Modicon M580 Hot Standby

System Planning Guide for Frequently Used Architectures

(Original Document)

03/2019

NHA58880.06

www.schneider-electric.com

Schneider Electric

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

You agree not to reproduce, other than for your own personal, noncommercial use, all or part of this document on any medium whatsoever without permission of Schneider Electric, given in writing. You also agree not to establish any hypertext links to this document or its content. Schneider Electric does not grant any right or license for the personal and noncommercial use of the document or its content, except for a non-exclusive license to consult it on an "as is" basis, at your own risk. All other rights are reserved.

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

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

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

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

© 2019 Schneider Electric. All rights reserved.

Table of Contents

Safety Information 5
About the Book 9
Chapter 1 Introducing the M580 Hot Standby System 15
Primary and Standby PACs 16
PAC Switchover in an M580 Hot Standby System 17
Physical Description of M580 Hot Standby CPUs 22
Standards and Certifications 26
Hot Standby System States 27
Configuration Compatibility 30
Chapter 2 Hardware Components in an M580 Hot Standby System 33
M580 Hot Standby Local Racks 34
M580 (e)X80 RIO Drops 39
Quantum RIO Drops 41
SD Memory Card 42
Chapter 3 Planning a Typical M580 Hot Standby Topology 45
Project Life Cycle 46
Modicon M580 Typical System Introduction 47
Planning an M580 Hot Standby Topology 51
Frequently Used M580 Hot Standby Topologies 54
M580 Hot Standby Topologies Using BMENOC0321 Module for Control Network Connection 71
Managing Flat Ethernet Networks with M580 Hot Standby 80
Chapter 4 Working with M580 Hot Standby Applications 93
Modicon M580 Hot Standby Programming Rules 94
M580 Hot Standby System Configuration 97
Configuring an M580 Hot Standby CPU 98
Change Configuration On The Fly (CCOTF) 102
Modifying an SFC Section Online 104
Configuring IP Addresses for an M580 Hot Standby System 105
Configuring Data Variables for an M580 BMEH58•040(S) Hot Standby Application 107
Configuring Hold Up Time for Drops and Devices 109
Transferring M580 Hot Standby Projects 110
Offline Application Modification with Allowed Application Mismatch . . 112
Restoring and Backing Up Projects 115

Chapter 5 Managing M580 Hot Standby Data Exchanges 117
Exchanging M580 Hot Standby Data 118
Hot Standby DDT Data Structure 121
Data Storage Elementary Functions 128
Chapter 6 M580 Hot Standby System Operation 131
Starting an M580 Hot Standby System 132
Hot Standby State Assignments and Transitions 135
Hot Standby System State Examples 138
Executing Hot Standby Commands 146
Memory Usage 149
Chapter 7 M580 Hot Standby System Performance 151
System Performance 152
Calculating the Minimum Task Cycle Time 154
Application Response Time for M580 Hot Standby CPUs 157
Chapter 8 M580 Hot Standby Diagnostics 163
8.1 M580 Hot Standby CPU LEDs 164
LED Diagnostics for M580 Hot Standby CPUs 165
Memory Card Access LED 169
8.2 Control Expert M580 Hot Standby Diagnostics 171
M580 Hot Standby System Diagnostics in Control Expert 172
Synchronizing Configuration of Distributed Equipment 174
8.3 M580 Hot Standby CPU Web Pages 176
Introducing the M580 Hot Standby CPU Web Pages 177
Status Summary (Hot Standby CPUs) 179
HSBY Status 182
Rack Viewer 185
8.4 M580 Hot Standby System Diagnostics 188
M580 Hot Standby System Diagnostics 188
8.5 M580 System Words 190
Modicon M580-specific System Words %SW132 to %SW167 190
Chapter 9 Updating Hot Standby System Firmware 191
Firmware Update 191
Chapter 10 Replacing M580 Hot Standby CPUs 195
Replacing Hot Standby Hardware Modules 195
Chapter 11 Verifying the Network Configuration 199
Using the Ethernet Network Manager 199
Glossary 203
Index 221

Safety Information

Important Information

NOTICE

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



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



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

⚠ DANGER

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

⚠ WARNING

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

⚠ CAUTION

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

NOTICE

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

PLEASE NOTE

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

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

BEFORE YOU BEGIN

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

⚠ WARNING

UNGUARDED EQUIPMENT

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

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

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

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

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

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

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

START-UP AND TEST

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

⚠ WARNING

EQUIPMENT OPERATION HAZARD

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

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

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

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

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

Before energizing equipment:

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

OPERATION AND ADJUSTMENTS

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

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

About the Book

At a Glance

Document Scope

PlantStruxure 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 PlantStruxure 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 simple 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 is valid for the Modicon M580 Hot Standby system when used with EcoStruxure™ Control Expert 14.0 or later.

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

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

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

Related Documents

Title of Documentation	Reference Number
Modicon M580 Hardware Reference Manual	EIO0000001578 (English), EIO0000001579 (French), EIO0000001580 (German), EIO0000001581 (Spanish), EIO0000001582 (Italian), EIO0000001583 (Chinese)
Modicon M580 Standalone System Planning Guide for Frequently Used Architectures	HRB62666 (English), HRB65318 (French), HRB65319 (German), HRB65320 (Italian), HRB65321 (Spanish), HRB65322 (Chinese)
Modicon M580 System Planning Guide for Complex Topologies	NHA58892 (English), NHA58893 (French), NHA58894 (German), NHA58895 (Italian), NHA58896 (Spanish), NHA58897 (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)

Title of Documentation	Reference Number
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 Remote I/O Modules Installation and Configuration Guide	EIO0000001584 (English), EIO0000001585 (French), EIO0000001586 (German), EIO0000001587 (Spanish), EIO0000001588 (Italian), EIO0000001589 (Chinese)
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 Change Configuration on the Fly User Guide	EIO0000001590 (English), EIO0000001591 (French), EIO0000001592 (German), EIO0000001593 (Spanish), EIO0000001594 (Italian), EIO0000001595 (Chinese)
Modicon M340/X80 BMXNRP0200/0201 Fiber Converter Module User Guide	EIO0000001108 (English), EIO0000001109 (French), EIO0000001110 (German), EIO0000001111 (Spanish), EIO0000001112 (Italian), EIO0000001113 (Chinese)
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)

Title of Documentation	Reference Number
Modicon M340/X80 with EcoStruxure™ Control Expert Analog Input/Output Modules User Manual	35011978 (English), 35011979 (German), 35011980 (French), 35011981 (Spanish), 35011982 (Italian), 35011983 (Chinese)
Modicon M340/X80 with EcoStruxure™ Control Expert Discrete Input/Output Modules User Manual	35012474 (English), 35012475 (German), 35012476 (French), 35012477 (Spanish), 35012478 (Italian), 35012479 (Chinese)
Modicon M340/X80 with EcoStruxure™ Control Expert BMXEHC0200 Counting Module User Manual	35013355 (English), 35013356 (German), 35013357 (French), 35013358 (Spanish), 35013359 (Italian), 35013360 (Chinese)
Grounding and Electromagnetic Compatibility of PLC Systems Basic Principles and Measures User Manual	33002439 (English), 33002440 (French), 33002441 (German), 33002442 (Spanish), 33003702 (Italian), 33003703 (Chinese)
Control Panel Technical Guide: How to protect a machine from malfunctions due to electromagnetic distgurbance	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)

Title of Documentation	Reference Number
EcoStruxure™ Control Expert Installation Manual	35014792 (French), 35014793 (English), 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 https://www.schneider-electric.com/en/download

Chapter 1
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 CPU modules
- LEDs and switches
- modes of operation

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

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Primary and Standby PACs	16
PAC Switchover in an M580 Hot Standby System	17
Physical Description of M580 Hot Standby CPUs	22
Standards and Certifications	26
Hot Standby System States	27
Configuration Compatibility	30

Primary and Standby PACs

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

The other PAC acts as the standby PAC. The primary PAC updates the standby PAC 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 PACs are running, either PAC can enter the primary state. When one running PAC becomes primary, the other running PAC may be in the standby or wait state.

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

Role of the Hot Standby PAC

The system monitors itself continuously. If a triggering event occurs *(see page 17)*, the Hot Standby system switches control to the standby, which then becomes the primary PAC. If the standby PAC stops communications, the primary continues to operate without a backup.

PAC Switchover in an M580 Hot Standby System

Introduction

The purpose of a Hot Standby system is to be ready to perform a switchover, if needed. A switchover is the immediate transfer of control of the network from the primary PAC to the standby PAC. The transfer needs to be swift and seamless.

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 PAC and the standby PAC check the health of the system.

The primary PAC checks the health of the following:
- the Ethernet RIO network link
- the Hot Standby link between the primary and standby CPUs

The standby PAC checks the following:
- the health of the primary PAC
- the identity of modules in both the primary and standby racks
- application versions running in the primary and standby CPUs
- firmware versions of the primary and standby CPUs
- the health of the Hot Standby link between the primary and standby CPUs

Before each MAST task, the primary PAC transfers to the standby PAC system, status and I/O data *(see page 118)*, including date and time data. On switchover, the standby PAC applies this time data and continues the same time stamping sequence. The maximum amount of transferable Hot Standby data depends on the CPU. *(see page 34)*

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

Switchover Causes

Any one of the following events will cause a switchover:
- The primary PAC has encountered a blocking condition *(see Modicon M580, Hardware, Reference Manual)* and entered the HALT state.
- The primary PAC has detected an unrecoverable hardware or system error.
- The primary PAC has received a STOP command from Control Expert or the DDDT.
- An application program is being transferred to the primary CPU.
- Primary PAC power is turned off; a power cycle occurs.
- The following events simultaneously occur:
  - The primary PAC loses communication to all RIO drops.
  - The Hot Standby link is healthy.
  - The standby PAC maintains communication with at least one RIO drop.

Similar to a switchover, a swap is a controlled event that transfers control of the network from the primary PAC to the standby PAC. A swap can be caused by:

- Execution of the DDDT `CMD_SWAP` command by either program logic, or an animation table **Force** command.
- Manually clicking the **HSBY Swap** button in the **Task** tab of the CPU **Animation** window in Control Expert.

Events that Do Not Cause Switchover

These events **DO NOT** cause a switchover:

- simultaneous interruption of communication with all RIO drops by both the primary and the standby PAC
- partial interruption of communication with the RIO drops by the primary PAC
- a Modbus connection break
- overload broadcast traffic generated by a peer (for example, SCADA, or another PAC)
- a BMENOC0301/11 module that stops operating
- removal of an SD memory card *(see page 42)*
- for a Hot Standby safety system, if the primary PAC is partially (either the SAFE program or the PROCESS program) in the HALT state, and not all of the tasks in the standby PAC are in RUN

Switchover Execution Time

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

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

- 15 ms for the I/O driven by the MAST task.
- 15 ms + T_{TASK} for the I/O driven by the FAST or the SAFE task, where T_{TASK} is the configured execution period for that task.

The application response time for a swap *(see page 160)* or a switchover *(see page 161)* can be calculated.

After the switchover, the former standby PAC becomes the primary. In the worst case, the new primary PAC operates with data of scan cycle N, while the outputs have received (from the former primary PAC) data of scan cycle N+1. The new primary PAC re-evaluates outputs beginning with scan N+1.

Because the Hot Standby switchover evaluation occurs during the MAST task, some FAST task program execution may be skipped.

Switchover Effect on Main IP Address Assignments

Distributed equipment uses the **Main IP address** setting, configured in the **IPConfig** tab *(see page 105)*, to communicate over an Ethernet network with the primary CPU. On switchover, the **Main IP address** setting is automatically transferred from the former primary CPU to the former standby – now the new primary – CPU. Similarly, on switchover the **Main IP address + 1** setting is automatically transferred from the former standby CPU to the new standby.

In this way, the configured links between the distributed equipment and the primary CPU 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 means of the A/B/Clear rotary switch *(see page 23)* on the back of the CPU, and are not affected by a change in primary or standby Hot Standby status.
- When connecting Control Expert to the Hot Standby system, use **IP address A** or **IP address B** to maintain the connection on a switchover. Avoid using the **Main IP address**, because on switchover this becomes **Main IP address + 1** and will disconnect Control Expert.

Switchover Effect on Remote Outputs

For RIO drops, the switchover is bumpless: the state of outputs is not affected by the switchover. During Hot Standby operations, each PAC maintains an independent, redundant owner connection with each RIO drop. Each PAC makes this connection via **IP address A** or **IP address B**, depending on the A/B/Clear rotary switch *(see page 23)* designation for its CPU. When a switchover occurs, the new primary PAC continues to communicate with I/O via its pre-existing redundant owner connection.

NOTE: The switchover may not be bumpless with respect to 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 will most likely go to fallback when the connection with the primary PAC is interrupted, and will recover their state after reconnecting with the new primary PAC.

To achieve bumpless behavior, the outputs need to support a sufficiently long hold up time *(see page 109)*.

Switchover Effect on CCOTF Changes

After the standby PAC becomes the new primary, it operates using both the firmware and the application previously configured in it. If CCOTF *(see page 102)* changes were previously made to the former primary PAC that were not transferred to the former standby PAC, these changes are not included in the configuration running in the new primary PAC.

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

Switchover Effect on Program Logic Changes

A logic mismatch condition exists when changes have been made to the application in the primary CPU, but not to the standby CPU. If the `LOGIC_MISMATCH_ALLOWED` *(see page 122)* flag is set, the standby CPU can continue to operate as standby while a logic mismatch exists. In this case, if a switchover occurs, the new primary CPU executes its own, different application using data received from the former primary CPU.

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

Modification to initial primary CPU logic:	Effect on new primary CPU program execution:
Only code is changed (no changes to variables).	All variable values exchanged between the controllers remain the same (EQUAL).
New variables were added.	The new variables are not used by the new primary CPU.
Existing variables were deleted.	The new primary CPU includes the deleted variables in program execution, and applies the most recent values to these variables.

Switchover Effects on Time Management

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

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

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

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

- time of day
- application counters
- free running counter

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

Switchover Effects on IPsec Connections

On switchover, the former primary BMENOC0301/11 module closes all connections that use its main IP address. These connections are re-opened on the new primary BMENOC0301/11 module using the main IP address after the two modules swap their main and main+1 IP addresses.

Because IPsec connections take a relatively long time to establish, it can take up to 5 minutes to re-establish an IPSEC connection that uses the main IP address.

Switchover Effect on Safety Operating Mode

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

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

Recovery of Former Primary PAC

The former primary PAC may or may not become the standby PAC, depending on cause of switchover.

If the switchover was caused by:	Make the former primary PAC the standby by:
Primary halt (non-safety PAC)	performing an `INIT` command and RUN the PAC
Primary halt (safety PAC - Process and/or SAFE task)	performing an `INIT` command (Process task) and/or an `INIT_SAFETY` command (SAFE task), and then RUN the PAC
PAC stop in a non-safety PAC, or in both the Process and SAFE tasks of a safety PAC	running the PAC
Primary error detected	performing a CPU `RESET` command
Application transfer on Primary	completing the transfer and RUN the application
Primary power off	powering up the PAC
Loss of all RIO drops (if any) while HSBY link is still healthy and Standby CPU has access to the drops	causing the PAC to recover RIO drops
DDDT command	The former primary automatically becomes the standby, provided the necessary preconditions exist, 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.
Control Expert **HSBY Swap** button	

Physical Description of M580 Hot Standby CPUs

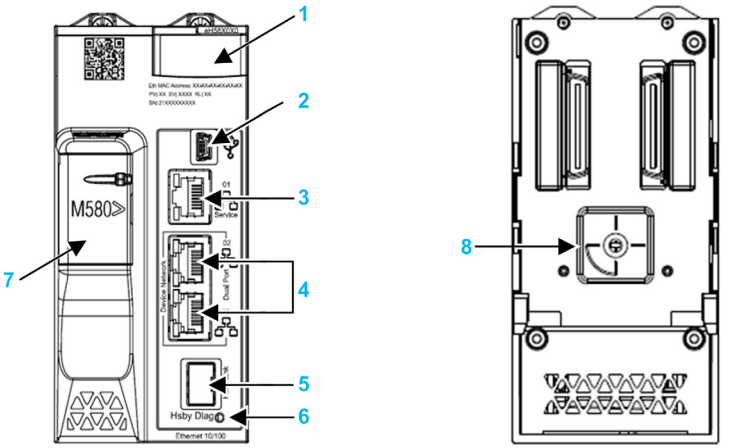
PAC Hot Standby CPU Modules

These M580 CPU modules support M580 Hot Standby systems:

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

CPU Module Front and Back Views

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

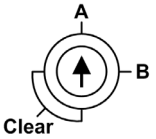


1 LED diagnostic display panel
2 Mini-B USB port for module configuration via PC running Control Expert
3 RJ45 Ethernet service port connector
4 RJ45 connectors that together serve as a dual port to the Ethernet network
5 SFP socket for copper or fiber-optic Hot Standby link connection
6 Hot Standby status link LED
7 SD memory card slot
8 A/B/Clear rotary selector switch, used to designate the PAC as either PAC A or PAC B, or to clear the existing Control Expert application

NOTE: The only visible difference between safety and non-safety CPUs is that safety CPUs are colored red.

Rotary Selector Switch

Use the rotary switch on the back of each M580 Hot Standby CPU to designate the role that the CPU plays in the M580 Hot Standby configuration:



Use the screwdriver provided with the CPU to set the rotary switch according to its role in a Hot Standby system:

Position	Result
A	• Designates the PAC as PAC A *(see page 133)*, as referenced in Control Expert and the `T_M_ECPU_HSBY` *(see page 122)* DDDT. • Assigns the PAC **IP address A** on Ethernet RIO network.
B	• Designates the PAC as PAC B *(see page 133)*, as referenced in Control Expert and the `T_M_ECPU_HSBY` DDDT. • Assigns the PAC **IP address B** on Ethernet RIO network.
Clear	• Clears the application in the PAC, and places the PAC into the `NO_CONF` operational state. • If an SD memory card is inserted in the PAC, the application in the card is also cleared, **NOTE:** Setting the switch for each Hot Standby PAC to the same A/B position can cause a a conflict of PAC roles *(see page 133)*.

Clearing CPU Memory

To clear a CPU memory, follow these steps:

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

When you next power up the PAC, if the remote PAC is primary, the primary PAC transfers the application to the local PAC.

SFP Socket

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



To insert a transceiver:

Step	Action
1	Check that the CPU is powered off.
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, check the orientation of the transceiver and repeat these steps.

To remove a transceiver:

Step	Action
1	Check that the CPU is powered off.
2	Pull out the latch to unlock the transceiver.
3	Pull on the transceiver to remove it.

NOTICE

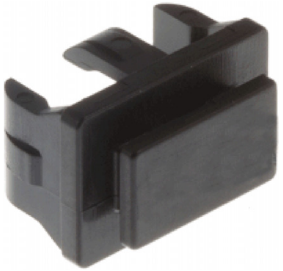
POTENTIAL EQUIPMENT DAMAGE

Do not Hot Swap the SFP transceiver. Insert or remove the transceiver only when there is no power to the CPU.

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 CPU Hot Standby link transceivers *(see page 36)*.

Each module 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



Grounding Considerations

⚠ ⚠ DANGER

ELECTRICAL SHOCK HAZARD

- Switch off the power supply at both ends of the PAC connection, and lock out and tag out both power sources before you insert or remove an Ethernet cable.
- In case lock out and tag out are not available, ensure that the power sources cannot be inadvertently switched on.
- Use suitable insulation equipment when you insert or remove an Ethernet cable.

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

Do not apply power to a Modicon X80 rack until connections are made at both ends of the Ethernet cable. For example, make these connections before you turn on the power:

- Connect the Ethernet cable to the BMECRA31210 performance EIO adapter module and another device (adapter module) or dual-ring switch (DRS). (Refer to the *Modicon M580 System Planning Guide for Complex Topologies (see Modicon M580, System Planning Guide for, Complex Topologies)* for details about DRSs.)
- Connect the copper Ethernet cable to both SFP transceivers when you use 490NAC0100 copper transceivers.

Use fiber-optic cable to establish a communications link when it is not possible to master the potential between the two grounds.

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 the standards and certifications (PDF format) that apply to the modules in this product line:

Language	
English	*Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications*
French	*Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications*
German	*Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications*
Italian	*Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications*
Spanish	*Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications*
Chinese	*Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications*

Safety-Specific System Certifications

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

Hot Standby System States

PAC State Versus Hot Standby System State

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

PAC Operating State	Hot Standby System State
INIT	INIT
STOP	STOP
RUN	PRIMARY with standby counterpart
	PRIMARY without standby counterpart
	STANDBY
	WAIT

This list describes the Hot Standby states:

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

  In a Hot Standby network, both PACs can be primary if both the Hot Standby and Ethernet RIO links are not functioning. When either of these two links is restored, the PAC does one of the following:
  - Remains in the primary state.
  - Transitions to the standby state.
  - Transitions to the wait state.
- **Standby**: The standby PAC maintains a state of readiness. It can take control of system processes and devices if the primary PAC cannot continue to perform these functions:
  - It reads the data and the I/O states from the primary PAC.
  - It does not scan distributed equipment, but receives this information from the primary PAC.
  - It executes program logic. You can configure the standby PAC to execute:
    - The first section of program logic (the default setting); or
    - Specified sections of program logic, including all MAST and FAST task sections.

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

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

- **Wait**: The PAC is in RUN mode, but cannot act as either primary or standby. The PAC 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 rotary selection switch on the rear of the CPU.
  - 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 PAC continues to communicate with other modules on the local rack, and can execute program logic, if configured to do so. You can configure a PAC in wait state to execute:
  - Specific sections of program logic in a non-safety PAC (or process program logic in a safety PAC), specified in the **Condition** tab of the **Properties** dialog for each section.
  - The first section of program logic in a non-safety PAC (or the first section of process program logic in a safety PAC).
  - No program logic for a non-safety PAC (or no process program logic for a safety PAC).
- **INIT**: Both the PAC and the Hot Standby system are initializing.
- **STOP**: The PAC is in STOP mode. On the STOP to RUN transition, the PAC 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 rotary selection switch on the rear of the CPU.

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

PAC Functions by Hot Standby System State

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

PAC functions	Hot Standby system states		
	Primary	Standby	Wait
RIO drops	YES	NO	NO
Distributed equipment	YES	NO	NO
Execution of program logic (non-safety PAC) or process task logic (safety PAC)	YES	Depending on configuration, STANDBY PAC can execute: ● First section (default) ● Specified sections (which can include all MAST and FAST sections) ● None	Depending on configuration, WAIT PAC can execute: ● First section (default) ● Specified sections (which can include all MAST and FAST sections) ● None
Execution of safe logic (safety PAC)	YES	NO	NO
Program Data Exchange (non-safety PAC) or Process Data Exchange (safety PAC)	YES	YES	NO
Safe Data Exchange (safety PAC)	YES	YES	NO
1. Data exchange is controlled by the **Exchange on STBY** attribute.			

Configuration Compatibility

Control Expert Version Requirement

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

PAC Hardware

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

- CPU *(see page 34)*
- Rack *(see page 36)*
- Power supply *(see page 37)*
- Some communication modules *(see page 38)*

NOTE: No I/O modules can be mounted onto the local rack. Refer to the topic The Modicon M580 Hot Standby Local Rack *(see page 34)* for a description of modules that can be added to the local rack.

CPU Compatibility

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

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

For example:

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

The following table depicts the compatibility of applications among non-safety M580 Hot Standby CPUs:

An application built for:	Can be downloaded to and executed by the following CPUs:		
	BMEH582040	BMEH584040	BMEH586040
BMEH582040	X	X	X
BMEH584040	–	X	X
BMEH586040	–	–	X
X: Can receive and execute the application. –: Cannot receive and execute the application.			

The following table depicts the compatibility of applications among M580 safety CPUs:

An application built for:	Can be downloaded to and executed by the following CPUs:				
	BMEP582040S	BMEP584040S	BMEH582040S	BMEH584040S	BMEH586040S
BMEP582040S	1	2	2	2	2
BMEP584040S	3	1	3	2	2
BMEH582040S	2	2	1	2	2
BMEH584040S	3	2	3	1	2
BMEH586040S	3	2	3	3	1
1: Fully compatible 2: Compatible, if CPU is upgraded in Control Expert and the application is fully rebuilt. 3.: Compatible, if CPU is upgraded in Control Expert and the application is fully rebuilt, and there is no limitation as to memory size.					

CPU Firmware Mismatch

An M580 Hot Standby system can continue operating when there is a mismatch of firmware versions in the primary and standby CPUs, if each CPU firmware can execute the application. This makes it possible to upgrade (or downgrade) CPU firmware without having to stop the operation of the Hot Standby system. To permit Hot Standby operations to continue in this case, use an animation table or program logic to set the `FW_Mismatch_Allowed` attribute of the `T_M_ECPU_HSBY` *(see page 122)* to **True**.

Application Mismatch

An M580 Hot Standby system cannot operate if the primary and standby CPUs are equipped with fundamentally different applications. In this case, the primary PAC operates as a standalone PAC, and the standby PAC enters the stop state.

To restore Hot Standby system operations, confirm that the same application is installed in both the primary and standby PACs.

Logic Mismatch

An M580 Hot Standby system can continue operating if the primary and standby CPUs are running different revisions of the same application. In this case, both CPUs were initially configured with the same application, but the logic in one CPU – usually the primary CPU – was subsequently revised.

For Hot Standby operations to continue when a logic mismatch exists, use an animation table or program logic to set the `Logic_Mismatch_Allowed` attribute of the `T_M_ECPU_HSBY` *(see page 122)* DDT to **True**.

For Hot Standby operations to continue when a logic mismatch exists, do both of the following:

- Select **Online modification in RUN or STOP** in the **Configuration** tab of the CPU.
- Set the **Number of modifications** in the **Configuration** tab of the CPU.
- Use an animation table or program logic to set the `Logic_Mismatch_Allowed` attribute of the `T_M_ECPU_HSBY` *(see page 122)* DDT to **True**.

NOTE: If the **Number of modifications** is set to 0, setting the `Logic_Mismatch_Allowed` attribute has no effect.

SFC Mismatch

A sequential function chart (SFC) mismatch occurs when the applications in the primary and standby CPUs include graphic symbols that define sequential program steps, where differences exists in at least one SFC section.

Refer to the topic *Modifying an SFC Section Online* *(see page 104)* for the recommended procedure for making online modifications to an SFC section.

Chapter 2
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.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
M580 Hot Standby Local Racks	34
M580 (e)X80 RIO Drops	39
Quantum RIO Drops	41
SD Memory Card	42

M580 Hot Standby Local Racks

Local Rack Modules

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

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

Both the primary and standby local rack may also include:

- up to six communication modules, depending on your CPU *(see page 34)*
- an SD memory card for each CPU

NOTE: An M580 Hot Standby local rack:

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

CPUs

The Modicon M580 Hot Standby CPUs include these features:

Feature	CPU: BMEH58...					
	2040	2040S	4040	4040S	6040	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)
BMENOC0301/11 Ethernet scanners	2	2	4	4	4	4

1. The BMEH586040 CPU supports the sum of program and data up to the stated maximum.
2. Application Program (non-safe) + Application Data (non-safe non-retain data only) + Application Program (safe) + Application Data (Safe) is less than 64Mbytes. There is a global memory pool of 64 Mbytes on BMEH586040S CPU for Application Program and Application Data.
3. This data is included in both the safe and non-safe data areas.
4. 2 GB without an external memory card.

Feature		CPU: BMEH58...					
		2040	2040S	4040	4040S	6040	6040S
Memory	Non-safe program (MB)	8 MB	8 MB	16 MB	16 MB	64 MB[1]	64 MB[2]
	Non-safe data (KB)	768 KB	768 KB	2048 KB	2048 KB		up to 64 MB[2]
	Safe 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
	Safe data (non-retained data) (MB)	–	512 KB	–	1024 KB	–	1024 KB[4]
	Max configurable safe redundant transfer data	–	512 KB	–	1024 KB	–	1024 KB[5]
	Shared: Global -> Safe	–	16 KB[3]	–	16 KB[3]	–	16 KB[3]
	Shared: Safe -> 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 CPU supports the sum of program and data up to the stated maximum.
2. Application Program (non-safe) + Application Data (non-safe non-retain data only) + Application Program (safe) + Application Data (Safe) is less than 64Mbytes. There is a global memory pool of 64 Mbytes on BMEH586040S CPU for Application Program and Application Data.
3. This data is included in both the safe and non-safe data areas.
4. 2 GB without an external memory card.

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

CPU Hot Standby Link SFP Transceivers

NOTICE

UNINTENDED EQUIPMENT OPERATION

When installing modules with fiber optic transceivers, do the following to help prevent dust and pollution from disrupting light production into the fiber optic cable.

- 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 optical transceivers.

Failure to follow these instructions can result in equipment damage.

Each M580 Hot Standby CPU 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

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 *(see page 42)* for Hot Standby CPUs. 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
BMENOC0301	Ethernet communication module for distributed equipment	up to four communication modules (depends on CPU *(see page 34)*)
BMENOC0311	FactoryCast Ethernet communication module for distributed equipment	
BMENOC0321[2]	Ethernet communication module that provides Ltransparency 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 CPUs 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 CPU tasks.

NOTE: M580 Hot Standby CPUs 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 your choice of CPU *(see page 34)*.

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 0 (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 *(see page 34)* 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 –: not 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.
- 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 System Planning Guide for Frequently Used Architectures (see Modicon M580 Standalone, System Planning Guide for, Frequently Used Architectures)*.

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 PAC.

Quantum RIO Drops

Introduction

The BMEH584040(S) and BMEH586040(S) Hot Standby CPUs support Quantum RIO drops. The Quantum RIO drops can contain only non-safety 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 CPU tasks.

NOTE: M580 Hot Standby CPUs 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 your choice of CPU *(see page 34)*.

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 any 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 PAC.

SD Memory Card

BMXRMS004GPF SD Memory Card

The SD memory card is an option you can use for storing your Hot Standby application. When you power up the CPU, the application in the SD memory card is loaded into flash memory.

The SD memory card slot in both the BMEH58•040 and the BMEH58•040S CPU housing is located behind a door *(see page 22)*. Use only a BMXRMS004GPF memory card in your CPU. It is a 4 GB, Class A card rated for industrial use. Other memory cards, such as those used in the M340 CPUs, are not compatible with the M580 CPUs.

NOTE:
If you insert an incompatible SD memory card, then power up the CPU:

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

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

Memory Card Characteristics

Characteristic	Description
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, wearout, and other internal mechanisms, the actual available capacity of the memory card is slightly lower than its global size.

Supported Functions

The SD memory card supports read-only data storage functions *(see page 128)*.

NOTE: In addition to these read-only data storage functions, you can also read and write to the SD memory card using the following Control Expert project management *(see Modicon M580, Hardware, Reference Manual)* commands located in the **PLC → Project Backup** menu:

- **Backup Compare**
- **Backup Restore**
- **Backup Save**

Formatting the Memory Card is Unnecessary

The SD memory card comes pre-formatted from the factory. There is no need to manually format the SD memory card using your PC. If you attempt to format the SD memory card, you may alter the formatted structure of the card, thereby rendering the card unusable.

Chapter 3
Planning a Typical M580 Hot Standby Topology

Overview

This chapter describes typical Hot Standby topologies.

What Is in This Chapter?

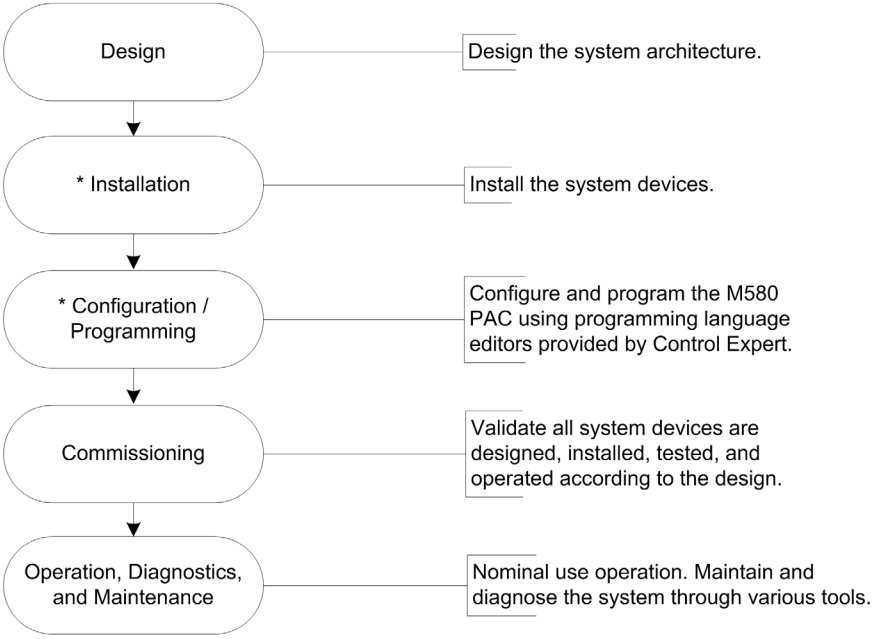
This chapter contains the following topics:

Topic	Page
Project Life Cycle	46
Modicon M580 Typical System Introduction	47
Planning an M580 Hot Standby Topology	51
Frequently Used M580 Hot Standby Topologies	54
M580 Hot Standby Topologies Using BMENOC0321 Module for Control Network Connection	71
Managing Flat Ethernet Networks with M580 Hot Standby	80

Project Life Cycle

Project Life Cycle

Before you turn to the topic of planning your network topology, it may be helpful to see the life cycle of a project within the M580 system.



*** NOTE**: 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 Standalone, System Planning Guide for, Frequently Used Architectures)* and the ability to extend to other local racks
- RIO drops *(see Modicon M580 Standalone, System Planning Guide for, Frequently Used Architectures)* that support Ethernet and X Bus communications across the backplane
- Ethernet distributed equipment *(see Modicon M580 Standalone, System Planning Guide for, Frequently Used Architectures)*
- network option switch modules that attach RIO drops and distributed equipment to the M580 system *(see Modicon M580 Standalone, System Planning Guide for, Frequently Used Architectures)*
- control network module that creates transparency between the device network and the control network *(see page 71)*
- 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

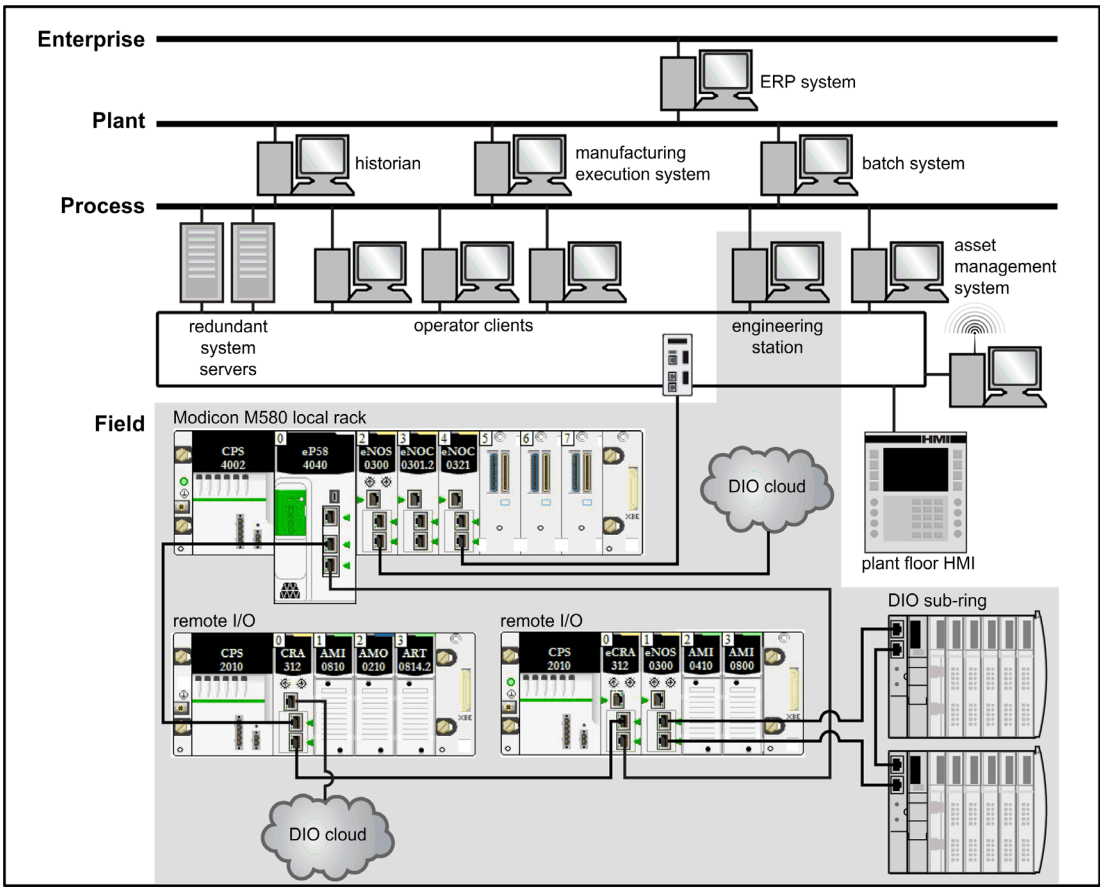
An M580 system provides automatic network recovery of less than 50 ms and *deterministic* RIO performance.

An M580 system uses Modicon X80 I/O modules, many of which are 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 M580 System Planning Guide for Complex Topologies *(see Modicon M580 Standalone, System Planning Guide for, Frequently Used Architectures)*.

Typical M580 Architecture

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



⚠ WARNING

UNEXPECTED EQUIPMENT BEHAVIOR

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

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

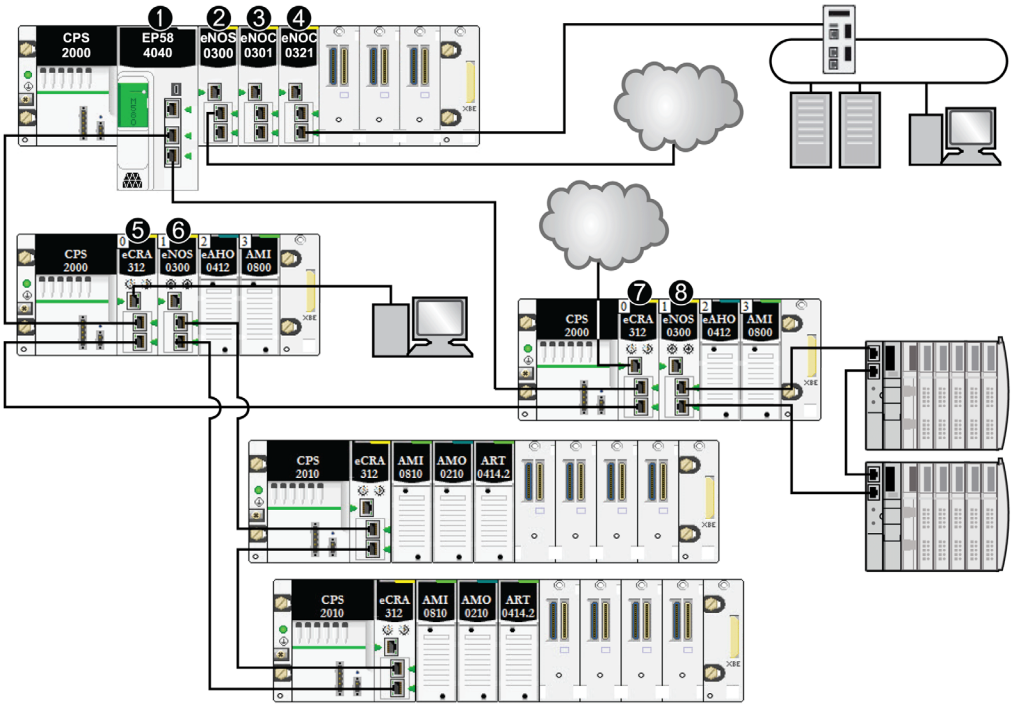
M580 Life Cycle

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 Simple 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 CPU 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 CPU 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/BMENOC0311 Ethernet communication module, connected to the CPU 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 CPU **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 PACs

The BMEH58•040 and BMEH58•040S primary and standby PACs 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 PACs. Although the system can function without an Ethernet link connecting the primary and standby PACs, 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 System Planning Guide for Frequently Used Architectures*.

Hot Standby Link Between Primary and Standby PACs

NOTICE

UNINTENDED EQUIPMENT OPERATION

When installing modules with fiber optic transceivers, do the following to help prevent dust and pollution from disrupting light production into the fiber optic cable:

- 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 optical transceivers.

Failure to follow these instructions can result in equipment damage.

Each BMEH58•040 and BMEH58•040S CPU is fitted with an SFP socket *(see page 36)* for an SFP transceiver, which can be either:

- An SFP copper wire transceiver with RJ45 connector.
- An SFP single mode fiber optic transceiver.

Your choice of connector determines the maximum distance of the Hot Standby physical link.

SFP transceiver name	Media	Maximum Hot Standby Link Length
490 NAC 01 00	CAT 5e copper wire[1]	100 m (328 ft)
490 NAC 02 01	Single mode fiber optic cable	up to 15 km (9.3 mi)
1. Compatible with Gigabit data transfer rate.		

NOTE: BMXNRP020• fiber converter modules cannot be used on the Hot Standby link.

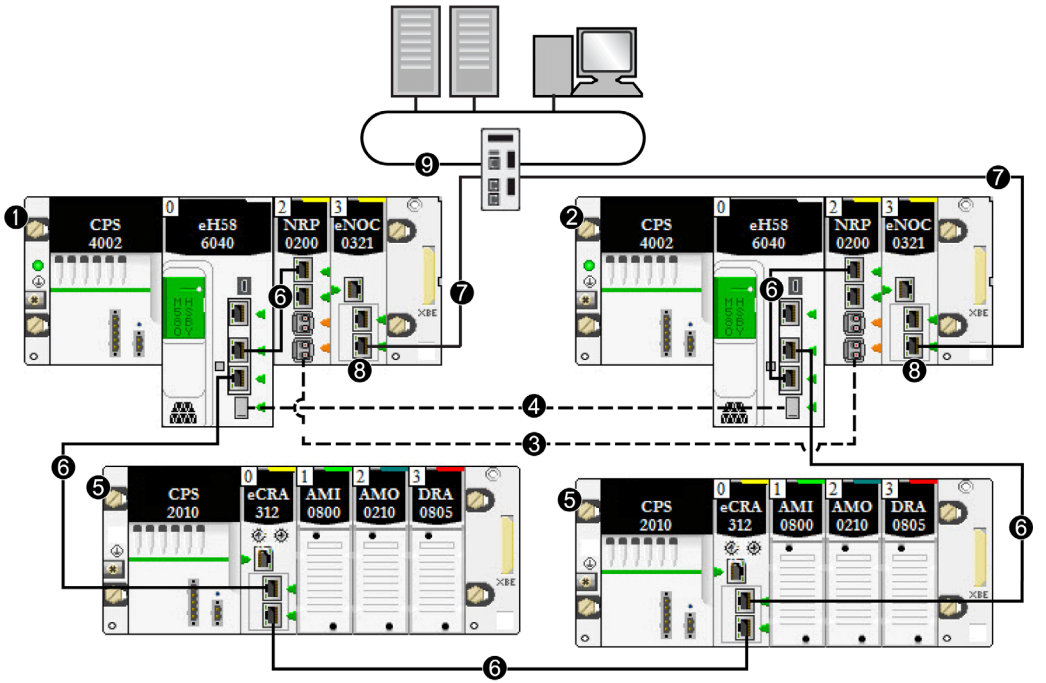
Ethernet Link Between Primary and Standby PACs

The primary and standby BMEH58•040 or BMEH58•040S PACs 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 CPU is connected.

Each BMEH58•040 and BMEH58•040S CPU 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 PACs, connect one of the dual Ethernet ports on the primary CPU to a dual Ethernet port on the standby CPU. You can do this in one of the following ways:

- Directly connect the primary and standby CPUs using CAT-5e copper wire.
- Place a BMENOS0300 switch module *(see page 56)* in each PAC, or place a BMENOC0301/11 communication module *(see page 38)* in each local rack, then connect the two modules.
- Connect each CPU 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 CPU
2 standby local rack with standby CPU
3 Ethernet RIO fiber optic link between primary and standby CPUs (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

A simple M580 Hot Standby system includes, at minimum, two Hot Standby local racks, each containing a Hot Standby CPU 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:

- a simple RIO main ring
- a DIO ring connected to:
  - a BMENOS0300 module, in a design that supports up to 64 devices
  - a BMENOC0301/11 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 CPUs' Ethernet backplanes. If your system contains an RIO main ring or a DIO ring that is connected to a non-isolated BMENOC0301/11 module on the local rack, verify that you isolate (disable the modules' Ethernet backplane ports) other BMENOC0301/11 modules to which you connect a DIO ring.

⚠ CAUTION

RISK OF UNINTENDED OPERATION

Do not connect together the service ports of the Hot Standby CPUs. Connecting together the service ports of the primary and standby CPUs can cause unintended system operation.

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

⚠ 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. Connecting more than one module to both the backplane and an Ethernet network can cause a broadcast storm. ● Use only one module in each local rack to connect an Ethernet network to the Ethernet backplane. That module can be: ❍ the CPU, when RIO modules are used ❍ one BMENOS0300 module ❍ one BMENOC0301/11 module **Failure to follow these instructions can result in injury or equipment damage.**

You can have multiple BMENOC0301/11 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.

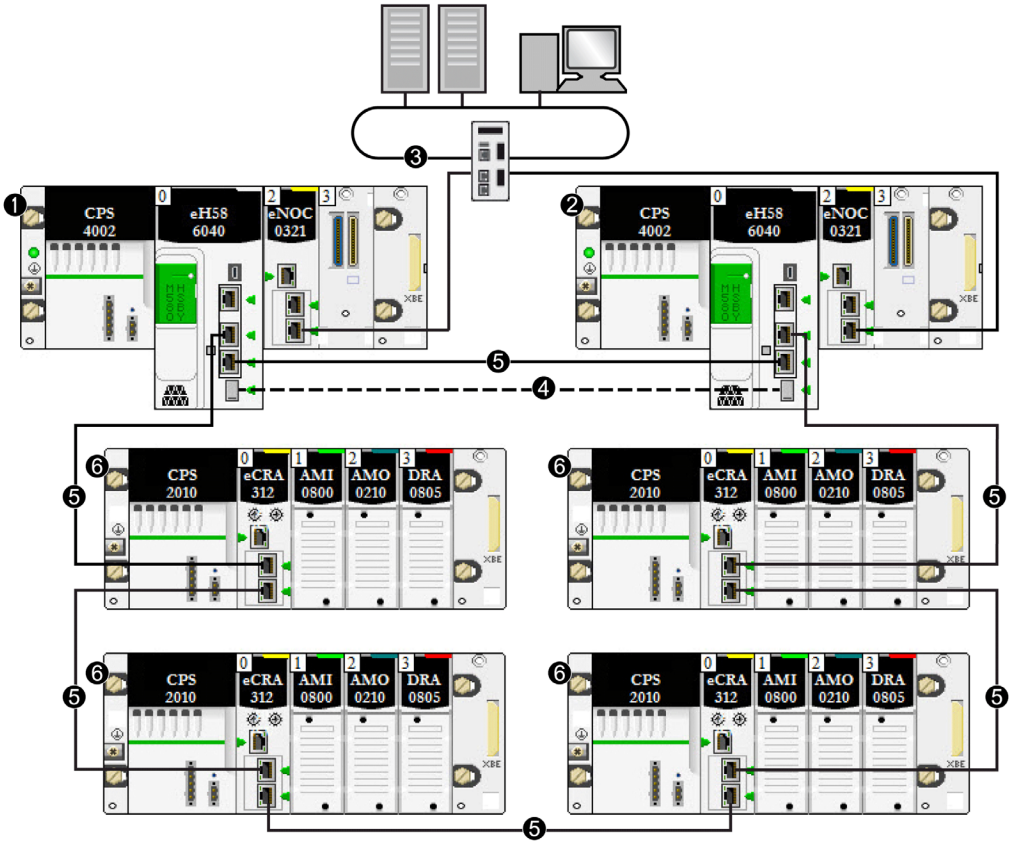
Simple RIO Main Ring Architecture

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

- two local Hot Standby racks, each containing a Hot Standby CPU 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 CPU
2 standby local rack with standby CPU
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 BMENOC0301/11 module. In this design, the CPU alone 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 CPU via the BMENOS0300 modules.

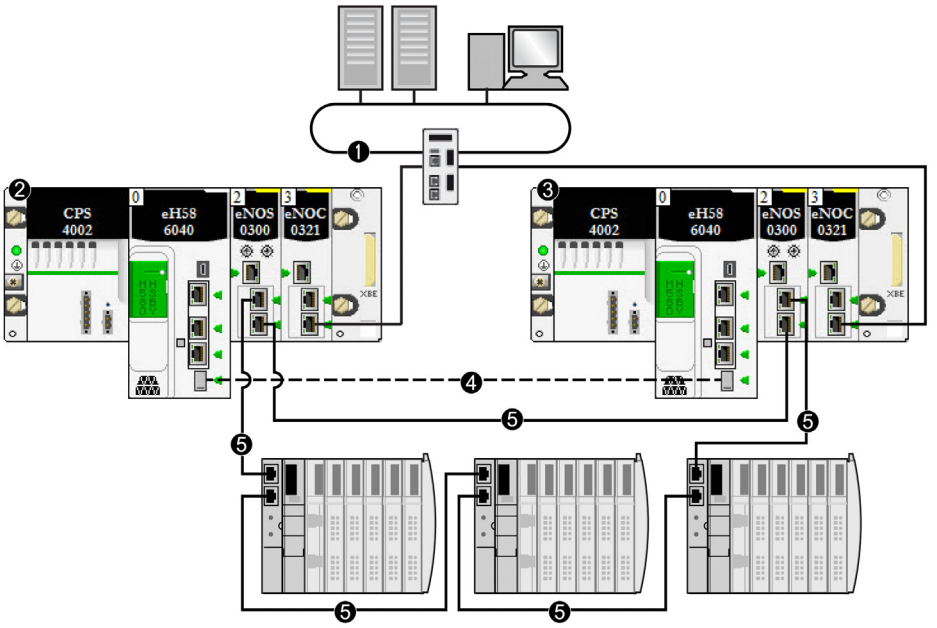
⚠ CAUTION

RISK OF BROADCAST STORM

- Do not connect more than one module in a local rack to both the Ethernet backplane and an Ethernet network. Connecting more than one module to both the backplane and an Ethernet network can cause a broadcast storm.
- Use only one module in each local rack to connect an Ethernet network to the Ethernet backplane. That module can be:
  - the CPU, when RIO modules are used
  - one BMENOS0300 module
  - one BMENOC0301/11 module

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

Because the Ethernet backplane port of each BMENOS0300 module is enabled, connecting a dual Ethernet port on the primary CPU to a dual Ethernet port on the standby CPU in this case 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 CPU
3 standby local rack with standby CPU
4 Hot Standby communication link
5 DIO ring

NOTE: In this design:

- Up to 64 distributed devices can be included.
- Connect the twoBMENOS0300 modules via a direct link.
- If there is a break 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 break occurs.
- No link can be made connecting the dual Ethernet ports on the primary CPU to the dual Ethernet ports on the standby CPU.
- 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 BMENOC0301/11 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.

⚠ CAUTION

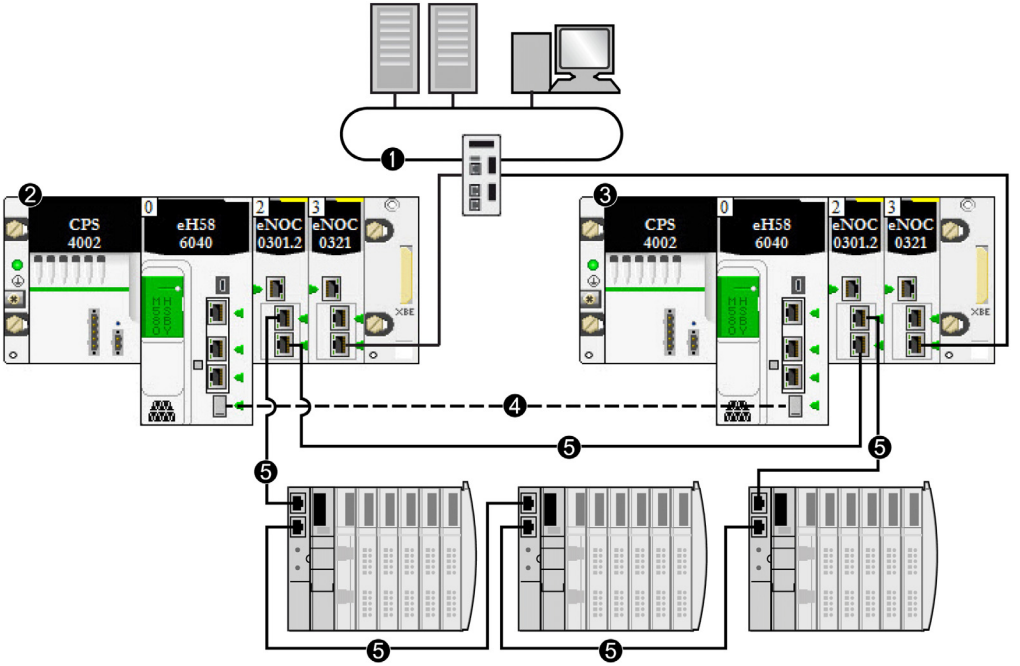
RISK OF UNINTENDED OPERATION

If the Hot Standby CPUs support both an RIO main ring and distributed equipment, you need to disable the Ethernet backplane port of each BMENOC0301/11 module. Enabling the CPU backplane ports in such a design can cause the Hot Standby system to cease functioning.

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

Distributed equipment in a DIO ring can be connected to the service port of a BMENOC0301/11 module. However, if the system also supports an RIO main ring, disable the backplane port of the BMENOC0301/11 module.

This example presents a DIO ring of distributed equipment scanned by BMENOC0301/11 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 CPU
3 standby local rack with standby CPU
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.

  Refer to the Modicon M580 BMENOC0301/11 Ethernet Communications Module Installation and Configuration Guide *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* for more details.
- Connect the two BMENOC0301/11 modules via a direct link.
- In the **Services → RSTP** page of the BMENOC0301/11 DTM in Control Expert, set the **Bridge Priority** to **Root**.

- If there is a break in the direct connection between the two BMENOC0301/11 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 BMENOC0301/11 modules so that the system can remain operational in case a second break occurs.
- Because this design employs a non-isolated BMENOC0301/11 module with its Ethernet backplane port enabled no link can be made connecting the dual Ethernet ports on the primary CPU to the dual Ethernet ports on the standby CPU.
- Up to five additional BMENOC0301/11 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.

Simple RIO Main Ring with DIO Daisy Chain Architecture

You can expand a simple 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.

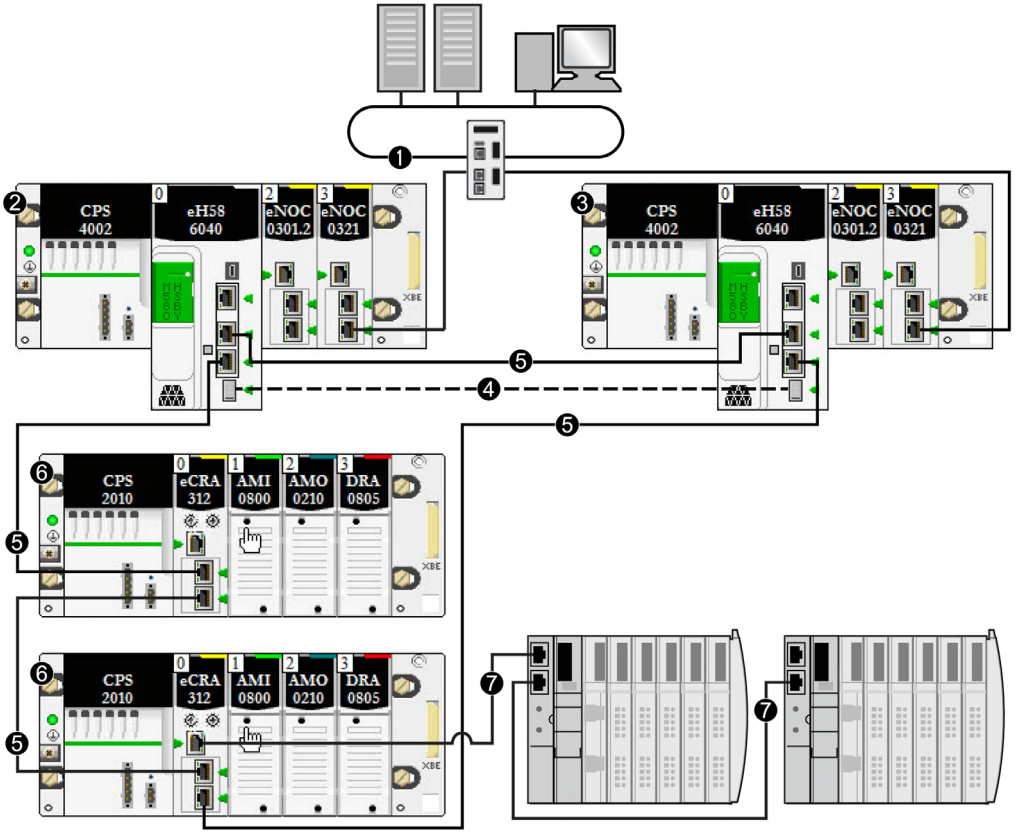
⚠ CAUTION

RISK OF UNINTENDED OPERATION

Because the Ethernet backplane port of each BMENOC0301/11 module is enabled, do not connect the Ethernet RIO ports of the primary CPU to the Ethernet RIO ports of the standby CPU. Connecting the Ethernet RIO ports of the primary and standby CPUs in this design can cause the Hot Standby system to cease functioning.

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

In this example, the distributed equipment is connected to the service port of an 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 CPU
3 standby local rack with standby CPU
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 BMENOC0301/11 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 CPUs.
- Up to 128 distributed devices can be scanned by non-isolated BMENOC0301/11 modules.

Of these 128 connections:
- 16 are reserved for local slaves.
- 112 are available for distributed equipment to be scanned.

Refer to the Modicon M580 BMENOC0301/11 Ethernet Communications Module Installation and Configuration Guide *(see Modicon M580, BMENOC0301/0311 Ethernet Communications Module, Installation and Configuration Guide)* for more details.

Simple 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 CPU scans the RIO main ring and a BMENOC0301/11 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.

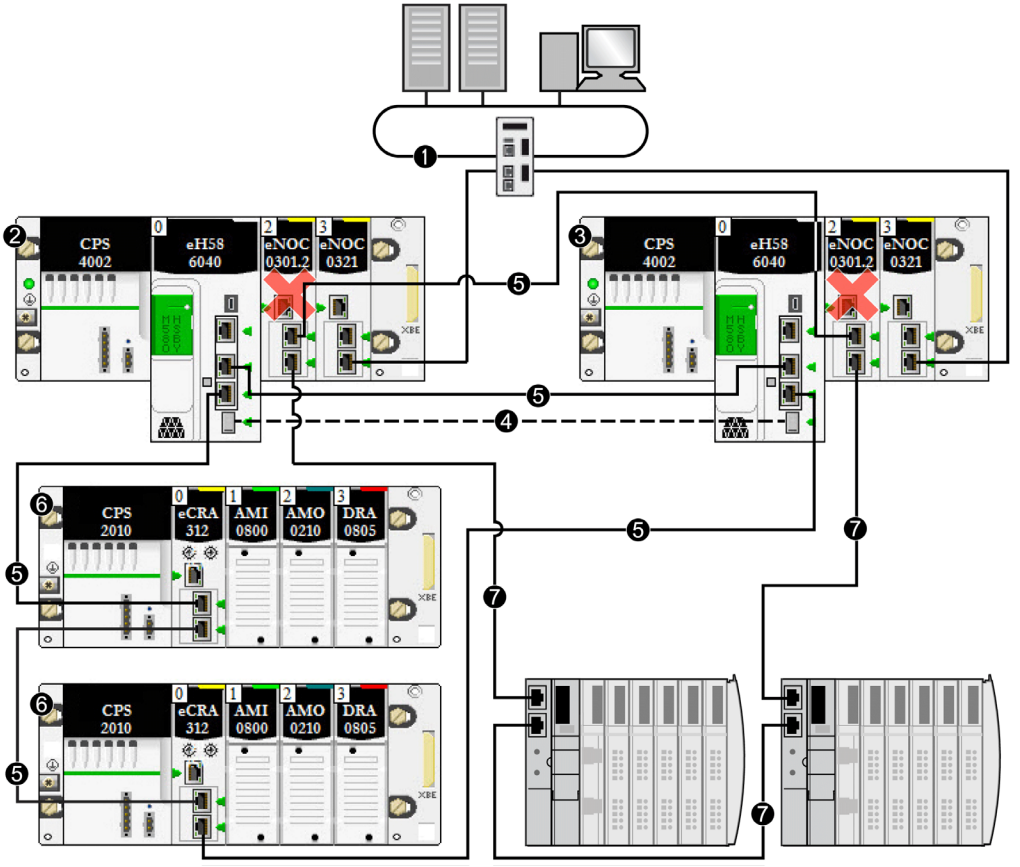
⚠ CAUTION

RISK OF BROADCAST STORM

- Do not connect more than one module in a local rack to both the Ethernet backplane and an Ethernet network. Connecting more than one module to both the backplane and an Ethernet network can cause a broadcast storm.
- Use only one module in each local rack to connect an Ethernet network to the Ethernet backplane. That module can be:
  - the CPU, when RIO modules are used
  - one BMENOS0300 module
  - one BMENOC0301/11 module

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

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 CPU
3 standby local rack with standby CPU
4 Hot Standby communication link
5 RIO main ring
6 (e)X80 RIO drop
7 DIO ring

Simple 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 CPU 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.

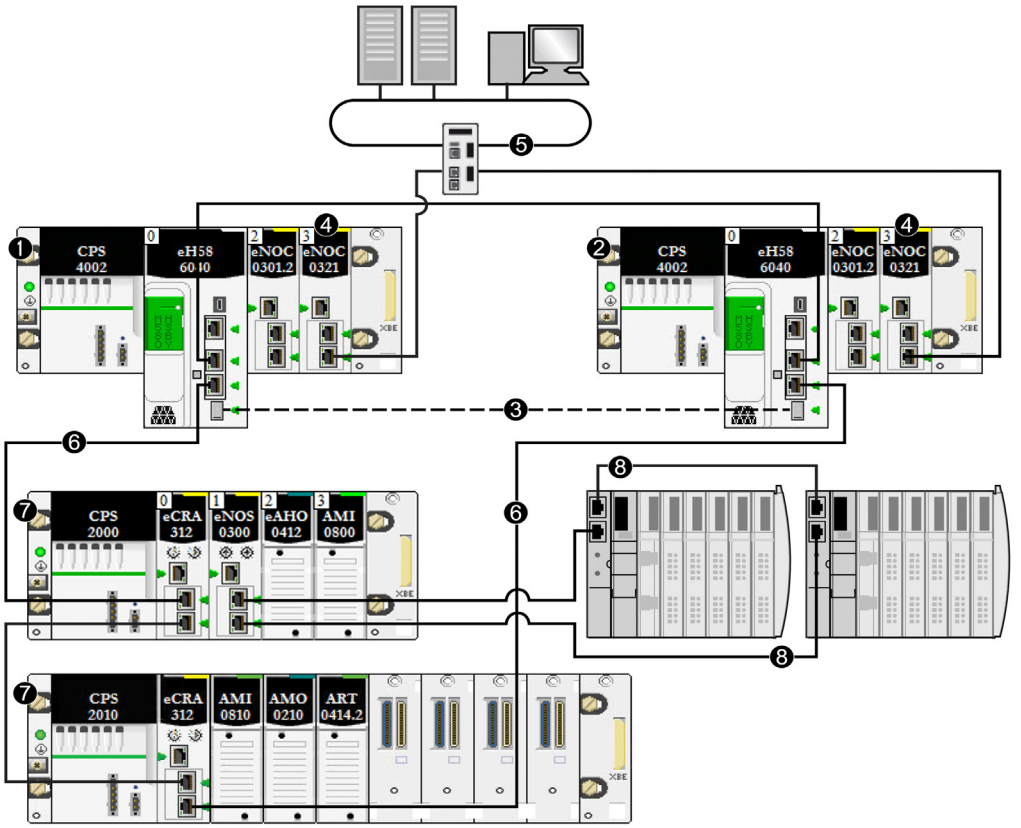
⚠ CAUTION

RISK OF BROADCAST STORM

Do not connect the paired Ethernet ports of the primary BMENOC0301/11 module to the Ethernet ports of the standby BMENOC0301/11 module. Connecting these ports can result in the occurrence of a broadcast storm, which can prevent the Hot Standby network from transporting intended network communications.

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

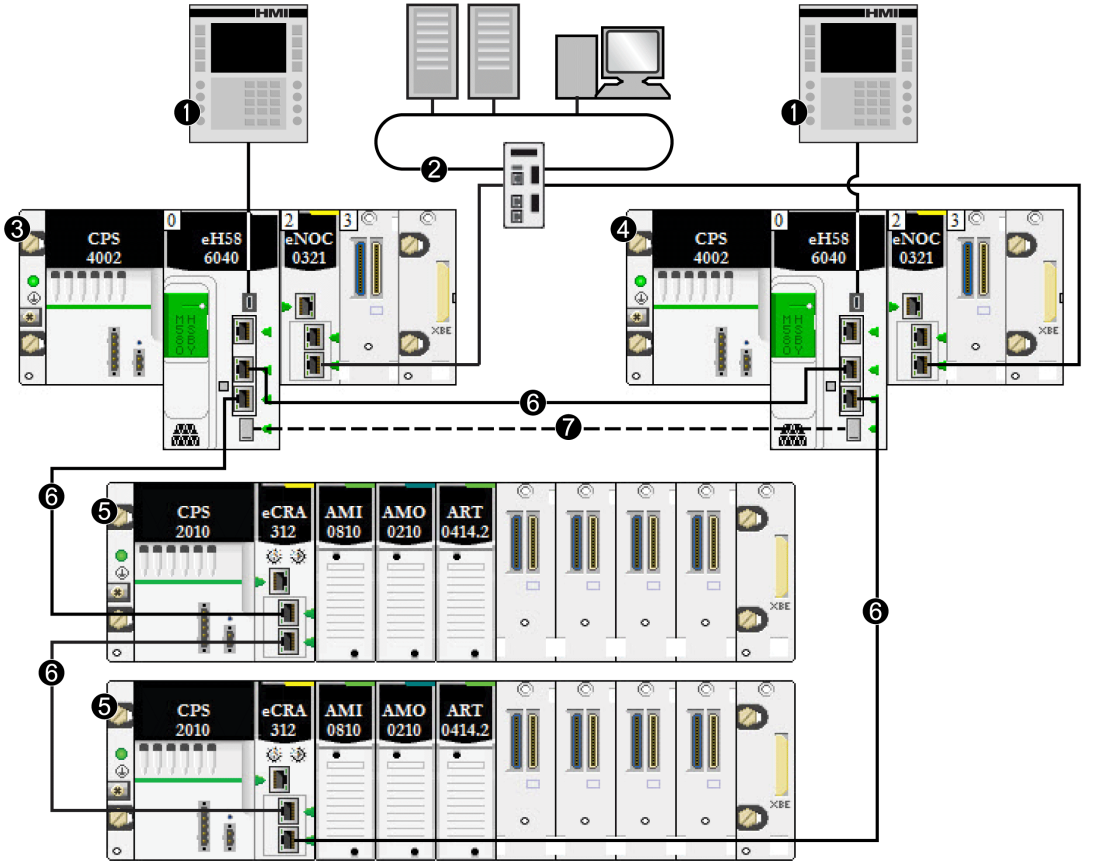
Unlike the prior example, the BMENOC0301 module scanning the distributed equipment is not isolated. Confirm that its Ethernet backplane port is enabled:



1 primary local rack with primary CPU
2 standby local rack with standby CPU
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

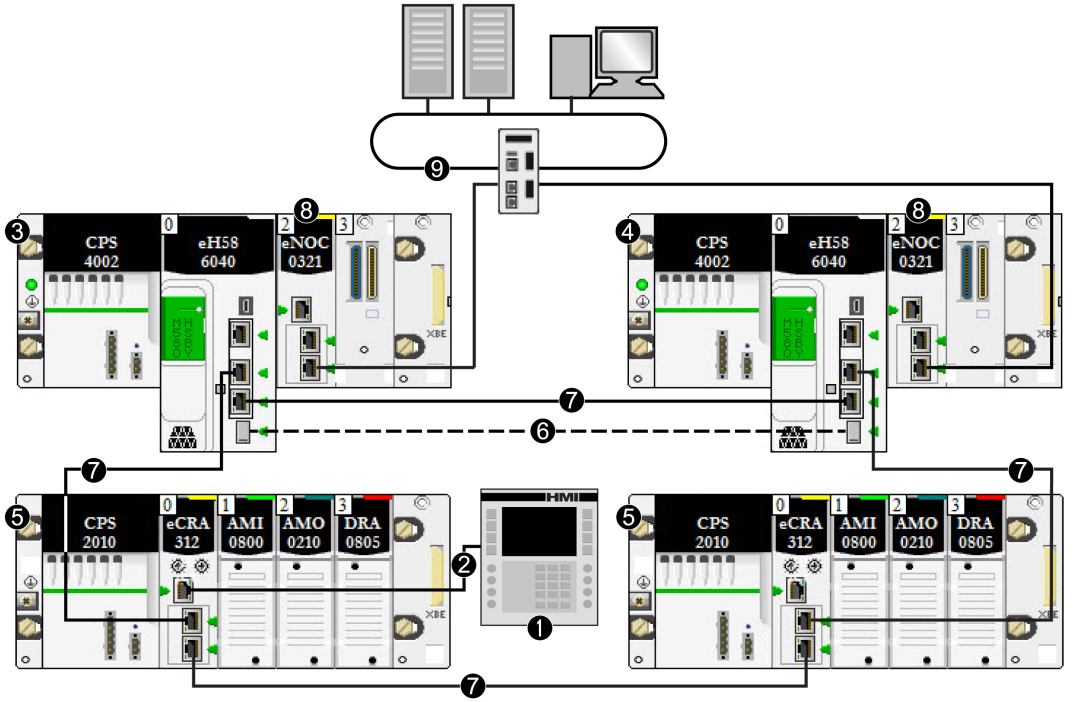
You can connect an HMI directly to a Hot Standby CPU. To access the local PACs (for diagnostics and control), connect an HMI to both the primary and standby CPUs. In the following example, the connection is made via the USB port of each CPU. This design allows access for diagnostics and control to the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



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 CPU
4 standby local rack with standby CPU
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 CPU. 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 CPUs can be accessed for diagnostics and control:

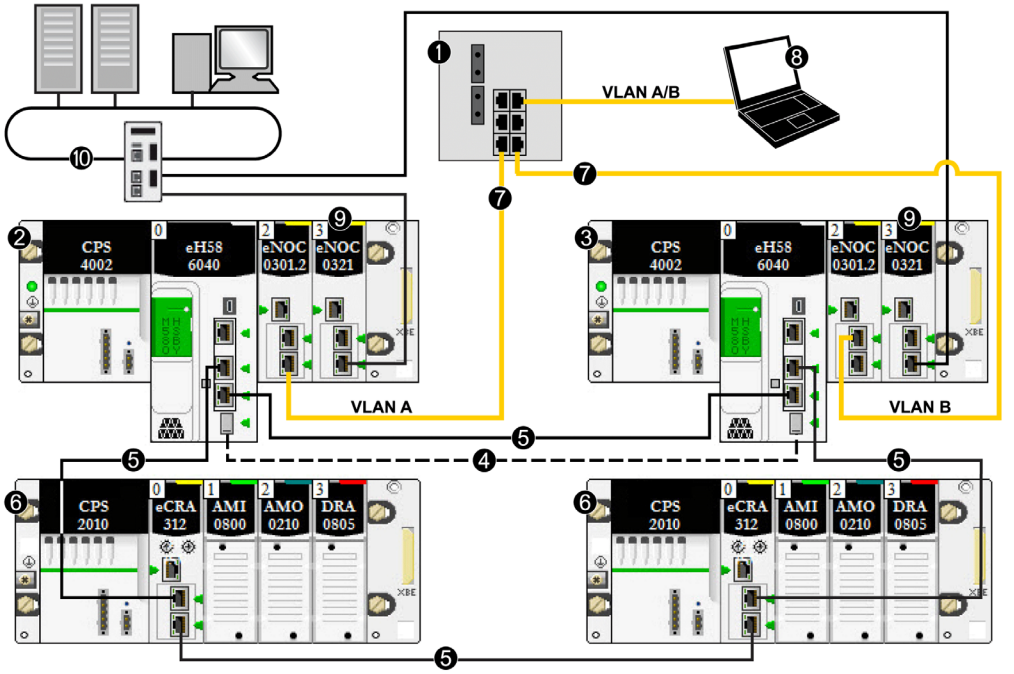


1 HMI
2 Ethernet connection to the service port of the BMECRA312•0 module
3 primary local rack with primary CPU
4 standby local rack with standby CPU
5 (e)X80 RIO drop
6 Hot Standby communication link
7 RIO main ring
8 BMENOC0321 module on the local rack providing transparency between the device network and the control network
9 control network

Simple Solution for Connecting an Engineering Tool to a Hot Standby Topology

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

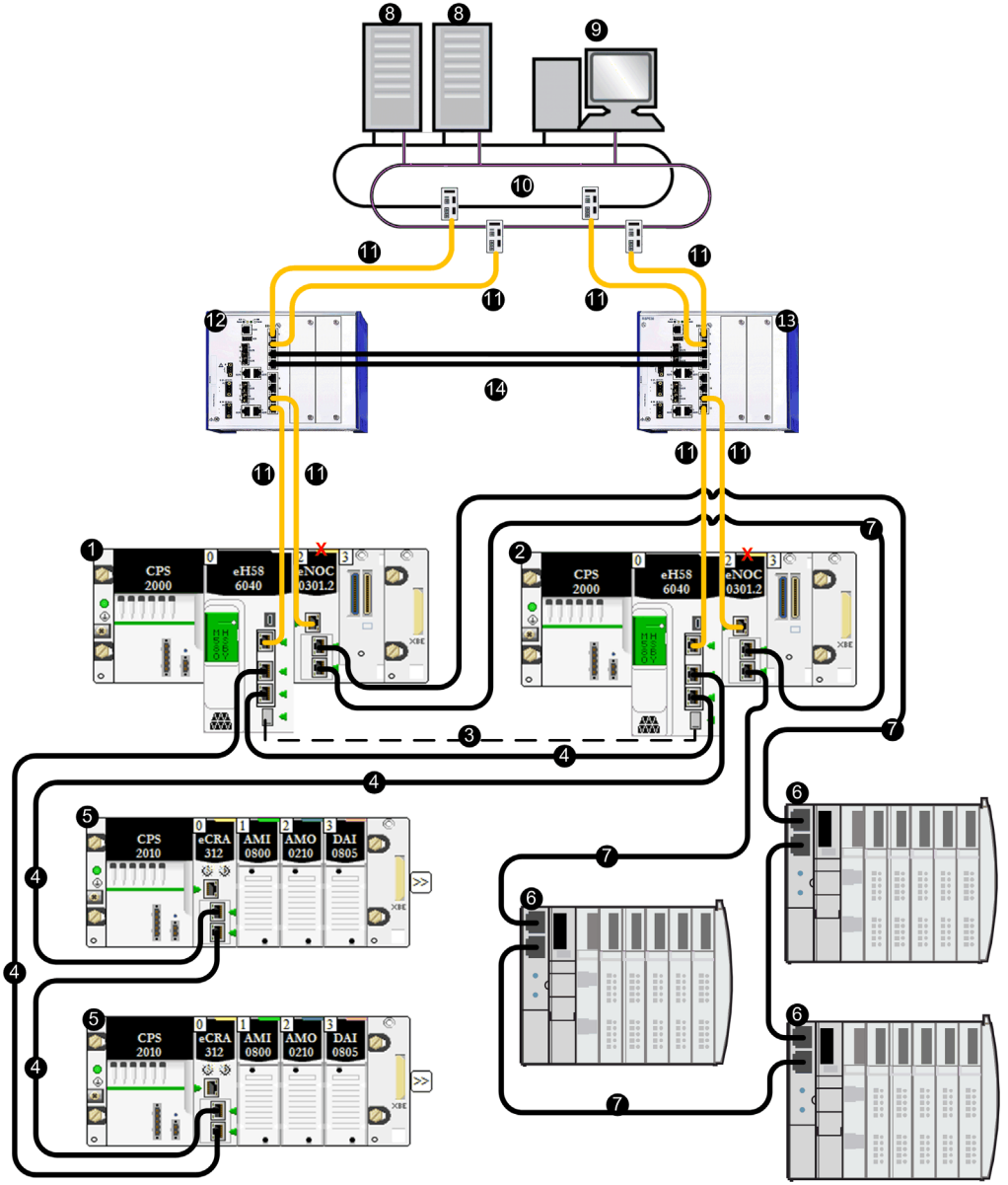
- The PC is equipped with a network interface card (NIC) that supports communication on multiple VLANs, in this case VLAN A and VLAN B.
- The BMENOC0301/11 module in the primary PAC belongs to VLAN A.
- The BMENOC0301/11 module in the standby PAC belongs to VLAN B.



1 layer 2 switch linking control network to the RIO main ring
2 primary local rack with primary CPU
3 standby local rack with standby CPU
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 CPU A
2 local rack with CPU 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 backpl.ane

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 CPU).

Schneider Electric recommends the installation of 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.

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)*.

⚠ CAUTION

RISK OF UNINTENDED OPERATION

Do not connect together the service ports of the Hot Standby CPUs. Connecting together the service ports of the primary and standby CPUs can cause unintended system operation.

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

⚠ 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. Connecting more than one module to both the backplane and an Ethernet network can cause a broadcast storm.
- Use only one module in each local rack to connect an Ethernet network to the Ethernet backplane. That module can be:
  - the CPU, when RIO modules are used
  - one BMENOS0300 network option switch module
  - one BMENOC0301/BMENOC0311 communication module

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

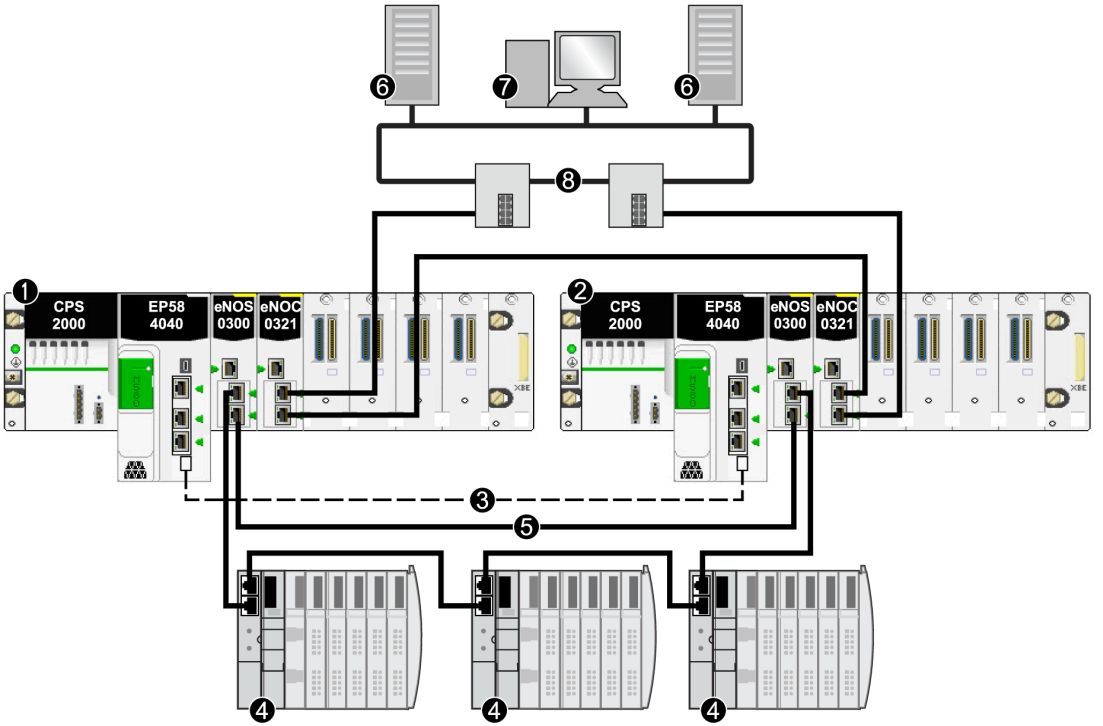
Device Network with DIO Ring and Control Network Architecture

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 CPU 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 CPU
2 standby local rack with standby CPU
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 a break 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 break occurs.
- No link can be made connecting the dual Ethernet ports on the primary CPU to the dual Ethernet ports on the standby CPU.
- 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 CPU scans the RIO drops and the distributed equipment.

⚠ CAUTION

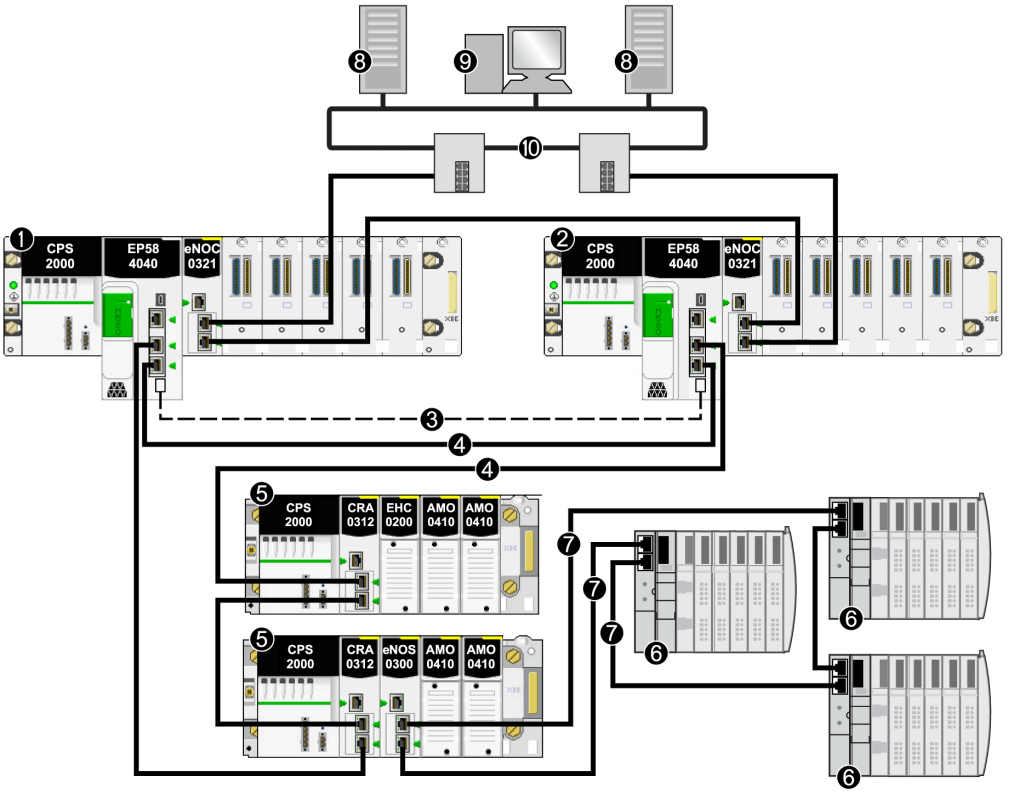
RISK OF BROADCAST STORM

- Do not connect more than one module in a local rack to both the Ethernet backplane and an Ethernet network. Connecting more than one module to both the backplane and an Ethernet network can cause a broadcast storm.
- Use only one module in each local rack to connect an Ethernet network to the Ethernet backplane. That module can be:
  - the CPU, when RIO modules are used
  - one BMENOS0300 module
  - one BMENOC0301/11 module

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

Because the Ethernet backplane port of each BMENOS0300 module is enabled, connecting a dual Ethernet port on the primary CPU to a dual Ethernet port on the standby CPU in this case 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 CPU.



1 primary local rack with primary CPU
2 standby local rack with standby CPU
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 CPU to the dual Ethernet ports on the standby CPU.
- 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.

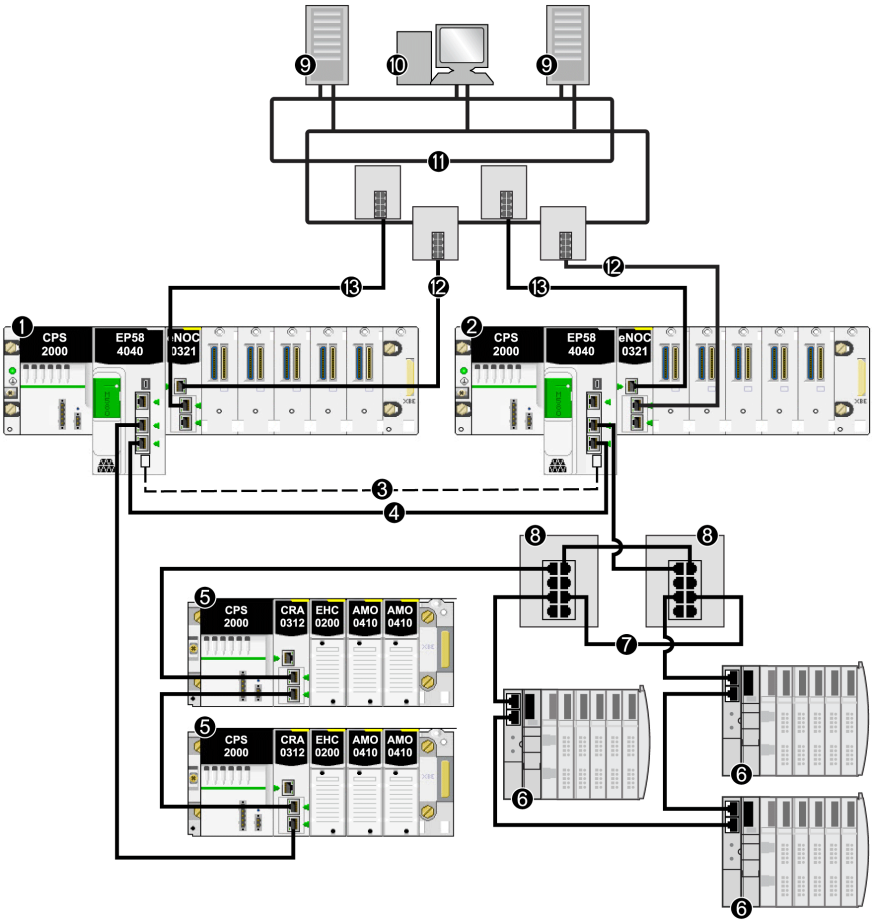
Device Network with RIO Main Ring, DIO Sub-Ring, DRSs, and Control Network Architecture

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, System Planning Guide for, Complex Topologies)*. If one DRS loses communication, 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 CPU:



1 primary local rack with primary CPU

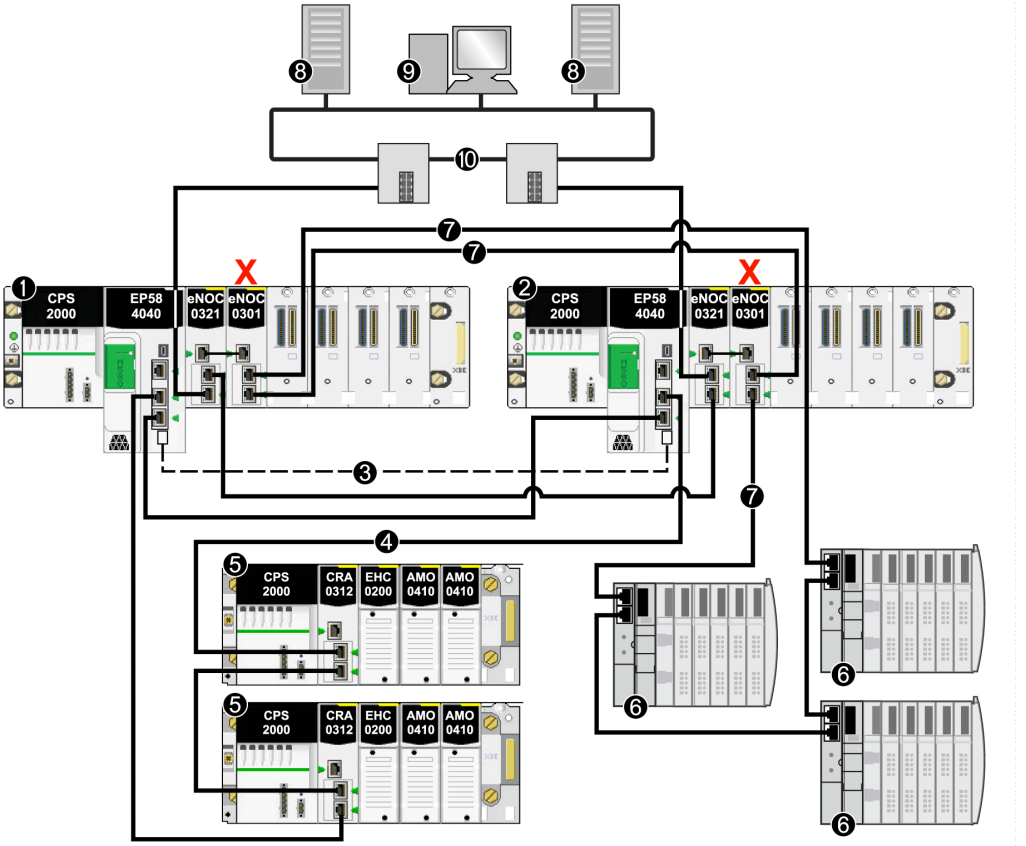
2 standby local rack with standby CPU
3 Hot Standby communication link
4 RIO main ring
5 (e)X80 RIO drop
6 distributed equipment
7 DIO sub-ring
8 dual-ring switch (DRS) with C9/C10 downloaded pre-configuration file
9 SCADA server
10 engineering workstation
11 control room network (Gb RSTP ring)

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

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

You can expand a simple 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 CPU:



1 primary local rack with primary CPU
2 standby local rack with standby CPU
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.

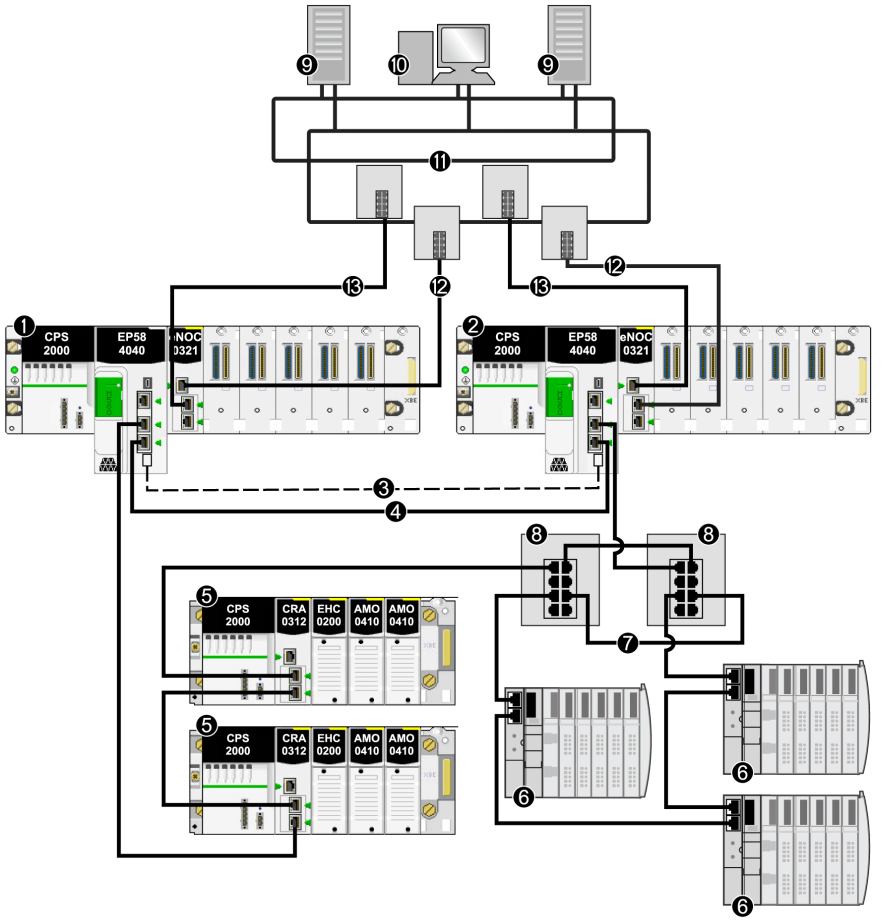
Device Network with RIO Main Ring, DIO Sub-Ring, DRSs, and Dual Control Network Architecture

This topology is similar to the one previously discussed, 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 BMENOC0321module 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 CPU



1 primary local rack with primary CPU
2 standby local rack with standby CPU
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

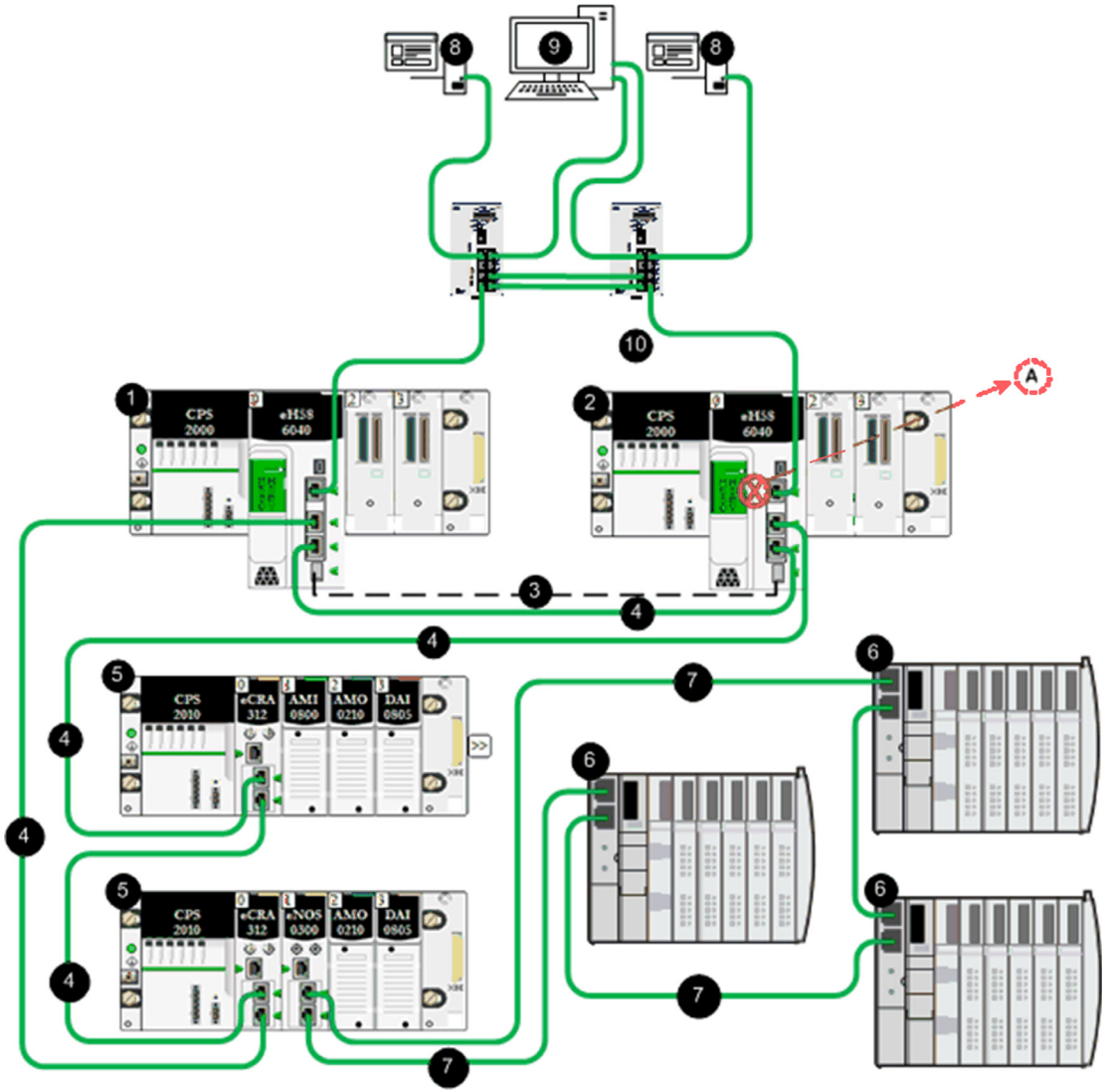
In an M580 Hot Standby configuration, some topologies may unintentionally create a loop that interferes with network communication. These topologies are essentially related to the management of flat networks, i.e., topologies in which the control network, the remote I/O network, and/or the device network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

To help avoid creating an unintentional loop caused by the connection to the service port, use the automatic blocking of service port feature on the CPU 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.



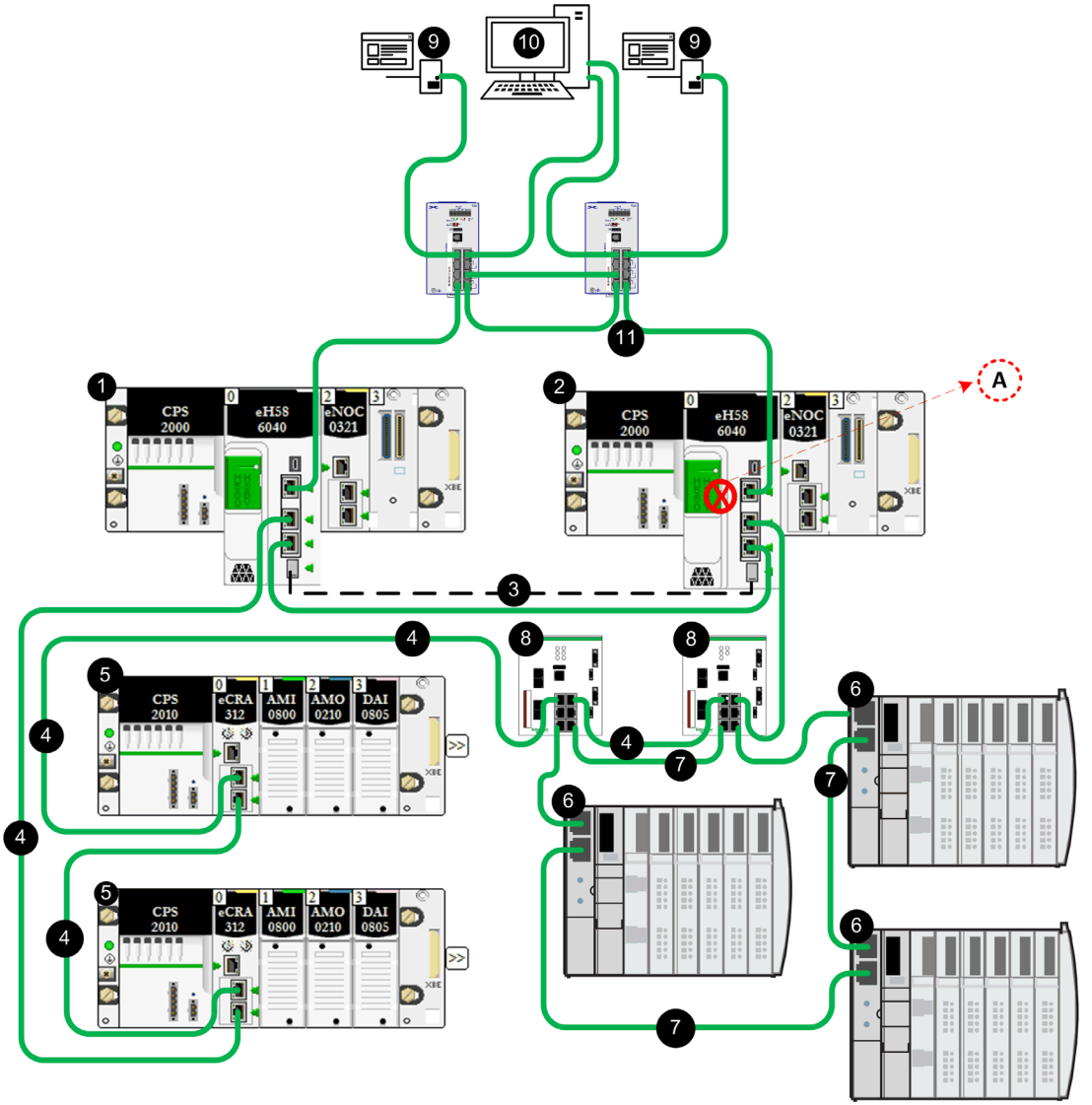
A In this topology, where SCADA is connected to the Hot Standby system via the primary CPU service port and the standby CPU service port, confirm that you select the **Automatic blocking of service port on Standby CPU** check box in the **ServicePort** configuration tab to help avoid network communication loss (formed by cables 4 and 10).

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

1. primary local rack with primary CPU
2. standby local rack with standby CPU
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.



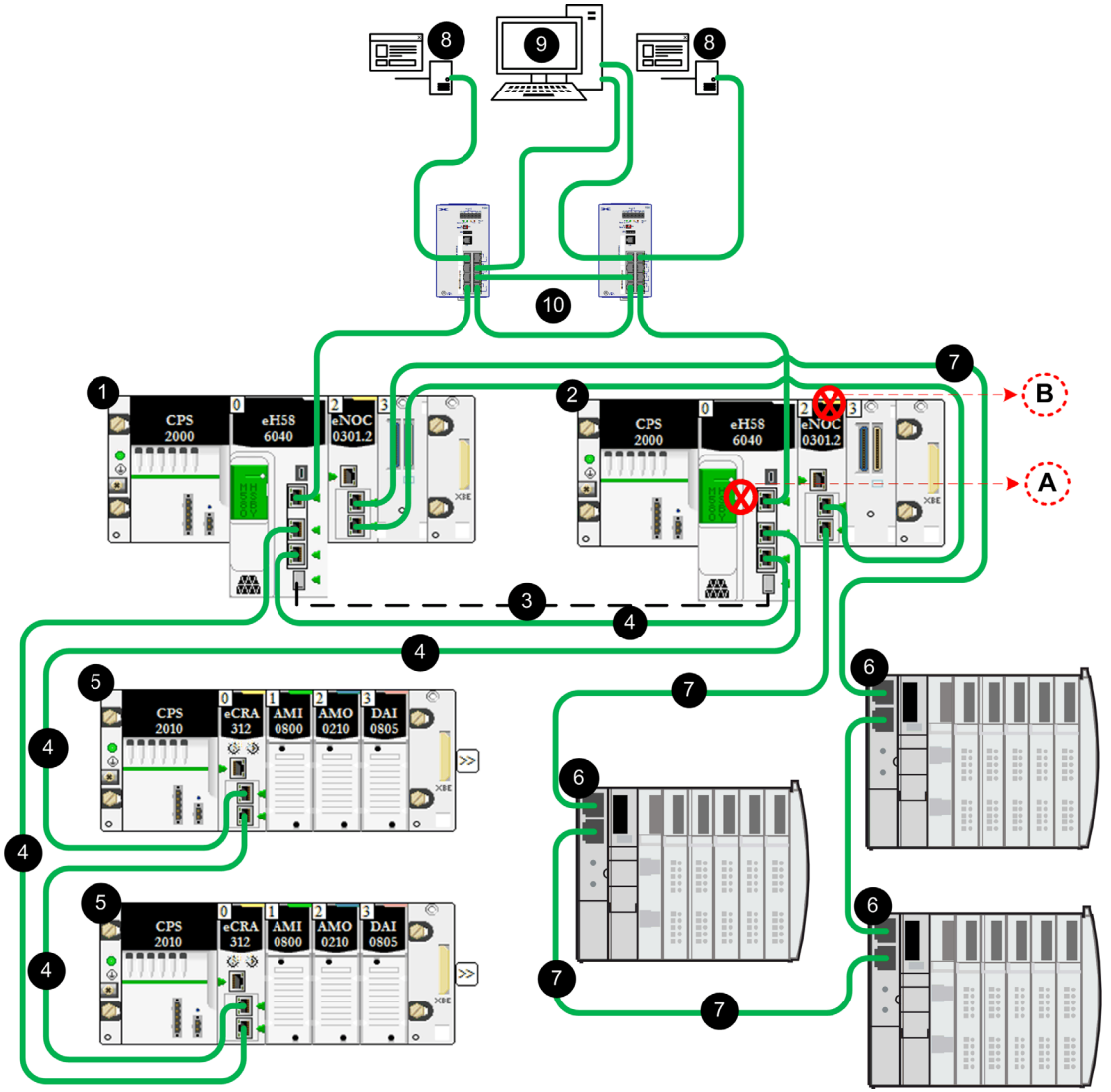
A In this topology, where SCADA is connected to the Hot Standby system via the primary and standby CPU service ports, confirm that you select the **Automatic blocking of service port on Standby CPU** check box in the **ServicePort** configuration tab to help avoid network communication loss (formed by cables 4 and 11).

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

1 primary local rack with primary CPU
2 standby local rack with standby CPU
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 CPUs.



A In this topology, where SCADA is connected to the Hot Standby system via service ports of the CPUs, confirm that you select the **Automatic blocking of service port on Standby CPU** check box in the **ServicePort** configuration tab to help avoid network communication loss (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 help avoid network communication loss (formed by cables 4 and 7).

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

1 primary local rack with primary CPU and BMENOC0301 module

2 standby local rack with standby CPU 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

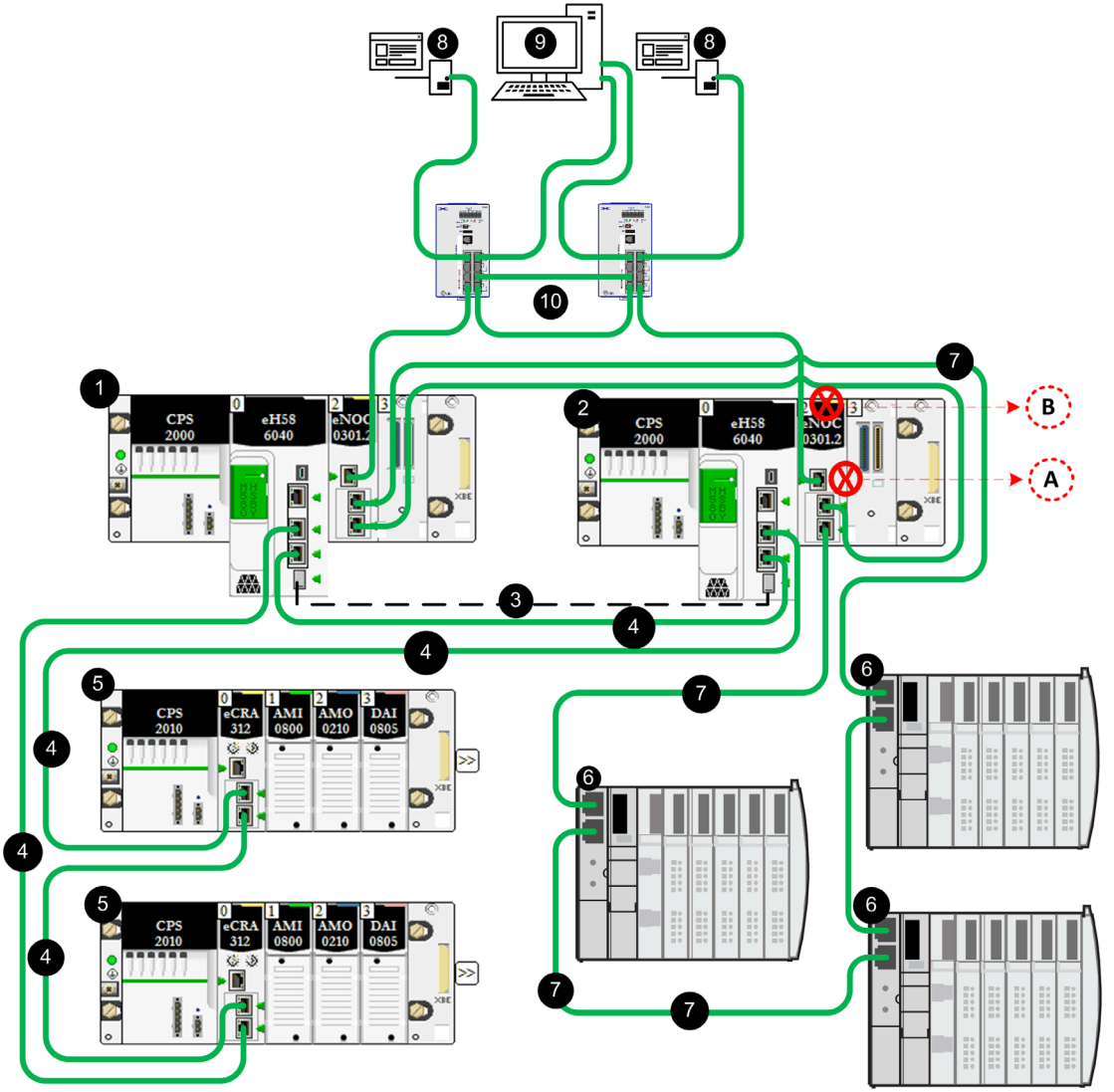
8 SCADA server

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 help avoid network communication loss (formed by cables 7and 10).

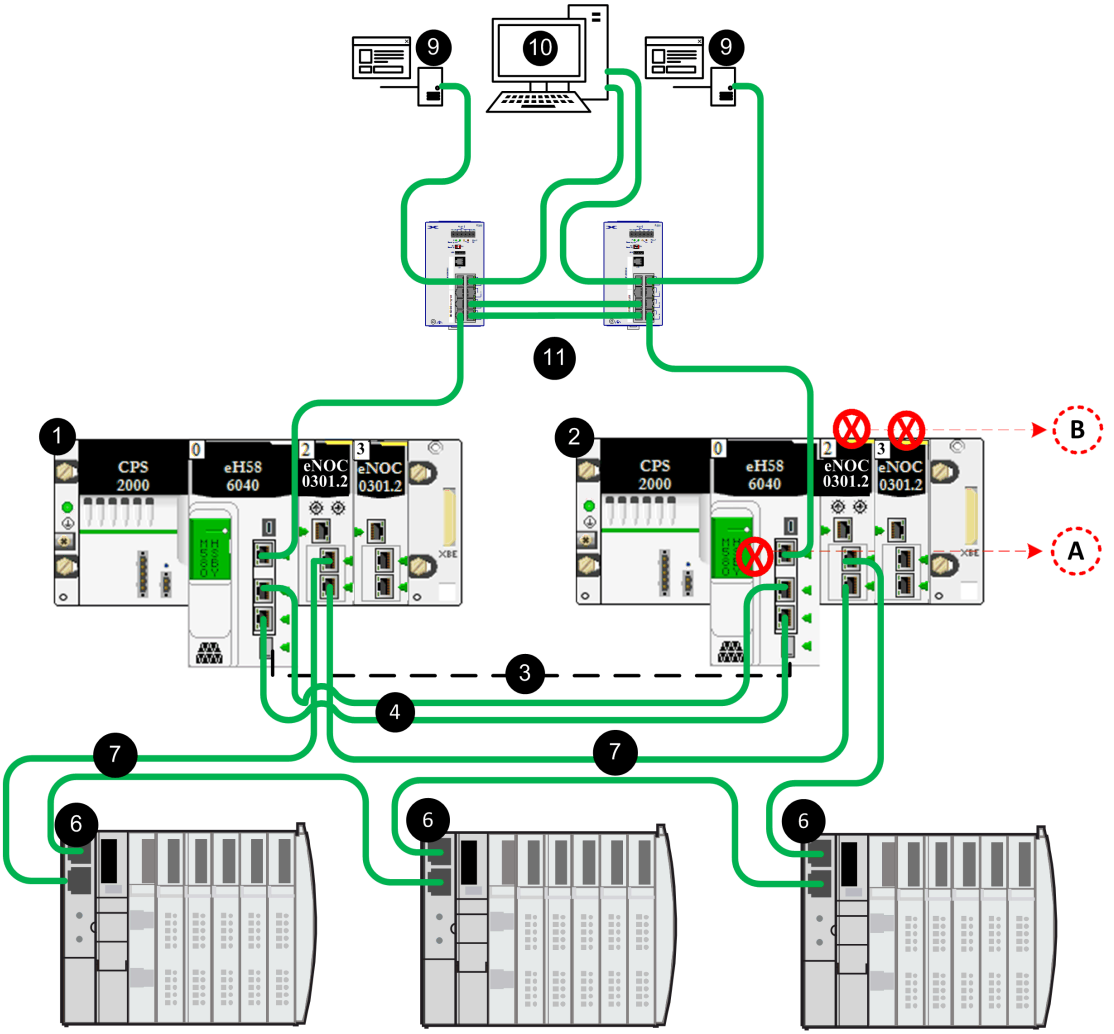
B In this topology, where a DIO ring/cloud network communicates with the control network via the BMENOC0301 modules, the Ethernet backplane port of the standby NOC is automatically disabled to help avoid network communication loss (formed by cables 4 and 7).

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

1 primary local rack with primary CPU and BMENOC0301 module
2 standby local rack with standby CPU 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 CPUs.



A In this topology, where SCADA is connected to the Hot Standby system via service ports of the primary and standby CPUs, confirm that you select the **Automatic blocking of service port on Standby CPU** check box in the **ServicePort** configuration tab help avoid network communication loss (formed by cables 4 and 11).

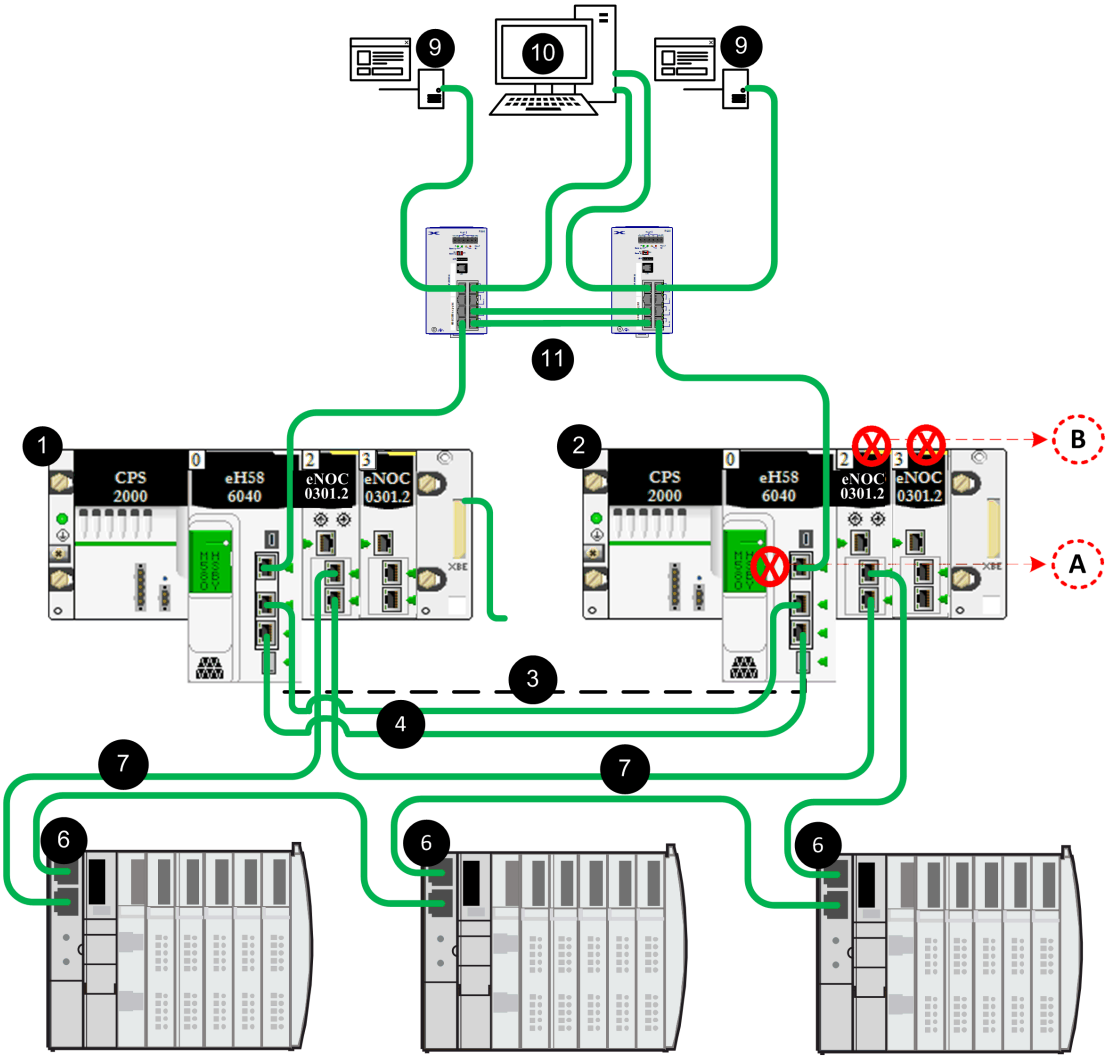
B In this topology, where a DIO ring network communicates with the control network via the CPU, the Ethernet backplane port of the standby NOCs are automatically disabled to help avoid network communication loss (formed by cables 4 and 7).

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

1 primary local rack with primary CPU and BMENOC0301 module
2 standby local rack with standby CPU 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 CPUs.



A In this topology, where SCADA is connected to the Hot Standby system via service ports of the primary and standby CPUs, confirm that you select the **Automatic blocking of service port on Standby CPU** check box in the **ServicePort** configuration tab to help avoid network communication loss (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 help avoid network communication loss (formed by cables 4 and 7).

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

1 primary local rack with primary CPU and BMENOC0301 module

2 standby local rack with standby CPU 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

Chapter 4
Working with M580 Hot Standby Applications

Overview

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

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Modicon M580 Hot Standby Programming Rules	94
M580 Hot Standby System Configuration	97
Configuring an M580 Hot Standby CPU	98
Change Configuration On The Fly (CCOTF)	102
Modifying an SFC Section Online	104
Configuring IP Addresses for an M580 Hot Standby System	105
Configuring Data Variables for an M580 BMEH58•040(S) Hot Standby Application	107
Configuring Hold Up Time for Drops and Devices	109
Transferring M580 Hot Standby Projects	110
Offline Application Modification with Allowed Application Mismatch	112
Restoring and Backing Up Projects	115

Modicon M580 Hot Standby Programming Rules

At a Glance

For Modicon M580 Hot Standby applications, some of the programming functionality you may have used does not apply to redundant operations. This section summarizes some of the code features and programming rules of a Modicon M580 Hot Standby application.

Error Correcting Code (ECC) Feature

M580 Hot Standby CPUs with firmware version 2.50 and higher include an error correcting code (ECC) feature. ECC enhances reliability by reducing the likelihood of memory random access errors, when a Hot Standby CPU 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 Hot Standby M580 CPU 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 CPU.
- For a non-safety CPU 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.

⚠ CAUTION

RISK OF UNINTENDED OPERATION

Using the save operation, that is invoked using system bit %S94, on the primary CPU does not also apply to the standby CPU.

If a swap or switchover occurs after a CCOTF has been performed on the primary CPU, and the application has not been transferred to the standby CPU, then the behavior of the application is unpredictable.

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

Section executed on Standby Restrictions

The following restrictions apply to sections executed on the Standby PLC, first section, or all sections depending on the configuration *(see page 99)*:

- Derived Function Blocks (DFB) may not be executed on Standby PLC sections.
- **R_TRIG, F_TRIG, TRIGGER, TON, TOF, TP** functions blocks may not be executed on Standby PLC sections.
- Asynchronous communication procedures may not be executed on Standby PLC sections.
- Asynchronous communication function blocks may not be executed on Standby PLC sections.

Asynchronous Communication Procedures

During a switchover event, asynchronous communication procedures: **READ_VAR, WRITE_VAR, DATA_EXCH, INPUT_CHAR, INPUT_BYTE, PRINT_CHAR,** do not automatically resume operation on the new Primary PLC without special care.

The following procedure should be used to allow asynchronous communication EFs to automatically resume operation after a switchover:

- Program your application so that all EFs management parameters are not exchanged with Standby PLC. 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 send through the CPU, the typical timeout value is 500 ms.
  - If the communication function is send through a NOC module, the typical timeout value is 2 s.

NOTE: If for some reason you are unable to follow this procedure, 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 CPU.

Asynchronous Communication Function Blocks

During a switchover event, asynchronous communication function blocs which use internal management parameters: **GET_TS_EVT_M, 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,** do not automatically resume operation on the new Primary PLC without special care.

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

- Program your application so that all EFBs instances are not exchanged with Standby PLC. To do this, de-select the **Exchange on STBY** attribute for the EFB instance.

Other Functions

While the use of the functions listed above is restricted, you are advised to use care even when employing permitted functions that are capable of writing to memory areas that are not part of the Hot Standby database transfer, such as Data Storage *(see page 128)* function blocks for instance.

Debugging

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

- First, you debug the application on a single Hot Standby PLC as if it was a standalone application. This allows you to use all of the powerful debugging features available in Control Expert, such as watchpoints, and so on.
- Next, you debug your application when it has been uploaded to two Hot Standby PLCs 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 *(see page 163)* for further details on debugging your Hot Standby application program.

PME UCM 0202 Universal Communication Module

Do not use a **PME UCM 0202** Universal Communication Module in a Drop of a Modicon M580 Hot Standby configuration.

M580 Hot Standby System Configuration

Control Expert Configuration Tool

The exclusive configuration tool for an M580(S) Hot Standby system is:

- Version 11.0 and later of Unity Pro L (for the BMEH582040 module).
  NOTE: Unity Pro is the former name of Control Expert for version 13.1 or earlier.
- Version 11.0 and later of Unity Pro XL (for the BMEH584040 and BMEH586040 modules).
- Version 14.0 and later of Control Expert XL Safety (for the BMEH582040S, BMEH584040S, and BMEH586040S.

Programming Application Languages and Libraries

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

Application language / library	Non-Safety CPUs		Safety CPUs			
	CPU: BMEH58...		CPU: 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 CPU

Introduction

This topic shows you how to configure the Hot Standby functionality of an M580 BMEH58•040 CPU. For information on how to configure the non-Hot Standby functions for the CPU, refer to the *M580 Hardware Reference Manual (see Modicon M580, Hardware, Reference Manual)*.

NOTE: The same procedure, as described below, can also be applied to the configuration of an M580 BMEH58•040S safety CPU.

Accessing the M580 CPU Hot Standby Configuration Tab

Use the **Hot Standby** tab of an M580 BMEH58•040 CPU to configure its Hot Standby function. To access this tab:

Step	Action
1	Add a BMEH58•040 CPU to your project.
2	In the **Project Browser**, navigate to and select **Configuration → PLC Bus → <rack> → <CPU>**.
3	Click the right mouse button, then 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** **Controller B Online**	Specify if a PAC A and PAC B operates online at the next start-up: ● True (default): The PAC attempts to operate online at next start-up. Depending on the other conditions, the PAC may act as the primary or standby. ● False: The PAC transitions to either the Wait or Stop state at next start-up.
Standby On Logic Mismatch	**Number of modifications**	The maximum number of online build changes from 0...50 that can be performed on the primary PAC. 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` *(see page 122)* flag has no effect. ● This setting cannot be edited via CCOTF.
Behavior of the CPU in Wait and Standby mode	**CPU executes**	Specify the sections of the MAST task the standby PAC executes in Wait state: ● All sections (default) ● First section ● No section at all When Control Expert is connected to the standby controller, all **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 CPU 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 PAC, sections of the SAFE task are not executed when the PAC is in WAIT or STANDBY state. You can also individually specify the sections of the MAST task the standby CPU 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 CPU 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 PAC)

Configuring Controller Online State

Controller A is the CPU with the A/B/Clear rotary selector switch *(see page 23)* (located on the back of the CPU) set to “A”. Controller B is the CPU with the A/B/Clear rotary selector switch set to “B”.

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

- To specify the PAC that will be primary on a cold start. For example, set **Controller A Online** to **True** and **Controller B Online** to **False**. PAC A powers up as primary, and PAC B powers up in wait state. After power up, you can manually set **Controller B Online** to **True**.
- To avoid an unintended switchover. For example, if PAC A is primary and PAC B is standby, set **Controller B Online** to **False**. PAC 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 immediately on the next project build (or re-build).

No Local I/O Configuration

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

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

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

- Use a BMENOC0301 or BMENOC0311 communication module and the IPsec protocol to provide a secure connection to the CPU.
- Then use the `CMD_RUN_REMOTE` or `CMD_STOP_REMOTE` commands of the `T_M_ECPU_HSBY` DDT to change a remote CPU operating state.

Enabling FDR Server Synchronization in a Hot Standby System

In an M580 Hot Standby system, a BMEH58•040 CPU, a BMENOC0311 CPU, or a BMENOC0301/11 Ethernet communication module can perform the role of an FDR server. To permit the synchronization of the FDR server in the primary PAC with the FDR server in the standby PAC, 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> → <CPU> → 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 CPU. **NOTE:** Select the **TFTP** column for each entry.
5	**Validate** and **Save** your edits.

NOTE: The FDR server cannot synchronize the primary and standby CPUs 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 CPU, so that this function will be executed by both the primary and standby CPUs.

Change Configuration On The Fly (CCOTF)

CCOTF Rules for Hot Standby

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

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

⚠ CAUTION

RISK OF UNINTENDED OPERATION

Confirm that the following preconditions exist before starting a CCOTF operation:
- The application running in the Hot Standby system does not trigger a swap.
- No condition exists that could forseeably cause a switchover.

Always apply a CCOTF transaction on the Primary CPU.

If a swap or switchover occurs after a CCOTF has been performed on the primary CPU, and the application has not been transferred to the standby CPU, then the behavior of the application is unpredictable.

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

NOTE: Always apply a CCOTF transaction on the primary CPU.

CCOTF allows modifications of a Hot Standby primary PAC configuration in RUN mode. The changes that can be made in the primary PAC are as follows:
- Add a discrete or analog module in a free slot.
- Delete 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 in a free slot.
- Delete a discrete or analog module.
- Modify the configuration and adjustment parameters of a module.

Any CCOTF changes made to the primary PAC configuration are not automatically transmitted to the standby PAC. Instead, the standby PAC 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 Hot Standby PAC configuration:

- A single CCOTF change can include multiple edits to multiple configuration objects.
- Edits to configuration objects are atomic: 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 change may be made to a single RIO drop. Before an additional CCOTF change can be made to the same RIO drop, transfer the application program from the primary PAC to the standby PAC.

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 PAC, the `Logic_Mismatch_Allowed` flag in the `T_M_ECPU_HSBY` DDT determines if the standby PAC can continue to operate online. If logic mismatches are not allowed, the standby PAC transitions to wait state.

CCOTF changes can be made to the primary PAC 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 PAC. The **Build → Build Changes** command in Control Expert is disabled.
- You need to transfer the application program in the primary PAC to the standby PAC *(see page 110)*.

Modifying an SFC Section Online

Precautions for Modifying an SFC Section Online

⚠ WARNING

RISK OF UNINTENDED OPERATION

Transfer the application from primary PAC to the standby PAC 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.

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

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 PAC 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 PAC.

4	Transfer the application from the primary PAC to the standby PAC. 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 CPU, refer to the *M580 Hardware Reference Manual (see Modicon M580, Hardware, Reference Manual)*.

Accessing the M580 CPU Hot Standby Animation task Tab

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

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

Assigning IP Addresses to Modicon M580 BMEH58•040 or BMEH58•040S CPUs

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 CPU for communication with distributed equipment. **NOTE:** Because this setting is always assigned to the primary CPU, it can be associated with either the A or B CPU. When a switchover occurs (for example, when CPU B becomes primary) the main IP address assignment is transferred from CPU A to CPU B.
Main IP address + 1	The Control Expert auto-generated IPv4 IP address used by the standby CPU 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 sole purpose is to provide seamless communication transitions on Hot Standby CPU 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 name	Description
IP address A	The configurable IPv4 IP address for the CPU with its A/B/Clear rotary selector switch *(see page 23)* set to “A”. CPU A uses this IP address for communication on the Ethernet RIO network.
IP address B	The configurable IPv4 IP address for the CPU with its A/B/Clear rotary selector switch *(see page 23)* set to “B”. CPU 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.
Gateway address	The configurable IP address of the default gateway to which messages for other networks are transmitted.

NOTICE
UNINTENDED EQUIPMENT OPERATION 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 may communicate with the Hot Standby system. A duplicate IP address condition, causing unintended equipment operation, can occur. **Failure to follow these instructions can result in 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*

  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 Hot Standby CPUs support the following data attributes:

Attribute	CPU	
	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 CPU, 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 CPU 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 *Unity Pro Operating Modes (see EcoStruxure™ Control Expert, Operating Modes)* manual.

Retain

All BME•586040 CPUs present the **Retain** variable attribute. This attribute determines whether the variable value will persist after a warm start of the CPU. 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 standalone Modicon M580 CPUs, this attribute is read-only. It is selected by default and cannot be de-selected.

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

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

The amount of maximum configurable retained data *(see page 34)* varies, depending on the CPU.

For the BME•586040 CPUs, you cannot edit the **Retain** attribute for a variable that existed at CPU 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 Hot Standby CPU exchanges data with the standby CPU. 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 CPU 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 Hot Standby transfer data *(see page 34)* varies, depending on the CPU.

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.

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 current 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 CPU 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, Schneider Electric recommends setting 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 current states after a communication disruption and before taking their fallback values. Because distributed devices are not connected to the primary CPU during a switch-over, 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 CPU and the standby CPU begin by operating the same application. CCOTF changes that are made to the application running in the primary CPU are not also made to the standby CPU. This causes a logic mismatch to exist between the two CPUs.

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

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

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

Transferring the Application from the Primary CPU to the Standby CPU

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

- **Automatic transfer**: If the non-primary CPU is in a non-configured state, the primary CPU automatically transfers the application program and data to the non-primary CPU when it powers up. There are several ways a CPU 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 selector switch *(see page 23)* was set to “Clear”, powered-up, then re-set to “A” or “B” (depending on the A/B designation of the primary CPU).
    NOTE: To place the standby CPU into run mode on restart, set the `CMD_RUN_AFTER_TRANSFER` *(see page 122)* DDDT command to true before power-up.
- **Transfer from PC to the standby CPU**: If your PC with Control Expert has open the same application as the one running in the primary CPU, you can transfer the application from your PC to the standby CPU. To do this, connect your PC to either the Ethernet service port or USB port of the standby CPU, then use the **PLC → Transfer Project to PLC** command to make the transfer.
  NOTE: If the standby PAC is connected to a configuration tool, such as Control Expert, only the connected configuration tool can transfer an application to the standby PAC. In this case, the primary PAC cannot transfer an application to the standby.

- **Transfer from primary CPU to standby CPU**: With Control Expert connected to the primary CPU, and with both the primary and standby CPUs 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` *(see page 122)* flag is set, the standby CPU automatically starts to run after completion of the transfer, reducing down time for the standby CPU.

  In each case, if both the primary and standby CPUs are equipped with SD memory cards, the application is transferred to both the standby CPU and its SD memory card.
- **SD memory card**: If the primary CPU includes an SD memory card with the current application, take the SD card from the primary CPU, place it into the standby CPU, then reboot the standby.

In each case:

- The application is transferred only if the application in the standby CPU is different from the application being transferred to it.
- If the application running in the primary CPU 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 CPU to the primary CPU.
- 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 CPU, then use the `CMD_SWAP` DDT command to make the standby CPU the primary CPU. Thereafter, you can transfer the application from new primary CPU (formerly the standby) to the standby CPU (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` *(see page 146)*, the primary PAC automatically begins to run immediately upon completion of the transfer.

Offline Application Modification with Allowed Application Mismatch

Procedure

⚠ WARNING
UNEXPECTED EQUIPMENT BEHAVIOR Before transferring a modified application to the Standby CPU: ● Examine carefully all the impacts of the modifications on the application. ● Check that the modified application does not have adverse effects on the process. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

To make offline modifications to an application program in either CPU, follow these steps:

Step	Action
1	Verify the following: ● The `HSBY_BUILD_OFFLINE` *(see EcoStruxure™ Control Expert, System, Block Library)* function block is implemented in the application program in both Primary and Standby CPU. ● Application program is equal in Primary and Standby CPU. ● The **Standby On Logic Mismatch** parameter is set in the **Hot Standby** configuration tab *(see page 98)*.
2	Connect Control Expert to the Primary CPU.
3	Set to 1 the `ALLOW_MISMATCH` bit of the `HSBY_BUILD_OFFLINE` function block. This setting authorizes the CPU to remain synchronized with its pair if a program is modified offline. **NOTE:** Verify that the section where the function block resides is executed by the Primary and the Standby CPU (verify the CPU section execution settings in the **Hot Standby** tab).
4	Confirm that the logic mismatch *(see page 31)* is enabled.
5	Disconnect Control Expert from the CPU.
6	Modify the application program offline. **NOTE:** Only modifications within the scope of the application code and/or some items under the DTM browser modifications are working as valid offline build modification for the Standby CPU. Any other modifications (configuration changes for example) are not taken into account by the `HSBY_BUILD_OFFLINE` function block.
7	Perform a **Build Changes** and save. **NOTE:** Do not perform a **Rebuild All Project** because the Standby CPU may not switch to the `RUN STANDBY` state after the program download and **RUN**. The swap from Standby to Primary cannot be performed.
8	Connect Control Expert to the Standby CPU.
9	Open the modified application program.

Step	Action
10	Download the program in the Standby CPU.
11	Select **RUN**. **NOTE:** Check that the CPU is now in `WAIT` state. **NOTE:** If the CPU does not transition to `WAIT` state, proceed as indicated in the following **Workaround** topic *(see page 114)*.
12	For safety Hot Standby CPUs, check if the safe part of the new application has been modified (`SAFETY_LOGIC_MISMATCH` bit = 1). If so, set the operating mode of the standby PAC to maintenance mode.
13	On the Standby CPU set to 1 the `ALLOW_MISMATCH` bit of the `HSBY_BUILD_OFFLINE` function block. This setting authorizes the CPU to remain synchronized with its pair if a program is modified offline. **Result:** The Standby CPU switches from `WAIT` to `RUN STANDBY` state. **NOTE:** Verify that the section where the function block resides is executed by the Standby CPU (verify the CPU section execution settings in the **Hot Standby** tab).
14	Verify that: ● The Primary CPU is in `RUN PRIMARY`. ● The Standby CPU is in `RUN STANDBY`.
15	Perform a switchover using the `CMD_SWAP` command *(see page 146)*, or by clicking **Animation → Task → Swap controllers → Primary <-> Standby** in the CPU configuration window in Control Expert. **NOTE:** Verify that the Standby CPU switched to Primary CPU.
16	Perform an application transfer to the Standby CPU *(see page 110)*.
17	Perform an application **RUN** to the Standby CPU *(see page 110)*.
18	On the Standby and Primary CPU reset to 0 the `ALLOW_MISMATCH` bit of the `HSBY_BUILD_OFFLINE` function block.

NOTE: Application mismatch topic is discussed in the configuration compatibility *(see page 31)* section.

Workaround When the Standby CPU Does Not Transition to WAIT State

If the Standby CPU does not transition to WAIT state after the **RUN** command in step 11 (a **Rebuild All Project** has been performed for example), the initial program and configuration need to be transferred in the Standby CPU.

Step	Action
1	Connect Control Expert to the Primary CPU
2	Upload the application program from the Primary CPU for future offline modifications. **NOTE:** The modifications done previously to the application program in Control Expert are lost.
3	Perform an application transfer to the Standby CPU *(see page 110)*.
4	Perform an application **RUN** to the Standby CPU *(see page 110)*.
5	Disconnect Control Expert from the CPU.
6	Modify the application program and repeat the **Procedure** actions from step 6 *(see page 112)*.

Use Case

In an existing Hot Standby system, the process to modify an application offline and transfer it to the Primary and Standby CPU follows these macro steps (refer to the preceding detailed procedure for more information):

- Using the CCOTF online modification *(see page 102)*, insert the HSBY_BUILD_OFFLINE function block in the application program of the Primary and Standby CPU. The function block needs one input bit for control and provides a status output.
- Allow the application mismatch in the Primary CPU by setting to 1 the ALLOW_MISMATCH input bit of the HSBY_BUILD_OFFLINE function block in the Primary CPU.
- Modify the application program offline.
- **Build Changes** (do not perform a **Rebuild All Project**).
- Transfer the modified application program in the Standby CPU.
- Allow the application mismatch in the Standby CPU by setting to 1 the ALLOW_MISMATCH input bit of the HSBY_BUILD_OFFLINE function block in the Standby CPU.
- Perform a switchover.
- Transfer the application in the new Standby CPU.
- Reset to 0 the ALLOW_MISMATCH input bit of the HSBY_BUILD_OFFLINE function block in the Primary and Standby CPU.

Restoring and Backing Up Projects

Restoring and Backing Up Projects

The CPU application RAM *(see Modicon M580, Hardware, Reference Manual)* and the CPU flash memory automatically and manually perform the following:

- Restore a project in the CPU from the flash memory (and the memory card if inserted):
  - Automatically after a power cycle
  - Automatically on a warm restart
  - Automatically on a cold start
  - Manually with a Control Expert command: **PLC → Project Backup → Backup Restore**

  NOTE: If a memory card is inserted with a different application than the application in the CPU, the application is transferred from the memory card to the CPU application RAM when the restore function is carried out.
- Save the CPU project in the flash memory (and the memory card if inserted):
  - Automatically after an online modification is performed in the application RAM
  - Automatically after a download
  - Automatically on detection of `%S66` system bit rising edge
  - Manually with a Control Expert command: **PLC → Project Backup → Backup Save**

  NOTE: Backup begins after the completion of the current MAST cycle and before the start of the next MAST cycle.

  Because MAST is configured as periodic for all Hot Standby CPUs, set the MAST period to a value larger than the actual MAST execution time. This lets the processor complete an entire backup without interruption.

  If the MAST period is set to a value less than the actual MAST execution time, backup processing is fragmented and requires a longer time to finish.
- Compare the CPU project and the flash memory project:
  - Manually with a Control Expert command: **PLC → Project Backup → Backup Compare**

  NOTE: When a valid memory card is inserted *(see page 42)* with a valid application, the application backup and restore operations are performed as follows:

- The application backup is performed on the memory card first and then on the flash memory.
- The application restore is performed from the memory card to the CPU application RAM first and then copied from the application RAM to the flash memory.

Chapter 5
Managing M580 Hot Standby Data Exchanges

Overview

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

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Exchanging M580 Hot Standby Data	118
Hot Standby DDT Data Structure	121
Data Storage Elementary Functions	128

Exchanging M580 Hot Standby Data

Periodic Data Exchanges

The Hot Standby CPUs perform two periodic data exchanges:

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

Data Transmitted Each MAST Cycle

Before each MAST task, the primary CPU transmits data to the standby CPU in two ways. The primary CPU 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 communication is lost on the Hot Standby link, the standby CPU does not receive updated I/O data and application variables. If communication is lost for three (3) seconds or more, the standby CPU enters wait state.

Your application needs to regularly check the data synchronization of the MAST, FAST, and SAFE (for safety PACs) 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 CPU with the standby CPU 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 CPUs 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 CPUs (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 `DWORD`s (64 `WORD`s of data).

The maximum amount of data that can be exchanged depends on your choice of CPU *(see page 34)*. If the amount of data in your Hot Standby system exceeds the maximum amount the CPU can transmit, you can:

- Use a CPU 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 CPU 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 CPU and RIO drops (both Quantum and M580).
- FAST tasks are optional for all Hot Standby CPUs, 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.
- Safe data are automatically associated only with the SAFE task.

Preconditions for Data Exchange: Primary and Standby CPUs

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

A single break *(see page 142)* in the Ethernet RIO main ring will not cause an interruption of Ethernet RIO communication between the primary and standby CPUs. The CPUs continue to function as primary and standby respectively. The primary CPU continues to exchange data with the standby over both the Hot Standby and the Ethernet RIO links.

Two breaks *(see page 143)* in the Ethernet RIO main ring (depending on their location) can cause a loss of Ethernet RIO communication between the primary and standby CPUs. However, even if the two CPUs are isolated from each other on the Ethernet RIO ring, they can still communicate over the Hot Standby link. If both CPUs continue to communicate with RIO drops *(see page 144)*, the CPUs continue to function as primary and standby respectively. The primary CPU 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 PAC, those changes are not applied to the configuration of the standby PAC. The exchange of Hot Standby application variables from the primary to the standby is affected, as follows:

- Data objects added to the primary PAC configuration do not exist in the standby PAC. 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 upwards), of data sent by the primary PAC but rejected by the standby PAC.
- Data objects deleted from the primary PAC configuration continue to exist in the standby PAC. No updates can be exchanged for these data objects. In this case, the standby PAC 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 upwards), of data that is retained by the standby PAC but not updated.
- Unchanged data objects remain common to both the primary PAC and the standby PAC, and continue to be included in the data exchange.

The data structure of the primary PAC and standby PAC will be equalized 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 CPU. The DDT instance should appear as: ECPU_HSBY_1.

NOTICE
RISK OF UNINTENDED OPERATION 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 PAC. Data is both auto-generated by the Hot Standby system, and provided by the application. This data is exchanged with the remote PAC.
- REMOTE_HSBY_STS: Provides information about the remote PAC, and contains the image of the last received exchange from the counterpart PAC. The validity of this information is represented by the REMOTE_STS_VALID flag in the common part of this DDT.
  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 PACs.
- 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 checking.
  - 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 PAC versus Remote PAC

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

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

Data Boundary Alignment

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

T_M_ECPU_HSBY DDT

⚠ CAUTION

RISK OF UNINTENDED OPERATION

Before you execute a swap command (either by application logic or in the Control Expert GUI) confirm that the standby PAC is ready to assume the primary role by verifying that the value of its REMOTE_HSBY_STS.EIO_ERROR bit is 0.

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

The `T_M_ECPU_HSBY` DDT consists of these objects:

Element	Type	Description	Written by
REMOTE_STS_VALID	BOOL	• True: Both HSBY_LINK_ERROR and HSBY_SUPPLEMENTARY_LINK_ERROR are 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 PACs is different. (Default = false)	System
LOGIC_MISMATCH_ ALLOWED	BOOL	• True: The standby remains standby in case of logic mismatch. • False (default): The standby goes into wait state in case of logic mismatch.	Application
LOGIC_MISMATCH	BOOL	Different revisions of the same application exist in the two PACs. (Default = false)	System
SFC_MISMATCH	BOOL	• True: The applications in the primary PAC and the standby PAC 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 PACs are running different revisions of the same application.In this condition: • A data exchange between the two PACs may not be possible. • A swap or switchover may not be bumpless. • Neither PAC can be standby (Default = false)	System
APP_BUILDCHANGE_DIFF	UINT	The number of build change differences between the applications in the primary PAC versus the standby PAC. Evaluated by the primary.	System

Element	Type	Description	Written by
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 CPUs: ● True: the standby remains standby in case of FW mismatch. ● False (default): the standby goes into wait state in case of FW mismatch. (Default = false)	Application
FW_MISMATCH	BOOL	The OS are different in the two PACs. (Default = false)	System
DATA_LAYOUT_MISMATCH	BOOL	The Data layout are different on the two PACs. 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 2 Hot Standby PACs are equal. **NOTE:** The backup application resides in flash memory or on the SD memory card of the PAC. 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	PAC A is configured to enter the primary or standby state. (Default = true) **NOTE:** Executable only on PAC A.	Configuration
PLCB_ONLINE	BOOL	PAC B is configured to enter the primary or standby state. (Default = true) **NOTE:** Executable only on PAC B.	Configuration

Element	Type	Description	Written by
CMD_SWAP	BOOL	• Set to 1 by program logic or animation table to initiate a switchover. The primary goes into wait, then the standby goes primary, finally the wait goes standby. The command is ignored if there is no standby. **NOTE:** Executable on both primary and standby. • Reset to 0 (default) by the system on switchover completion or if there is no standby. **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
CMD_APP_TRANSFER	BOOL	• Set to 1 by program logic or animation table to start an application transfer from the primary to the standby. Executable only on the primary. **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[0...2]	• Set to 1 by program logic or animation table to automatically start in Run after a transfer. **NOTE:** Executable only on the primary. • Reset to 0 (default) by the system after transfer completion and: ❍ remote PAC is in Run ❍ PAC is not primary ❍ by animation table or logic command	Application / System
CMD_RUN_REMOTE	BOOL	• Set to 1 by program logic or animation table to run the remote PAC. 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 PAC enters standby or wait state.	Application / System

Element	Type	Description	Written by
CMD_STOP_REMOTE	BOOL	• Set to 1 by program logic or animation table to stop the remote PAC. **NOTE:** Executable on the primary, the standby, or a stopped PAC. • Reset to 0 (default) by the application to end 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 PACs. **NOTE:** Executable on both primary and standby 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 (false): if the initial values for exchanged variables are identical.	System
MAST_SYNCHRONIZED (1)	BOOL	• True: if the exchanged data from the previous MAST cycle was received by the standby. • False (default): if the exchanged data from at least the previous MAST cycle was not received by the standby. **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. • False (default): if the exchanged data from at least the previous FAST cycle was not received by the standby. **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

Element	Type	Description	Written by
SAFE_SYNCHRONIZED	BOOL	• True: if the exchanged data from the last SAFE cycle was received by the standby. • False (default): if, at least, the exchanged data from the last SAFE cycle was not received by the standby.	System
SAFETY_LOGIC_MISMATCH	BOOL	• True: the SAFE logic part of the application is different in the two PACs. • False (default): the SAFE logic part of the application is identical in the two PACs. **NOTE:** The content for this element is determined by comparing system word %SW169 for each PAC.	–
LOCAL_HSBY_STS	T_M_ECPU_HSBY_STS	Hot Standby status for the local PAC	(see below)
REMOTE_HSBY_STS	T_M_ECPU_HSBY_STS	Hot Standby status for the remote PAC	(see below)

(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 PACs 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 in EcoStruxure™ Control Expert, System Bits and Words, Reference Manual).
- Example of consequence: upon an Application Program Transfer (APT) command, the primary PAC might not be able to transfer the program to the standby PAC.

T_M_ECPU_HSBY_STS Data Type

The `T_M_ECPU_HSBY_STS` data type presents the following elements:

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 PAC is in Run state but waiting to go primary or standby. • False: The PAC is in standby, primary or stop state.	System
RUN_PRIMARY	BOOL	• True: The PAC is in primary state. • False: The PAC is in standby, wait or stop state.	System
RUN_STANDBY	BOOL	• True: The PAC is in standby state. • False: The PAC is in primary, wait or stop state.	System

Element	Type	Description	Written by
STOP	BOOL	● True: The PAC is in stop state. ● False: The PAC is in primary, standby or wait state.	System
PLC_A	BOOL	● True: the PAC A/B/Clear switch *(see page 23)* is in “A” position. ● False: the PAC switch is not in “A” position.	System
PLC_B	BOOL	● True: the PAC A/B/Clear switch *(see page 23)* is in “B” position. ● False: the PAC switch is not in “B” position.	System
EIO_ERROR	BOOL	● True: The PAC does not detect any of the configured Ethernet RIO drops. ● False: The PAC detects at least one configured Ethernet RIO drop. **NOTE:** This bit is always 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: Not existent ● 1: Stop ● 2: Run ● 3: Breakpoint ● 4: Halt	System
FAST_TASK_STATE	BYTE	State of the FAST task: ● 0: Not existent ● 1: Stop ● 2: Run ● 3: Breakpoint ● 4: Halt	System
SAFE_TASK_STATE	BYTE	State of the SAFE task: ● 0: Not existent ● 1: Stop ● 2: Run ● 3: Breakpoint ● 4: Halt	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 Hot Standby CPUs, and for process tasks in the M580 BMEH58•040S safety Hot Standby CPUs.

EF	Hot Standby CPU 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 CPU, using an elementary function, are not replicated in the SD card of the other CPU 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.

Chapter 6
M580 Hot Standby System Operation

Overview

This chapter describes operation of the M580 Hot Standby system.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Starting an M580 Hot Standby System	132
Hot Standby State Assignments and Transitions	135
Hot Standby System State Examples	138
Executing Hot Standby Commands	146
Memory Usage	149

Starting an M580 Hot Standby System

Preconditions

During the start-up sequence, each PAC 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 *(see page 23)* position
- Operating state (Run or Stop) of the CPU

On initial start-up, confirm that the:

- Hot Standby link is connected.
- PAC you start first has been fully programmed.
- A/B/Clear rotary switches on the back of the two Hot Standby CPUs 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 switch *(see page 23)* set to “A”.
2	Connect your PC with both Control Expert and the program you want to download.
3	Download the program to the controller.
4	Start the controller in that backplane. If all necessary preconditions exist, the controller becomes the primary Hot Standby CPU.
5	Turn on power to the second backplane. **NOTE:** In this example, this is the backplane with the controller A/B/Clear switch set to “B”.
6	If necessary, repeat steps 2 and 3 for the second controller, and download the program to it. **NOTE:** If the second controller is not configured, the primary CPU automatically downloads the program to the second controller, which becomes the standby.
7	Start the second controller.

Step	Action		
8	Check the LED display for each CPU. If both CPUs are operating as intended, the LEDs will appear as follows:		
	LED	**First CPU** (A)	**Second CPU** (B)
	RUN	Solid Green	Solid Green
	REMOTE RUN	Solid Green	Solid Green
	ETH MS	Solid Green	Solid Green
	ETH MS	Solid Green	Solid Green
	A	Solid Green	OFF
	B	OFF	Solid Green
	PRIM	Solid Green	OFF
	STBY	OFF	Solid Green
	SRUN (safety PAC)	Solid Green	Solid Green
	SMOD (safety PAC)	Solid Green	Solid Green

NOTE:
For a description of:
- BMEH58•040 CPU LEDs, refer to *LED Diagnostics* *(see page 165)*.
- Startup states of the BMEH58•040 CPU, refer to *Hot Standby State Assignments* *(see page 135)*.

A/B/Clear Rotary Switch Role Assignment

The A/B/Clear rotary switch *(see page 23)* assignment does not by itself determine the Hot Standby primary or standby role of a CPU. 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 rotary switch settings determine the role of a CPU only in the case of a simultaneous power up. In that case:
- The CPU set to “A” becomes primary.
- The CPU set to “B” becomes secondary.

Conflicting A/B/Clear Rotary Switch Role Assignment

If you mistakenly set the A/B/Clear rotary switch *(see page 23)* to the same setting – “A” or “B” – for both Hot Standby CPUs, the first CPU to power up becomes the primary, and the second CPU to power up enters wait state.

If you mistakenly set the A/B rotary switch to “Clear” for both CPUs, both CPUs remain non-configured.

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

If both A/B CPU Switches set to:	LED	First CPU to power-up	Second CPU to power-up
A	A	Blink Green	Blink Green
	B	OFF	OFF
	PRIM	Blink Green	OFF
	STBY	OFF	OFF
B	A	OFF	OFF
	B	Blink Green	Blink Green
	PRIM	Blink Green	OFF
	STBY	OFF	OFF
Clear	A	Blink Green	Blink Green
	B	Blink Green	Blink Green
	PRIM	OFF	OFF
	STBY	OFF	OFF

NOTE: If the A/B rotary switches for both CPUs are set to the same position (“A” or “B”), and if both CPUs start-up simultaneously, both CPUs enter wait state.

Hot Standby State Assignments and Transitions

Hot Standby State Assignments

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

- The health of the Hot Standby link between the PACs.
- The health of the Ethernet link between the PACs over the Ethernet RIO main ring.
- The existence of one or more Ethernet connections between each PAC and configured devices via the Ethernet RIO main ring.
- The online state *(see page 100)* of PAC A and PAC B.
- The A/B/Clear rotary switch *(see page 23)* selection on the rear of the CPU.
- The PAC state (RUN or STOP).

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

Network preconditions				Initial state		Final state	
EIO link[1]	RIO device connections[2]		Hot Standby link	PAC_A	PAC_B	PAC_A	PAC_B
	PAC_A	PAC_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
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

1.The supplementary link between PAC A and PAC B over the RIO or DIO ring.
2. The connection between a PAC and RIO drop over the ERIO network. OK indicates the CPU recognizes at least one drop. Not OK indicates the PAC recognizes no drops for 3 seconds.
3. Priority is given to PAC designated “A” via A/B rotary selection switch on the rear of the CPU.
4. Priority is given to PAC that recognizes at least one RIO drop.

Network preconditions				Initial state		Final state	
EIO link[1]	RIO device connections[2]		Hot Standby link	PAC_A	PAC_B	PAC_A	PAC_B
	PAC_A	PAC_B					
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 PAC A and PAC B over the RIO or DIO ring.
2. The connection between a PAC and RIO drop over the ERIO network. OK indicates the CPU recognizes at least one drop. Not OK indicates the PAC recognizes no drops for 3 seconds.
3. Priority is given to PAC designated “A” via A/B rotary selection switch on the rear of the CPU.
4. Priority is given to PAC that recognizes at least one RIO drop.

Hot Standby PAC State Transitions During Operations

A PAC 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: ● PAC is in RUN state. ● PAC is operating online *(see page 100)*. ● Connected to a primary PAC 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: ● PAC is operating online *(see page 100)*. ● PAC is allowed to enter primary state (PAC transitions from STOP to RUN, or warm start in RUN). ● PAC is controlling the Ethernet RIO link, or connected via the Hot Standby link to a counterpart PAC that is not in RUN state.

1. While the PAC is switching from Primary to Standby state, the PAC will pass to an intermediate Wait state for a duration of at least one cycle.

Transition	This transition occurs when...
Standby to Primary	One of the following exists: ● The counterpart PAC enters wait or standby state. ● Communication with the counterpart PAC is interrupted on both the Ethernet RIO link and the Hot Standby link. ● The counterpart PAC is in primary state and receives a swap command.
Standby to Wait	The following exists: ● Communication is interrupted with the counterpart PAC over the Hot Standby link for more than 3 seconds. ● The ERIO link between the 2 PACs 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 PACs only: Online modification mismatch is allowed, if modifications have been made in the safe part of the application (SAFETY_LOGIC_MISMATCH = 1) and maintenance mode has not been set on either the Primary PAC or Standby PAC (i.e. each PAC is operating in safety mode).
Primary to Wait	One of the following exists: ● The PAC has lost communication with all (e)X80 EIO adapter modules, and the counterpart PAC is in standby state and continues to communicate with at least one (e)X80 EIO adapter module. ● The PAC is designated “B” via the A/B/Clear rotary selector switch *(see page 23)*, and the counterpart PAC (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 PAC is disconnected from all (e)X80 EIO adapter modules. ❍ The standby PAC remains connected to at least one (e)X80 EIO adapter module. ❍ The Hot Standby link between PAC A and PAC B remains healthy. ● The primary is in Halt (because at least one task is in Halt) and the counterpart PAC is in Standby state with all tasks in RUN. ● The primary PAC receives a swap command, and the counterpart PAC is in standby state. ● 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.
Primary/Standby/Wait to Stop	● The PAC transitions from RUN to STOP state.

1. While the PAC is switching from Primary to Standby state, the PAC 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 CPU A and CPU B.
- Ethernet RIO link between CPU A and CPU B.
- Ethernet RIO connections between each CPU and one or more (e)X80 EIO adapter modules over the RIO main ring.

In each example, CPU A is the module with its A/B/Clear rotary switch *(see page 23)* set to “A”; CPU B is the module with its A/B rotary switch set to “B”.

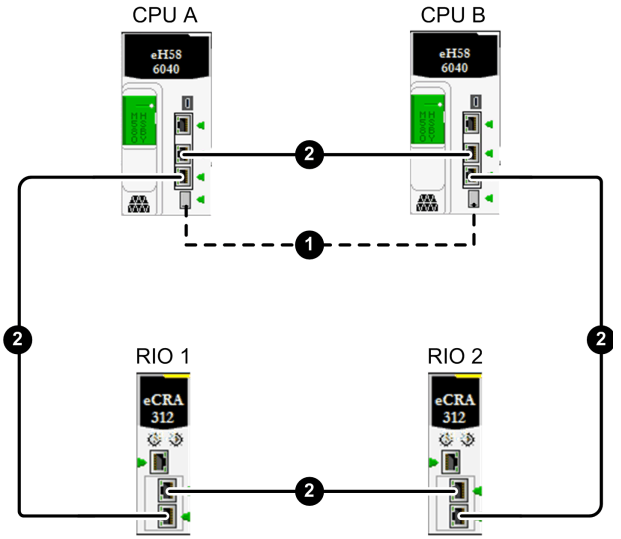
Each example presumes that every other necessary precondition exists for Hot Standby system operation. For example:

- 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 PACs only: If a logic mismatch and safe logic mismatch exist, the `LOGIC_MISMATCH_ALLOWED` flag, the **Online modification in RUN or STOP** parameter, and the Maintenance mode are set.

All Communication Links are OK for both CPUs

In this example, all Hot Standby system connections are operational:

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



1 Hot Standby fiber optic link between CPU A and CPU B
2 Ethernet RIO main ring

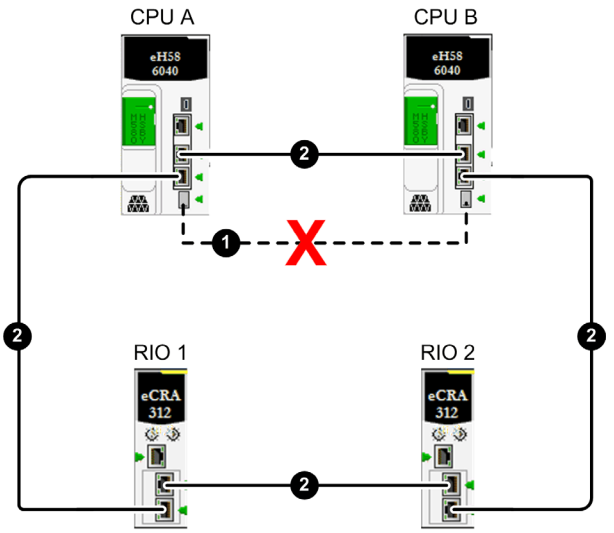
In this example, CPU A and CPU B enter the following Hot Standby states:

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

Hot Standby Link is Not OK for both CPUs

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

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



1 Hot Standby fiber optic link between CPU A and CPU B
2 Ethernet RIO main ring
X Indicates a broken communication link

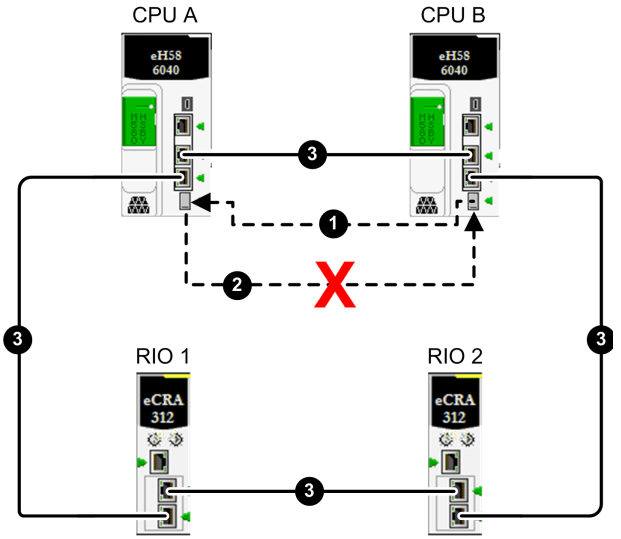
In this example, CPU A and CPU B enter the following Hot Standby states:

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

Hot Standby Link is Not OK for One CPU and is OK for the Other CPU

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

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



1 Operational Hot Standby fiber optic link from CPU B to CPU A
2 Broken Hot Standby fiber optic link from CPU A to CPU B
3 Ethernet RIO main ring
X Indicates a broken communication link

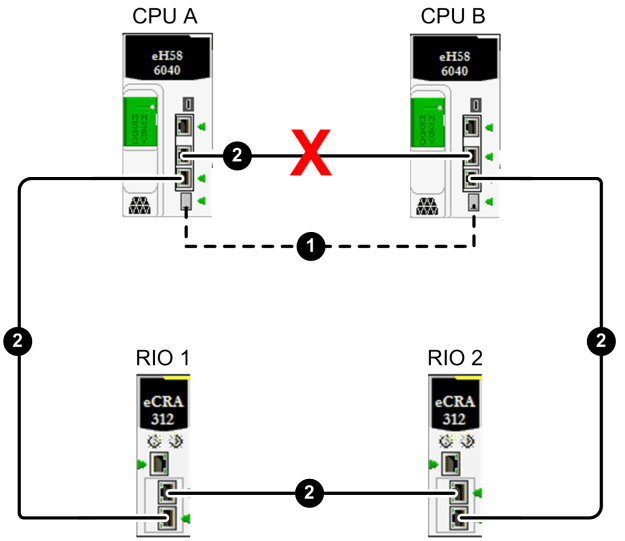
In this example, CPU A and CPU B enter the following Hot Standby states:

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

One Break Exists 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 CPUs, in this example, the break could be located at any point along the Ethernet RIO main ring (2). All other Hot Standby system connections are functioning:

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



1 Hot Standby fiber optic link between CPU A and CPU B
2 Ethernet RIO main ring
X Indicates a broken communication link

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

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

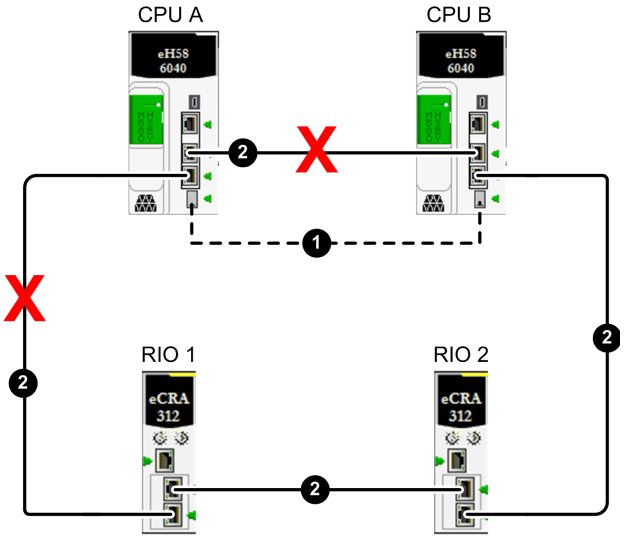
Two Breaks in the Ethernet RIO Main Ring Isolate One CPU

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

- The loss of the Ethernet RIO link between the CPUs.
- The isolation of CPU A from the (e)X80 EIO adapter modules on the Ethernet RIO main ring.

The Hot Standby link remains operational.

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



1 Hot Standby fiber optic link between CPU A and CPU B
2 Ethernet RIO main ring
X Indicates a broken communication link

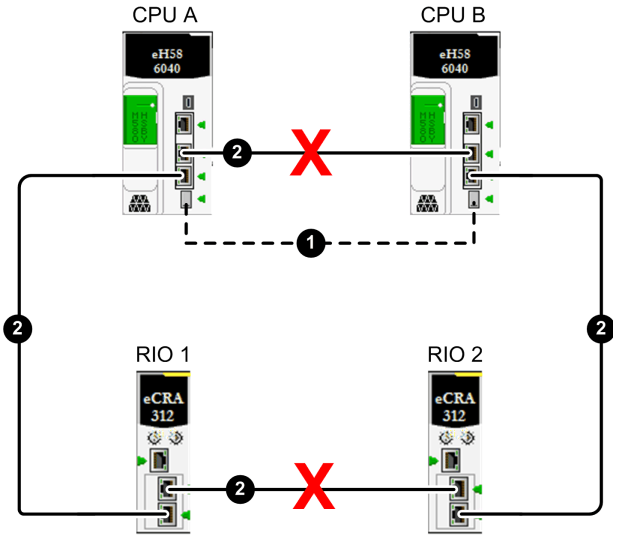
In this example, CPU A and CPU B enter the following Hot Standby states:

If this Hot Standby system state arises during:	CPU A and CPU B perform the following roles:
Sequential start-up of CPU A and CPU B	• The first CPU to start up is primary. • When CPU B starts up: ❍ CPU B becomes primary. ❍ If CPU A was the first to start-up, CPU A enters standby state. **NOTE:** CPU B becomes primary because CPU A has no connections to (e)X80 EIO adapter modules on the Ethernet RIO main ring. • If CPU A starts up starts up after CPU B, CPU A enters standby state.
Simultaneous start-up of CPU A and CPU B	• CPU B is primary. • CPU A enters standby state.
Run-time	• CPU B remains or becomes primary. • CPU A enters standby state.

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

In this example, two breaks exist in the Ethernet RIO main ring, causing the loss of the Ethernet RIO link between CPU A and CPU B. The location of the breaks cause each CPU 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	CPU A	CPU B
Hot Standby link between CPU A and CPU B	OK	OK
Ethernet RIO link between CPU A and CPU B	Not OK	Not OK
Ethernet RIO connections between CPU and one or more (e)X80 EIO adapter modules	OK	OK



1 Hot Standby fiber optic link between CPU A and CPU B
2 Ethernet RIO main ring
X Indicates a broken communication link

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

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

- The Control Expert graphical user interface CPU 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 CPU, refer to the *M580 Hardware Reference Manual (see Modicon M580, Hardware, Reference Manual)*.

Hot Standby Commands

⚠ CAUTION

RISK OF UNINTENDED OPERATION

Before executing a swap, either by application logic or in the Control Expert GUI, verify that the standby PAC is ready to assume the primary role.

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.

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

The M580 BMEH58•040 and BMEH58•040S CPUs 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 PAC to the standby PAC. **NOTE:** The backup application resides in flash memory or on the SD memory card of the PAC. 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 PAC into RUN operating mode upon completion of transfer of application to standby PAC.	Primary only	X	–
CMD_RUN_REMOTE	Places the remote[1] PAC into RUN operating mode. Executable only on the primary CPU.	Primary only	X	X[3]
CMD_STOP_REMOTE	Places the remote[1] PAC 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. Executable on both the primary and the standby CPU. **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.	Both	X	X[3]

X: Command is supported.
–: Command is not supported.
1. *Remote* refers to the PAC to which your PC and Control Expert is not connected.
2. In the CPU configuration window **Hot Standby** tab.
3. In the CPU configuration window **Animation → Task** tab.
4. These commands can be executed only if the remote CPU is also the standby CPU.

Command	Description	Executable on Primary or Standby	Supported by:	
			DDDT	GUI
FW_MISMATCH_ALLOWED	When changes have been made to the firmware in the primary CPU, this command lets the standby CPU 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 CPU (for example, as a a result of CCOTF changes), this command lets the standby CPU continue to operate as standby. If this command is set to 0, the standby goes into wait state.	Primary only	X	–
PLCA_ONLINE	Lets the CPU with its A/B/Clear rotary switch *(see page 23)* set to “A” serve as either primary or standby, depending on other operating conditions. If set to 0, PAC A goes into either wait or stop state.	PAC A only	X	X[2]
PLCB_ONLINE	Lets the CPU with its rotary switch set to “B” serve as either primary or standby, depending on other operating conditions. If set to 0, PAC B goes into either wait or stop state.	PAC B only	X	X[2]

X: Command is supported.
–: Command is not supported.
1. *Remote* refers to the PAC to which your PC and Control Expert is not connected.
2. In the CPU configuration window **Hot Standby** tab.
3. In the CPU configuration window **Animation → Task** tab.
4. These commands can be executed only if the remote CPU is also the standby CPU.

Memory Usage

Introduction

The memory usage function is used to view:

- The physical distribution of the PAC memory.
- The space taken up 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 access the memory usage details of the PAC:

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) taken up by user data (objects relating to configuration): ● **saved Data**: located data associated with the processor (%M, %MW, %S, %SW, etc.) or the input/output modules. This data is retained by the CPU in the event of a CPU warm start. ● **saved Declared Data**: unlocated data (declared in the data editor) that is retained by the CPU in the event of a CPU warm start. ● **unsaved Declared Data** unlocated data (declared in the data editor) that is not retained by the CPU in the event of a CPU warm start.

Parameter	Description
User program	This field indicates the memory space (in words) taken up by the project program: ● **Constants**: static constants associated with the processor (%KW) and the input/output modules; initial data values, ● **Executable code**: executable code of the 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) taken up by other data relating to the configuration and the project structure: ● **Configuration**: other data relating to configuration (Page0 for a Quantum PAC, 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....)
Internal memory	This field shows the organization of the PAC's 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 (Even if the PAC is in Run or in Stop).

NOTE: Certain blocks cannot be moved in online mode. You will attain a lower level of fragmentation by re-organizing the memory in offline mode.

Chapter 7
M580 Hot Standby System Performance

Overview

Creating a deterministic Hot Standby system requires the use of network components and designs that support efficient 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.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
System Performance	152
Calculating the Minimum Task Cycle Time	154
Application Response Time for M580 Hot Standby CPUs	157

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 CPU	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		
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 *(see page 34)* for the selected CPU.

Minimum / Maximum System Channels

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

Calculating the Minimum Task Cycle Time

Introduction

By configuring sufficiently large task cycle times, the CPU 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 CPU will force the task to over-run.

There are two main conditions to be satisfied, to avoid over-run events. By using the formulas (set forth below) to compute the minimum task time periods for your system, you can avoid a task over-run situation.

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 CPUs 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 CPU 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 PAC Processor Load

Each application task uses a percentage of the processor resources, referred to here as the task bandwidth ($Task_{BW}$). 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 PAC manages a set of user and system tasks and schedules them using a fixed priority strategy. It is necessary to reserve a minimum of spare processor bandwidth to permit the PAC 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 CPU resources that are used by a task per time unit. To be evaluated, it is necessary to know or evaluate the following elementary 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 PAC is not connected to a standby PAC. (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. It is recommended to 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 PAC, the time needed to transfer (T_{TFR}) the data with the remote PAC, 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 CPU module 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 PAC 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 *(see page 122)*).
2. Some applications have a very unstable execution time because the code executed is slightly different from one task cycle to the next. There is no general rule to decide which value for minimum task bandwidth should be considered in this case.

Application Response Time for M580 Hot Standby CPUs

Introducing Application Response Time

Each Ethernet RIO input signal packet travels from an RIO drop to the CPU, and the CPU sends an output signal back to the RIO drop. The time it takes for the CPU 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 CPU uses to resolve an RIO logic scan.

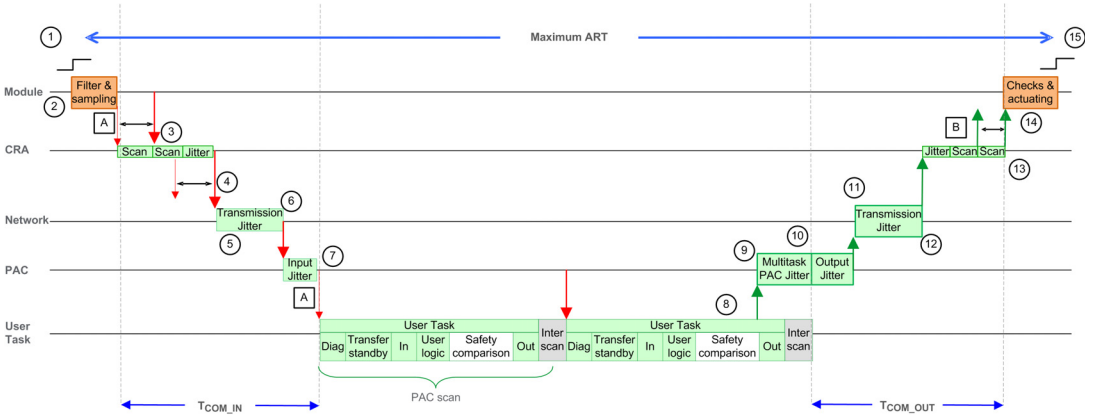
Calculating ART for M580 Hot Standby CPUs

The *Modicon M580 Standalone System Planning Guide for Frequently Used Architectures* describes both a simplified method and a more complex method of calculating ART for a standalone CPU.

To calculate a maximum ART for an M580 Hot Standby CPU, it is necessary to add to the standalone CPU ART calculation estimates of the maximum time for:

- A switchover event, during which the standby PAC takes over the role of primary PAC, after the original primary PAC ceases to be operational or loses communication.
- A swap event, during which the standby PAC takes over the role of primary PAC in response to a user or application command.

ART: General Calculation for an M580 Safety Hot Standby PAC 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 PAC multitasking
1: input turns ON	10: CPU output jitter
2: input filtering and sampling (Safe 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 (Safe modules)
6: network jitter	15: output applied
7: CPU input jitter	–

The more complex method of ART calculation, in the *Modicon M580 Standalone System Planning Guide for Frequently Used Architectures* 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 PAC is not overloaded, and that total bandwidth of all tasks is less than 80% *(see page 154)*.

- Hot Standby effect: An additional time (Transfer Standby) is necessary to exchange the user data from the primary PAC to the standby PAC. This time is included in the execution time displayed in the **PLC → Animation** screen if the primary PAC is connected to the standby.
- Safety effect:

- An additional time (Safety Comparison) is necessary to exchange and compare the safe data between the PAC and the safety coprocessor. This time is always included in the execution time displayed in the **PLC → Animation** screen.
- All the safety 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.

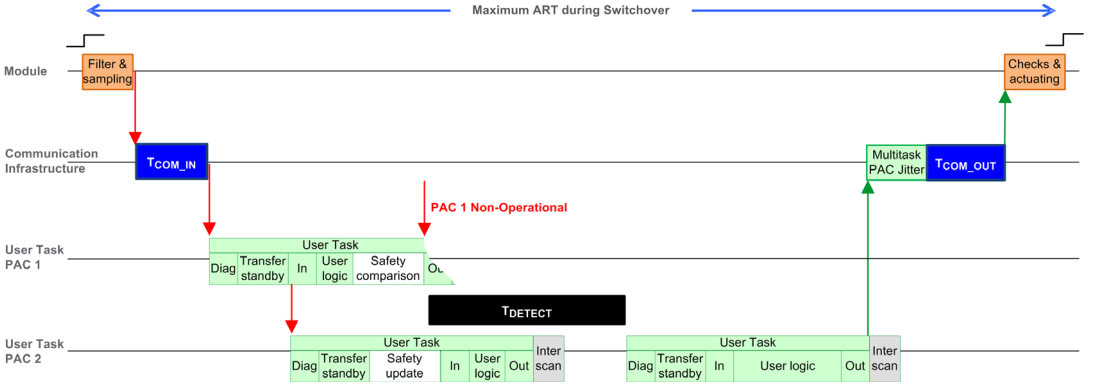
The following elements are necessary for the ART evaluation:

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 too late, 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.		See[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 Standalone System Planning Guide for Frequently Used Architectures*.
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 Hot Standby PAC in a Multi-task Configuration During a Switchover

A switchover occurs in a Hot Standby system when the primary PAC ceases to be operational or loses communication. The standby PAC, after a detection time, will restart the MAST task in its role as the new primary PAC. Thereafter, the SAFE and FAST tasks will be able to start in the new primary PAC. 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 too late, after the begin of the scan.		T_{PER}[2]
T_{DETECT}	Time used by the standby PAC to detect and confirm the primary PAC 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

1. For a description of these elements, refer to the more complex ART calculation method in the *Modicon M580 Standalone System Planning Guide for Frequently Used Architectures*.
2. Refer to the **Processor Load** topic for a discussion of these elements.
3. The configured cycle period for the respective task (SAFE, FAST).

Name	Description		Value
Additional_jitter	Jitter introduced by the multi-task system to restart the task on the new PAC.	Mast task	0
		Safe task	T_{SAFE}[3]
		Fast task	T_{FAST}[3]
T_{COM_OUT}	Sum of all the time used by the communication with output modules.		See[1]
T_{OUTPUT}	Time used by the safety output modules for the diagnostics and for actuating the external signal.		6 ms

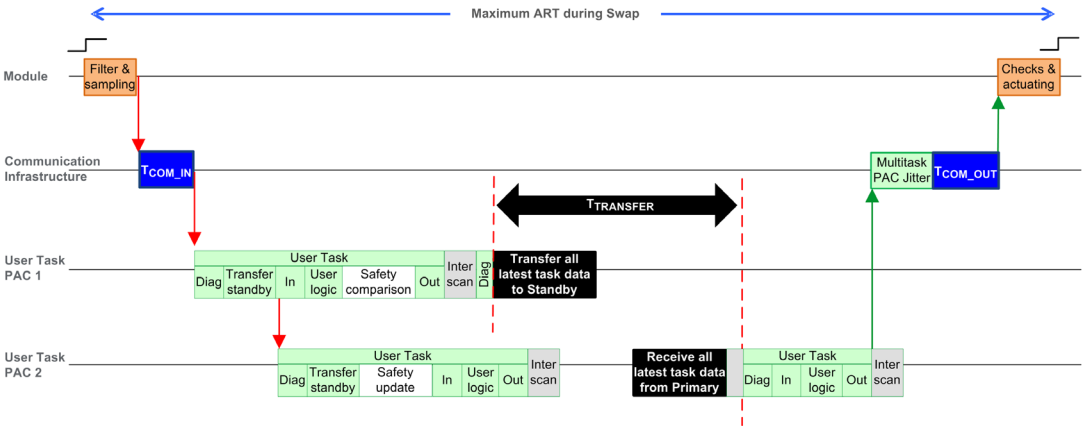
1. For a description of these elements, refer to the more complex ART calculation method in the *Modicon M580 Standalone System Planning Guide for Frequently Used Architectures*.
2. Refer to the **Processor Load** topic for a discussion of these elements.
3. The configured cycle period for the respective task (SAFE, FAST).

ART: M580 Safety Hot Standby PAC in a Multi-task Configuration During a Swap

A swap 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 PAC checks that all the conditions necessary to authorize a swap are met, then confirms that all the tasks have updated the standby PAC with the latest data. The primary PAC then goes into wait mode. The remote PAC switches to the primary mode, starting the MAST task first and then the other tasks SAFE and FAST). In the meantime, the other PAC (i.e., the original primary PAC) goes to standby mode.

The worst-case swap 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 too late, after the begin of the scan.		T_{PER}[2]
$T_{TRANSFER}$	During the diagnostics of the MAST task, the PAC accepts the Swap command and begins to perform the transfer of all the latest data for each task.		Refer to the formula, below.
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 PAC.	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.		See[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 Standalone System Planning Guide for Frequently Used Architectures*.
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:

$\max((K3 \times (MAST_{KB} + 2 \times SAFE_{KB} + FAST_{KB}) + K4 \times (MAST_{DFB} + 2 \times SAFE_{DFB} + FAST_{DFB})) / 1000, T_{SAFE})$

Where:

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

Coefficient	BMEH582040S	BMEH584040S or BMEH586040S
K3	46.4	14.8
K4	34.5	11.0

Chapter 8
M580 Hot Standby Diagnostics

Overview

This chapter describes M580 Hot Standby diagnostic tools provided by the:

- BMEH58•040 CPU Hot Standby LEDs
- Control Expert graphical user interface

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
8.1	M580 Hot Standby CPU LEDs	164
8.2	Control Expert M580 Hot Standby Diagnostics	171
8.3	M580 Hot Standby CPU Web Pages	176
8.4	M580 Hot Standby System Diagnostics	188
8.5	M580 System Words	190

Section 8.1
M580 Hot Standby CPU LEDs

Overview

This section describes LEDs on the M580 BMEH58•040 Hot Standby CPU modules.

What Is in This Section?

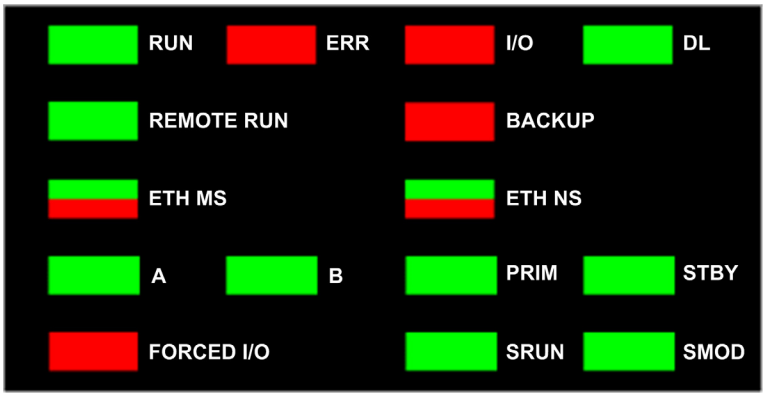
This section contains the following topics:

Topic	Page
LED Diagnostics for M580 Hot Standby CPUs	165
Memory Card Access LED	169

LED Diagnostics for M580 Hot Standby CPUs

LED Panel

The front face of a BMEH58•040 Hot Standby CPU 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 CPUs.

- For a description of the safety PAC LEDs **SRUN** and **SMOD**, refer to the topic *LED Displays for the M580 Safety CPU and Copro (see Modicon M580, Safety System Planning Guide)* in the *Modicon M580, Safety System Planning Guide*.
- For a presentation of LED diagnostics for safety CPUs, refer to the topic *M580 Safety CPU LED Diagnostics (see Modicon M580, Safety Manual)* in the *Modicon M580, Safety Manual*.

Hot Standby Panel LEDs

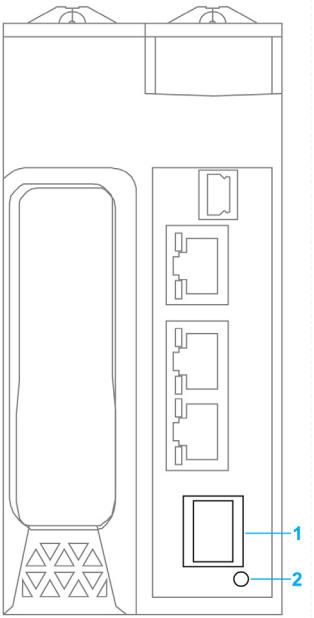
The BMEH58•040 Hot Standby CPU LEDs present the following Hot Standby system diagnostics:

LED	Description
A	• **ON** (green) indicates: ❍ The local CPU A/B/Clear rotary switch *(see page 23)* is set to “A” , and ❍ The remote CPU A/B/Clear rotary switch is set to “B”. • **BLINKING** (green) indicates: ❍ If LED B is **OFF**: - The local CPU A/B/Clear rotary switch is set to “A” , and - The remote CPU A/B/Clear rotary switch is also set to “A”. ❍ If LED B is also **BLINKING** green: - The local CPU A/B/Clear rotary switch is set to “Clear”. • **OFF**: Indicates local CPU A/B/Clear rotary switch is not set to “A” or to “Clear”.

LED	Description
B	• **ON** (green) indicates: ❍ The local CPU A/B/Clear rotary switch is set to “B” , and ❍ The remote CPU A/B/Clear rotary switch is set to “A”. • **BLINKING** (green) indicates: ❍ If LED A is **OFF**: - The local CPU A/B/Clear rotary switch is set to “B” , and - The remote CPU A/B/Clear rotary switch is also set to “B”. ❍ If LED A is also **BLINKING** green: - The local CPU A/B/Clear rotary switch is set to “Clear.” • **OFF**: Indicates local CPU A/B/Clear rotary switch is not set to “B” or “Clear”.
REMOTE RUN	Indicates the RUN status of the remote PAC: • **ON**: (green): The remote PAC is in RUN state. • **BLINKING**: (green): The remote PAC is in STOP state. • **OFF**: The local PAC cannot read the state of the remote PAC. Both the Hot Standby link and the Ethernet RIO link are lost.
PRIM	Indicates the primary status of the PAC: • **ON** (green) The local PAC is primary, but the remote PAC is not in standby state. • **BLINKING**: The local PAC is in wait state; the STBY LED is also **BLINKING**. • **OFF**: The local PAC is not primary. **NOTE:** • If CPU is in RUN mode and both PRIM and STBY LEDs are **OFF**, CPU is in wait state. • If both CPUs are in RUN mode, and one CPU is primary and the other CPU is in wait state: - On Primary: PRIM LED is ON, STBY LED is BLINK. - On Wait: PRIM LED is OFF, STBY LED is BLINK
STBY	Indicates the standby status of the PAC: • **ON** (green): Indicates the PAC is in standby state. • **BLINKING** (green) indicates either: ❍ The local PAC is primary, but the remote PAC is not in standby state. ❍ The local PAC is in wait state; the PRIM LED is also **BLINKING**. • **OFF**: Indicates local PAC is not in standby state. **NOTE:** • If CPU is in RUN mode and both PRIM and STBY LEDs are **BLINKING**, the CPU is in wait state. • If one CPU is primary and the other CPU is in wait state: - On Primary: PRIM LED is **ON**, STBY LED is **BLINKING**. - On Wait: PRIM LED is **OFF**, STBY LED is **BLINKING**.

Hot Standby Link LED

A Hot Standby link LED is located on the front of the BMEH58•040 CPU:



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 PAC.
blinking	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 CPUs* in the Modicon M580 Hardware Reference Manual *(see Modicon M580, Hardware, Reference Manual)* for standalone, non-safety LEDs.
- *M580 Safety CPU LED Diagnostics* in the M580 M580 Safety Manual *(see Modicon M580, Safety Manual)*, for safety LEDs.

Memory Card Access LED

Introduction

The green memory card access LED underneath the SD memory card door indicates the CPU access to the memory card when a card is inserted. This LED can be seen when the door is open.

Dedicated LED States

By itself, the **memory card access** LEDs indicate these states:

LED Status	Description
ON	The memory card is recognized, but the CPU is not accessing it.
flashing	The CPU is accessing the memory card.
blinking	The memory card is not recognized.
OFF	The memory card can be removed from the CPU slot or the CPU does not recognize the memory card.

NOTE: Confirm that the LED is OFF before you remove the card from the slot.

Combined LED Meanings

The access card LED operates together with the **BACKUP** LED *(see Modicon M580, Hardware, Reference Manual)*. Their combined patterns indicate the following diagnostic information:

Memory Card Status	Conditions	CPU State	Memory Card Access LED	BACKUP LED
no memory card in the slot	—	no configuration		
memory card not OK	—	no configuration		
memory card without project	—	no configuration		
memory card with a non-compatible project	—	no configuration		
– no specific circumstances or CPU state				

Memory Card Status	Conditions	CPU State	Memory Card Access LED	BACKUP LED
memory card with a compatible project	An error is detected when the project is restored from the memory card to the CPU RAM.	no configuration	during transfer:  end of transfer: 	during transfer:  end of transfer: 
	No error is detected when the project is restored from the memory card to the CPU RAM.	—	during transfer:  end of transfer: 	during transfer:  end of transfer: 
– no specific circumstances or CPU state				

This legend shows the different LED patterns:

Symbol	Meaning	Symbol	Meaning
	off		steady red
	steady green		blinking green

Section 8.2
Control Expert M580 Hot Standby Diagnostics

Overview

This sections described diagnostic tools for the M580 BMEH58•040(S) Hot Standby CPUs that are available in Control Expert.

What Is in This Section?

This section contains the following topics:

Topic	Page
M580 Hot Standby System Diagnostics in Control Expert	172
Synchronizing Configuration of Distributed Equipment	174

M580 Hot Standby System Diagnostics in Control Expert

Introduction

Control Expert provides M580 Hot Standby System diagnostic information in the following GUI screens:

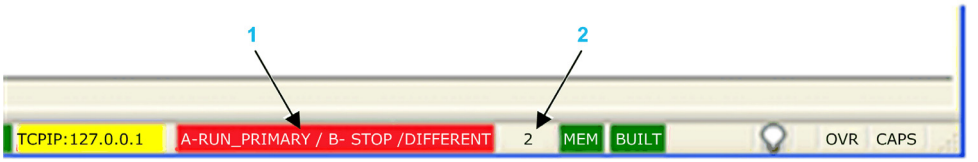
- the Hot Standby status viewer embedded in the Control Expert Task Bar.
- the **Information** tab of the CPU **Animation** window

Hot Standby Status Viewer

When Control Expert is connected to the Hot Standby system, it displays the Hot Standby status of each PAC, including:

- The status of CPUs A and B.
- The comparative state of logic running in the standby PAC.
- If a logic mismatch exists, the number of modifications *(see page 99)* made to the application running in the primary PAC.

The Hot Standby Status Viewer looks like this:



1 Hot Standby status
2 Number of changes

The status values for PACs A and B include:

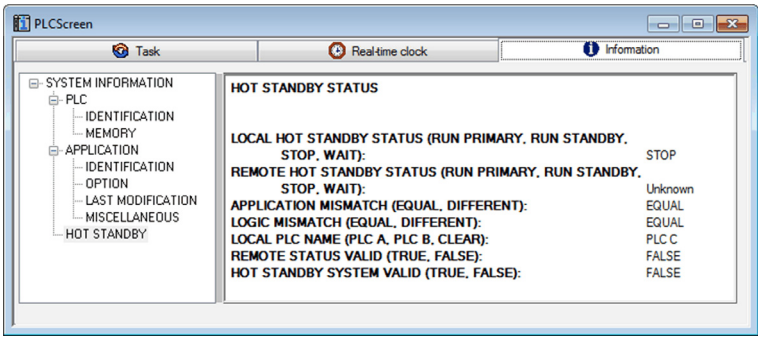
- RUN_PRIMARY
- RUN_STANDBY
- STOP
- WAIT

Also presented is the logic state of the standby PAC, which can be either:

- EQUAL (green background): There is no logic mismatch.
- DIFFERENT (red background): Online changes have been made to the primary PAC application that have not been transferred to the standby PAC.

Hot Standby Information Tab

Use the CPU Configuration window **Animation → Information** tab to view the status of the Hot Standby system:



The **Information** tab contains one word of status data, as follows:

Hot Standby status of the local PAC: ● Primary ● Standby ● Stop ● Wait	Local PAC name (position of A/B/Clear rotary switch *(see page 23)*): ● PLC A ● PLC B ● CLEAR
Hot Standby status of the remote PAC: ● Primary ● Standby ● Stop ● Wait	Remote status valid: ● True ● False
Application mismatch status: ● Equal ● Different	Hot Standby system valid: ● True ● False
Logic mismatch status: ● Equal ● Different	–

Synchronizing Configuration of Distributed Equipment

Introduction

The M580 BMEH58•040(S) Hot Standby CPU DTMs include a **Hot Standby Synchronization** page where you can synchronize the storage of configuration (.prm) files for distributed equipment in the primary and standby CPUs. Distributed equipment configuration files stored in Hot Standby CPUs are used by the fast device replacement (FDR) service.

Use this page to:

- View the synchronization status of distributed equipment configuration files stored by the Hot Standby system CPUs.
- Stop synchronization.
- Force a manual synchronization.

The standby CPU synchronizes with the primary CPU by pulling data every 10 seconds to verify that the data in the standby has been updated in the primary. If the standby unsuccessfully synchronizes with the primary, it continues polling the primary every 10 seconds.

If the data in the standby and primary PACs are different, an application mismatch *(see page 31)* condition exists. In this case, synchronization stops and a synchronization error is detected in the standby CPU.

NOTE:

- When the standby CPU is offline, it does not synchronize.
- If you disable the TFTP service, Hot Standby synchronization cannot be performed, because this function is based on TFTP.

Accessing the Hot Standby Synchronization Page

To access the CPU **Hot Standby Synchronization** page, follow these steps:

Step	Action
1	In Control Expert, open the **DTM Browser** (**Tools → DTM Browser**).
2	Right-click the CPU in the **DTM Browser**.
3	Select **Connect**.
4	Right-click the CPU in the **DTM Browser**.
5	Select **Device menu → Diagnosis**.
6	Click the **Hot Standby Synchronization** tab.

Using the Hot Standby Synchronization Page

The **Hot Standby Synchronization** page presents the following parameters and controls:

Parameter	Description
Refresh Every 500ms	Select this to display synchronization data in this page, and refresh displayed data every 500ms.
Status area:	
Synchronizing	• True: Synchronization is executing. • False: Synchronization is not executing.
Synchronized	• True: Data in both primary and standby are synchronized. • False: Data in both primary and standby are not synchronized.
Error Status	• Green: No synchronization error is detected. • Red: A synchronization error has been detected.
Manual Synchronization → Stop Synchronization area:	
Stop Synchronization Service	Select this then click **Send** to stop the synchronization service. **NOTE:** To re-start the synchronization service, select one of the **Force Manual Synchronization** options (below), then click **Send**.
Manual Synchronization → Force Manual Synchronization area:	
Copy Files from Standby to Primary	Select this then click **Send** to push DIO device configuration (.prm) files from the standby CPU to the primary.
Copy Files from Primary to Standby	Select this then click **Send** to pull DIO device configuration (.prm) files from the primary CPU to the standby.
Clear Files in Primary	Select this then click **Send** to delete the DIO device configuration (.prm) files from the primary. If synchronization is enabled, the standby CPU synchronizes with the primary and any DIO device configuration files in the standby are also deleted.

Section 8.3
M580 Hot Standby CPU Web Pages

Overview

This section describes the diagnostic web pages for the M580 BMEH58•040(S) Hot Standby CPU modules.

What Is in This Section?

This section contains the following topics:

Topic	Page
Introducing the M580 Hot Standby CPU Web Pages	177
Status Summary (Hot Standby CPUs)	179
HSBY Status	182
Rack Viewer	185

Introducing the M580 Hot Standby CPU Web Pages

Introduction

The M580 BMEH58•040(S) Hot Standby CPUs includes an embedded web server that provide monitoring and diagnostic functions. All web pages are read-only.

These web pages are included:

- Module:
  - Status Summary (Hot Standby) *(see page 179)*
  - HSBY Status *(see page 182)*
  - Performance *(see Modicon M580, Hardware, Reference Manual)*
  - Port Statistics *(see Modicon M580, Hardware, Reference Manual)*
- Connected Devices:
  - I/O Scanner *(see Modicon M580, Hardware, Reference Manual)*
  - Messaging *(see Modicon M580, Hardware, Reference Manual)*
- Services:
  - QoS *(see Modicon M580, Hardware, Reference Manual)*
  - NTP *(see Modicon M580, Hardware, Reference Manual)*
  - Redundancy *(see Modicon M580, Hardware, Reference Manual)*
- System:
  - Alarm Viewer *(see Modicon M580, Hardware, Reference Manual)*

In addition, for the BMEH584040, BMEH586040, BMEH584040S, and BMEH586040S CPUs, a Rack Viewer *(see page 185)* page is also included.

This section describes the web pages that are unique to the M580 Hot Standby CPUs: the *Status Summary* and *HSBY Status* web pages. For all other web pages, refer to the M580 CPU Embedded Web Pages *(see Modicon M580, Hardware, Reference Manual)* topics of the *Modicon M580 Hardware Reference Manual*.

Browser Access Requirements

The embedded web pages are accessible using the following operating system and browser combinations:

Operating system	Browser
Android OS v4 mini	Chrome mobile minimum version 35.0.1916.141
iOS6	Safari v6
iOS7	
Windows 7	Internet Explorer v8.0.7601.17514
Windows 8	
Windows 8.1	
Windows 8.1 RT	Internet Explorer minimum v8
Windows Phone OS	Internet Explorer Mobile v10

The embedded web site is accessible via WiFi, using a smartphone or tablet equipped with a:

- Schneider Electric WiFi dongle, called the *wifer*, part number TCSEGWB13FA0.
- PMXNOW0300 wireless module.

Status Summary (Hot Standby CPUs)

Introduction

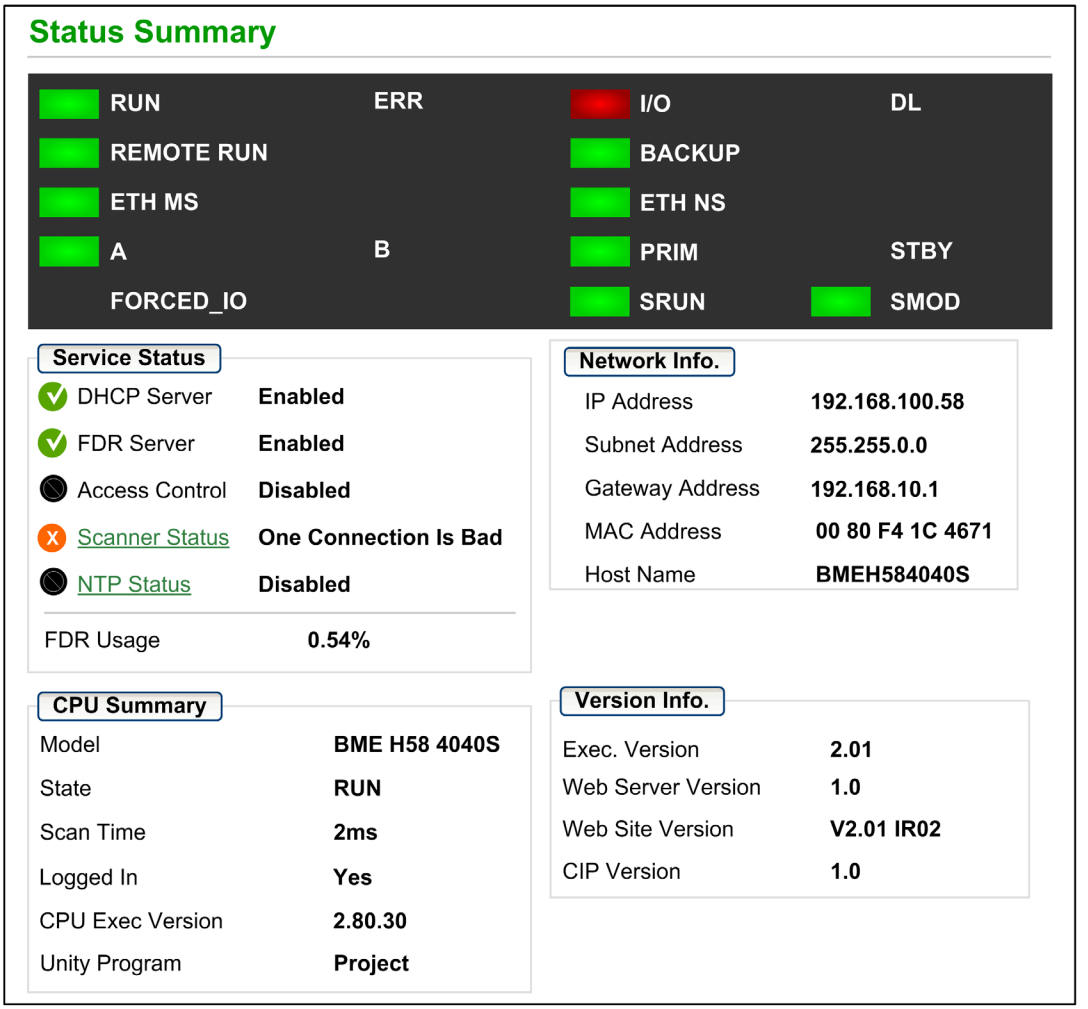
The **Status Summary** web page provides this information about the CPU:

- Ethernet service diagnostic information
- Version descriptions for installed firmware and software
- CPU 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 (**Menu → Module → Status Summary**):



Diagnostic and Status Information

The **Status Summary** web page provides this information:

Parameters	Description	
LEDs	The web page displays the state of these LEDs:	
	• RUN • ERR • I/O • DL • REMOTE RUN • BACKUP • ETH MS • ETH MS	• A • B • PRIM • STBY • FORCED_IO • SRUN (safety PAC) • SMOD (safety PAC)
	NOTE: The LEDs on the web page behave the same as the LEDs on the CPU *(see page 165)*.	
Service Status	This area presents information describing the status of CPU Ethernet services. The colored icons appearing to the left of some items indicate the following status:	
	green	The available service is operational and running.
	red	An error is detected in an available service.
	black	The available service is not present or not configured.
	The status of these Ethernet services is included:	
	• DHCP Server • FDR Server • Access Control	• Scanner Status • NTP Status • FDR Usage
Version Info.	This area describes the software versions that are running on the CPU, including:	
	• Executable Version • Web Server Version	• Web Site Version • CIP Version
CPU Summary	This area describes the CPU hardware and the applications that are running on the CPU, including: • Model • State • Scan Time	
Network Info.	This field contains IP addressing settings for the CPU, including: • IP Address • Subnet Address • Gateway Address	

HSBY Status

Introduction

The **HSBY Status** web page provides this information about the Hot Standby system:

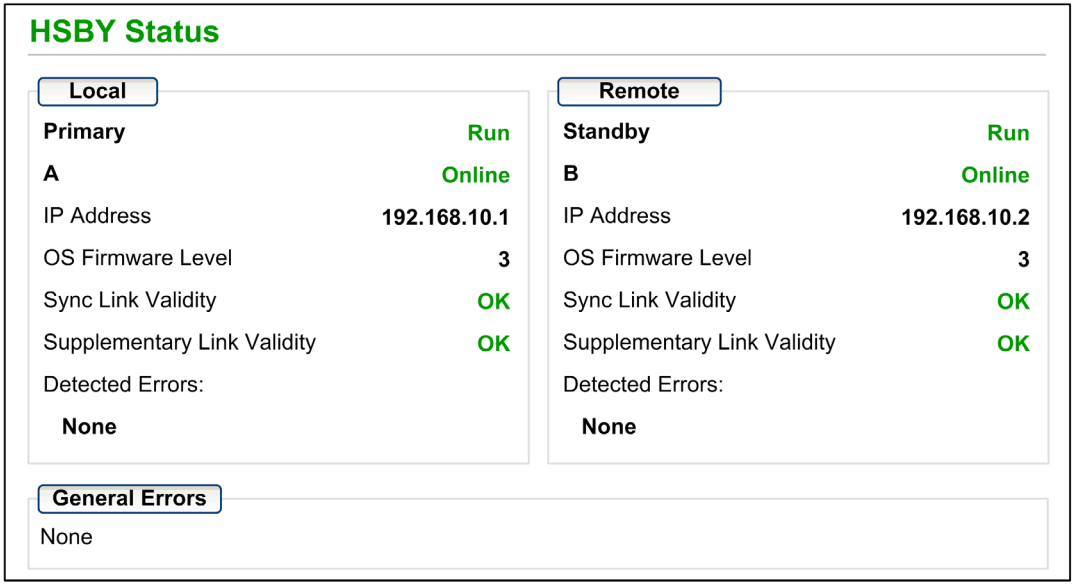
- Hot Standby role and status of the **Local** CPU
- Hot Standby role and status of the **Remote** CPU
- General errors detected for the Hot Standby system

NOTE:

- The local CPU is the CPU 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 (**Menu → Module → HSBY Status**):



Diagnostic and Status Information

The **HSBY Status** web page provides this information:

Area	Description	
Local/Remote	This area displays the state of Hot Standby settings for the local and remote CPUs:	
	<Hot Standby Role>	The Hot Standby system role of the CPU. Valid values include: ● Primary ● Standby ● Wait
	<Operating State>	The operating state of the CPU. Valid values include: ● RUN ● STOP ● NoConf ● HALT
	A/B switch setting	The designation of the CPU, defined by the rotary switch *(see page 23)* on the back of the CPU. Valid values include: ● A ● B
	<Run Mode>	The designation of the CPU, defined by the rotary switch on the back of the CPU. Valid values include: ● Online ● Wait
	IP Address	The IP address used to communicate with the CPU for web page access: ● For the primary Hot Standby CPU, this is the **Main IP Address** setting. ● For the standby Hot Standby CPU, this is the **Main IP Address** setting + 1.
	OS Firmware Level	Firmware version of the CPU operating system.
	Sync Link Validity	The status of the Hot Standby link *(see page 51)*: ● OK: the link is operational. ● NOK: the link is not operational.
	Supplementary Link Validity	The status of the Ethernet RIO link *(see page 52)*: ● OK: the link is operational. ● NOK: the link is not operational.
	Detected Errors	Detected errors for the CPU, including: ● HSBY link error detected ● RIO link error detected (the connection between PAC A and PAC B over the Ethernet RIO network) ● RIO error detected (the connection between a PAC and (e)X80 EIO adapter modules over the Ethernet RIO network.

Area	Description
General Errors	Detected errors for the Hot Standby system, including: ● Application mismatch ● Logic mismatch ● Firmware mismatch ● Data layout mismatch ● Backup application mismatch ● Safety logic mismatch (for safety PACs)

Rack Viewer

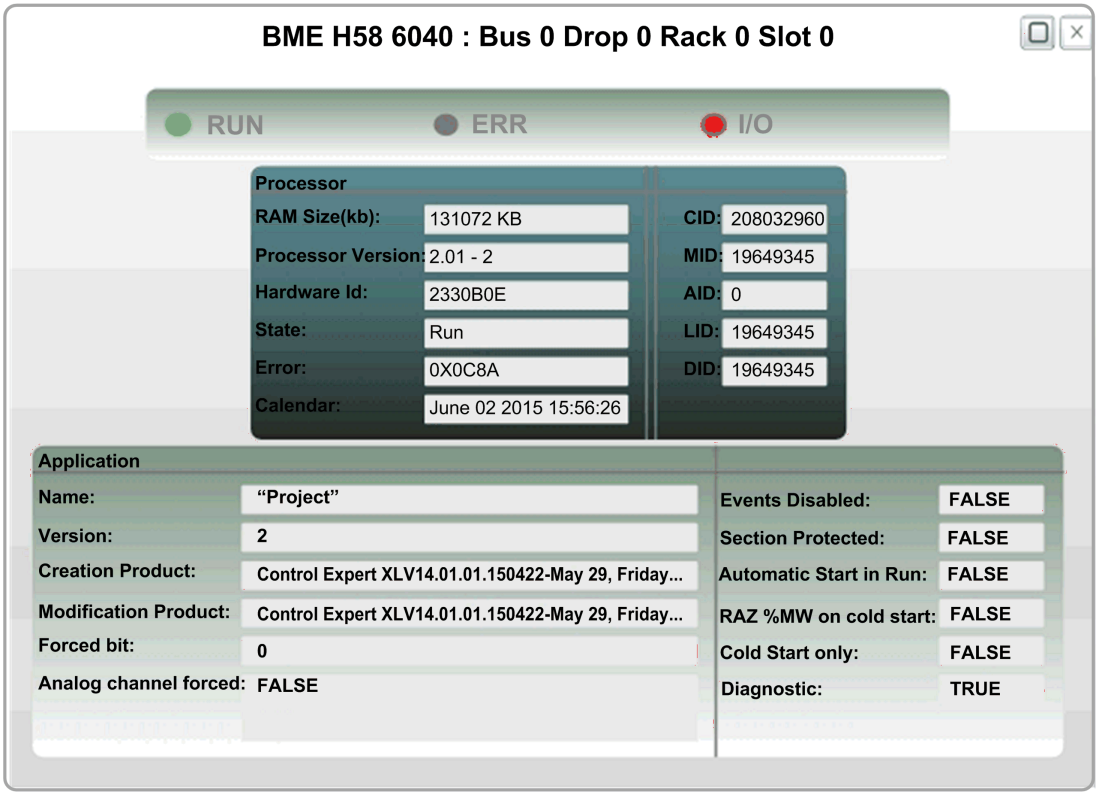
Introducing the CPU Status Page

The BMEH584040(S) and BMEH586040(S) Hot Standby CPUs include a **Rack Viewer** web page. Use this page to view CPU information, including:

- LEDs status
- processor identification
- application signature identification
- select application configuration settings

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 Data

The **Rack Viewer** page for M580 Hot Standby CPUs displays the following data:

Data Field	Description
Processor	
RAM size (kb)	The size of processor RAM in KB
Processor Version	Firmware version
Hardware ID	An identifier for the module hardware. OS Loader checks this value to determine compatibility between the hardware and the operating system.
State	The operating state of the processor: ● NO CONFIGURATION ● IDLE ● STOP ● RUN ● HALT ● INITIALIZING ● ERROR ● OS LOADER
Error	The identity of the last detected error
Calendar	Date and time of last detected error
Signature	
CID	*Creation ID*: Random number generated when an application is created. The number remains a constant.
MID	*Modification ID*: Random number generated on each application modification and rebuild, either partial or global. When an application is created, MID = CID.
AID	*AutoModification ID*: A new random value is generated for AID by the PAC after one of the following minor modifications to the application: ● a Control Expert request to modify %KW ● a P_Unit request that performs a save_param request or replaces init value When an application is created or built in local mode, AID = 0.
LID	*Layout ID*: Random number generated after a modification of the variable layout. LID does not change as a result of a runtime change either adding or deleting a data block.LID changes only on when the global rebuild of the application. LID addresses the needs of Hot Standby. It permits the transfer of a memory block from the primary PAC to the standby so that application variables (except for deleted or new ones) exist at the same location. LID = CID = MID when the application is created.
DID	*Data ID*: Indicates that a block of data has been freed. Also used for the special case of remapping a symbol from unlocated to located.
Application	
Name	Name of the Control Expert project
Version	Project version
Creation Product	Includes both: ● Version and build of Control Expert used to create the project. ● Date and time the project was created.

Data Field	Description
Modification Product	Includes both: ● Version and build of Control Expert used to edit the project. ● Date and time the project was last edited.
Events Disabled	Indicates if all event processing has been disabled: ● **True** indicates all event processing has been disabled. ● **False** indicates event processing has not been disabled. **NOTE:** Events can be enabled/disabled by using: ● The **Enable or Disable all** command *(see EcoStruxure™ Control Expert, Operating Modes)* in the **Task** tab of the CPU. ● The `MASKEVT` and `UNMASKEVT` functions. ● System bit %S38.
Forced bit	The number of forced bits in the application.
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.
Last Stop	The event that last caused the application to stop. Values include: ● Changeover from RUN to STOP by the terminal or dedicated input ● Stop on software detected fault (task overrun or SFC overrun) ● Power loss detected ● Stop on hardware detected fault ● Stop on HALT instruction
Last Stop Date	Date an event last caused the application to stop.
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 PAC goes into RUN operational mode: ● **True** indicates the application automatically starts. ● **False** indicates the application does not automatically start.
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.
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.

Section 8.4
M580 Hot Standby System Diagnostics

M580 Hot Standby System Diagnostics

Introduction

The M580 Hot Standby system continuously monitors the system state, and adds to its diagnostic buffer an entry for each detected error or change of state event. You can view and handle this collection of events using the following tools:

- **Alarm Viewer** web page *(see Modicon M580, Hardware, Reference Manual)*, for events relating to the selected CPU.
- **Diagnostic Viewer** in Control Expert *(see EcoStruxure™ Control Expert, Operating Modes)*, for detected events relating to the Hot Standby system.

M580 Hot Standby System Messages

Each detected system event presents:

- A message describing the event type.
- An explanatory text symbol entry, more particularly describing the event.
- A numeric decimal identifier, representing the combination of message and symbol.

The M580 Hot Standby system can display the following messages

ID (dec)	Message (Event Text)	Symbol (Event Type)	Possible Cause
14101	Switch from Wait to Primary	No Error	–
14102	Switch From Wait to Standby	Linked to Primary. No Error	–
14103	Switch from Standby to Primary	No remote PLC connection	No Hot Standby link and EIO link between CPUs.
14104	Switch from Standby to Primary	Remote PLC not Primary	• Loss of power on former primary. • Former primary stopped. • Error detected on former primary.
14105	Switch from Standby to Wait	Hsby Link Error	• Break in Hot Standby link cable • Transceiver inoperable in either CPU.
14106	Switch from Standby to Stop	PLC not in RUN	Standby CPU stopped.
14107	Switch from Primary to Wait	Loc RIO err and no peer RIO err	Former primary CPU lost connection to all (e)X80 EIO adapter modules; former standby (now primary) CPU maintains connection to at least one (e)X80 EIO adapter module.
14108	Switch from Primary to Wait	Swap Command	Former primary CPU received swap command.

ID (dec)	Message (Event Text)	Symbol (Event Type)	Possible Cause
14109	Switch from Primary to Stop	PLC not in RUN	Former primary CPU stopped (PAC in STOP or one task in HALT)
14110	Switch from Primary to Wait	PLC_B linked to Primary	–
14111	Peer PLC disconnection on RIO Link	RIO Link Error	Two breaks in Ethernet RIO cable have isolated the remote CPU.
14112	Peer PLC disconnection on Hsby Link	Hsby Link Error	• Break in Hot Standby link cable • Transceiver inoperable in either CPU.
14113	Mismatch Error	FW mismatch	Different firmware versions in each CPU.
14114	Mismatch Error	Logic mismatch	Different application logic revisions running in each CPU.
14115	Mismatch Error	Application mismatch	Different applications running in each CPU.
14116	Degraded Hsby Data transfer	Data Layout mismatch	Online changes to data structure have been made to primary CPU, but not transferred to standby.
14117	Bad peer rotary switch config	Not in a PLC_A and PLC_B config	Rotary switch settings do not specify an A and a B PAC.
14118	Power supply error	Loss of redundancy	One of the BMXCPS4002 redundant power supply units is no longer functioning.

Section 8.5
M580 System Words

Modicon M580-specific System Words %SW132 to %SW167

Diagnostic System Words

⚠ WARNING

UNEXPECTED APPLICATION BEHAVIOR

Do not use system objects (%Si, %SWi) as variable when they are not documented.

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**: CPU MAC Address.
- **%SW135** to **%SW137**: CPU 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*.

Chapter 9
Updating Hot Standby System Firmware

Firmware Update

Introduction

You can update firmware for modules in an M580 Hot Standby system by installing a new firmware version using Unity Loader.

The firmware installation can be performed for the following modules:

- Hot Standby CPUs
- (e)X80 EIO adapter modules
- Ethernet communication modules
- Ethernet network option switch modules

NOTE: In a Hot Standby system, Schneider Electric recommends that you update the firmware in system communication modules (for example, BMENOC0301/11) before updating the firmware in the CPUs.

Refer to the *Unity Loader, User Manual* for a description of the download procedure.

Firmware File

The firmware file is a **.ldx* file.

Updating CPU Firmware Without Stopping the Hot Standby System Process

You can update firmware in the two Hot Standby CPUs without interrupting the system process if the following pre-conditions exist:

- One CPU is running as the primary.
- The other CPU 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` *(see page 146)* property.
- Firmware mismatch is allowed for the Hot Standby system, by setting the `FW_MISMATCH_ALLOWED` *(see page 146)* property
- The new firmware to be installed is not fundamentally different from the old firmware, and therefore will not trigger an `APP_MISMATCH` *(see page 122)* condition.

When updating the CPU firmware, install the standby CPU firmware first, then install the primary CPU firmware.

To update Hot Standby CPU module firmware, follow these steps:

Step	Action
1	Install on your PC the Unity Loader software.
2	Connect the PC that is running Unity Loader to one of the following ports on the standby CPU module: ● the CPU mini-B USB port, or ● the CPU Ethernet service port
3	Launch Unity Loader.
4	Click **Firmware** tab.
5	In the **PC** list box, select the *.ldx* file that contains the firmware file.
6	Check that transfer sign is green to allow transfer from PC to the module.
7	Click **Transfer**.
8	After the transfer of firmware is complete, click **Close**. **NOTE:** Upon completion of the firmware update, the standby CPU remains the standby CPU.
9	Confirm that the installation of firmware did not create an application mismatch condition.
10	Repeat steps 2 through 8, above, for the primary CPU. **NOTE:** ● When you begin the transfer, the standby CPU immediately becomes the primary. ● The primary CPU becomes the standby upon completion of the firmware update.

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 (running Unity Loader) 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 (running Unity Loader) to the CPU to perform this update, the process executing in the CPU stops until the update is complete.

Follow these steps to update the firmware to other modules in the Hot Standby system:

Step	Action
1	Install Unity Loader software.
2	Connect the PC that is running Unity Loader to the service port of the target module, which can be: ● a (e)X80 EIO adapter module ● an Ethernet communications module ● an Ethernet network option switch module
3	Launch Unity Loader.
4	Click **Firmware** tab.
5	In the **PC** list box, select the *./dx* file that contains the firmware file.
6	Check that transfer sign is green to allow transfer from PC to the module.
7	Click **Transfer**.
8	Click **Close**.

Chapter 10
Replacing M580 Hot Standby CPUs

Replacing Hot Standby Hardware Modules

Overview

Replace the modules in this order:
- Standby PAC (PAC B in this example)
- Primary PAC (PAC A in this example)

Replacing PAC B Procedure

Replace the modules in the standby PAC:

Step	Action
1	Confirm that the application program running on the M580 Hot Standby PAC has been exported in the ZEF format and is available on the computer. If not, upload the application program from one of the two PACs to Control Expert.
2	Export the application in the ZEF format on the Control Expert workstation.
3	If not yet installed, install Unity Pro XL version 11.0 (or higher). **NOTE:** Unity Pro is the former name of Control Expert for version 13.1 or earlier.

⚠ WARNING

SYSTEM NO LONGER ACTIVE NOR REDUNDANT

Before stopping the system, confirm that no critical operation is in progress. The system is inactive and non-redundant.

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

Step	Action
4	Stop the standby PAC (PAC B) and power it off. **NOTE:** At this point, the system is no longer operating redundantly.
5	Disconnect the Hot Standby sync link cable from PAC B.
6	Replace hardware or upgrade *(see Quantum using EcoStruxure™ Control Expert, Change Configuration On The Fly, User Guide)* the PAC B firmware with version 2.10 or higher.

Step	Action
7	Confirm that there is no program in PAC B: **a.** Set the rotary switch *(see page 23)* to **Clear**. **b.** Power up the PAC. **c.** Wait approximately one minute, until LEDs A and B are blinking. **d.** Power down the PAC. **e.** Set the rotary switch to **B**.
8	Power on PAC B.
9	If using an SD memory card, insert the card in PAC B. (Refer to the SD memory card instructions for information about existing programs on the card.) **NOTE:** Confirm that the PAC is in a NOCONF state *(see Modicon M580, Hardware, Reference Manual)*.
10	Import the ZEF file of the application.
11	In the **PLC Bus** editor, replace the current version of the PAC with the new firmware PAC version.
12	Select the **Online modification in RUN or STOP** check box in the PAC **Configuration** tab to enable the configuration change.
13	Rebuild the application (**Build → Rebuild All Project**) and download into PAC B. The PAC is in STOP mode.
14	Connect the Hot Standby sync link cable to PAC B.
15	Connect Control Expert to PAC A.

⚠ WARNING

LOSS OF COMMUNICATION

Before changing the mode of PAC A to STOP, confirm that no critical operation is in progress. Once confirmed, the system is considered inactive or non-redundant.

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

16	Stop PAC A. **NOTE:** The system is no longer active nor redundant.
17	Connect Control Expert to PAC B.

⚠ WARNING
UNEXPECTED APPLICATION BEHAVIOR - LOSS OF DATA At the end of the application download, all application data in the PAC B have their initial value. Before you change the mode of PAC B to RUN, confirm that the application can restart with the initial values. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

18	Put PAC B in RUN mode.
19	Confirm that PAC B is now the primary.

Replacing PAC A Procedure

After you replace PAC B *(see page 195)*, follow these steps to replace PAC A:

Step	Action
1	Power off PAC A, which is in STOP mode. **NOTE:** At this point, the system is no longer operating redundantly.
2	If using an SD memory card, remove it.
3	Disconnect the Hot Standby sync link cable from PAC A.
4	Replace hardware or upgrade *(see Quantum using EcoStruxure™ Control Expert, Change Configuration On The Fly, User Guide)* the PAC B firmware with version 2.10 or higher.
5	Power on PAC A.
6	If using an SD memory card, insert it in PAC A. **NOTE:** Confirm that the PAC is in a **No Conf** state.
7	Connect the Hot Standby sync link cable to PAC A.
8	An automatic transfer from primary to standby occurs.
9	Execute a RUN command on PAC A.
10	Confirm that PAC A is now the standby.

Chapter 11
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 PACs. Only devices enabled in the address server (DHCP) are controlled.

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 against 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)
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).

Parameter	Description
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 - the device Name,
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**. A preliminary, read-only global view of your network displays.
2	Check for settings with a red background, indicating the tool has detected a configuration error.
3	Click **OK** to close the **Network Inspector** tool.
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 controlled.
6	Configure all 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 checks 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 up 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 CPU 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%.

Glossary

!

%I

According to the CEI standard, %I indicates a language object of type discrete IN.

%IW

According to the CEI standard, %IW indicates a language object of type analog IN.

%M

According to the CEI standard, %M indicates a language object of type memory bit.

%MW

According to the CEI standard, %MW indicates a language object of type memory word.

%Q

According to the CEI standard, %Q indicates a language object of type discrete OUT.

%QW

According to the CEI standard, %QW indicates a language object of type analog OUT.

%SW

According to the CEI standard, %SW indicates a language object of type system word.

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 CPU application takes to react to a given input. ART is measured from the time a physical signal in the CPU 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

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.

connection originator

The EtherNet/IP network node that initiates a connection request for I/O data transfer or explicit messaging.

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

(*distributed I/O*) Legacy term for distributed equipment. DRSs use DIO ports to connect distributed equipment.

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 CPU 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 CPU with DIO scanner service on the local rack. DIO network traffic is delivered after RIO traffic, which takes priority in an RIO network.

distributed equipment

Any Ethernet device (Schneider Electric device, PC, servers, or third-party devices) that supports exchange with a CPU 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.schneider-electric.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 simple graphical user interface (GUI) for setting device parameters to a highly sophisticated application capable of performing complex real-time calculations for diagnosis and maintenance purposes. In the context of a DTM, a device can be a communications module or a remote device on the network.

See FDT.

E

EDS

(*electronic data sheet*) EDS are simple text files that describe the configuration capabilities of a device. EDS files are generated and maintained by the manufacturer of the device.

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.

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.

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

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.

Ethernet DIO scanner service

an embedded scanner service of M580 CPUs (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 CPUs (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.

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.

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.

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.

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.

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`.

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)).

IODDT

(*input/output derived data type*) A structured data type representing a module, or a channel of a CPU. 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.

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 CPU 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 CPU 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.

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.

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

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.

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

A scanner acts as the originator of I/O connection requests for implicit messaging in EtherNet/IP, and message requests for Modbus TCP.

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.

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 CPU
- 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 CPU. This graphical description of the CPU'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 CPU 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.

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

(*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.

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.

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`.

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.

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.

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

A

application response time, *157*
architecture, *48*
ART, *157*
  swap, *161*
  switchover, *160*

B

backup, *115*
block
  service port, *80*
BMENOC0321
  topologies, *71*
BMXRMS004GPF, *42*

C

CCOTF, *102*
certifications, *26*
Change
  Hot Standby PLC, *195*
clear
  application, *23*
communication loss
  loop, *80*
compatibility, *30*
configuration
  service port, automatically block, *80*
Control Expert
  application languages, *97*
  libraries, *97*
control network
  topologies, *71*
CPU
  clear, *23*
  configuring, *98*

D

data attribute
  Exchange On STBY, *108*
  Retain, *107*
data exchange, *118*
DDT
  LOCAL_HSBY_STS, *121*
  REMOTE_HSBY_STS, *121*
  T_M_ECPU_HSBY, *121*
diagnostics
  Control Expert status viewer, *172*
  Hot Standby LEDs, *165*
  memory card, *169*
  system, *188*
  web pages, *177*
download, *115*

E

elementary functions (EFs), *128*
Ethernet loop
  loss of communication, *80*
Ethernet network manager, *199*
Ethernet RIO link, *52*
eX80 RIO drop, *39*
Exchange On STBY, *108*

F

firmware
  update, *191*
  upgrade, *191*
flat network
  service port block, *80*

H

hardware
  local rack, *34*
Hold up time, *109*

Host Standby system
commands, *146*
Hot Standby link, *51*
Hot Standby PLC
Change, *195*
Hot Standby system
PAC state examples, *138*
starting, *132*
HSBY status web page
CPU, *182*

I

I/O memory, *152*
IP address
A, *106*
B, *106*
configuring, *105*
main, *105*
main + 1, *105*

L

LEDs
Hot Standby, *165*
life cycle, *46*, *49*
local rack
hardware, *34*
LOCAL_HSBY_STS, *121*
loop
loss of communication, *80*

M

memory card, *42*
diagnostics, *169*
memory consumption, *149*
mismatch
application, *31*
firmware, *31*
logic, *31*

N

network, flat
service port block, *80*

P

PAC
state transitions, *136*
states, *135*
physical description
CPU, *22*
project
transfer, *110*

Q

Quantum RIO drop, *41*

R

REMOTE_HSBY_STS, *121*
restore, *115*
Retain, *107*

S

service port
automatically block, *80*
SFC section
online changes to, *104*
SFP transceiver, *36*
standards, *26*
status summary web page
CPU, *179*
status viewer, *172*
switchover, *17*
synchronization, *174*
system
diagnostics, *188*
system states
Hot Standby, *27*

T

T_M_ECPU_HSBY, *121*
task
    associating with variable, *119*
task cycle time
    calculating, *154*
topologies
    frequently used, *54*
    using BMENOC0321, *71*
transfer project, *110*

U

update
    firmware, *191*
upgrade
    firmware, *191*

W

web pages, *177*
    rack viewer, *185*

X

X80 RIO drop, *39*