST de-synchronization occurs.

Task Load Using System Words %SW110 to %SW116

Beginning with firmware version 2.7, the processor load is calculated and averaged in real time by the controller. The values are available in the system words

- %SW110 is the processor load used by the system for internal service.
- %SW111 is the processor load used by the MAST task.
- %SW112 is the processor load used by the FAST task.
- %SW113 is the processor load used by the SAFE task.
- %SW114 is the processor load used by the AUX0 task.
- %SW115 is the processor load used by the AUX1 task.
- %SW116 is the total processor load (sum of %SW110 to %SW115).

For each task, the value is the average load of the task:

$$\%SW11x = \frac{\sum_{CurrentCycle-99}^{CurrentCycle} T_{Taskx_{cycle}}}{(100 * P_{Taskx})}$$

The values of the significant task load are displayed in the **PLCScreen** for Control Expert version 14.1 and later.

This directly provides the global processor load, and the individual task load without tedious measures and calculations. The individual task loads can be used to identify the major bandwidth consumers and therefore help tune the application.

NOTE: tasks except MAST, works only if the MAST task is in Stop or is disabled. When the MAST task is running, the other task loads are underestimated. The size of the discrepancy is greater when the global processor load is small.

Therefore:

- To get precise information on task execution, you need to measure %SW112 to % SW115 while the MAST task is stopped or disabled and calculate by program the global load (%SW116 is not accurate).
- To program a notification if the processor load is approaching 100%, test the % SW116 value.

For firmware version 3.20 and later, the measures are correct and can be used as intended.

Helping to Prevent Task Overrun

Tuning

It may be necessary to tune the application to reach a state where there is no Overrun, all the tasks are synchronized, and the global processor load is less than 80%. The different control levers are:

1. Limit the code executed. By default, a controller executes all the user code at each cycle. By using a structured code, for some application types, it is possible to limit code execution to what is minimally necessary in each particular context using, for example: IF_THEN_ELSE, EN_ENO, SFC language, section condition.
2. Relax the timing constraints on task periods. This is particularly efficient for high frequency tasks. For example, moving the Fast task period from 5 to 6 milliseconds may have very little impact on the process, but frees 20% of processor load.

3. Limit the usage of DFB and nested DFB. At each boundary (in, out, inout) of a DFB, the controller must copy the external variables to/from internal variables. It consumes both time and memory.

Identifying Task Execution Time in Control Expert

The **Task** tab of the **PLCScreen** dialog in Control Expert displays measures of the task execution envelopes. For each task, it gives three execution values - or pairs of execution values:

- Minimum Duration: Shortest period of task execution, measured from the last cold restart or reinitialization.
- Current Duration: Most recent period of task execution.
- Maximum Duration: Longest period of task duration, measured from the last cold restart or reinitialization.

The measure can be re-initialized using the **Init Duration** command.

When a pair of value is displayed in the format: T_1 (T_2):

- T_1 is the envelope execution time, including the program execution, the communication, the time left for higher priority tasks.
- T_2 is everything except the execution time. Thus:

  $T_1 = T_{exe} + T_2$.

Efficient Application Transfer

Hot Standby and Safety-related Controllers

For Hot Standby and/or Safety-related controller, limit the number of variables that will be copied to Standby and/or Safety Copro:

- Purge unused variables.
- Whenever possible, use a single instance of DFB, for example, when a DFB does not contain state information. This can have a large impact on transfer time.
- De-select **Exchange On Standby** for variables that are evaluated at each cycle and are not a state. In a Switchover, the value copied from the primary will be overwritten during the first cycle execution, prior to processing any output. Use naming rules to identify these variables in the program, to ease the design and allow multiple modifications simultaneously.

Glossary

A

adapter:

An adapter is the target of real-time I/O data connection requests from scanners. It cannot send or receive real-time I/O data unless it is configured to do so by a scanner, and it does not store or originate the data communications parameters necessary to establish the connection. An adapter accepts explicit message requests (connected and unconnected) from other devices.

advanced mode:

In Control Expert, advanced mode is a selection that displays expert-level configuration properties that help define Ethernet connections. Because these properties should be edited only by people with a good understanding of EtherNet/IP communication protocols, they can be hidden or displayed, depending upon the qualifications of the specific user.

architecture:

Architecture describes a framework for the specification of a network that is constructed of these components:

- physical components and their functional organization and configuration
- operational principles and procedures
- data formats used in its operation

ARRAY:

An ARRAY is a table containing elements of a single type. This is the syntax: ARRAY [<limits>] OF <Type>

Example: ARRAY [1..2] OF BOOL is a one-dimensional table with two elements of type BOOL.

ARRAY [1..10, 1..20] OF INT is a two-dimensional table with 10x20 elements of type INT.

ART:

(*application response time*) The time a controller application takes to react to a given input. ART is measured from the time a physical signal in the controller turns on and triggers a write command until the remote output turns on to signify that the data has been received.

AUX:

An (AUX) task is an optional, periodic processor task that is run through its programming software. The AUX task is used to execute a part of the application requiring a low priority. This task is executed only if the MAST and FAST tasks have nothing to execute. The AUX task has two sections:

- IN: Inputs are copied to the IN section before execution of the AUX task.
- OUT: Outputs are copied to the OUT section after execution of the AUX task.

B

BCD:

(*binary-coded decimal*) Binary encoding of decimal numbers.

BOOL:

(*boolean type*) This is the basic data type in computing. A `BOOL` variable can have either of these values: 0 (`FALSE`) or 1 (`TRUE`).

A bit extracted from a word is of type `BOOL`, for example: `%MW10.4`.

BOOTP:

(*bootstrap protocol*) A UDP network protocol that can be used by a network client to automatically obtain an IP address from a server. The client identifies itself to the server using its MAC address. The server, which maintains a pre-configured table of client device MAC addresses and associated IP addresses, sends the client its defined IP address. The BOOTP service utilizes UDP ports 67 and 68.

broadcast:

A message sent to all devices in the subnet.

C

CCOTF:

(*change configuration on the fly*) A feature of Control Expert that allows a module hardware change in the system configuration while the system is operating. This change does not impact active operations.

CIP™:

(*common industrial protocol*) A comprehensive suite of messages and services for the collection of manufacturing automation applications (control, safety, synchronization, motion, configuration and information). CIP allows users to integrate these manufacturing applications with enterprise-level Ethernet networks and the internet. CIP is the core protocol of EtherNet/IP.

class 1 connection:

A CIP transport class 1 connection used for I/O data transmission via implicit messaging between EtherNet/IP devices.

class 3 connection:

A CIP transport class 3 connection used for explicit messaging between EtherNet/IP devices.

connected messaging:

In EtherNet/IP, connected messaging uses a CIP connection for communication. A connected message is a logical relationship between two or more application objects on different nodes. The connection establishes a virtual circuit in advance for a particular purpose, such as frequent explicit messages or real-time I/O data transfers.

connection originator:

The EtherNet/IP network node that initiates a connection request for I/O data transfer or explicit messaging.

connection:

A virtual circuit between two or more network devices, created prior to the transmission of data. After a connection is established, a series of data is transmitted over the same communication path, without the need to include routing information, including source and destination address, with each piece of data.

connectionless:

Describes communication between two network devices, whereby data is sent without prior arrangement between the two devices. Each piece of transmitted data also includes routing information, including source and destination address.

control network:

An Ethernet-based network containing PACs, SCADA systems, an NTP server, PCs, AMS, switches, etc. Two kinds of topologies are supported:

- flat: All modules and devices in this network belong to same subnet.
- 2 levels: The network is split into an operation network and an inter-controller network. These two networks can be physically independent, but are generally linked by a routing device.

CPU:

(*central processing unit*) The CPU, also known as the processor or controller, is the brain of an industrial manufacturing process. It automates a process as opposed to relay control systems. CPUs are computers suited to survive the harsh conditions of the industrial environment.

D

DDT:

(*derived data type*) A derived data type is a set of elements with the same type (`ARRAY`) or with different types (structure).

determinism:

For a defined application and architecture, you can predict that the delay between an event (change of value of an input) and the corresponding change of a controller output is a finite time *t*, smaller than the deadline required by your process.

Device DDT (DDDT):

A Device DDT is a DDT predefined by the manufacturer and not modifiable by user. It contains the I/O language elements of an I/O module.

device network:

An Ethernet-based network within an RIO network that contains both RIO and distributed equipment. Devices connected on this network follow specific rules to allow RIO determinism.

DFB:

(*derived function block*) DFB types are function blocks that can be defined by the user in ST, IL, LD or FBD language.

Using these DFB types in an application makes it possible to:

- simplify the design and entry of the program
- make the program easier to read
- make it easier to debug
- reduce the amount of code generated

DHCP:

(*dynamic host configuration protocol*) An extension of the BOOTP communications protocol that provides for the automatic assignment of IP addressing settings, including IP address, subnet mask, gateway IP address, and DNS server names. DHCP does not require the maintenance of a table identifying each network device. The client identifies itself to the DHCP server using either its MAC address, or a uniquely assigned device identifier. The DHCP service utilizes UDP ports 67 and 68.

DIO cloud:

A group of distributed equipment that is not required to support RSTP. DIO clouds require only a single (non-ring) copper wire connection. They can be connected to some of the copper ports on DRSs, or they can be connected directly to the controller or Ethernet communications modules in the *local rack*. DIO clouds **cannot** be connected to *sub-rings*.

DIO network:

A network containing distributed equipment, in which I/O scanning is performed by a controller with DIO scanner service on the local rack. DIO network traffic is delivered after RIO traffic, which takes priority in an RIO network.

DIO:

(*distributed I/O*) Legacy term for distributed equipment. DRSs use DIO ports to connect distributed equipment.

distributed equipment:

Any Ethernet device (Schneider Electric device, PC, servers, or third-party devices) that supports exchange with a controller or other Ethernet scanner service.

DNS:

(*domain name server/service*) A service that translates an alpha-numeric domain name into an IP address, the unique identifier of a device on the network.

domain name:

An alpha-numeric string that identifies a device on the internet, and which appears as the primary component of a web site's uniform resource locator (URL). For example, the domain name *schneider-electric.com* is the primary component of the URL *www.se.com*.

Each domain name is assigned as part of the domain name system, and is associated with an IP address.

Also called a host name.

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.

DSCP:

(*differentiated service code points*) This 6-bit field is in the header of an IP packet to classify and prioritize traffic.

DST:

(*daylight saving time*) DST is also called *summer time* and is a practice consisting of adjusting forward the clock near the start of spring and adjusting it backward near the start of autumn.

DT:

(*date and time*) The `DT` type, encoded in BCD in a 64-bit format, contains this information:

- the year encoded in a 16-bit field
- the month encoded in an 8-bit field
- the day encoded in an 8-bit field
- the time encoded in an 8-bit field
- the minutes encoded in an 8-bit field
- the seconds encoded in an 8-bit field

NOTE: The eight least significant bits are not used.

The `DT` type is entered in this format:

DT#<Year>**-**<Month>**-**<Day>**-**<Hour>**:**<Minutes>**:**<Seconds>

This table shows the upper/lower limits of each field:

Field	Limits	Comment
Year	[1990,2099]	Year
Month	[01,12]	The leading 0 is displayed; it can be omitted during data entry.
Day	[01,31]	For months 01/03/05/07/08/10/12
	[01,30]	For months 04/06/09/11
	[01,29]	For month 02 (leap years)
	[01,28]	For month 02 (non-leap years)
Hour	[00,23]	The leading 0 is displayed; it can be omitted during data entry.
Minute	[00,59]	The leading 0 is displayed; it can be omitted during data entry.
Second	[00,59]	The leading 0 is displayed; it can be omitted during data entry.

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 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.

EFB:

(*elementary function block*) This is a block used in a program which performs a predefined logical function.

EFBs have states and internal parameters. Even if the inputs are identical, the output values may differ. For example, a counter has an output indicating that the preselection value has been reached. This output is set to 1 when the current value is equal to the preselection value.

EF:

(*elementary function*) This is a block used in a program which performs a predefined logical function.

A function does not have any information on the internal state. Several calls to the same function using the same input parameters will return the same output values. You will find information on the graphic form of the function call in the [*functional block (instance)*]. Unlike a call to a function block, function calls include only an output which is not named and whose name is identical to that of the function. In FBD, each call is indicated by a unique [number] via the graphic block. This number is managed automatically and cannot be modified.

Position and configure these functions in your program in order to execute your application.

You can also develop other functions using the SDKC development kit.

EIO network:

(*Ethernet I/O*) An Ethernet-based network that contains 3 types of devices: a local rack, an X80 EIO drop, and a ConneXium extended dual-ring switch (DRS). Distributed equipment may also participate in an EIO network via connection to DRSs or the service port of X80 EIO adapter modules.

EN:

`EN` stands for **EN**able; it is an optional block input. When the `EN` input is enabled, an `ENO` output is set automatically.

If `EN` = 0, the block is not enabled; its internal program is not executed, and `ENO` is set to 0.

If `EN` = 1, the block's internal program is run and `ENO` is set to 1. If a runtime error is detected, `ENO` is set to 0.

If the `EN` input is not connected, it is set automatically to 1.

ENO:

ENO stands for **E**rror **NO**tification; this is the output associated with the optional input EN.

If ENO is set to 0 (either because EN = 0 or if a runtime error is detected):

- The status of the function block outputs remains the same as it was during the previous scanning cycle that executed correctly.
- The output(s) of the function, as well as the procedures, are set to 0.

Ethernet DIO scanner service:

an embedded scanner service of M580 controllers (BMEP581020, BMEP582020, BMEP583020, BMEP584020, BMEH582040, BMEH582040S, BMEH584040, BMEH584040S, BMEH586040, BMEH586040S) that manages distributed equipment only on an M580 device network

Ethernet I/O scanner service:

an embedded scanner service of M580 controllers (BMEP582040, BMEP583040, BMEP584040, BMEH582040, BMEH582040S, BMEH584040, BMEH584040S, BMEH586040, BMEH586040S) that manages distributed equipment **and** RIO drops on an M580 device network

EtherNet/IP™:

A network communication protocol for industrial automation applications that combines the standard internet transmission protocols of TCP/IP and UDP with the application layer common industrial protocol (CIP) to support both high speed data exchange and industrial control. EtherNet/IP employs electronic data sheets (EDS) to classify each network device and its functionality.

Ethernet:

A 10 Mb/s, 100 Mb/s, or 1 Gb/s, CSMA/CD, frame-based LAN that can run over copper twisted pair or fiber optic cable, or wireless. The IEEE standard 802.3 defines the rules for configuring a wired Ethernet network; the IEEE standard 802.11 defines the rules for configuring a wireless Ethernet network. Common forms include 10BASE-T, 100BASE-TX, and 1000BASE-T, which can utilize category 5e copper twisted pair cables and RJ45 modular connectors.

explicit messaging client:

(*explicit messaging client class*) The device class defined by the ODVA for EtherNet/IP nodes that only support explicit messaging as a client. HMI and SCADA systems are common examples of this device class.

explicit messaging:

TCP/IP-based messaging for Modbus TCP and EtherNet/IP. It is used for point-to-point, client/server messages that include both data, typically unscheduled information between a client and a server, and routing information. In EtherNet/IP, explicit messaging is considered class 3 type messaging, and can be connection-based or connectionless.

F

FAST:

An event-triggered (FAST) task is an optional, periodic processor task that identifies high priority, multiple scan requests, which is run through its programming software. A FAST task can schedule selected I/O modules to have their logic solved more than once per scan. The FAST task has two sections:

- IN: Inputs are copied to the IN section before execution of the FAST task.
- OUT: Outputs are copied to the OUT section after execution of the FAST task.

FBD:

(*function block diagram*) A graphical programming language that works like a flowchart. By adding simple logical blocks (`AND`, `OR`, etc.), each function or function block in the program is represented in this graphical format. For each block, the inputs are on the left and the outputs on the right. Block outputs can be linked to inputs of other blocks in order to create complex expressions.

FDR:

(*fast device replacement*) A service that uses configuration software to replace an inoperable product.

FDT:

(*field device tool*) The technology that harmonizes communication between field devices and the system host.

FTP:

(*file transfer protocol*) A protocol that copies a file from one host to another over a TCP/IP-based network, such as the internet. FTP uses a client-server architecture as well as separate control and data connections between the client and server.

full duplex:

The ability of two networked devices to independently and simultaneously communicate with each other in both directions.

function block diagram:

See FBD.

G

gateway:

A gateway device interconnects two different networks, sometimes through different network protocols. When it connects networks based on different protocols, a gateway converts a datagram from one protocol stack into the other. When used to connect two IP-based networks, a gateway (also called a router) has two separate IP addresses, one on each network.

H

harsh environment:

Resistance to hydrocarbons, industrial oils, detergents and solder chips. Relative humidity up to 100%, saline atmosphere, significant temperature variations, operating temperature between - 10°C and + 70°C, or in mobile installations.

HART:

(*highway addressable remote transducer*) A bi-directional communication protocol for sending and receiving digital information across analog wires between a control or monitoring system and smart devices.

HART is the global standard for providing data access between host systems and intelligent field instruments. A host can be any software application from a technician's hand-held device or laptop to a plant's process control, asset management, or other system using any control platform.

high-capacity daisy chain loop:

Often referred to as HCDL, a high-capacity daisy chain loop uses dual-ring switches (DRSs) to connect device sub-rings (containing RIO drops or distributed equipment) and/or DIO clouds to the Ethernet RIO network.

HMI:

(*human machine interface*) System that allows interaction between a human and a machine.

HTTP:

(*hypertext transfer protocol*) A networking protocol for distributed and collaborative information systems. HTTP is the basis of data communication for the web.

I

I/O scanner:

An Ethernet service that continuously polls I/O modules to collect data, status, event, and diagnostics information. This process monitors inputs and controls outputs. This service supports both RIO and DIO logic scanning.

%I:

According to the CEI standard, `%I` indicates a language object of type discrete IN.

IEC 61131-3:

International standard: programmable logic controllers

Part 3: programming languages

IGMP:

(internet group management protocol) This internet standard for multicasting allows a host to subscribe to a particular multicast group.

IL:

(*instruction list*) This language is a series of basic instructions. It is very close to assembly language used to program processors. Each instruction is made up of an instruction code and an operand.

implicit messaging:

UDP/IP-based class 1 connected messaging for EtherNet/IP. Implicit messaging maintains an open connection for the scheduled transfer of control data between a producer and consumer. Because an open connection is maintained, each message contains primarily data, without the overhead of object information, plus a connection identifier.

inter-controller network:

An Ethernet-based network that is part of the control network, and provides data exchange between controllers and engineering tools (programming, asset management system (AMS)).

INT:

(*INTeger*) (encoded in 16 bits) The upper/lower limits are as follows: -(2 to the power of 15) to (2 to the power of 15) - 1.

Example: `-32768`, `32767`, `2#1111110001001001`, `16#9FA4`.

IODDT:

(*input/output derived data type*) A structured data type representing a module, or a channel of a controller. Each application expert module possesses its own IODDTs.

IP address:

The 32-bit identifier, consisting of both a network address and a host address assigned to a device connected to a TCP/IP network.

IPsec:

(*internet protocol security*) An open set of protocol standards that make IP communication sessions private and secure for traffic between modules using IPsec, developed by the internet engineering task force (IETF). The IPsec authentication and encryption algorithms require user-defined cryptographic keys that process each communications packet in an IPsec session.

isolated DIO network:

An Ethernet-based network containing distributed equipment that does not participate in an RIO network.

%IW:

According to the CEI standard, `%IW` indicates a language object of type analog IN.

L

LD:

(*ladder diagram*) A programming language that represents instructions to be executed as graphical diagrams very similar to electrical diagrams (contacts, coils, etc.).

literal value of an integer:

A literal value of an integer is used to enter integer values in the decimal system. Values may be preceded by the "+" and "-" signs. Underscore signs (_) separating numbers are not significant.

Example:

`-12, 0, 123_456, +986`

local rack:

An M580 rack containing the controller and a power supply. A local rack consists of one or two racks: the main rack and the extended rack, which belongs to the same family as the main rack. The extended rack is optional.

local slave:

The functionality offered by Schneider Electric EtherNet/IP communication modules that allows a scanner to take the role of an adapter. The local slave enables the module to publish data via implicit messaging connections. Local slave is typically used in peer-to-peer exchanges between PACs.

M

M580 Ethernet I/O device:

An Ethernet device that provides automatic network recovery and deterministic RIO performance. The time it takes to resolve an RIO logic scan can be calculated, and the system can recover quickly from a communication disruption. M580 Ethernet I/O devices include:

- local rack (including a controller with Ethernet I/O scanner service)
- RIO drop (including an Ethernet X80 EIO adapter module)
- DRS switch with a predefined configuraton

main ring:

The main ring of an Ethernet RIO network. The ring contains RIO modules and a local rack (containing a CPU with Ethernet I/O scanner service) and a power supply module.

MAST:

A master (MAST) task is a deterministic processor task that is run through its programming software. The MAST task schedules the RIO module logic to be solved in every I/O scan. The MAST task has two sections:

- IN: Inputs are copied to the IN section before execution of the MAST task.
- OUT: Outputs are copied to the OUT section after execution of the MAST task.

MB/TCP:

(*Modbus over TCP protocol*) This is a Modbus variant used for communications over TCP/IP networks.

%M:

According to the CEI standard, `%M` indicates a language object of type memory bit.

MIB:

(*management information base*) A virtual database used for managing the objects in a communications network. See SNMP.

Modbus:

Modbus is an application layer messaging protocol. Modbus provides client and server communications between devices connected on different types of buses or networks. Modbus offers many services specified by function codes.

multicast:

A special form of broadcast where copies of the packet are delivered to only a specified subset of network destinations. Implicit messaging typically uses multicast format for communications in an EtherNet/IP network.

%MW:

According to the CEI standard, `%MW` indicates a language object of type memory word.

N

network:

There are two meanings:

- In a ladder diagram:

  A network is a set of interconnected graphic elements. The scope of a network is local, concerning the organizational unit (section) of the program containing the network.

- With expert communication modules:

  A network is a set of stations that intercommunicate. The term *network* is also used to define a group interconnected graphic elements. This group then makes up part of a program that may comprise a group of networks.

NIM:

(*network interface module*) A NIM resides in the first position on an STB island (leftmost on the physical setup). The NIM provides the interface between the I/O modules and the fieldbus master. It is the only module on the island that is fieldbus-dependent — a different NIM is available for each fieldbus.

NTP:

(*network time protocol*) Protocol for synchronizing computer system clocks. The protocol uses a jitter buffer to resist the effects of variable latency.

O

O->T:

(*originator to target*) See originator and target.

ODVA:

(*Open DeviceNet Vendors Association*) The ODVA supports network technologies that are based on CIP.

operation network:

An Ethernet-based network containing operator tools (SCADA, client PC, printers, batch tools, EMS, etc.). Controllers are connected directly or through routing of the inter-controller network. This network is part of the control network.

originator:

In EtherNet/IP, a device is considered the originator when it initiates a CIP connection for implicit or explicit messaging communications or when it initiates a message request for un-connected explicit messaging.

P

PAC:

programmable automation controller. The PAC is the brain of an industrial manufacturing process. It automates a process as opposed to relay control systems. PACs are computers suited to survive the harsh conditions of the industrial environment.

port 502:

Port 502 of the TCP/IP stack is the well-known port that is reserved for Modbus TCP communications.

port mirroring:

In this mode, data traffic that is related to the source port on a network switch is copied to another destination port. This allows a connected management tool to monitor and analyze the traffic.

Q

%Q:

According to the CEI standard, `%Q` indicates a language object of type discrete OUT.

QoS:

(*quality of service*) The practice of assigning different priorities to traffic types for the purpose of regulating data flow on the network. In an industrial network, QoS is used to provide a predictable level of network performance.

%QW:

According to the CEI standard, `%QW` indicates a language object of type analog OUT.

R

rack optimized connection:

Data from multiple I/O modules are consolidated in a single data packet to be presented to the scanner in an implicit message in an EtherNet/IP network.

RIO drop:

One of the three types of RIO modules in an Ethernet RIO network. An RIO drop is an M580 rack of I/O modules that are connected to an Ethernet RIO network and managed by an Ethernet RIO adapter module. A drop can be a single rack or a main rack with an extended rack.

RIO network:

An Ethernet-based network that contains 3 types of RIO devices: a local rack, an RIO drop, and a ConneXium extended dual-ring switch (DRS). Distributed equipment may also participate in an RIO network via connection to DRSs.

RPI:

(requested packet interval) The time period between cyclic data transmissions requested by the scanner. EtherNet/IP devices publish data at the rate specified by the RPI assigned to them by the scanner, and they receive message requests from the scanner at each RPI.

RSTP:

(*rapid spanning tree protocol*) Allows a network design to include spare (redundant) links to provide automatic backup paths if an active link stops working, without the need for loops or manual enabling/disabling of backup links.

S

scanner class device:

A scanner class device is defined by the ODVA as an EtherNet/IP node capable of originating exchanges of I/O with other nodes in the network.

scanner:

A scanner acts as the originator of I/O connection requests for implicit messaging in EtherNet/IP, and message requests for Modbus TCP.

service port:

A dedicated Ethernet port on the M580 RIO modules. The port may support the following major functions (depending on the module type):

- port mirroring: for diagnostic use
- access: for connecting HMI/Control Expert/ConneXview to the controller
- extended: to extend the device network to another subnet
- disabled: disables the port, no traffic is forwarded in this mode

SFC:

(*sequential function chart*) Used to graphically represent in a structured manner the operation of a sequential controller. This graphical description of the controller's sequential behavior and of the various resulting situations is created using simple graphic symbols.

simple daisy chain loop:

Often referred to as SDCL, a simple daisy chain loop contains RIO modules only (no distributed equipment). This topology consists of a local rack (containing a controller with Ethernet I/O scanner service), and one or more RIO drops (each drop containing an RIO adapter module).

SMTP:

(*simple mail transfer protocol*) An email notification service that allows controller-based projects to report alarms or events. The controller monitors the system and can automatically create an email message alert with data, alarms, and/or events. Mail recipients can be either local or remote.

SNMP:

(*simple network management protocol*) Protocol used in network management systems to monitor network-attached devices. The protocol is part of the internet protocol suite (IP) as defined by the internet engineering task force (IETF), which consists of network management guidelines, including an application layer protocol, a database schema, and a set of data objects.

SNTP:

(*simple network time protocol*) See NTP.

SOE:

(*sequence of events*) The process of determining the order of events in an industrial system and correlating those events to a real-time clock.

ST:

(*structured text*) The structured literal language is a developed language similar to computer programming languages. It can be used to organize a series of instructions.

sub-ring:

An Ethernet-based network with a loop attached to the main ring, via a dual-ring switch (DRS) on the main ring. This network contains RIO drops or distributed equipment.

subnet mask:

The 32-bit value used to hide (or mask) the network portion of the IP address and thereby reveal the host address of a device on a network using the IP protocol.

%SW:

According to the CEI standard, `%SW` indicates a language object of type system word.

switch:

A multi-port device used to segment the network and limit the likelihood of collisions. Packets are filtered or forwarded based upon their source and destination addresses. Switches are capable of full-duplex operation and provide full network bandwidth to each port. A switch can have different input/output speeds (for example, 10, 100 or 1000Mbps). Switches are considered OSI layer 2 (data link layer) devices.

T

T->O:

(*target to originator*) See target and originator.

target:

In EtherNet/IP, a device is considered the target when it is the recipient of a connection request for implicit or explicit messaging communications, or when it is the recipient of a message request for un-connected explicit messaging.

TCP/IP:

Also known as *internet protocol suite*, TCP/IP is a collection of protocols used to conduct transactions on a network. The suite takes its name from two commonly used protocols: transmission control protocol and internet protocol. TCP/IP is a connection-oriented protocol that is used by Modbus TCP and EtherNet/IP for explicit messaging.

TCP:

(*transmission control protocol*) A key protocol of the internet protocol suite that supports connection-oriented communications, by establishing the connection necessary to transmit an ordered sequence of data over the same communication path.

TFTP:

(*trivial file transfer protocol*) A simplified version of *file transfer protocol* (FTP), TFTP uses a client-server architecture to make connections between two devices. From a TFTP client, individual files can be uploaded to or downloaded from the server, using the user datagram protocol (UDP) for transporting data.

TIME_OF_DAY:

See TOD.

TOD:

(*time of day*) The TOD type, encoded in BCD in a 32-bit format, contains this information:

- the hour encoded in an 8-bit field
- the minutes encoded in an 8-bit field
- the seconds encoded in an 8-bit field

NOTE: The eight least significant bits are not used.

The TOD type is entered in this format: xxxxxxxx: **TOD#**<Hour>**:**<Minutes>**:**<Seconds>

This table shows the upper/lower limits of each field:

Field	Limits	Comment
Hour	[00,23]	The leading 0 is displayed; it can be omitted during data entry.
Minute	[00,59]	The leading 0 is displayed; it can be omitted during data entry.
Second	[00,59]	The leading 0 is displayed; it can be omitted during data entry.

Example: TOD#23:59:45.

trap:

A trap is an event directed by an SNMP agent that indicates one of these events:

- A change has occurred in the status of an agent.
- An unauthorized SNMP manager device has attempted to get data from (or change data on) an SNMP agent.

TR:

(*transparent ready*) Web-enabled power distribution equipment, including medium- and low-voltage switch gear, switchboards, panel boards, motor control centers, and unit substations. Transparent Ready equipment allows you to access metering and equipment status from any PC on the network, using a standard web browser.

U

UDP:

(*user datagram protocol*) A transport layer protocol that supports connectionless communications. Applications running on networked nodes can use UDP to send datagrams to one another. Unlike TCP, UDP does not include preliminary communication to establish data paths or provide data ordering and checking. However, by avoiding the overhead required to provide these features, UDP is faster than TCP. UDP may be the preferred protocol for time-sensitive applications, where dropped datagrams are preferable to delayed datagrams. UDP is the primary transport for implicit messaging in EtherNet/IP.

UTC:

(*coordinated universal time*) Primary time standard used to regulate clocks and time worldwide (close to former GMT time standard).

V

variable:

Memory entity of type `BOOL`, `WORD`, `DWORD`, etc., whose contents can be modified by the program currently running.

VLAN:

(*virtual local area network*) A local area network (LAN) that extends beyond a single LAN to a group of LAN segments. A VLAN is a logical entity that is created and configured uniquely using applicable software.

Index

W

X

Schneider Electric
35 rue Joseph Monier
92500 Rueil Malmaison
France

+ 33 (0) 1 41 29 70 00

www.se.com

As standards, specifications, and design change from time to time, please ask for confirmation of the information given in this publication.

NHA58880.09