

# Premium

## Hot Standby with Unity

### User Manual

04/2015

---

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.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Schneider Electric.

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.

© 2015 Schneider Electric. All rights reserved.

---

# Table of Contents

---



	<b>Safety Information</b> .....	<b>7</b>
	<b>About the Book</b> .....	<b>9</b>
<b>Part I</b>	<b>Introduction</b> .....	<b>11</b>
<b>Chapter 1</b>	<b>Hot Standby Concepts</b> .....	<b>13</b>
	Terminology .....	<b>14</b>
	Purpose and Features .....	<b>15</b>
	Overview .....	<b>17</b>
	Redundant Hardware .....	<b>19</b>
	Core Hot Standby Hardware .....	<b>24</b>
	Configuration Requirements .....	<b>26</b>
	Establishing Redundancy .....	<b>30</b>
	Revised Operation Modes .....	<b>33</b>
	Programming Differences .....	<b>34</b>
	Hot Standby Restricted Functions .....	<b>38</b>
<b>Chapter 2</b>	<b>Hot Standby Overview</b> .....	<b>45</b>
	Introduction to the Controller .....	<b>46</b>
	Operating Limits .....	<b>52</b>
	Certifications and Standards .....	<b>53</b>
<b>Chapter 3</b>	<b>Hot Standby Systems</b> .....	<b>55</b>
3.1	Minimum Configurations by I/O Type .....	<b>56</b>
	Minimum Configuration for Redundant Discrete I/O .....	<b>57</b>
	Minimum Configuration for Redundant Analog I/O (Inputs Only) .....	<b>60</b>
	Minimum Configuration for Redundant Analog I/O (Outputs Only) .....	<b>62</b>
	Minimum Configuration for Ethernet I/O .....	<b>66</b>
	Minimum Configuration for Redundant Modbus I/O .....	<b>71</b>
	Adding HMI / SCADA to the ETY-sync link .....	<b>76</b>
3.2	Compatible Equipment .....	<b>78</b>
	Overview .....	<b>79</b>
	Premium Racks and Rack Accessories .....	<b>81</b>
	Premium Power Supplies .....	<b>82</b>
	In-rack Communication Modules: Ethernet .....	<b>83</b>
	In-rack Communication Modules: Modbus .....	<b>84</b>
	In-rack I/O Modules: Discrete .....	<b>85</b>
	In-rack I/O Modules: Analog .....	<b>86</b>
	In-rack I/O Modules: Safety .....	<b>87</b>

	Connection Devices: Discrete I/O . . . . .	88
	Connection Devices: Main Rack Analog I/O . . . . .	89
	Allowed Devices: Connected by Ethernet . . . . .	90
	Allowed Devices: Connected by Modbus . . . . .	91
	Ethernet Network Devices . . . . .	92
	Modbus Network Devices and Cables . . . . .	93
	Maximum Configuration . . . . .	94
3.3	Example Hot Standby Systems . . . . .	97
	System with Multiple ETYs Running I/O Scanning Services . . . . .	98
	System with Redundant I/O and SCADA Network Services . . . . .	100
	System with Mixed Ethernet and Modbus . . . . .	102
<b>Chapter 4</b>	<b>PLC Communications and Switchover . . . . .</b>	<b>103</b>
4.1	Database Transfer Between Hot Standby PLCs . . . . .	104
	Understanding the Premium Hot Standby Database Transfer Process . . . . .	105
	Understanding System Scan Time in Premium Hot Standby . . . . .	107
<b>Chapter 5</b>	<b>Switchover and Swap in Premium Hot Standby . . . . .</b>	<b>111</b>
	Ethernet Service Switchover Latencies . . . . .	112
	In-rack I/O Switchover Latencies . . . . .	113
	Presentation for Swapping in Premium Hot Standby . . . . .	114
	Operating recommendations for Swapping in a Premium Hot Standby system . . . . .	115
<b>Chapter 6</b>	<b>Compatibility with PL7 Systems . . . . .</b>	<b>119</b>
	Considerations When Upgrading From PL7 . . . . .	120
	Using the PL7-Unity Pro Converter . . . . .	121
<b>Part II</b>	<b>Configuration and Use . . . . .</b>	<b>123</b>
<b>Chapter 7</b>	<b>Configuring in Unity Pro . . . . .</b>	<b>125</b>
7.1	Configuring a System with the Unity Pro Tabs and Dialogs . . . . .	126
	Introducing Unity Pro . . . . .	127
	Accessing the Base Configuration . . . . .	128
	Using the Overview Tab . . . . .	129
	Using the Configuration Tab . . . . .	130
	Using the Animation Tab and PLC Screen Dialogs . . . . .	132
	Using the Premium Hot Standby Tab . . . . .	138
	Configuring In-rack I/O . . . . .	139
	Configuring the PCMCIA Cards . . . . .	140
	Swapping Network Addresses at Switchover . . . . .	142

7.2	Configuring TSX ETY 4103/5103 Modules . . . . .	<b>146</b>
	Overview of Premium Hot Standby TSX ETY . . . . .	<b>147</b>
	ETY Operating Modes and Premium Hot Standby . . . . .	<b>150</b>
	IP Address Assignment . . . . .	<b>153</b>
	Network Effects of Premium Hot Standby . . . . .	<b>155</b>
7.3	Configuring Registers . . . . .	<b>158</b>
	Understanding System Words and Bits . . . . .	<b>159</b>
	Understanding the Non-Transfer Area and Reverse Transfer Words . . . . .	<b>160</b>
	Understanding the Unity Command Register . . . . .	<b>161</b>
	Understanding the Unity Status Register . . . . .	<b>163</b>
	Using Initialized Data . . . . .	<b>166</b>
	Synchronization of Realtime Clocks . . . . .	<b>167</b>
<b>Chapter 8</b>	<b>Programming/Debugging . . . . .</b>	<b>169</b>
8.1	Developing Your Hot Standby Application . . . . .	<b>170</b>
	Programming Method . . . . .	<b>171</b>
	How to Program a Premium Hot Standby Application . . . . .	<b>176</b>
	Structure of Database . . . . .	<b>178</b>
	Transferring Your Program to the Primary and Standby PLCs . . . . .	<b>185</b>
8.2	Debugging Your Hot Standby Application . . . . .	<b>186</b>
	Debugging . . . . .	<b>187</b>
	Adjusting MAST Task Properties . . . . .	<b>190</b>
	Using the Hotstandby system with the HSBY_SWAP DFB . . . . .	<b>194</b>
<b>Chapter 9</b>	<b>Operating . . . . .</b>	<b>199</b>
9.1	Start/Stop System . . . . .	<b>200</b>
	Starting the Two PLCs . . . . .	<b>201</b>
	Stopping the Premium Hot Standby . . . . .	<b>203</b>
9.2	Switchover . . . . .	<b>204</b>
	Operating Modes Overview . . . . .	<b>205</b>
	Conditions for Switchover . . . . .	<b>207</b>
<b>Chapter 10</b>	<b>Maintaining . . . . .</b>	<b>213</b>
	Verifying the Health of a Premium Hot Standby . . . . .	<b>214</b>
	Premium Hot Standby Detection and Diagnostics . . . . .	<b>215</b>
	Detecting Inoperative Primary CPU- and ETY-sync Links . . . . .	<b>217</b>
	Detecting Inoperative Standby CPU- and ETY-sync Links . . . . .	<b>218</b>
	Detecting Inoperative CPU-sync Links . . . . .	<b>219</b>
	Checking for Identical Application Programs—Checksum . . . . .	<b>220</b>
	Replacing an Inoperative Module . . . . .	<b>221</b>
	Troubleshooting a Hot Standby PLC . . . . .	<b>223</b>

---

<b>Part III</b>	<b>Modifying and Upgrading</b> .....	<b>225</b>
<b>Chapter 11</b>	<b>Handling Application Modification</b> .....	<b>227</b>
	Understanding Premium Hot Standby Logic Mismatch .....	<b>228</b>
	Online/Offline Modifications to an Application Program .....	<b>229</b>
<b>Chapter 12</b>	<b>Handling PLC Firmware Upgrades</b> .....	<b>235</b>
	Overview of Premium Hot Standby Firmware Upgrades .....	<b>236</b>
	Executing the Firmware Upgrade Procedure .....	<b>237</b>
<b>Appendices</b>	.....	<b>239</b>
<b>Appendix A</b>	<b>Additional Information</b> .....	<b>241</b>
	Additional Premium Hot Standby Specifications .....	<b>242</b>
	TextIDs .....	<b>249</b>
<b>Appendix B</b>	<b>Detailed Behavior on Interruption of Power, Communications, or Device Capabilities</b> .....	<b>251</b>
	Overview .....	<b>252</b>
	Halt Events or Stop Command on PLC .....	<b>254</b>
	CPU Hardware or Firmware Becomes Inoperative .....	<b>257</b>
	Interruption of Supply Power to Main Rack .....	<b>260</b>
	ETY Hardware or Firmware (Monitored by Hot Standby CPU) Becomes Inoperative .....	<b>263</b>
	ETY Hardware or Firmware (Not Monitored by Hot Standby CPU) Becomes Inoperative .....	<b>266</b>
	Ethernet Copro Becomes Inoperative .....	<b>269</b>
	Interruption of CPU-sync link between Primary and Standby PLCs ..	<b>272</b>
	ETY-sync Link Cable Disconnection with I/O Scanner Active .....	<b>274</b>
	Full ETY I/O Link Disconnection (Both Switches for Monitored I/O Inoperative) .....	<b>278</b>
	Discrete I/O Module Becomes Inoperative .....	<b>280</b>
	SCP Card in SCY Module Becomes Inoperative .....	<b>283</b>
<b>Glossary</b>	.....	<b>287</b>
<b>Index</b>	.....	<b>289</b>

---

# 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, 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.

---

# About the Book

---



## At a Glance

### Document Scope

This manual presents information necessary to configure and operate your Premium Hot Standby System consisting of the Premium Hot Standby processor (TSX H57 24M or TSX H57 44M) and the Unity Pro software. It also discusses the implementation of redundant I/O consistent with the Hot Standby system, including Discrete, Analog, and Ethernet I/O using the TSX ETY 4103 / 5103 modules. Finally, this manual provides information regarding device communication using Modbus and other Premium Ethernet services, and places the Hot Standby in context of a larger distributed system involving HMI / SCADA and remote network devices.

While this manual describes how to set up and configure a Premium Hot Standby System you have already installed, it does not describe the basic physical installation of the Premium Hot Standby CPU, rack, power supply, or associated hardware. Nor does it provide related information such as operating limits, grounding, electromagnetic compatibility, or other environmental considerations. For details on these topics, please refer to the Related Documents.

### Validity Note

This documentation is valid for Unity Pro 10.0 or later.

### Related Documents

Title of Documentation	Reference Number
Premium and Atrium using Unity Pro, Processors, racks and power supply modules, Implementation manual	35010524 (Eng), 35010525 (Fre), 35006162 (Ger), 35006163 (Spa), 35012772 (Ita), 35012773 (Chs)
Grounding and Electromagnetic Compatibility of PLC Systems, Basic Principles and Measures, User Manual	33002439 (Eng), 33002440 (Fre), 33002441 (Ger), 33002442 (Spa), 33003702 (Ita), 33003703 (Chs)

You can download these technical publications and other technical information from our website at [www.schneider-electric.com](http://www.schneider-electric.com).

---

## Product Related Information

### **DANGER**

#### **HAZARD OF ELECTRIC SHOCK, EXPLOSION or ARC FLASH**

- Be familiar with the power requirements of all devices and accessories being installed, removed, or maintained in the Premium Hot Standby system.
- Always use a properly rated voltage-sensing device to confirm that power is off.
- Replace and secure all covers and elements of the system before reapplying power.
- Confirm that all affected PLCs are loaded with the correct application program before reapplying power.
- Use only the specified voltage for your TSX PSY series power supply when placing the system in operation.

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

### **WARNING**

#### **UNEXPECTED SYSTEM BEHAVIOR - INVALID CONTROL PATHS**

- The designer of any control scheme must consider the potential failure modes of control paths. He must provide a mean to achieve a safe state during and after a path failure for certain critical control functions, for instance emergency stop and overtravel stop.
- Separate or redundant control paths must be provided for critical control functions as well.
- System control paths must include communication links. Consideration must be given to the implications of unanticipated transmission delays or failures of the link.
- Each implementation of a Premium processor-based system must be individually and thoroughly tested for proper operation before being placed into service.

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

### **WARNING**

#### **UNINTENDED EQUIPMENT OPERATION**

The application of this product requires expertise in the design and programming of control systems. Only persons with such expertise should be allowed to program, install, alter, and apply this product.

Follow all local and national safety codes and standards.

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

---

# Part I

## Introduction

---

### Purpose of the Part

This part introduces the Premium Hot Standby controller and provides an overview of its use in various basic systems. It provides a description of the hardware, presents concepts necessary to understand the high availability of a Hot Standby implementation, and gives an overview of the operating modes and the transitions between operating modes.

### What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
1	Hot Standby Concepts	13
2	Hot Standby Overview	45
3	Hot Standby Systems	55
4	PLC Communications and Switchover	103
5	Switchover and Swap in Premium Hot Standby	111
6	Compatibility with PL7 Systems	119



---

# Chapter 1

## Hot Standby Concepts

---

### Purpose of this Chapter

This chapter begins with a discussion of the purpose and features of the Premium Hot Standby, and introduces some of the concepts you must understand to use the system properly.

### What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Terminology	14
Purpose and Features	15
Overview	17
Redundant Hardware	19
Core Hot Standby Hardware	24
Configuration Requirements	26
Establishing Redundancy	30
Revised Operation Modes	33
Programming Differences	34
Hot Standby Restricted Functions	38

## Terminology

### Terms and Acronyms

This manual uses many technical terms and acronyms. Some of the most commonly used are:

- *Application Program*: this term refers to the software program you write to provide monitoring and control for your application.
- *Controller or PLC*: this manual uses these terms interchangeably to refer to the Hot Standby Programmable Logic Controllers. Each Controller contains two important microprocessors, the CPU and the Copro.
- *CPU*: this is an acronym for Central Processing Unit, which is the microprocessor that performs general system functions and processes your application program.
- *Copro*: this term is short for Coprocessor. This manual uses the term Copro specifically to refer to the coprocessor that governs the exchange of data between the Hot Standby PLCs.
- *Switchover*: this refers to the moment when application control transfers from the Primary controller to the Standby controller. The Switchover event has a finite duration. It can be initiated manually, programmatically or automatically by system conditions.
- *Swap*: this refers to the moment when application control transfers from the Primary controller to the Standby controller. The swap event has a finite duration. It can be initiated programmatically using a specific DFB.
- *Main rack*: this is the rack that supports the processor.
- *In-rack I/O*: Because a Premium Hot Standby system requires that the primary and standby racks be identically configured ([see page 26](#)), both racks contain all the same I/O modules at the same rack locations and with the same firmware versions.  
In-rack I/O may be configured to be either redundant or local.  
Redundant in-rack I/O is operational only when the PLC in its rack is the Primary controller. Redundant I/O in a Standby rack does not operate.  
Each pair of redundant modules in the Primary and Standby racks is field-wired through a connection block or a Telefast device to the same field sensor or actuator.  
Local in-rack I/O always functions as long as its local PLC is online, regardless of whether the local PLC is the primary or the standby controller. Local in-rack I/O modules are independent field-wired to field sensors or actuators.  
Operations managed by local I/O are not considered part of the Hot Standby application because they do not function if the local PLC is offline.

## Purpose and Features

### Purpose

The Premium Hot Standby is an industrial control platform intended to provide automatic redundancy for a wide range of conditions. It assists you in meeting your system availability requirements at a reasonable cost. The main component of the system is a second PLC called the "Standby" PLC with an identical configuration to the main or "Primary" controller.

By detecting and responding programmatically to defined system conditions, the Premium Hot Standby can automatically transition from the Primary controller and its associated modules to the Standby controller and its identical modules. This transition, called the "Switchover".

Because the Premium Hot Standby detects and responds automatically to a wide range of error conditions, you will be able to reduce the length and complexity of your application programs. This in turn enables quicker implementations and lower development and maintenance costs.

### Features

The Premium Hot Standby system offers:

- increases the system availability of your treatment plants and remote stations, allowing you to conduct many maintenance operations while the system is operational
- Reduces your installation and operating expenses
- provides redundancy for I/O in the Premium racks and over Ethernet TCP/IP and Modbus networks
- requires no specialized modules or equipment other than the Hot Standby PLCs and Ethernet (TSX ETY ...) modules. You can reuse your Premium racks, power supplies, and I/O (analog, discrete).
- offers a user-friendly development environment compatible with IEC 61131-3
- allows creation of a redundant-ready application program almost as easily as for a standalone PLC, and requires few changes from your normal programming methods
- Hot Standby is a single-detected-fault-tolerant system. That is, the system continues operating even when one component of the system is inoperative.

### Use Restrictions

Monitored ETY modules (TSXETY4103/5103) are used in a Premium Hot Standby Multi-rack system for managing communication between the two main racks.

 **WARNING**

**UNEXPECTED SYSTEM OPERATION**

Do not use the following Ethernet services on Monitored ETY when using the Multi-rack function:

- NTP
- Messaging
- SNMP

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

Those services create overloads that can lead to malfunctions in the system and prevent the system from delivering the expected availability and quality.

The Premium Hot Standby Multi-rack has been tested and validated against power cuts. It is able to perform switchovers when power goes down on the rack where the Primary CPU is and is able to withstand several consecutive power cuts (provided power comes back on the rack that was powered off previously and the system restarts as Primary and Standby in a stable state, that is, a minimum of one minute is provided after the system has successfully rebooted).

The Premium Hot Standby Multi-rack system has **not** been designed to stand repetitive and frequent power cuts and may enter a state where the equipment under control is **not** managed (for example, Primary Offline/Standby Offline states).

## Overview

### System Redundancy

The Premium Hot Standby controller implements system redundancy using redundant hardware and by automatically switching over to the Standby (backup) hardware on detecting defined system events. While your prior PLC experience is very important to the proper use of this system, you will need to become familiar with new concepts, practices, and restrictions in order to properly implement and manage the Premium Hot Standby's redundancy. In this section, we present some of the concepts that are most important to developing this familiarity. This section does not offer a comprehensive discussion of these topics, but it should aid your understanding of this manual.

**NOTE:** Users of Premium PL7 systems should be aware that significant differences exist between Unity- and PL7-based systems. There are further differences to be aware of if you are a Premium PL7 Warm Standby user. Read and understand this manual before upgrading from a PL7 Warm Standby system. See [Compatibility with PL7 Systems \(see page 119\)](#) for additional details.

**NOTE:** Users of Quantum Hot Standby and other redundant systems should be aware that differences exist between the redundancy provided by these systems and that provided by the Premium Hot Standby system. The differences include terminology, the conditions for switching to the standby system, system requirements and restrictions, and more. Read and understand this manual before implementing or installing your Premium Hot Standby system.

### What a Redundant Configuration Looks Like

#### **WARNING**

##### **UNINTENDED EQUIPMENT OPERATION**

Do not use the following for Ethernet communication between the Primary and Standby CPUs:

- embedded CPU Ethernet ports
- any active components, especially fiber optic converters

Use a copper cross-over Ethernet cable for the inter-CPU connection.

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

The embedded CPU Ethernet ports are dedicated to data exchange between Primary PLC and Standby PLC. This is the "CPU Sync Link".

Redundancy is possible at several levels in an architecture: SCADA Clients, Data Servers, Control Network, PACs Station, Field Network, Field Devices, etc. Redundancy is used to enhance the possibility that the services provided by the different parts continue to operate, generally without loss of data, in case of interruption. It is possible to differentiate several levels of redundancy according to their performances in terms of availability.

For more information about redundancy, please refer to the "System Technical Guide - High Availability solutions" document available from Schneider-Electric, which provides detailed and practical information about high availability systems.

### Topologies and Architectures

Schneider-Electric recommends several types of redundant architectures, that may differ according to the chosen range (Premium Hot Standby, Quantum Hot Standby, etc.).

The Premium Hot Standby solution is based on Ethernet; therefore, all Ethernet-based architectures may be used (Bus, Star, Tree, Ring, etc.). Different network topologies bring different levels of redundancy.

A Schneider-Electric Hot Standby system is used when downtime cannot be tolerated. It delivers high availability through redundancy and always consists of two units with identical configurations. When set in the appropriate modes, the Standby processor is able to take over immediately in case the Primary processor becomes inoperative.

Several communication networks may be used:

- Ethernet modules
- Modbus modules
- serial Modbus using SCP cards

## WARNING

### UNINTENDED EQUIPMENT OPERATION

- Make an uninterrupted point-to-point connection between the Hot Standby CPU-sync link ports.
- Do not connect any other Ethernet devices (such as switches and hubs) so that they share the same network cabling as the CPU-sync link.
- Do not exceed maximum Ethernet cable lengths for the types of cable used.

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

Selection criteria for choosing between Modicon Premium and Modicon Quantum systems are provided in the "PAC station" section of the "System Technical Guide - High Availability solutions" available from Schneider-Electric.

There are many other criteria that can be part of the decision (including, but not limited to: cost, performances and space), but the major technical differences are having Remote I/O redundancy or I/O redundancy. Modicon Quantum systems are suitable for Remote I/O redundancy, whereas Modicon Premium systems are the best solution for Local I/O redundancy.

There are other technical differences. For instance, some System Words (for example, %SW60 and %SW61) do not share the same mapping. Also, the Non-Transfer Area cannot be managed in the same way, for more information, please refer to the "System Technical Guide - High Availability solutions" document from Schneider-Electric.

---

## Redundant Hardware

### Two Controllers: Primary and Standby

The Standby PLC also regularly communicates information back to the Primary PLC using a group of 16-bit system words known as the Reverse Transfer Registers. The content of these system words is configurable, but commonly they provide the Primary PLC additional information on the health of the Standby controller and its associated modules. The fundamental requirement for a Premium Hot Standby system is the use of two completely identical Hot Standby PLCs, either two TSX H57 24M or two TSX H57 44M controllers. These controllers require the same firmware versions, and the modules need to be installed in the same slots in both the primary and standby racks. Both controllers need to run the same application program.

In a system that is operating nominally, with both controllers are fully functional, each controller assumes one of these operating modes

- Run Primary Mode (Primary PLC)
- Run Standby Mode (Standby PLC)

The role of the Primary PLC is almost identical to that of a single PLC in a non-Hot Standby system. That is, it runs your application program and provides the normal control functions expected from a standalone PLC.

The major differences from a standalone PLC are: (a) the Primary Hot Standby controller will communicate regularly with its Standby PLC so that the Standby remains ready to assume the Primary role if required, and; (b) the Primary PLC will monitor itself and certain associated equipment for the specific conditions that dictate a Switchover to the Standby controller. In addition, the Primary PLC manages all Redundant In-rack and Ethernet I/O.

However, there are two major differences from a standalone PLC:

- The Primary PLC communicates regularly with the Standby PLC so that the Standby remains ready to assume the Primary role if required.
- The Primary PLC monitors itself and certain associated equipment for specific conditions that dictate a switchover to the Standby controller. In addition, the Primary PLC manages all redundant in-rack and Ethernet I/O.

The role of the Standby PLC is to assume control of the system but not to interfere with the control asserted by the Primary controller. To do so, the Standby controller regularly monitors the state of the redundant in-rack I/O and the distributed Ethernet I/O, which is being solved by the Primary controller, and does not duplicate the I/O control signals sent by the Ethernet and Modbus modules on the Primary rack. The Standby PLC may execute control logic for any local I/O that resides in the same rack with it. The standby PLC solves only the first section (section 0) of the logic program; it waits for and then applies the output images from the Primary controller to the I/O participating in the Hot Standby application.

For a Premium HSBY multi-rack configuration, the following 2 limitations must be taken into account:

- NTP service is not available
- Messaging service is not allowed on a configured ETY module.

For more information about the two RUN states in a Premium Hot Standby system, refer to Revised Operating Modes ([see page 33](#)) and to Operating Modes Overview ([see page 207](#)).

### Establishing the Primary and Standby Controllers

Provided you have properly configured the overall system, the first Hot Standby PLC to which power is applied will assume the role of the Primary controller. Therefore, you can determine controller roles by delaying the application of power to one PLC using a time-lag relay or some related means.

When you apply power simultaneously to two properly configured Hot Standby PLCs, the firmware automatically assigns the role of the Primary controller based on the MAC addresses of the two PLCs. The PLC with the lowest MAC address will become the Primary controller.

### Distinguishing Between Controllers

In this manual and in Unity Pro we employ the common practice of distinguishing between the two physical controllers by labeling one as PLC A and the other as PLC B. In the event of a Switchover, or if you replace one of the PLCs, the identification of PLC A and PLC B may not align with the Primary and Standby operating modes in the manner you would expect, where PLC A equates to the Primary controller. The same is true for any physical labels you might apply to your PLCs to distinguish them in your system.

## WARNING

### UNINTENDED EQUIPMENT OPERATION

- Never assume that a PLC is in a certain operating mode before installing, operating, modifying, or servicing it.
- Before acting on a PLC, always positively confirm the operating mode of both Hot Standby PLCs by viewing their LEDs and checking their System Status Words.

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

### Two Ethernet Modules: Monitored ETYs

In addition to requiring two identical controllers, every Premium Hot Standby system requires a minimum of two identical Premium Ethernet modules, one on each rack. These modules can be either two TSX ETY 4103's or two TSX ETY 5103's. Like the controllers, the rack positions and firmware versions of the ETY modules must be identical. Unlike the controllers, a minimum firmware version is required - your system will not be able to function redundantly unless both ETYs have matching firmware versions 4.0 or higher.

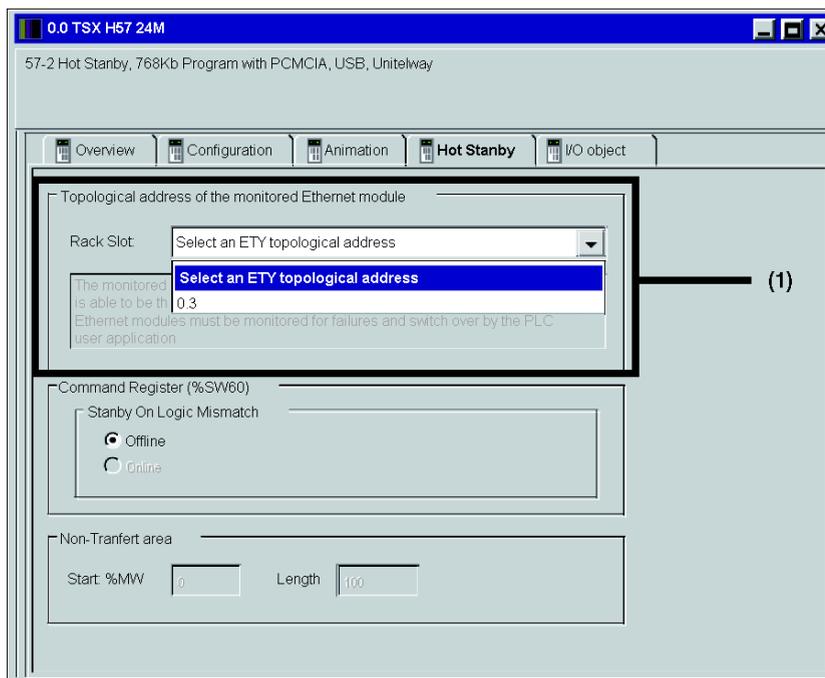
## Roles of the Monitored ETYs

The monitored ETY modules play two roles in Hot Standby high availability:

- They provide a channel for additional communication that helps the PLCs identify the cause of a switchover. For example, the monitored ETYs can send you information to help determine whether a Primary PLC is non-responsive due to a detected error in your application program or due to an interruption of supply power to the Primary rack.
- These monitored ETYs provide the basis for redundant control of I/O over an Ethernet TCP/IP network. To establish redundant Ethernet I/O, run the I/O Scanning service from the monitored ETY module in the Primary rack. Because all redundant Ethernet I/O is managed using the monitored ETY, it is usually referred to as monitored I/O.

## Selecting the Monitored ETYs

Of course, a Premium Hot Standby system can support more than one pair of ETY modules, provided they meet the requirements above. However, you must designate one pair as the “Monitored ETY” in Unity Pro. Unity Pro 3.1 (and above) has a special option to select the Monitored ETY, and is therefore the minimum version of Unity that can be used to configure Hot Standby systems. The Unity Pro screen used to select the Monitored ETYs appears as follows:



- (1) You can find the option to select one pair of ETYs to act as the Monitored ETYs on the Hot Standby tab of the controller’s configuration. You select the Monitored ETYs using their topological address (their position on the rack).

**NOTE:** For more information concerning the role of the Monitored ETYs in providing a redundant Ethernet I/O capability, see Minimum Configuration for Redundant Ethernet I/O ([see page 66](#)).

**NOTE:** For more information on configuring your Premium Hot Standby controller in Unity Pro, see Configuring a System with the Unity Pro Tabs and Dialogs ([see page 126](#)). For more information on configuring your TSX ETY modules in Unity Pro, see Minimum Configuration for Redundant Ethernet I/O ([see page 146](#)).

## Two Control Connections: the Sync Links

The matching PLCs and Monitored ETYs rely on two control connections, known as “sync links”. The direct connection between the Hot Standby controllers is called the “CPU-sync link”. The connection between the Monitored ETYs is called the “ETY-sync link”. These sync links have the following properties:

### CPU-sync link

The CPU-sync link is the main communications channel for providing Premium Hot Standby redundancy. It is established between the Ethernet-based Hot Standby (labeled “HSBY”) ports on the face of each controller. Each Hot Standby controller’s CPU provides the data passed over the CPU-sync link, but the management of the actual transmission of this data is the responsibility of each port’s coprocessor (Copro). It is important that you establish this sync link using an uninterrupted cable connection, and that you do not use it for any other purpose.

## WARNING

### UNINTENDED EQUIPMENT OPERATION

- Make an uninterrupted point-to-point connection between the Hot Standby CPU-sync link ports.
- Do not connect any other Ethernet devices so that they share the same network cabling as the CPU-sync link.
- Do not exceed maximum Ethernet cable lengths for the type of cable selected.

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

### ETY-sync link

The ETY-sync link is a channel for the transmission and receipt of additional Hot Standby-related information, including diagnostic information. In addition, you may use this link to establish redundant Ethernet I/O (Monitored I/O), and as a provider of other Ethernet services such as HTTP, FTP, TFTP, and SNMP. You can connect the ETY-sync link directly using a crossover cable if you do not plan to use Monitored I/O, but this sync link can also be connected using standard cables through two (2) or more network switches.

Disconnecting a crossover cable configured with I/O Scanning causes both PLCs to enter the Offline mode.

## WARNING

### UNINTENDED EQUIPMENT OPERATION

Do not configure the I/O Scanner service when you make a point-to-point ETY-sync link connection with a crossover cable.

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

### Establishing and Protecting the Sync Links

It is very important that you establish these two sync links properly and protect them from damage while the system is in service. If these cables are not connected when the system is started, both Hot Standby PLCs will start as the Primary PLC and attempt to control the system, potentially resulting in conflicting commands to system devices.

## WARNING

### UNINTENDED EQUIPMENT OPERATION

- Always confirm that both the CPU-sync link and ETY-sync link are physically connected before applying power.
- If communications equipment such as network switches is part of the ETY-sync link, confirm that these devices are ON, initialized, and operating properly before applying power.
- Route and protect the CPU-sync link and ETY-sync link cables so that a single accident cannot disconnect both cables.

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

If there is IO configured in the supervised ETY, consider the following scenarios:

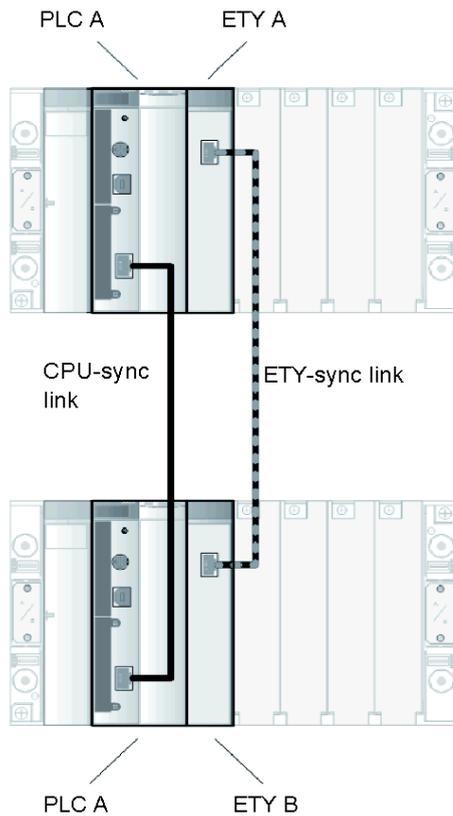
- If there is 1 or more I/O scanning lines configured in supervised ETY and there is no link, both CPUs will run offline.
- If there is no configured I/O scanning in the supervised ETY, both CPUs will run as Primary.

## Core Hot Standby Hardware

### Overview

Once you understand the requirement for two identical Premium Hot Standby PLCs, two identical TSX ETY 4103 / 5103 Ethernet modules, and the two sync links between them, you need only add two identical racks and power supplies to create the core Hot Standby system. This system is pictured below:

### Illustration



## Parts list

The parts list for this core system is as follows:

Name	Reference	Min. Vers.	No. Units
Premium Standard Racks	TSX RKY ***	—	2
Premium Power Supplies	TSX PSY ***	—	2
Premium Hot Standby Controller	TSX H57 24M or TSX H57 44M	—	2
Premium Hot Standby Ethernet Modules	TSX ETY 4103 or TSX ETY 5103	4.0	2
Premium Rack Line Terminators	TSX TLYEX	—	4
Premium Protective Covers	TSX RKA 02	—	as required
Schneider-Electric Cat 5e cables	EU versions: ● 490NTC00005 (5 m) ● 490NTC00015 (15 m) ● 490NTC00040 (40 m) ● 490NTC00080 (80 m)  LU versions: ● 490NTC00005U (5 m) ● 490NTC00015U (15 m) ● 490NTC00040U (40 m) ● 490NTC00080U (80 m)	—	2

**NOTE:** The above hardware is always required in Hot Standby systems, but it does not provide a useful redundant system, as it includes no redundantly managed I/O. For an introduction to different types of redundant systems (Ethernet I/O, Analog I/O, etc.), see *Hot Standby Systems*, [page 55](#).

## Configuration Requirements

### Identical Hardware and Software

In previous sections, we stated the requirement for identical controllers and Ethernet modules. In fact, the requirement for identical configurations extends to all equipment on both the Primary and Standby racks, and even to the application programs that you create. To create a working Hot Standby system, you must meet all of the following hardware/firmware requirements or your system will not be able to come online.

### Identical Hardware

- Identical Premium Hot Standby controllers, either two TSX H57 24M's or two TSX H57 44M's, with identical CPU and Copro firmware, identical memory cards and accessories, and occupying the same rack positions. (You can permit differing firmware versions on a temporary basis so that operational firmware upgrades ([see page 235](#)) are possible.
- Identical Premium Ethernet communication modules, either two TSX ETY 4103's or two TSX ETY 5103's, with identical firmware of version 4.0 or above, and occupying the same rack positions.
- Identical in-rack I/O, including identical firmware, hardware, revisions (if applicable), and rack positions. These conditions remain true whether the in-rack I/O is redundant or local.
- Identical module cartridges and accessories. For In-rack communication and I/O modules that accept such accessories, any cartridges used must be identical, and identically positioned and configured.
- Identical Premium TSX RKY ••• racks (backplanes). Each PLC must contain the same number of racks, using the appropriate line terminations. The rack IDs used must be the same on each PLC.
- Identical Premium TSX PSY ••• power supplies, occupying the same rack positions, and, ideally, supplied by different feeder circuits.
- Identical cabling and cabling systems, fully shielded, and compliant with the length requirements for the type of fieldbus you employ.

### Identical Software

In addition, the following software requirement applies:

- Identical application programs must be loaded on both Premium Hot Standby controllers ([see page 34](#)).

## In-Rack Redundancy

In this manual, we refer to two types of in-rack I/O modules: redundant and local. For a pair of In-rack I/Os modules to function redundantly as part of the Hot Standby system, and therefore be designated as redundant in-rack I/O, the following must be true:

- Each input and output module in one rack must be identical to an input or output module in the other rack.
- You must connect each match I/O pair to a single field device using one of the connection blocks for discrete I/O (*see page 57*) or signal duplicator for analog I/O (*see page 60*). (For example: an ABE7 connection block is used for discrete I/O.)
- You must configure these identical I/O modules for continued parallel operation through switchover events using the appropriate fallback values for the outputs. The appropriate fallback values are dependent on your application and the I/O type. In addition, for discrete outputs, the fallback values depend on whether positive or negative logic is used.

In-rack I/O modules that do not meet the redundancy requirements are referred to as local in-rack I/O (or just local I/O). Remember that:

- While only the Primary controller can affect redundant in-rack I/O modules, the local in-rack I/O may be operated on by either the Primary or Standby controllers.
- Before implementing local in-rack I/O, refer to In-rack I/O Management (*see page 180*).

## Supported Hardware

- Multiple racks can be used in Premium HotStandby systems with PLC version 2.83 or higher (version in synchronization with Unity Pro 6.0). Systems with earlier version do not support (*see page 94*) the use of extended racks systems.
- Some modules (*see page 78*) available for other Premium PLCs cannot be used in Hot Standby systems. In general, the modules that may not be used are the expert function modules such as counters, etc.

## Programming Platform

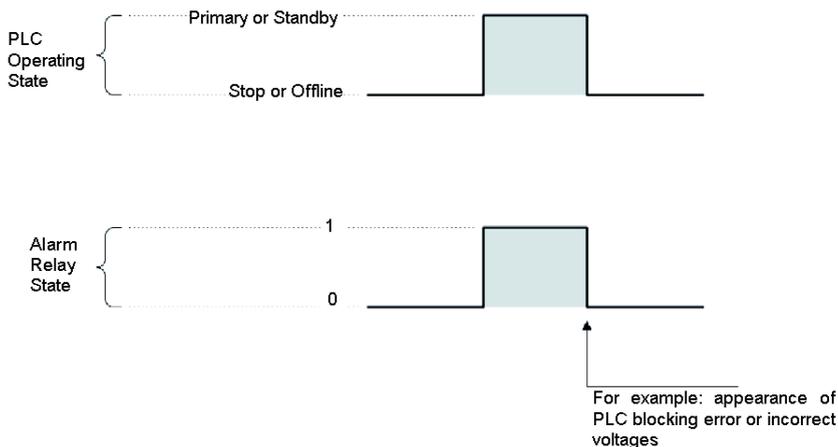
- Only Unity Pro version 3.1 or above can be used to configure Premium Hot Standby systems, and to manage application programs intended for use on Premium Hot Standby systems.
- You must configure at least one pair of ETY modules as the "Monitored ETY" in Unity Pro.

### Power Supply Alarm Relays

Many of the Premium TSX PSY ••• power supplies include alarm relays that provide a second method for determining the power supply status. The alarm relay for these power supplies is located on the terminal block as pictured here:



When your Hot Standby system is operating normally, with the PLCs in either the Run Primary or Run Standby operating modes ([see page 33](#)), the alarm relay is activated and its contact is closed (state 1). Whenever one of your PLCs enters the Stop or Offline operating modes, or when the power supply drops out or loses its supply power, the alarm relay falls back and its associated relay opens (state 0). This figure illustrates these behaviors:



The state of the alarm relay on each Premium main rack aligns to the Hot Standby controller's operating modes ([see page 33](#)) as follows:

Hot Standby PLCs Operating Mode	State of Alarm Relay on Associated Power Supply
Stop	Open
Offline	Open
Run Primary	Closed
Run Standby	Closed

In addition, the redundant power supplies in a Premium Hot Standby system offer three possible wiring designs for the alarm relays - the relays can be wired in series, in parallel, or independently. Therefore, in a Premium Hot Standby system, considering operating states and wiring, the number of possible configurations for the alarm relays is much higher than for a standalone Premium system. If you plan to use the power supply alarm relays, carefully consider which of these configurations is appropriate for your system in all operational states.

## Establishing Redundancy

### Database Exchange

he Premium Hot Standby provides redundancy by maintaining its Standby PLC and associated modules in a state where they can assume the Run Primary operating mode quickly. This means that the Standby PLC has all of the information necessary to mirror the I/O states present on the Primary PLC, and that this information is regularly updated. For the Premium Hot Standby, the collected information is called the database and the regular exchange of this database is referred to as the database transfer.

The database is created by the Primary controller's CPU just after this CPU has finished evaluating the input conditions (the %I and %MW values) and the Standby's Reverse Transfer System Status Words (%SW62 - %SW65). After the database creation is complete, the Primary PLC transfers the database to its Copro, which in turn transmits it over the CPU-sync link to the Standby controller's Copro. The Standby controller then applies the information in the database as required.

The database that is cyclically transferred from the Primary controller to the Standby controller (via the Copros and the CPU-sync link) includes both system data and user application data. In both cases, some of this data is located (addressable) data, and some is unlocated. The data transferred includes:

#### System Information:

- Located (a subset of the System Bits and Words)
  - Exchanged during every MAST Task:
    - System Bits: %S30, S31, %S38, %S50, %S59, %S94
    - System Words: %SW0, %SW1, %SW8, %SW9, %SW49...%SW53, %SW59, %SW60, %SW70, %SW108
  - Exchanged only during Switchover
    - %SD18 and %SD20
- Unlocated
  - A subset of the system data managed by the Primary PLC's operating system. This subset includes system counters used by function blocks such as TON, TOFF, and others.

#### User Application Data:

- Located
  - All %M, %MW, %MD, and %MF data from address 100 up to the maximum number of global address fields configured in Unity Pro's Configuration tab, but no more than 128 KB. The range below 100 (for example, %MW0 - %MW99) is not transferred.
  - The output (%Q) objects and any output forcing settings.
  - EDT / DDT when they are located by the user.
  - Sequential Function Chart (SFC) data types.
- Unlocated
  - EDT / DDT when they are located by the system.
  - Function Block (EFB / DFB) data types.

**NOTE:** In addition to the above, the Primary controller sends the values of all Forced Bits to the Standby as part of the regular database exchange.

**NOTE:** The maximum amount of located data that can be transferred in the database is 128 KB for both the TSX H57 24M and the TSX H57 44M. The maximum unlocated data is 120 KB for the TSX H57 24M and 300 KB for the TSX H57 44M.

**NOTE:** The maximum size of the entire database is approximately 165 KB for the TSX H57 24M and 405 KB for the TSX H57 44M.

**NOTE:** For specific information regarding the command words and adjustment parameters, and the maximum memory sizes of these areas, refer to the *Unity Pro Operating Modes manual*. For a detailed description of the System Bits and System Words that are exchanged, refer to the *Unity Pro System Bits and Words Reference Manual*.

For more information on the database transfer, including information regarding the application of this information by the Standby, refer to *Understanding the Premium Hot Standby Database Transfer Process*, [page 105](#).

### Synchronized Program Execution

By itself, the regular exchange of system and user application data is not enough to synchronize the Standby controller with the Primary controller. It is also important that the cyclical execution of tasks on each controller remains aligned, so that neither controller races ahead of the other if it is still processing its information. This means that the Primary controller will sometimes need to wait for the Standby to finish processing, and the Standby will sometimes wait for information from the Primary.

This requirement for aligned program execution requires in turn that the task execution cycle be deterministic in nature. For this reason, only MAST tasks are used when programming a Premium Hot Standby system. For more detail on the requirement for MAST tasks, and their execution in a Hot Standby context, see *General*, [page 34](#) and *Adjusting MAST Task Properties*, [page 190](#).

### Switchover Events

The term Switchover refers to the moment when system control is transferred from the Primary controller to the Standby controller. The Switchover event has a finite duration, and can be initiated manually (through Unity Pro or a physical PLC reset) or automatically by system conditions. The causes of Switchovers, and the behavior of a Premium Hot Standby system when a Switchover event occurs, is a complex topic covered throughout this manual, and extensively in *Detailed Behavior on Interruption of Power, Communications, or Device Capabilities*, [page 251](#).

While this manual covers Switchover events in some detail, a few general statements will aid in your understanding of these subsequent topics:

- Much of the benefit of the Premium Hot Standby system is its ability to detect various error conditions and, when warranted, initiate a Switchover. The type of error detected determines the duration of the Switchover event. For example:
  - If the Primary PLC remains online, and can communicate with the Standby PLC, but detects an error that requires a Switchover, it will command the initiation of a Switchover event. See *PLC Communications and Switchover*, [page 103](#), for further details.
  - If the Primary PLC is inoperative, or all communications between the Primary and Standby controllers are lost, an automatic Switchover occurs.
- The behavior of Redundant In-rack I/O during a Switchover event is straightforward. The requirement for identical hardware and for a recurring, synchronized database transfer contributes to this simplicity. There are, however, some points that are not immediately obvious:
  - The fallback settings of Redundant In-rack I/O become very important, and must be coordinated with the output type, output logic, and the expected application behavior.
  - There are important differences between the Switchover of Discrete and Analog I/O, See *Minimum Configurations by I/O Type*, [page 56](#) for further details.
- In the case of network-based I/O (Ethernet and Modbus), an additional layer of complexity is added during a Switchover event. This is because:
  - As part of the Switchover, the Hot Standby PLC reassigns the network address associated with the Primary PLC to the Standby PLC when a Switchover occurs. See *Swapping Network Addresses at Switchover*, [page 142](#) for further details.
  - In addition, other Ethernet services (HTTP, FTP) may be running at the time of a Switchover, and the PLC will close and reopen these services on the ETYs of the new Primary PLC during the Switchover event. See *Network Effects of Premium Hot Standby*, [page 155](#) for further details.
- Local I/O is not part of any automatic Switchover. You should manage your Local I/O in Section 0 of your application program if you would like it to continue to operate after a Switchover.

## Revised Operation Modes

In a normally operating Premium Hot Standby system, there are two PLCs running at any given time, one as the Primary PLC and one as the Standby PLC. Consequently, a Premium Hot Standby system requires additional, revised operating modes or states to reflect the system status. Further, the redundant nature of the system means that the relationships between operating modes will change. The following provides a quick summary of the Premium Hot Standby operating modes:

- Stop: The PLC has received a Stop command and has successfully stopped.
- Run Primary: The PLC has received a Run command and has assumed the Primary role. It did not detect another Hot Standby PLC acting as the Primary, or, if both PLCs were started simultaneously, it had the lower MAC address.
- Run Standby: The PLC has received a Run command and has assumed the Standby role. Either this PLC detected another Hot Standby controller already operating as the Primary, or, if both PLCs were started simultaneously, it had the higher MAC address.
- The PLC in question has received a Stop command or has responded to a detected error and has left one of the Run (Primary or Standby) operating modes.

### **WARNING**

#### **UNINTENDED EQUIPMENT OPERATION**

- Never assume that a PLC is in a certain operating mode before installing, operating, modifying, or servicing it.
- Before acting on a PLC, always positively confirm the operating mode of both Hot Standby PLCs by viewing their LEDs and checking their System Status Words.

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

**NOTE:** In certain circumstances, such as when no valid application is loaded on a PLC, a Premium Hot Standby controller will enter and report itself as being in a "Non-Conf" or non-configured state. This state is not considered an operating mode.

A more in-depth description of the Premium Hot Standby operating modes, including a state / state transitions diagram, can be found in section *Conditions for Switchover*, [page 207](#).

## Programming Differences

### General

In general, programming a Premium Hot Standby controller with Unity Pro is very similar to programming any other standalone Premium controller using Unity Pro. Unity Pro provides a user-friendly, IEC 61131-3 compatible development environment, and most of your programming skills in other development environments and for other devices will be applicable for the Premium Hot Standby.

However, there are some important considerations:

- Only Unity Pro version 3.1 or above can be used to configure Premium Hot Standby systems, and to manage application programs intended for use on Premium Hot Standby systems.
- The application programs on both PLCs must be identical, or the PLC will report a "logic mismatch".
  - If the Hot Standby PLCs are operational at the time a logic mismatch occurs, the Standby controller will enter the Offline operating state.
  - If a logic mismatch exists during a simultaneous startup of both the Hot Standby PLCs, one PLC will start as the Primary, and the other PLC will remain in the Offline operating mode.
  - If the controllers are started sequentially and a logic mismatch exists, the second PLC that attempts to start will start in the Offline state.
  - When the Hot Standby controllers test for a logic mismatch, they normally check three conditions:
    - Whether the application program you have loaded on both PLCs is the same.
    - Whether the Unity Pro animation tables for both PLCs are the same.
    - Whether the Unity Pro comments for both PLCs are the same.
  - If the application programs on each PLC are different, this will result in a logic mismatch.
  - By default, if the animation tables and comments on the PLCs differ, a logic mismatch will occur. However, you can override this behavior; see Understanding Premium Hot Standby Logic Mismatch ([see page 228](#)) for details.
  - Some changes to your application programs are possible while online; other changes require an offline update. See Online/Offline Modifications to an Application Program ([see page 229](#)) for more information.
- When connecting Unity Pro to a Hot Standby system, keep in mind that:
  - Generally, the information you can see in Unity Pro will be the same whether you connect to the Primary PLC or to the Standby PLC. Most registers on the Standby PLC will reflect the values provided by the Primary PLC during each MAST task.
  - However, some differences between the data on the Primary PLC and the Standby PLC do exist. These exceptions include the located System Words and User Application data maintained independently on each PLC (%SW61, %MW0 - %MW99).
  - If you attempt to write values to the Standby PLC's registers, this will usually be ineffective. The next database transfer from the Primary PLC will usually overwrite any values you commanded.

- Users who have programmed PL7 Warm Standby systems or other Hot Standby systems will notice that many of the events that had to be managed in the application program are now automatic. However, your present system requirements may require the programming of redundant operations. For example, if you want a switchover to be triggered by the detection of an error in an ETY module that is not configured as the Monitored ETY, you have to manage this in your application.

### Application Task Types

In a Premium Hot Standby system, the Standby controller must remain ready to assume the role of the Primary controller. This requires that both controllers run identical applications, and that the Standby controller is provided with current application data and state information from the Primary controller once per scan. The synchronous and deterministic transfer of the Primary controller data and state information to the Standby controller is achieved by using MAST tasks.

### Exclusive Use of MAST Tasks

MAST tasks should be used exclusively in Premium Hot Standby systems because the transfer of Primary's system and user application data to the Standby controller is synchronized with this task. Preemptive, asynchronous, or interrupt-driven tasks and programming methods, including the use of FAST tasks, events, and edge triggers, etc., should not be used. They can impact the performance of the MAST tasks and cause discrepancies between Primary and Standby output values in the event of a Switchover.

Only MAST tasks support data synchronization between the Primary and Standby Controllers.

## WARNING

### UNINTENDED EQUIPMENT OPERATION

Do not use asynchronous, preemptive, or interrupt-driven tasks to program the outputs of your Premium Hot Standby System.

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

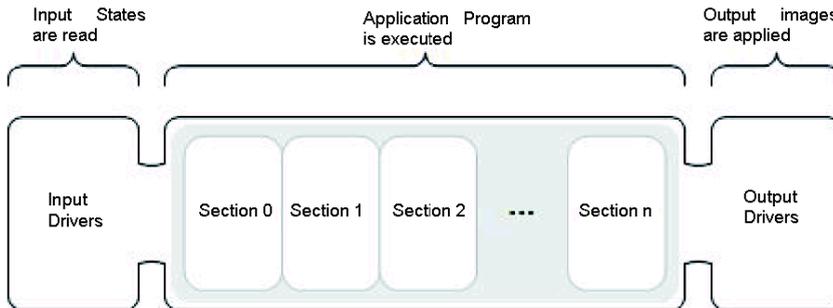
### How Hot Standby MASK Tasks Differ

Hot Standby MAST tasks are different from the normal MAST tasks you are familiar with from your experience programming other Premium PLCs. In a Premium Hot Standby PLC, the execution of a MAST task involves extra steps necessary to support redundancy. These additional steps provide the following:

- Database creation (see Database Exchange ([see page 30](#))).
- Database transmission.
- Wait states to synchronize MAST task execution (see Synchronized Program Execution ([see page 31](#))).
- Application of output images (as required).

### A Typical Premium MAST Task

The following illustrations provide a comparison between a normal MAST task and a Hot Standby MAST task. The normal MAST task appears here:

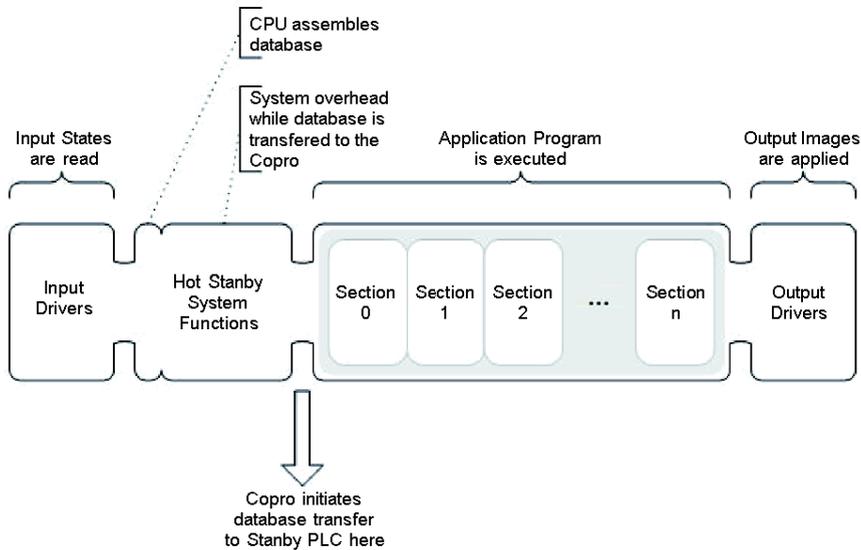


### The Hot Standby MAST Task

The Hot Standby version of the MAST task introduces an additional step for "Hot Standby System Functions", which include:

- The assembly of the database by the CPU.
- The transmission of the database from the CPU to the Copro.

The Hot Standby version of the MAST task appears below:



Typically, the durations of the Input Driver, Application Program, and Output Driver stages are similar to those found in standalone Premium PLCs.

The time required for the CPU to assemble the database is normally negligible. However, the time required to transfer the database to the Copro, and for the Copro to communicate this information to the Standby, scales linearly with the size of the database. For more information on Hot Standby MAST tasks actions and durations, refer to Database Transfer Between Hot Standby PLCs ([see page 104](#)) and Adjusting MAST Task Properties ([see page 190](#)).

## Hot Standby Restricted Functions

### At a Glance

For Premium Hot Standby applications, some of the programming functionality you may have used in the past does not apply to redundant operations. This section summarizes these restrictions.

### PL7 Warm Standby Functions

The following legacy function blocks support Premium PL7 Warm Standby behavior, and are now inconsistent with Hot Standby redundancy. Use of these function blocks can lead to unintended changes in output states, either immediately or at Switchover. These function blocks must not be used:

- PL7\_COUNTER
- PL7\_DRUM
- PL7\_MONOSTABLE
- PL7\_REGISTER\_32
- PL7\_REGISTER\_255
- PL7\_TOF
- PL7\_TON
- PL7\_TP
- PL7\_3\_TIMER

## WARNING

### UNINTENDED EQUIPMENT OPERATION

Do not use the PL7 Warm Standby function blocks listed above in a Premium Hot Standby system.

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

### Data Exchange DFBs

The following Derived Function Blocks (DFBs) were specifically used for data exchange, and are now inconsistent with Hot Standby redundancy. These function blocks were not ported to Unity Pro, and are therefore not available for use:

- Ha\_db\_basic
- Ha\_db\_cycle\_opt
- Ha\_db\_size\_opt

## Expert Functions for SFC / Grafcet Programming

The following Expert Functions (EFs) previously used to provide data and context exchanges in Sequential Function Chart (SFC) / Grafcet programming are now inconsistent with Hot Standby redundancy. These function blocks were not ported to Unity Pro, and are therefore not available for use:

- Get\_stat\_chart
- Set\_stat\_chart

**NOTE:** For more information on SFC and Grafcet programming, refer to the *Startup Guide for Unity Pro*, reference 35008402, and to the *Unity Pro PL7 Application Converter User Manual*, reference 35006148.

## Runtime Modification of Expert Function Parameters

You are no longer permitted to modify Expert Function (EF) parameters at runtime using your application or the Unity Pro debug screen. Expert Functions (typically used for process control) write their parameter values in memory ranges that are not part of the Hot Standby database transfer. Therefore, if EF parameters are modified when the system is operational, this could result in different operating states between the Primary and the Standby PLCs when a Switchover occurs.

### **WARNING**

#### **UNINTENDED EQUIPMENT OPERATION**

- Do not program your application so that it changes Expert Function parameters unless you also program your application to transfer these changes to the Standby PLC during each MAST task.
- Do not manually modify Expert Function parameters using the Unity Pro debug screen while the system is operational.

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

### SAVE\_PARAM Function

The use of the SAVE\_PARAM function is not permitted in a Hot Standby application. This function overwrites the initial value of a module parameter that is stored in the program code area. This area is not transferred from the Primary to the Standby in the database.

## WARNING

### UNINTENDED EQUIPMENT OPERATION

Do not use the SAVE\_PARAM function in a Premium Hot Standby system.

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

### T\_COM\_MB Derived Data Type

The T\_COM\_MB IODDT (I/O Derived Data Type) may cause unpredictable behavior when used to query the high byte of the communications PROTOCOL variable. Only the low byte should be queried using this function.

## WARNING

### UNINTENDED EQUIPMENT OPERATION

When using the T\_COM\_MB IODDT function to determine the Modbus protocol in use, do not query the high byte of the PROTOCOL variable.

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

### Changing Declared Variables

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

## WARNING

### UNINTENDED EQUIPMENT OPERATION

Do not change the initial values of declared variables using the System Bit %S94.

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

## Internal Control Loops

To avoid risks of output bumps when switching, do not use the internal control loops. Instead, use EFBs from control library.

### **WARNING**

#### **UNINTENDED EQUIPMENT OPERATION**

Do not use the internal control loops.

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

## Section 0 Restrictions

The following restrictions apply only to programming the first section (Section 0) of your application:

- Derived Function Blocks (DFB) may not be used in Section 0.

### **WARNING**

#### **UNINTENDED EQUIPMENT OPERATION**

Do not use TON, TOFF, and TP function blocks in Section 0 of your application program.

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

## Asynchronous Communication Function Blocks

During a Switchover event, asynchronous communication function blocks (for example, WRITE\_VAR) will not automatically resume operation on the new Primary PLC without special care.

### **WARNING**

#### **UNINTENDED EQUIPMENT OPERATION**

Follow the suggested procedure below when using asynchronous communication function blocks.

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

The following procedure should be used to allow asynchronous communication function blocks to automatically resume operation after a Switchover:

- Program your application so that it stores the values of all function block management parameters in the Non-Transfer Memory Area (%MW0...%MW99).
- Initialize the Length parameter each time the function block is called.
- Use a separate Timer function block as a replacement for the communication function block's Timeout parameter.

**NOTE:** If for some reason you are unable to follow this procedure, and a Switchover renders your communication function block inoperative, write your application program so that it sets the function block's activity bit to 0 before restarting the function block in the new Primary CPU.

### 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. For example, the explicit instructions WRITE\_CMD and WRITE\_PARAM are both capable of writing non-transferable values and have to be used carefully. Consider the following example:

If the WRITE\_CMD is related to a "Modbus change to character mode" command in a TSX SCP 114 module, this change will only be done in the Primary PLC. If a Switchover occurs, the new Primary will restart with the Modbus mode rather than the Character mode.

### 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 were a standalone application. This allows you to use all of the powerful debugging features available in Unity Pro, such as watchpoints, etc.
- 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 Unity Pro's debug features can be used during this stage.

**NOTE:** See Debugging Your Hot Standby Application ([see page 186](#)) for further details on debugging your Hot Standby application program.

### Primary vs. Standby Execution

In a Premium Hot Standby system, your application is executed differently depending on whether it is running on the Primary PLC or on the Standby PLC. The main difference is that the full application program is executed on the Primary controller, while the Standby only runs the first section (also known as "Section 0").

This is important because some system behaviors must be commanded in Section 0. Examples include:

- Local I/O, if it is meant to be run from the Standby PLC. This includes switching between Redundant In-rack Analog I/O signals; see Minimum Configuration for Redundant Analog I/O (Outputs Only ([see page 62](#)) for further details.
- Population of the Standby PLC's Reverse Transfer Registers (%SW62 - %SW65) with custom diagnostic information for use by the full application program on the Primary PLC.

Other system behaviors must not be commanded in Section 0. For example:

- You should not change the values of redundantly controlled discrete outputs in Section 0. The Standby PLC executes the first section (Section 0) of your application program, and then later applies the %Q / %QW images received from the Primary PLC. If you alter discrete output bits in Section 0, the commanded output values for the Standby PLC's redundant In-rack outputs might be changed twice in a single MAST task, and the resulting physical state might be inconsistent with that directed by the Primary PLC.

## WARNING

### UNINTENDED EQUIPMENT OPERATION

Do not change discrete output bit values for redundant outputs in the first section (section 0) of your application program.

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



---

# Chapter 2

## Hot Standby Overview

---

### Purpose of this Chapter

In this chapter, you will find a brief overview of the Premium Hot Standby controller. This chapter begins by describing the physical and display characteristics of the controller, specifically those that make the TSX H57 ••• distinct from other Premium PLCs including the TSX P57 •••. The chapter concludes with information regarding the operating limits and certifications and standards of the Premium Hot Standby controller.

### What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Introduction to the Controller	46
Operating Limits	52
Certifications and Standards	53

## Introduction to the Controller

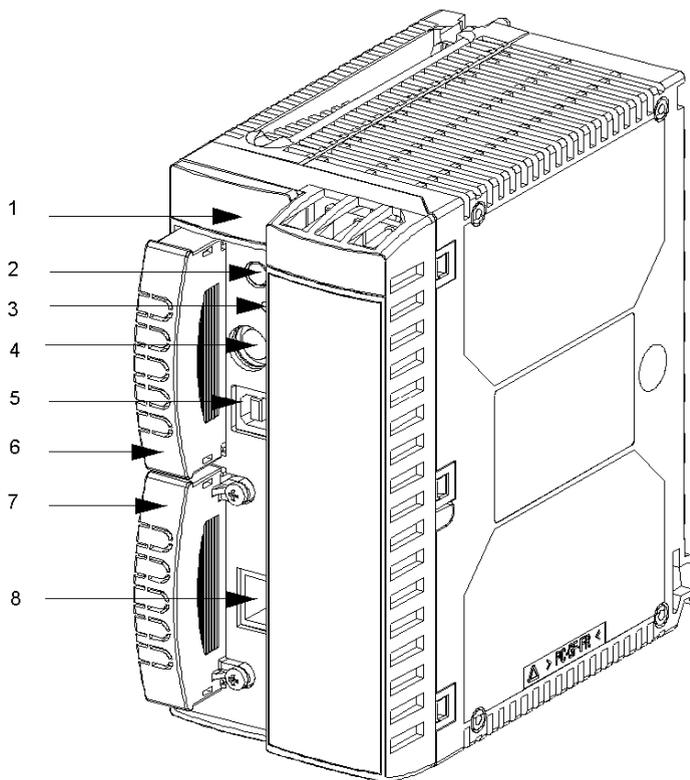
### Overview

The TSX H57 ... Hot Standby controllers are very similar to the Premium TSX P57 ... controllers. The major changes are firmware-related, primarily affecting the operational behavior of the controller. However, the changes in the operational behavior also dictate that the physical indicators, controls, and terminal ports on the front face of the device perform differently.

For example:

- The Display Block LEDs behave differently and have different meanings.
- The Ethernet port on the front-face of the device is dedicated to the CPU-sync link.
- Operating the Cold Start Reset Button has new consequences.
- Removing or inserting PCMCIA cards in an operational system will have new consequences.

### Illustration



- 1 Display Block (status LEDs)
- 2 DOS File Memory Extract Button (not used)

- 3 Cold Start Reset Button
- 4 Uni-Telway Terminal Port (programming connection, HMI)
- 5 USB Terminal Port (programming connection)
- 6 PCMCIA Slot for Application Memory Card Extension (Slot A)
- 7 PCMCIA Slot for Data Storage Card (Slot B)
- 8 Dedicated Port for CPU-sync link connection

**NOTE:** Pressing the Cold Start reset button will cause the affected PLC to reboot using default values instead of cached system and application data. If the affected PLC is the Primary, a Switchover will occur. If the affected PLC is the Standby, it will return to the Standby role after re-initializing.

**NOTE:** Attempting to remove or insert a PCMCIA card while your Hot Standby system is operational will cause the affected PLC to restart. If the affected PLC is the Primary, a Switchover will occur. If the affected PLC is the Standby, it will return to the Standby role after re-initializing.

**NOTE:** Ensure that the PCMCIA card storing your application program is loaded into Slot A on both PLCs. If the card containing your application program is inserted into Slot B on either or both PLCs, the system will not start.

### Uni-Telway Port

The Uni-Telway port on the face of the Premium Hot Standby PLC can be used for Unity Pro and HMI / SCADA connections. However, the Uni Telway port is not managed redundantly by the Premium Hot Standby system. It will remain operable as long as the PLC is operable, but its address and status do not change during or after a Switchover event.

Therefore, the following points must be understood:

- In master mode (default), the Uni-Telway port provides a point-to-point connection with the Unity Pro workstation or HMI terminal. This physical connection is normally established between the Unity Pro or HMI station and the designated Primary controller. If a Switchover occurs, the Unity Pro or HMI station will now be connected to either the new Standby PLC or to an Offline PLC. In such circumstances, either the connection will be nonfunctional or it may not be immediately apparent that a Switchover has occurred because the controllers run identical programs and have similar values in memory. In this second case, attempts to control the Hot Standby system through the Unity Pro or HMI station will not work as expected.
- Even when a Hot Standby controller is connected as a Uni-Telway slave on a larger Uni-Telway network, no redundant management of the port is possible. The Uni-Telway port's assigned slave address will not be automatically swapped during a Switchover. The communications master, whether a Unity Pro workstation, HMI terminal, or other device, will continue to address the affected PLC at the old address as if a Switchover had not occurred. Therefore, if you plan to use the Uni-Telway port for operational purposes, ensure that your system will respond appropriately if a Switchover occurs.

## USB Port

The USB port on the face of the PLC can only be used for a point-to-point slave connection with a Unity Pro workstation. As with a point-to-point Uni-Telway connection, there is no redundant management of the USB port. Like the Uni-Telway port, if a Switchover occurs, Unity Pro will now be connected either to the new Standby PLC or to an Offline PLC. Again, it may not be immediately apparent that a Switchover has occurred.

### **WARNING**

#### **UNINTENDED EQUIPMENT OPERATION**

Do not use a connection to the Uni-Telway or USB ports as your primary means of controlling a Premium Hot Standby system.

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

### **WARNING**

#### **UNINTENDED EQUIPMENT OPERATION**

- Verify that a PLC is in the appropriate operating mode before installing, operating, modifying, or servicing it.
- Before acting on a PLC, always positively confirm the operating mode of both Hot Standby PLCs by viewing their LEDs and checking their System Status Words.

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

## Cold Start Button Differences

Pressing the Cold Start reset button will cause the affected PLC to reboot using default values instead of cached system and application data. If the affected PLC is the Primary, a Switchover will occur. If the affected PLC is the Standby, it will return to the Standby role after re-initializing.

## PCMCIA Differences

Attempting to remove or insert a PCMCIA card while your Hot Standby system is operational will cause the affected PLC to restart. If the affected PLC is the Primary, a Switchover will occur. If the affected PLC is the Standby, it will return to the Standby role after re-initializing.

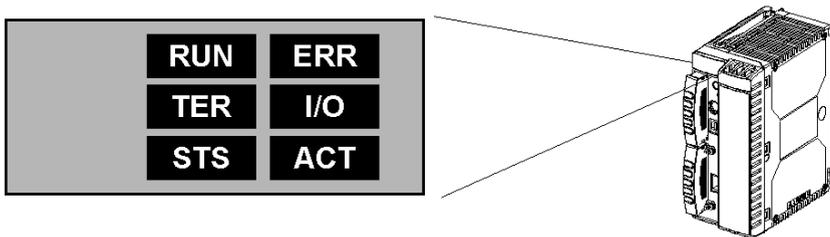
Ensure that the PCMCIA card storing your application program is loaded into Slot A on both PLCs. If the card containing your application program is inserted into Slot B on either or both PLCs, the system will not start.

**NOTE:** For general information regarding the use of PCMCIA cards in Premium PLCs, refer to *Premium and Atrium using Unity Pro Processors, racks, and power supply modules Implementation manual*, reference 35010524, in the chapters *Installation* and *Diagnostics*.

### Hot Standby Display Block

The Display Block LEDs on a Premium Hot Standby controller are similar in appearance to those on other Premium PLCs, but they do not have identical meanings. Please familiarize yourself with the following figure and table to understand the differences.

### Location and Appearance



### LED States and Meaning

LED	Meaning	Steady ON	Flashing	Steady OFF
RUN (green)	Displays the Hot Standby operating mode	PLC running in Primary Mode, executing the full application program	<ul style="list-style-type: none"> <li>2.5 s ON, 500 ms OFF: PLC running in Standby mode, executing only the first section (section 0) of the application program.</li> <li>500 ms ON, 2.5 s OFF: PLC running in Offline mode, no application program execution.</li> <li>500 ms ON, 500 ms OFF: PLC is in Stop mode or has detected a blocking software error.</li> </ul>	<ul style="list-style-type: none"> <li>PLC has not been configured</li> <li>Application program missing or invalid</li> </ul>
ERR (red)	Reports errors not related to I/O modules	PLC is not performing according to specifications or has become inoperative	<ul style="list-style-type: none"> <li>PLC not configured.</li> <li>Application program missing or invalid.</li> <li>PLC has detected a blocking software error.</li> <li>A memory card battery error has been detected.</li> <li>X-Bus error detected (See Note 1).</li> </ul>	Normal state

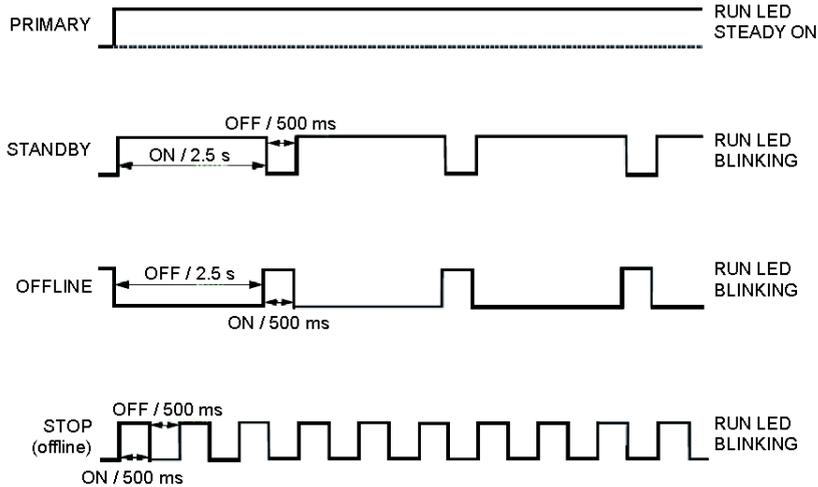
LED	Meaning	Steady ON	Flashing	Steady OFF
I/O (red)	Reports errors related to I/O modules	<ul style="list-style-type: none"> <li>In-rack I/O is not configured or operating properly</li> <li>An Ethernet device monitored by the I/O Scanning utility notifies the PLC that it has become inoperative</li> </ul>	X-Bus error detected (See Note 1)	Normal state
TER (yellow)	Reports activity on the Uni-Telway terminal port	—	Terminal port link active. The rate of flashing is relative to the amount of traffic.	Link not active
STS (yellow)	Displays the status of the CPU-sync link Coprocessor	The system is not redundant. This state usually occurs when the Coprocessor is booting, but should cease when the Coprocessor self-tests conclude.	Normal state. Data is being cyclically exchanged between the Primary and Standby Controllers.	The Coprocessor did not meet its self-test specifications or has become inoperative
ACT (yellow)	Reports activity on the CPU-sync link	(See Note 2)	CPU-sync link active. This is the normal state when system is operational. The rate of flashing is relative to the amount of traffic. (See Note 2)	Link not active

**NOTE: 1** - When an X-Bus error is detected, it is signaled by simultaneous flashing of the ERR and I/O LEDs.

**NOTE: 2** - The ACT LED indicates the communication activity between the Primary and Standby PLCs. Because this LED will illuminate during each database exchange (once per MAST task), this light may appear to be continuously illuminated.

### Depiction of Run LED States

The Run LED can flash three different patterns to help you distinguish between the Premium Hot Standby controller's operating modes. Those patterns are noted in the table above. They are repeated below in pictorial format for your convenience.



**NOTE:** For more information regarding the use of LEDs in Hot Standby and non-Hot Standby Premium controllers, refer to *Premium and Atrium using Unity Pro Processors, racks and power supply modules Implementation manual*, reference 35010524, in the chapter *TSX P57 / TSX H57 processors diagnostic*.

## Operating Limits

### Environmental

The environmental validation and certification of the Premium Hot Standby controller was identical to that performed for the TSX P57 \*\*\* PLCs. The environmental qualifications, standards, and limits for the Premium Hot Standby can be found in the *Premium and Atrium using Unity Pro Processors, racks and power supply modules Implementation manual*, reference number 35010524.

### Mechanical

The mechanical validation and certification of the Premium Hot Standby controller was identical to that performed for the TSX P57 \*\*\* PLCs. The mechanical qualifications, standards, and limits for the Premium Hot Standby can be found in the *Premium and Atrium using Unity Pro Processors, racks and power supply modules Implementation manual*, reference number 35010524.

### Electrical

The electrical validation and certification of the Premium Hot Standby controller was identical to that performed for the TSX P57 \*\*\* PLCs. The electrical qualifications, standards, and limits for the Premium Hot Standby can be found in the *Premium and Atrium using Unity Pro Processors, racks and power supply modules Implementation manual*, reference number 35010524.

### EMC

The electromagnetic compatibility and emissions validation of the Premium Hot Standby controller was identical to that performed for the TSX P57 \*\*\* PLCs. The electromagnetic compatibility and emissions qualifications, standards, and limits for the Premium Hot Standby are in the *Premium and Atrium using Unity Pro Processors, racks and power supply modules Implementation manual*, reference 35010524.

### Power Supply

In addition to providing the qualifications, standards, and limits for the Premium Hot Standby controller, the *Premium and Atrium using Unity Pro Processors, racks and power supply modules Implementation manual*, reference 35010524, provides electrical validation and certification information for the TSX PSY \*\*\* power supplies.

---

## Certifications and Standards

### Agency Certifications

Schneider Electric submitted this product for independent testing and qualification by third-party listing agencies. These agencies have certified this product as meeting the following standards.

#### North America

- UL508, Industrial Control Equipment
- CSA Hazardous Locations (Zones C11 Div2 C22.2, No. 213, Non-Incendive Electrical Equipment for Use in Class I, Division 2 Hazardous Locations)
- Canadian Standards Association, Specification C22.2, No. 142, Process Control Equipment

**NOTE:** Agency schedules for certifying and listing products are subject to change. For current information on third-party product certifications, please consult our website [www.telemecanique.com](http://www.telemecanique.com).

### Compliance Standards

Schneider Electric tested this product for compliance with the following compulsory standards.

#### North America

- Federal Communications Commission, FCC Part 15

#### Europe

- CE / IEC
  - Programmable Controllers: IEC 61131-2
  - EMI: EN55011 (Group 1, Class A)
  - EMS: EN 61000-6-2
- CE / European Directives
  - Low Voltage: N° 2006/95/EC
  - Electromagnetic Compatibility: N° 2004/108/EC

#### Maritime

- Bureau Veritas (BV)
- Det Norske Veritas (DNV)
- Lloyd's Register of Shipping (LR)
- Germanischer Lloyd (GL)
- Russian Maritime Register of Shipping (RMRS)
- Royal Institution of Naval Architects (RINA)
- American Bureau of Shipping (ABS)

### Voluntary Standards

Schneider Electric voluntarily tested this product to additional standards. The additional tests performed, and the standards under which the tests were conducted, are specifically identified in the Chapter *Operating Standards and Conditions* in the *Premium and Atrium using Unity Pro Processors, racks and power supply modules Implementation manual*, reference 35010524.

### **CE Compliance Note**

The products described in this manual comply with European Directives concerning Electromagnetic Compatibility and Low Voltage (CE marking) when used as specified in the relevant documentation, in applications for which they are specifically intended, and in connection with approved third-party products.

---

# Chapter 3

## Hot Standby Systems

---

### Purpose of this Chapter

This chapter provides an overview of various I/O, communications, and HMI / SCADA implementations in a Premium Hot Standby system. For simplicity's sake, and to aid your understanding, the minimum configuration for each type of I/O is presented independent of all other I/O. Each of these examples states the distinguishing features and required equipment for that minimum configuration.

After the presentation of the minimum configurations, a listing of all modules that can be used in a Premium Hot Standby system (both In-rack and distributed) is provided.

### What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
3.1	Minimum Configurations by I/O Type	56
3.2	Compatible Equipment	78
3.3	Example Hot Standby Systems	97

---

## Section 3.1

### Minimum Configurations by I/O Type

---

#### Purpose of this Section

This section describes the minimum configurations for all I/O types supported by the Premium Hot Standby.

#### What Is in This Section?

This section contains the following topics:

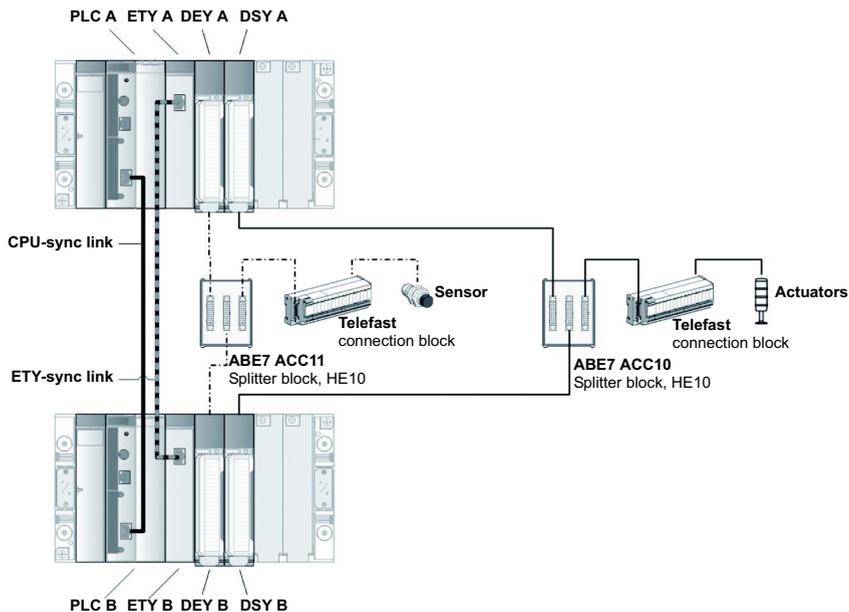
Topic	Page
Minimum Configuration for Redundant Discrete I/O	57
Minimum Configuration for Redundant Analog I/O (Inputs Only)	60
Minimum Configuration for Redundant Analog I/O (Outputs Only)	62
Minimum Configuration for Ethernet I/O	66
Minimum Configuration for Redundant Modbus I/O	71
Adding HMI / SCADA to the ETY-sync link	76

## Minimum Configuration for Redundant Discrete I/O

The minimum configuration necessary to support Redundant Discrete I/O appears below. The distinguishing features of this configuration are:

- The discrete input signals and output values are implemented using ABE7 connection blocks (using HE10 connectors and cables).
- These signals and values are in turn multiplexed / de-multiplexed using a Telefast connection block.
- Errors detected on Discrete I/O cannot cause an automatic Switchover event.
- The key design considerations are whether to use positive or negative logic at the outputs, the configuration of fallback modes based on this decision, and the minimization of output chatter (see next page).

### Illustration



## Parts List

Name	Reference	Min. Vers.
All equipment in the <i>Core Hot Standby Hardware</i> , <a href="#">page 24</a> plus...		
Premium Discrete Input Modules (HE10 connectors)	TSX DEY *** K	—
Premium Discrete Output Modules (HE10 connectors)	TSX DSY *** K	—
ABE7 Input Connection Blocks, 16 Channels	ABE7 ACC11	—
ABE7 Output Connection Blocks, 16 Channels	ABE7 ACC10	—
Telefast Connection Blocks	Various	—
HE10 cables, pre-assembled	TSX CDP **3	—
HE10 cables, pre-assembled or self-assembled	TSX CDP **3 (preferred) or ABF H20H008	—

## Discrete Outputs and Section 0 Programming

Because the Standby PLC executes the first section (section 0) of your application program, and then later applies the %Q object image received from the Primary PLC, avoid changing the state of redundant outputs in section 0. If you alter output bits in section 0, the output image for the Standby PLC's in-rack output modules might be changed twice in a single MAST task, and the resulting physical state might be inconsistent with that directed by the Primary PLC.

### **WARNING**

#### **UNINTENDED EQUIPMENT OPERATION**

Do not change discrete output bit values for redundant outputs in the first section (section 0) of your application program.

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

## Screw Terminals / Negative Logic

While positive-logic connections using the ABE7 ACC10 and ABE7 ACC11 are the preferred solution, it is possible to use other input / output modules with screw terminals and / or negative logic. In these cases, it will no longer be possible to use the ABE7 ACC10 and ABE7 ACC11 connection blocks. If you decide to use connection blocks with screw terminals and / or negative logic, observe the fallback instructions below and, if necessary, protect the channels according to the instructions in the related user manuals.

---

## Discrete I/O Fallback Modes

In addition to the guidance in these manuals, please be aware that proper configuration of I/O fallback modes is essential in providing the redundancy of a Hot Standby system. In general, outputs should be configured to fallback to their present state to prevent unintended equipment operation in the short period after the Primary has become inoperative and before the Standby has assumed the Primary role.

More specifically, in the case of discrete outputs, improper configuration can result in these outputs becoming locked in the state they last held when one of the PLCs becomes inoperative. To prevent discrete outputs from freezing when one of the controllers enters an inoperative state, the output modules using positive logic should use fallback mode 0 while those using negative logic should use fallback mode 1.

### **WARNING**

#### **UNINTENDED EQUIPMENT OPERATION**

- Configure your output module fallback modes to prevent changes in output states during Switchover.
- Use fallback mode 0 for all positive logic discrete output modules.
- Use fallback mode 0 when output modules are cabled in parallel using ABE7 ACC1• connection blocks.
- Use fallback mode 1 for all negative logic discrete output modules.

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

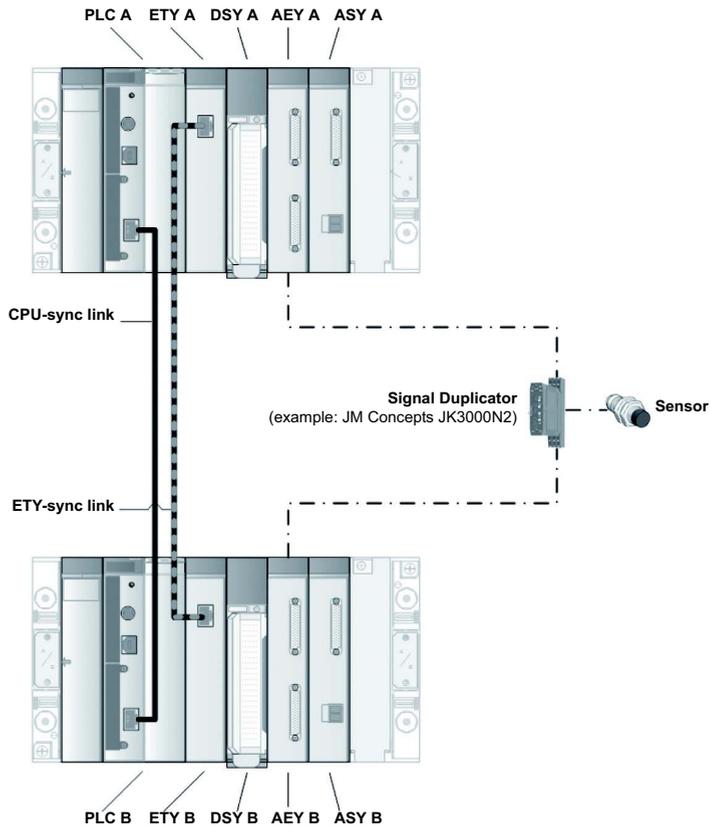
**NOTE:** For additional important details on the behavior of Discrete I/O and fallback modes during Switchover, see *Switchover When Primary Becomes Inoperative*, [page 209](#).

## Minimum Configuration for Redundant Analog I/O (Inputs Only)

The minimum configuration necessary to support Redundant Analog Inputs appears below. The distinguishing features of this configuration are:

- The use of a signal duplicator to ensure that the TSX AEY \*\*\* modules on the Primary side and on the Standby side both receive the correct input signal.
- Errors detected on Analog I/O cannot cause an automatic Switchover event.

### Illustration



---

## Parts List

Name	Reference	Min. Vers.
All equipment in the <i>Core Hot Standby Hardware</i> , <a href="#">page 24</a> plus...		
Premium Analog Input Modules	TSX AEY ...	—
Premium Analog Output Modules	TSX ASY ...	—
Analog Signal Duplicator	For example: JM Concepts JK3000N2	—

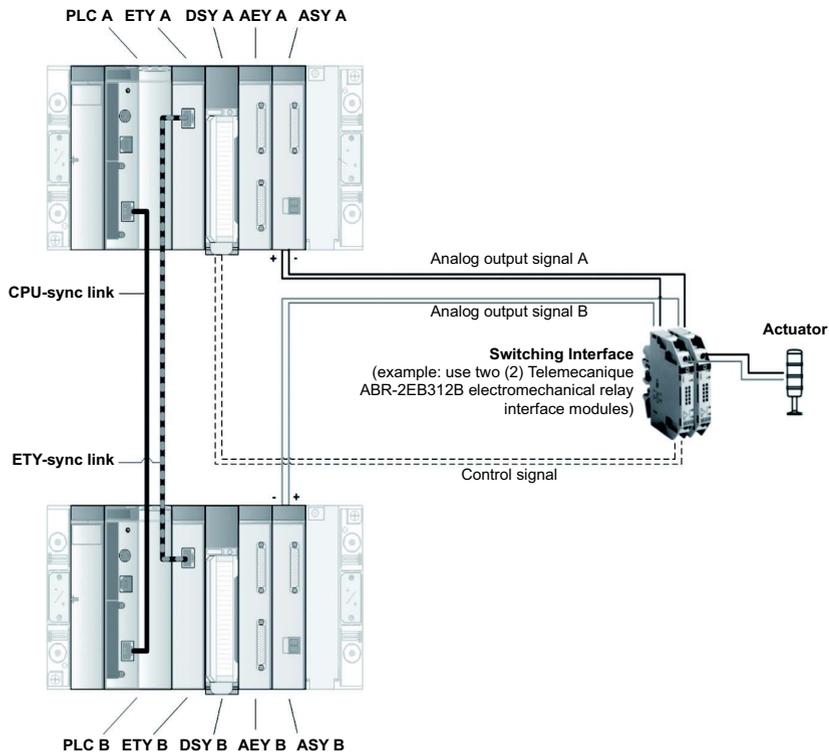
## Minimum Configuration for Redundant Analog I/O (Outputs Only)

### Introduction

The minimum configuration necessary to support Redundant Analog Outputs appears below. The distinguishing features of this configuration are:

- The use of a switching interface so that the TSX ASY ... modules do not simultaneously send output signals that lead or lag one another.
- Errors detected on Analog I/O cannot cause an automatic Switchover event.
- The key design consideration is the appropriate use of Section 0 to accomplish output signal switching.

### Illustration



## Parts List

Name	Reference	Minimum Version
All equipment in the core system ( <i>see page 24</i> ) plus...		
Premium Discrete Output Modules	TSX DSY ...	—
Premium Analog Input Modules	TSX AEY ...	—
Premium Analog Output Modules	TSX ASY ...	—
Analog Outputs Switching Interface	For example: Telemecanique ABR-2EB312B	—

## Analog Signal Conflicts

In order to enjoy the rapid Switchover provided by the Premium Hot Standby, it is necessary that the I/O output values be applied to both the Primary and Standby outputs once per MAST task. Otherwise, a Switchover might cause output values to change in an unexpected manner. When applying these output images, there will naturally be a small delay between the application to the outputs of the Primary PLC and to those of the Standby PLC. In the case of discrete I/O, the possible values of the outputs are limited to 1 and 0, and do not normally change twice within the period of one MAST task. In the analog case, however, the output values are not so limited and the application of two identical but staggered analog signals to a single actuator could create unexpected behaviors. This can be avoided by using a switching interface so that only one analog signal is applied at a time.

### WARNING

#### UNINTENDED EQUIPMENT OPERATION

Design your Hot Standby system so that only one analog output signal at a time is applied to an actuator.

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

## Analog Output Module Cabling

Therefore, the recommended configuration for Premium Hot Standby analog outputs uses two low-level switching interfaces (for example, the Telemecanique ABR-2EB312B or JM Concept GK3000D1) to switch between the analog output signals from the two PLCs. The operating inputs to the switching interfaces, which determine the switch configuration and hence the effective analog signal, are provided by an In-rack discrete output module managed by just one Hot Standby PLC.

---

## Analog Output Control Program (Section 0)

In the illustration on the previous page, note that the discrete output module controlling the switching interface is only connected to the Primary Hot Standby rack (PLC A). This is equivalent to saying that this discrete output module is locally managed by PLC A - its outputs are Local I/O. Therefore, as with all Local I/O, the behavior of these discrete outputs should be managed in the first section (Section 0) of your program. Here is an example in "pseudocode" of this Section 0 programming:

```
IF (I am the Primary) THEN
  Set the Switching Interface to PLC A's analog signal
ELSE
  Set the Switching Interface to PLC B's analog signal
END IF
```

In reality, implementing this pseudocode would mean testing the %SW61 status register to determine the local PLC's operating mode, and setting the discrete %Q output bits to 1 if in the Primary mode, or 0 if in any other mode. Therefore, in practice your code would appear more like:

```
IF (%SW61.0 = 0) AND (%SW61.1 = 1) THEN
  %QX.Y = 1
ELSE
  %QX.Y = 0
END IF
```

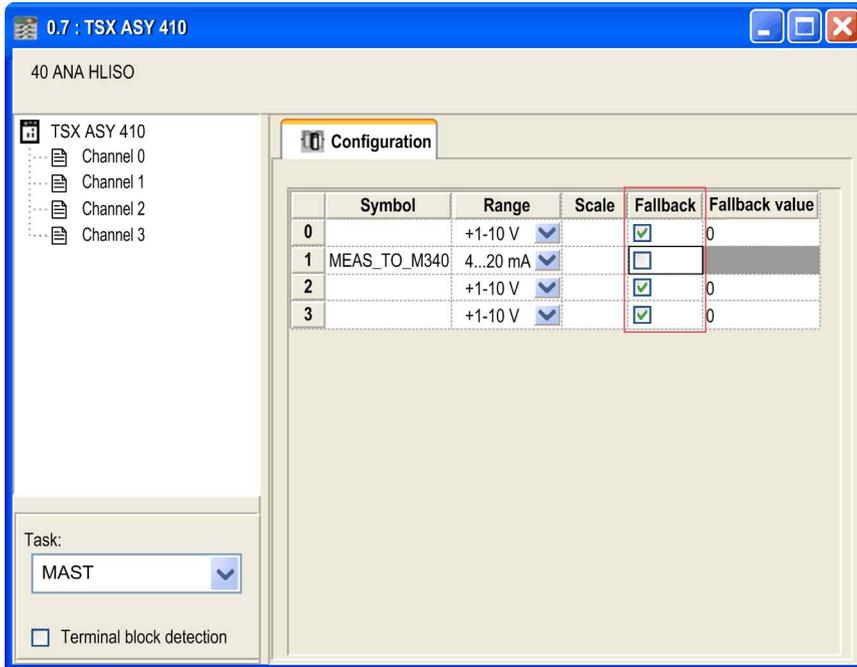
Now, revisit the illustration and consider the following:

- PLC A is the Primary. The code in section 0 of your program executes, and sets the discrete output bits to 1. The switching interfaces respond to this input by allowing the analog signal from PLC A to drive the actuator. During this same period, the section 0 code also runs on Standby PLC B, but to no effect as there is no connection between the discrete outputs on PLC B and the switching interfaces.
- Now, assume that a Switchover has occurred, and PLC A has entered the Standby mode, while PLC B is the new Primary. PLC A still runs section 0 of the application program, but now will assert a value of 0 at the discrete output connected to the switching interface. The switching interfaces will respond to this input by using the analog signal from the new Primary, PLC B, to drive the actuator.
- The last situation to consider is when the PLC controlling the Local I/O can no longer execute section 0 of the application program. In our example above, such a case would arise if PLC A detected an error and entered the Offline operating mode. This makes the importance of picking the proper fallback behavior for locally managed I/O very clear. In the example above, the proper fallback behavior would be "fallback to zero".

## Analog I/O Fallback Modes

Proper configuration of I/O fallback modes is essential in providing the redundancy of a Hot Standby system. Analog module outputs must not be configured to a fallback state to prevent unintended equipment operation in the short period after the Primary controller has become inoperative and before the Standby controller has assumed the Primary role.

For the output channels, make sure the **Fallback** checkboxes are not checked:



## WARNING

### UNINTENDED EQUIPMENT OPERATION

Uncheck the output module fallback check boxes to prevent changes in output states during Switchover.

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

---

## Minimum Configuration for Ethernet I/O

### At a Glance

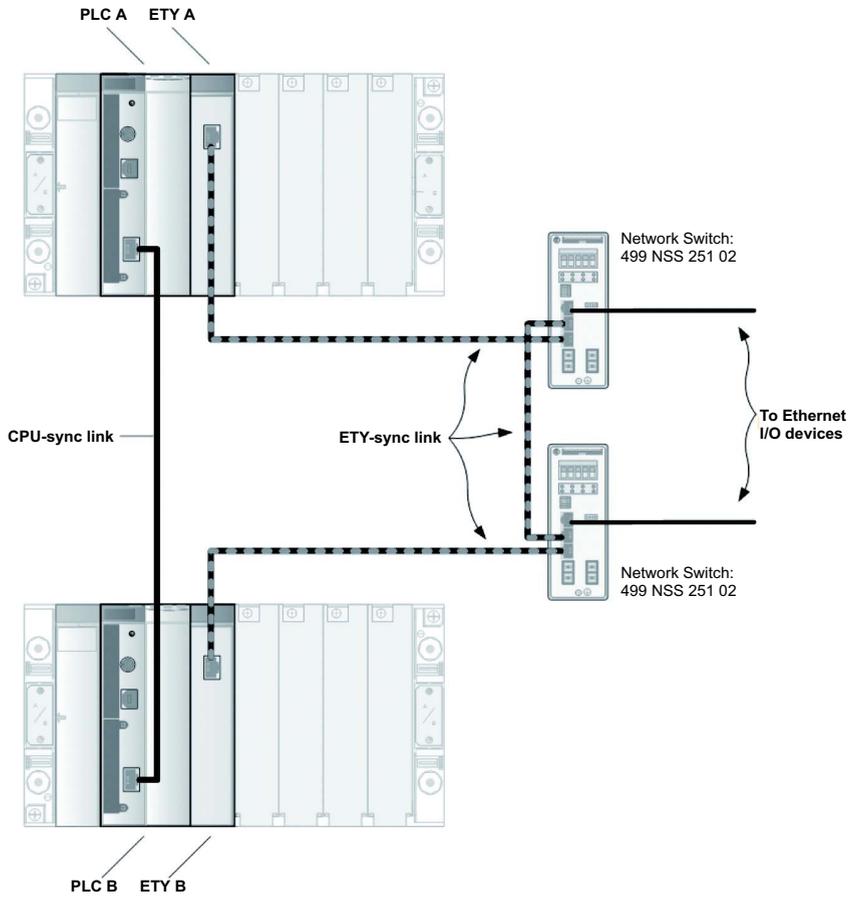
The minimum configuration necessary to support Redundant Ethernet-based I/O appears below.

The distinguishing features of this configuration are:

- Because only one pair of ETYs is shown in this configuration, these ETYs are by default the Monitored ETY modules.  
They must serve two roles:
  - Manage the ETY sync link.
  - Provide Redundant Ethernet I/O capability.
- If an I/O Scanning service is run on these Monitored ETYs, events such as loss of communication to the attached Ethernet I/O devices can trigger an automatic Switchover event.
- Redundant but non-Monitored Ethernet I/O cannot automatically cause a Switchover when it becomes inoperative. If this behavior is desired, create it in your application program.
- The topology of the Ethernet network connected to the Monitored ETYs can take many forms (tree, ring, etc.) depending on the type of network switches used.

**NOTE:** For switches in difference network topologies like star, tree or ring, refer to the ConneXium catalog and Transparent Ready technical publications.

## Illustration



## Parts List

Name	Reference	Minimum Version
All equipment in the core system ( <i>see page 24</i> ) plus...		
Network Switches	For example: 499 NSS 251 02	—
Ethernet Cables	Various	—

---

## ETY-sync link Diagnostics Example

The ETY-sync link plays an important role in transmitting diagnostic information between the Hot Standby controllers. The following example demonstrates how the controllers can use this information to diagnose specific events more accurately.

If the power supply to the Primary PLC is interrupted by some event that does not affect other modules on the same rack, the Standby PLC initially handles the event as a loss of communication on the CPU-sync link. This is the same diagnostic that is reported when the CPU-sync link cable is disconnected. To distinguish between these two cases, the Standby CPU requests from its local ETY module the status of the remote PLC. Using this secondary communications channel, the Standby can determine that the Primary controller has experienced a power loss, and assume the Primary role.

## IP Address Considerations

When used in a Premium Hot Standby System, the Ethernet TCP/IP network modules TSX ETY 4103/5103 support address swapping at switchover. If a Premium Hot Standby ETY primary module initially has the IP address  $xx.yy.zz.n$ , then the PLC automatically assigns its counterpart ETY on the standby rack an IP address of  $xx.yy.zz.n+1$ . Therefore, to avoid IP address conflicts, do not assign consecutive IP addresses to the ETY modules on the primary PLC's rack.

IP address  $xx.yy.zz.255$  in the primary rack is reserved for TCP/IP broadcast messages, and is not managed at switchover. You cannot assign IP address  $xx.yy.zz.254$  to an ETY module in the primary rack because a switchover would cause the corresponding ETY module in the other rack to take the broadcast IP address  $xx.yy.zz.255$ .

### **WARNING**

#### **UNINTENDED EQUIPMENT OPERATION**

- Do not assign consecutive IP addresses to ETY modules in the Primary rack. This will cause duplicate IP addresses to be assigned to the ETY modules on the Standby rack.
- Do not assign the IP addresses  $xx.yy.zz.254$  or  $xx.yy.zz.255$  to any ETY module in a Premium Hot Standby system.
- Do not assign IP address  $xx.yy.zz.0$  to a Monitored ETY, the Monitored ETY will remain in fault and have its default IP. The CPU thus remains in offline mode.
- The Primary and Standby ETYs must reside in the same network and subnetwork.

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

**NOTE:** If you assign an address of  $xx.yy.zz.254$  or  $xx.yy.zz.255$  to the Monitored ETY modules, the system will not come online when the Run command is issued, and diagnostic indications will appear on the PLC and ETY LEDs. However, the controller does not detect IP address conflicts between non-Monitored ETYs, so there is a potential for network conflicts with these modules.

---

**NOTE:** For more information on this topic, see [Swapping Network Addresses at Switchover \(see page 142\)](#) and [Ethernet Service Switchover Latencies \(see page 112\)](#).

### Number of Switches

If you will use the ETY-sync link to support Monitored Ethernet I/O, connect it through at least two network switches. If you do not connect it through at least two switches, a loss of communications on the ETY-sync link causes both PLCs to go Offline.

The sequence of events that would result in both PLCs going into Offline mode is:

1. The single network switch on the ETY-sync link becomes inoperative, and because the I/O Scanning service is running on the Monitored ETYs, a Switchover will occur.
2. The Primary PLC (PLC A) signals the Standby PLC (PLC B) to assume the Primary role. PLC A then enters the Offline mode.
3. PLC B assumes the Primary role, and restarts all configured Ethernet services, including the I/O Scanning service on the ETY-sync link.
4. Because the sole network switch is still inoperative, the new Primary (PLC B) cannot reconnect to the Monitored Ethernet I/O, and must take itself Offline as well.

## WARNING

### UNINTENDED EQUIPMENT OPERATION

Connect the ETY-sync link through at least two approved network switches if you plan to use the ETY-sync link to provide Monitored Ethernet I/O capabilities.

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

**NOTE:** ETY module connections are described further in: [The Sync Links \(see page 22\)](#). Configuration of Redundant Ethernet I/O in Unity Pro is described in [Configuring TSX ETY 4103/5103 Modules \(see page 146\)](#) and in the *Premium and Atrium Using Unity Pro Ethernet Network Modules User Manual*, reference 35006192.

### Hot Swapping ETY Modules

One advantage offered by Premium controllers is that many in-rack modules can be hot swapped, that is, replaced while power is applied to the Premium rack. In general, hot swapping operations are performed the same way in Hot Standby systems as in standalone Premium systems. The hot swapping of ETY modules in a Hot Standby system represents an important exception.

---

Consider these points before you attempt to hot swap an ETY module in a Premium Hot Standby system:

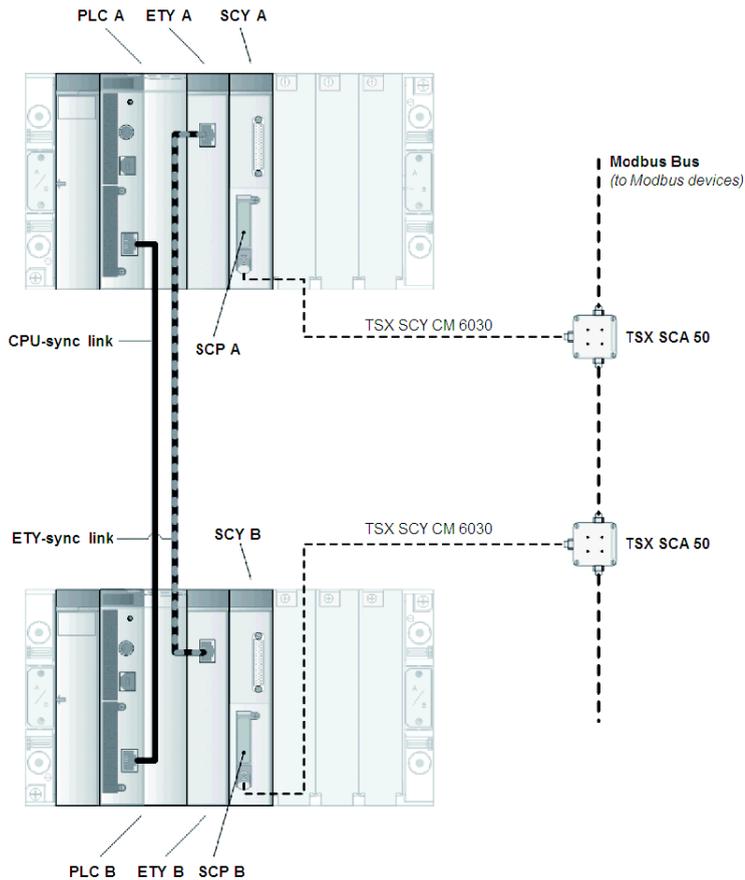
- Hot swapping a monitored ETY in the primary PLC's rack triggers a switchover. The primary PLC goes Offline and the standby PLC takes control of the system. You need to reinitialize the offline controller in order to restore hot standby capability.
- If you hot swap a Monitored ETY in the standby PLC's rack, a switchover does not occur. The standby PLC goes offline and the primary PLC remains the primary. The system is not redundant until you reinitializes the offline controller.
- If you hot swap a non-monitored ETY in either PLC's rack, the system behaves almost identically to a standalone Premium system. There are no automatic state changes, and your application program largely determines the resulting behavior.

## Minimum Configuration for Redundant Modbus I/O

The minimum configuration necessary to support Redundant Modbus-based I/O appears below. The distinguishing features of this configuration are:

- The Modbus addresses of In-rack Modbus modules are automatically reassigned during a Switchover event.
- Errors detected on Modbus I/O cannot cause an automatic Switchover event.

### Illustration



## Parts List

Name	Reference	Min. Vers.
All equipment in the <i>Core Hot Standby Hardware</i> , <a href="#">page 24</a> plus...		
Premium Communications Module	TSX SCY 21601 or TSX SCY 11601	2.3 1.1
Modbus Card for Communications Module	TSX SCP 114	1.7
Modbus Junction Box	TSX SCA 50	—
Modbus Cables (TSX SCP 114 to Junction Box)	TSX SCP CM 4030 or TSX SCY CM 6030	—
Modbus Cables (between Junction Boxes)	Various	—

## Modbus Communications Modes and Formats

Do not change the communications mode (Modbus, Uni-Telway, Character) of the TSX SCP 114 module while the Hot Standby system is operational. The Primary controller does not update the Standby controller when TSX SCP 114 configuration information changes. If this information is changed when the Hot Standby system is operating, and then a Switchover event occurs, the communications mode can change unexpectedly.

### **WARNING**

#### **UNINTENDED EQUIPMENT OPERATION**

Do not change the communications mode (Modbus, Uni-Telway, or Character) of the TSX SCP 114 module while your Hot Standby system is operating.

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

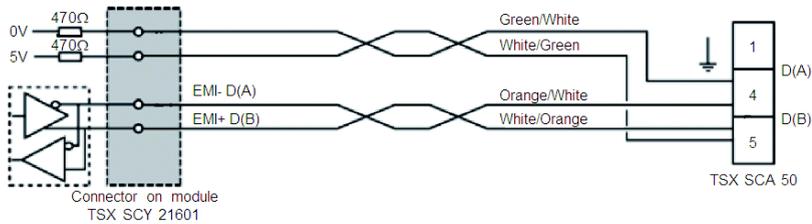
## Modbus Slave Link on RS485, Two Wires

Your Premium Hot Standby system is capable of acting as a Modbus slave when the TSX SCP 114 PCMCIA card is used in the TSX SCY 21601 module. The preferred Modbus cable when connecting to the TSX SCP 114 is the TSX SCP CM 4030. When you implement a Modbus network as part of your Premium Hot Standby system, configure the network so the Modbus master controls the polarization.

## Modbus Master Link on RS485, Two Wires

Only the Modbus master role is available when you use the integrated channel of the TSX SCY 21601 or TSX SCY 11601 modules. The Modbus master must establish the polarization. The preferred Modbus cable when connecting to the TSX SCP 11601 / 21601 is the TSX SCP CM 6030. The line can be adapted by positioning the corresponding connector to ON in the TSX SCA 50 Modbus junction boxes at the end of the RS485 line.

The following illustration displays the TSX SCY CM 6030 cord connection:



### Networked Modbus Modules

Not all Modbus devices have been tested for use with Premium Hot Standby systems. Only the following devices have been tested for use as Modbus TCP/IP devices in a Premium Hot Standby system:

- Advantys STB
- Advantys OTB
- Modicon Momentum I/O
- Telemecanique ATV61
- Magelis XBT G
- Magelis XBT GT
- Modicon Premium

Only the following devices have been tested for use as Modbus slaves on a network including a Premium Hot Standby system with Modbus capabilities:

- Advantys STB
- Advantys OTB
- Telemecanique ATV31
- Telemecanique TeSys U-Line

---

## Card Operations

Mounting, dismounting, and connecting / disconnecting cables to the TSX SCP 114 card while power is applied to the TSX SCY 21601 can damage the card.

### CAUTION

#### EQUIPMENT DAMAGE

- Remove all power from your TSX SCY 21601 device before attempting to insert or remove the TSX SCP 114 communications card.
- Remove all power from your TSX SCY 21601 device before attempting to connect or disconnect cables from the TSX SCP 114 communications card.
- Before mounting, dismounting, or connecting cables to the SCP 114, read and understand the *Micro/Premium PLCs TSX SCP 111/112/114 PCMCIA Communication Cards Quick Reference Guide*, reference 1590524.

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

## Lost Messages at Switchover

When a Switchover event occurs, there is a possibility that some Modbus messages will not reach the intended devices. Program your application to confirm that the station addressed on the Modbus link has received and responded correctly to a message before sending a new one. For example, this may involve buffering all outgoing Modbus commands, trapping the occurrence of a Switchover event, and using the buffered commands to check the values of the commanded registers for compliance.

### WARNING

#### UNINTENDED EQUIPMENT OPERATION

Create your application program to confirm the receipt and effect of Modbus commands issued just before and during a Switchover event.

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

## Modbus Address Considerations

If a Premium Hot Standby Modbus module initially has a slave address  $n$ , then the PLC automatically assigns its counterpart Modbus module on the Standby rack an address of  $n+1$ . Therefore, to avoid Modbus slave address conflicts, do not assign consecutive addresses to the Modbus modules on the Primary PLC's rack. Also, do not assign any Modbus module on the Primary rack the address 98 or the Switchover address swap will not work.

---

## **WARNING**

### **UNINTENDED EQUIPMENT OPERATION**

- Do not assign consecutive slave addresses to Modbus modules in the Primary rack. This will cause duplicate addresses to be assigned to the corresponding Modbus modules on the Standby rack, leading to unintended equipment operation.
- Do not assign the slave address 98 to any Modbus module in a Premium Hot Standby system. The automatic assignment of the n+1 address to the Standby Modbus slave will not work in this event.

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

**NOTE:** Unlike the Monitored ETY's IP addresses, there is no error detection preventing the use of Modbus slave address 98.

---

## Adding HMI / SCADA to the ETY-sync link

A switchover from the Primary PLC to the Standby PLC can occur in either of these instances:

- The Monitored ETY in the Primary rack becomes inoperative.
- The Ethernet link to the Monitored ETY breaks.

You may choose to use an HMI or SCADA device to remotely control your Premium Hot Standby system. You can maintain remote control through the HMI or SCADA device after a switchover only when that device is connected to the ETY's sync link (through a switch). As the figure below shows, this case requires the connection of the ETY sync link to at least two switches.

**NOTE:** A failure of the HMI or SCADA device does not cause a switchover.

### **WARNING**

#### **UNINTENDED EQUIPMENT OPERATION**

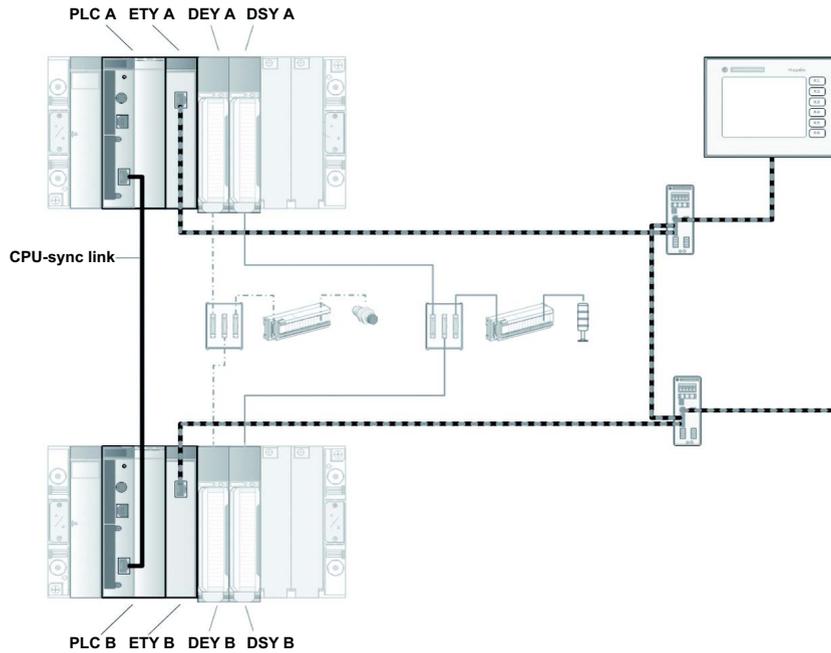
If the Primary PLC will be addressed by HMI or SCADA terminals for the purpose of system control:

- Make the HMI/SCADA connection to the ETY-sync link.
- Always run the I/O Scanning service on the Monitored ETY modules.
- Use a minimum of two network switches on the ETY-sync link.

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

**NOTE:** The HMI or SCADA device itself may become inoperable, or the Ethernet link between the device and the Hot Standby system may be lost, but these conditions do not cause a switchover.

The following example shows a Magelis XBT GT HMI panel connected to two 499 NSS 251 02 Ethernet switches that are used for the sync link between the monitored ETY modules in the Primary and Standby racks. If one of these ETY modules becomes inoperable or if the Ethernet link is broken, the HMI panel can still monitor and control the system.



---

## Section 3.2

### Compatible Equipment

---

#### Purpose of this Section

This section describes the I/O and connection equipment compatible with the Premium Hot Standby.

#### What Is in This Section?

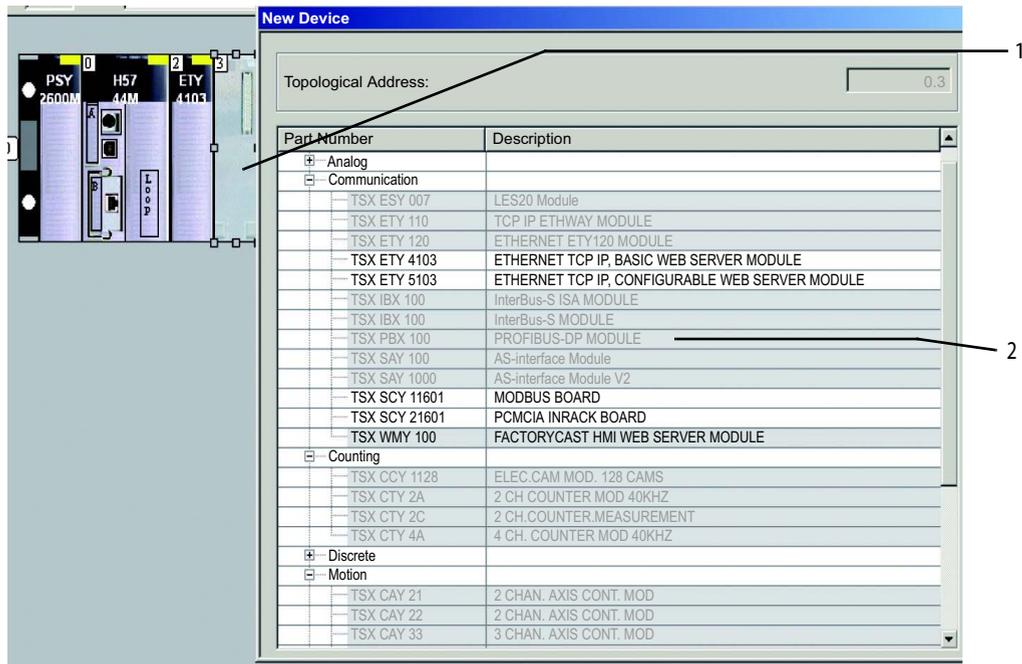
This section contains the following topics:

Topic	Page
Overview	79
Premium Racks and Rack Accessories	81
Premium Power Supplies	82
In-rack Communication Modules: Ethernet	83
In-rack Communication Modules: Modbus	84
In-rack I/O Modules: Discrete	85
In-rack I/O Modules: Analog	86
In-rack I/O Modules: Safety	87
Connection Devices: Discrete I/O	88
Connection Devices: Main Rack Analog I/O	89
Allowed Devices: Connected by Ethernet	90
Allowed Devices: Connected by Modbus	91
Ethernet Network Devices	92
Modbus Network Devices and Cables	93
Maximum Configuration	94

## Overview

This section lists all of the modules and equipment that have been tested for use with the Premium Hot Standby system.

**NOTE:** In most instances, if you try to put a module that is not on the approved list into a Premium Hot Standby rack, Unity Pro will recognize that it is a disallowed module and prevent you from configuring the device in question. The unapproved module will usually appear "grayed out" in both the pictorial and list representations as illustrated here:



1 : Gray in pictorial representation.

2 : Gray in list representation.

The Premium Hot Standby system has other measures to help you avoid the use of unapproved modules in your Premium racks. For example, if you try to insert a Hot Standby PLC in a rack where unapproved modules are already present, a dialog alerting you to this fact appears.

If you attempt to compile your application program with an unapproved module still in the rack, your program compiles as if the device was not there. If the application program is then downloaded to the PLCs, Unity Pro reports detected errors in the log file, but the program runs as if the slot with that device was empty. The ERR LED on such unconfigured modules typically illuminate.

---

Despite all of the protections noted above, some modules and configurations may compile without diagnostic messages but not run as intended. For example, some Modbus communication devices have a minimum firmware requirement that the Hot Standby controllers do not test. Therefore, you must only use the compatible equipment listed in this section.

 **WARNING**

**UNINTENDED EQUIPMENT OPERATION**

Use only the approved In-rack and network-connected modules listed in the following tables. If unapproved modules are installed, the Premium Hot Standby system may behave in an unexpected manner during Switchover events.

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

---

## Premium Racks and Rack Accessories

<b>Name</b>	<b>Reference</b>	<b>Min. Vers.</b>
Premium Extended Rack, 4 Positions, or	TSX RKY 4EX	—
Premium Standard or Extended Rack, 6 Positions, or	TSX RKY 6EX	—
Premium Standard or Extended Rack, 8 Positions, or	TSX RKY 8EX	—
Premium Standard or Extended Rack, 12 Positions, or	TSX RKY 12EX	—
Premium Rack Line Terminators	TSX TLYEX	—
Premium Protective Covers	TSX RKA 02	—

---

## Premium Power Supplies

Name	Reference	Min. Vers.
Premium Power Supplies	TSX PSY ...	—

**NOTE:** Any Premium TSX PSY ... power supply may be used with your Hot Standby system, provided the supplies are identical, identically positioned, and are capable of meeting the power requirements of your system under maximum load. To determine whether a Premium power supply will meet the needs of your system, refer to *Premium and Atrium using Unity Pro Processors, racks and power supply modules Implementation manual*, reference 35010524, section *TSX PSY power supply modules: breakdown of power consumption and power*. This reference provides a means of calculation Premium system power requirements.

**NOTE:** Unity Pro also has a feature that may help you estimate your system's power consumption. To access this feature, right-click on the power supply in the X-Bus configuration screen. Select the "Power Supply and I/O Budget..." menu item. Select the "Power supply" tab when the dialog box opens.

---

## In-rack Communication Modules: Ethernet

### Ethernet Modules

The following Ethernet communication modules have been tested for In-rack use in the Premium Hot Standby system:

Name	Reference	Min. Version
Premium Hot Standby Ethernet Modules	TSX ETY 4103 or TSX ETY 5103	V4.0 or higher for mono rack V5.0 or higher for multi-rack

**NOTE:** Please check with Schneider-Electric Technical Support for latest version numbers.

**NOTE:** Install Modbus communication modules in the main racks. Do not install communication modules in extended racks.

The communication channel of an Ethernet TSX ETY 4103 or 5103 module connects to a TCP/IP network that supports Modbus messaging on a TCP/IP profile.

Both devices support:

- Modbus messaging
- simple network management protocol (SNMP)
- electronic mail notification service (SMTP)
- input/output management over an Ethernet network (I/O Scanning)
- IP address management (BOOTP/DHCP)
- dynamic IP addressing (with version 3.2 or higher)
- access to the built-in Web server
- common data exchange between stations (Global Data)
- diagnostics from Web pages

The TSX ETY 5103 module provides two additional capabilities that are not available in the ETY 4103:

- FactoryCast, where you have the option to create user Web and TCP Open pages
- NTP, which provides time synchronization updates for controller clocks based on a Universal Time Coordinated (UTC) reference source

---

## In-rack Communication Modules: Modbus

### Modbus Modules

The following Modbus communication modules have been tested for In-rack use in the Premium Hot Standby system:

Name	Reference	Min. Vers.
Modbus Communication Module with Multiprotocol Communications Card (Slave or Master and character mode)	TSX SCY 21601 with TSX SCP 114	2.3 1.7
Modbus Communication Module (Modbus Master)	TSX SCY 11601	1.1

**NOTE:** Please check with Schneider-Electric Technical Support for latest version numbers.

**NOTE:** The combination of the TSX SCY 21601 base with the TSX SCP 114 multiprotocol card allows your Premium Hot Standby controllers to act either as the Modbus master or as a slave or as character mode. This configuration allows the use of third-party Modbus masters. The TSX SCY 11601 module can only operate as the Modbus master.

**NOTE:** Install Modbus communication modules in the main racks. Do not install communication modules in extended racks.

**NOTE:** It is recommended to use the TSX SCP 114 (see cabling example (see *Premium and Atrium Using Unity Pro, Asynchronous Serial Link, User Manual*)) with version 3.3 or higher when it is running with a TSX SCY 21601 coupler.

---

## In-rack I/O Modules: Discrete

The following Premium Discrete I/O modules have been tested for In-rack use in the Premium Hot Standby system:

<b>Name</b>	<b>Reference</b>	<b>Min. Vers.</b>
Discrete Input Modules (HE10 connectors)	TSX DEY ... K	—
Discrete Output Modules (HE10 connectors)	TSX DSY ... K	—
Discrete Input/Output Modules (HE10 connectors)	TSX DMY ... K	—
Discrete Input Modules (screw terminals)	TSX DEY ...	—
Discrete Output Modules (screw terminals)	TSX DSY ...	—

---

## In-rack I/O Modules: Analog

The following Premium Analog I/O modules have been tested for In-rack use in the Premium Hot Standby system:

Name	Reference	Min. Vers.
Analog Input Modules (screw terminals or SUB-D)	TSX AEY ...	—
Analog Output Modules (screw terminals or SUB-D)	TSX ASY ...	—

---

## In-rack I/O Modules: Safety

The following Preventa Safety modules have been tested for In-rack use in the Premium Hot Standby system:

Name	Reference	Min. Vers.
Preventa Safety Module (screw terminals and SUB-D)	TSX PAY ...	—

---

## Connection Devices: Discrete I/O

The Premium Hot Standby system relies on special I/O connections to enable redundancy. The following connection devices and cables have been tested for use as redundant Discrete I/O connections in Premium Hot Standby systems:

<b>Name</b>	<b>Reference</b>	<b>Min. Vers.</b>
ABE7 Input Connection Blocks, 16 Channels	ABE7 ACC11	—
ABE7 Output Connection Blocks, 16 Channels	ABE7 ACC10	—
Telefast Connection Blocks	Various	—
HE10 cables, pre-assembled	TSX CDP **3	—
HE10 cables, pre-assembled or self-assembled	TSX CDP **3 or ABF H20H008	—

---

## Connection Devices: Main Rack Analog I/O

### Special I/O Connections

The Premium Hot Standby system relies on special I/O connections to enable redundancy. The following connection devices and cables have been tested for use as redundant Analog I/O connections in Premium Hot Standby systems:

Name	Reference	Min. Vers.
Analog Inputs - Analog Signal Duplicator	For example: JM Concepts JK3000N2	—
Analog Outputs - Switching Interface	For example: Telemecanique ABR-2EB312B or JM Concept GK3000D1	—

---

## Allowed Devices: Connected by Ethernet

The following table presents the I/O modules and other devices that can be redundantly controlled by a Premium Hot Standby system over an Ethernet connection:

<b>Name</b>	<b>Reference</b>	<b>Min. Vers.</b>
Advantys STB I/O Modules (Discrete and Analog)	STB ****	—
Advantys FTB I/O Modules (Discrete and Analog)	FTB ****	—
Advantys FTM I/O Modules (Discrete and Analog)	FTM ****	—
Ethernet for Momentum	170E ****	—
Modicon Momentum I/O Modules (Discrete, Analog)	170 A ****	—
Twido I/O Modules (Discrete and Analog)	TWD ****	—
Telemecanique Altivar Variable Speed Drives	ATV **	—
Modicon Premium PLCs	TSX ****	—
Modicon Quantum PLCs	140 ****	—
Telemecanique Tesys U-Line	LU ****	—
Magelis HMI	XBT G / XBT GT	—

---

## Allowed Devices: Connected by Modbus

### Modbus TCP/IP

The following table presents the I/O modules and other devices that can be redundantly controlled by a Premium Hot Standby system over an Ethernet (Modbus TCP/IP) connection:

Name	Reference	Min. Vers.
Advantys STB	STB ....	—
Advantys OTB	OTB ....	—
Ethernet for Momentum	170E ....	—
Modicon Momentum I/O	170 A ....	—
Telemecanique ATV61	ATV61	—
Magelis HMI	XBT G / XBT GT	—
Modicon Premium PLCs	TSX ....	—

**NOTE:** These Modbus slaves can only be controlled via Ethernet TCP/IP if an appropriate Ethernet / Modbus gateway is used. The following Ethernet / Modbus gateways were tested with the Premium Hot Standby system:

- TSX ETG 100 gateway
- TSX ETG 1000 gateway
- 174 CEV 30020 gateway

### Modbus Slaves

These devices have been tested in the role of Modbus slaves in a Premium Hot Standby system:

Name	Reference	Min. Vers.
Advantys STB	STB ....	—
Advantys OTB	OTB ....	—
Telemecanique ATV31	ATV31	—
Telemecanique Tesys U-Line	LU ....	—

---

## Ethernet Network Devices

All products of the ConneXium family that are compatible with standard TSX ETY 4103 / 5103 Ethernet modules in a non-Hot Standby configuration are expected to be compatible with the TSX ETY 4103 / 5103 modules in a Hot Standby system. The following Ethernet devices have been tested for use with the Premium Hot Standby system:

### Network Switches, General Use

Name	Reference	Min. Vers.
ConneXium Network Switches	TCS ESM 043F1CS0	—
ConneXium Network Switches	TCS ESM 043F2CS0	—
ConneXium Network Switches	TCS ESM 083F23F0	—
ConneXium Network Switches	TCS ESM 0•3F2CU0	—
ConneXium Network Switches	499 N•S 27100	—

### Network Switches, Specific to I/O family

These devices have been tested in the role of Modbus slaves in a Premium Hot Standby system:

Name	Reference	Min. Vers.
Advantys STB Network Interface Module	STB NIP 2212	—
Modicon Momentum Network Interface Module	170 ENT 110 0•	—
Advantys OTB Ethernet switches ( with embedded I/O)	OTB •••	—

**NOTE:** when using a STB NIP 2212 in a Premium HotStandby application, the Link Failure Mode (in the STB Configuration Web pages) should be set to Fallback option, otherwise a bump could be seen on the outputs.

### Ethernet / Modbus Gateways

These devices have been tested in the role of Modbus slaves in a Premium Hot Standby system:

Name	Reference	Min. Vers.
Premium Ethernet / Modbus Gateway	TSX ETG 100	—
Premium Ethernet / Modbus Gateway	TSX ETG 1000	—
ConneXium Ethernet / Modbus Gateway	174 CEV 30020	—

---

## Modbus Network Devices and Cables

Name	Reference	Min. Vers.
Modbus Junction Box	TSX SCA 50	—
Modbus Cables (TSX SCP 114 to Junction Box)	TSX SCP CM 4030 or TSX SCY CM 6030	—
Modbus Cables (between Junction Boxes)	Various	—

## Maximum Configuration

The following tables provide a summary of the maximum system configurations possible using the Premium Hot Standby PLCs and their compatible equipment.

For a given type of module, the maximum number of usable modules can be evaluated as: the maximum number of usable channels divided by the number of channels for this type of module, provided that a sufficient number of slots is available and other potential limitations (communication load, etc.) are met.

### TSX H57 24M Controller

Characteristics		TSX H57 24M	
Maximum configuration	Maximum number of TSX RKY 12EX racks	8	
	Maximum number of TSX RKY 4EX/6EX/8EX racks	16	
	Maximum number of slots	111	
Functions	Maximum number of channels	In-rack discrete I/O	1024
		In-rack analog I/O	80
		Expert (counting, axis, motion, weighing)	0
		Modbus / Character Mode	16
	Maximum number of connections	Built-in Uni-Telway (terminal port)	1
		Network (Ethernet TCP-IP)	2
		Third party field bus	0
		AS-i field bus	0
	Savable real-time clock	yes	
	Process control channels	10	
Process control loops	30		
Memory	Savable internal RAM	192 kilobytes	
	PCMCIA memory card (maximum capacity)	768 kilobytes	
Application structure	MAST task	1	
	Event processing (1 has priority)	64	
Application code execution speed	Internal RAM	100% Boolean	15.75 Kins/ms
		65% Boolean + 35% Digital	11.40 Kins/ms
	PCMCIA card	100% Boolean	15.75 Kins/ms
		65% Boolean + 35% Digital	11.40 Kins/ms

Characteristics		TSX H57 24M
Execution time	Basic Boolean instruction	0.039/0.057 s
	Basic digital instruction	0.053/0.073 s
	Floating point instruction	0.55/0.63 s
System overhead	MAST task	1 ms

## TSX H57 44M Controller

Characteristics		TSX H57 44M	
Maximum configuration	Maximum number of TSX RKY 12EX racks	8	
	Maximum number of TSX RKY 4EX/6EX/8EX racks	16	
	Maximum number of slots	111	
Functions	Maximum number of channels	In-rack discrete I/O	2048
		In-rack analog I/O	256
		Expert (counting, axis, motion, weighing)	0
		Modbus / Character Mode	16
	Maximum number of connections	Built-in Uni-Telway (terminal port)	1
		Network (Ethernet TCP-IP)	4
		Third party field bus	0
		AS-i field bus	0
	Savable real-time clock	yes	
	Process control channels	20	
Process control loops	60		
Memory	Savable internal RAM	440 kilobytes	
	PCMCIA memory card (maximum capacity)	2048 kilobytes	
Application structure	MAST task	1	
	Event processing (1 has priority)	64	
Application code execution speed	Internal RAM	100% Boolean	15.75 Kins/ms
		65% Boolean + 35% Digital	11.40 Kins/ms
	PCMCIA card	100% Boolean	15.75 Kins/ms
		65% Boolean + 35% Digital	11.40 Kins/ms

---

Characteristics		TSX H57 44M
Execution time	Basic Boolean instruction	0.039/0.057 s
	Basic digital instruction	0.054/0.073 s
	Floating point instruction	0.55/0.63 s
System overhead	MAST task	1 ms

**NOTE:** For more technical information regarding the capacity and performance of the Premium Hot Standby PLCs, refer to *Additional Information*, [page 241](#) and to the *Premium and Atrium using Unity Pro Processors, racks, and power supply modules Implementation manual*, reference 35010524.

---

## Section 3.3

### Example Hot Standby Systems

---

#### Purpose of this Section

This section presents example Premium Hot Standby systems.

#### What Is in This Section?

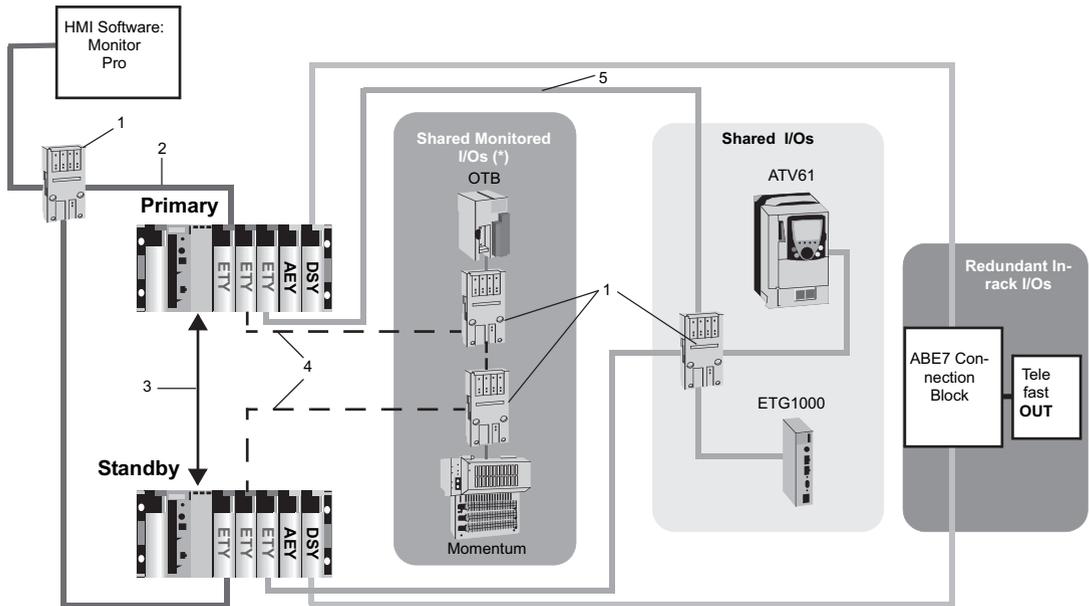
This section contains the following topics:

Topic	Page
System with Multiple ETYs Running I/O Scanning Services	98
System with Redundant I/O and SCADA Network Services	100
System with Mixed Ethernet and Modbus	102

## System with Multiple ETYs Running I/O Scanning Services

### Illustration

Now that the minimum configurations of each type of I/O and communications module have been introduced, we will consider a few representative sample systems. The following graphic shows a Premium Hot Standby System with multiple ETYs, and with three of these paired ETYs running I/O Scanning services:



(\*) "Monitored" means that an automatic Switchover will occur if the associated ETY become inoperative, or the cable to the first switch becomes disconnected.

**NOTE:** An Ethernet ring configuration will be non-functional unless you use managed switches. Other network configurations can use both managed and unmanaged switches.

---

## Parts List

No.	Name	Reference	Min. Vers.
All equipment in the pictured Premium Hot Standby rack plus...			
1	Ethernet Switch	499NES25100	—
2	Ethernet TCP/IP	490NTW000 ***	—
3	CPU-sync link	490NTC000 ***	—
4	Ethernet I/O Scanner #1	490NTW000 ***	—
5	Ethernet I/O Scanner #2	490NTW000 ***	—



---

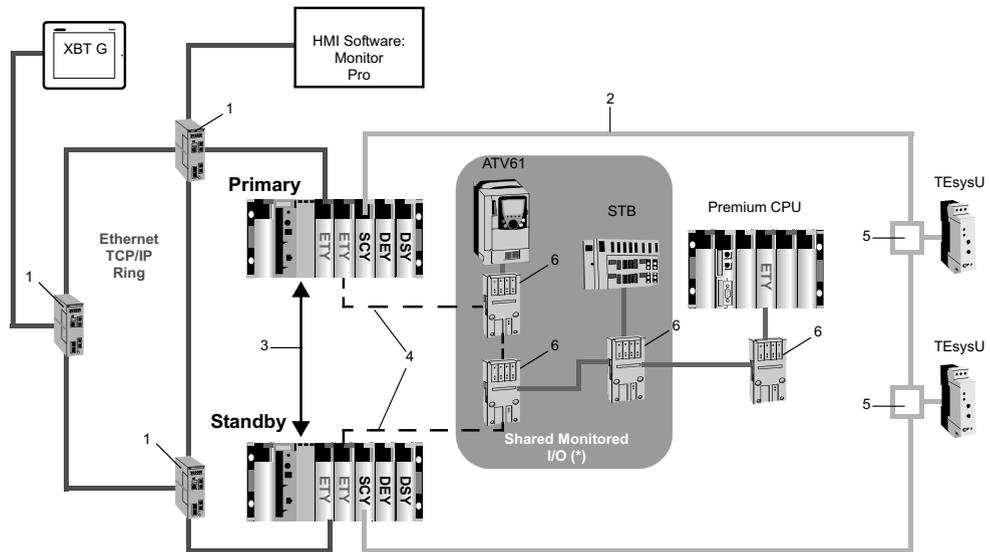
## Parts List

No.	Name	Reference	Min. Vers.
All equipment in the pictured Premium Hot Standby rack plus...			
1	Ethernet TCP/IP network #1	490NTW000***	—
2	Ethernet Switch	499NES25100	—
3	Ethernet TCP/IP network #2 and #3	490NTW000***	—
4	CPU sync-link	490NTC000***	—
5	ConneXium Ethernet Switch with Ring capability	TCSESM0•3	—
6	Modbus Gateway (example: TSX ETG 1000)	TSXETG1000	—
7	Monitored ETY	TSXETY•103	4.0
8	Bus X		—
9	Analog output signal A		—
10	Analog output signal B		—
11	Control signal		—

## System with Mixed Ethernet and Modbus

The following graphic shows a Premium Hot Standby System with mixed Ethernet and Modbus I/O.

### Illustration



(\*) "Monitored" means that an automatic Switchover will occur if the associated ETY or the first switch become inoperative, or if the cable to the first switch becomes disconnected.

**NOTE:** An Ethernet ring configuration will be non-functional unless you use managed switches. Other network configurations can use both managed and unmanaged switches.

### Parts List

No.	Name	Reference	Min. Vers.
All equipment in the pictured Premium Hot Standby rack plus...			
1	ConneXium Ethernet Switch with Ring capability	TCSESM0•3	—
2	Modbus RS485 cable	TSXSCYCM6030	—
3	CPU sync-link	490NTC000•••	—
4	Ethernet I/O scanner communications	490NTW000•••	—
5	Junction box	TSXSXA50	—
6	Ethernet Switch	499NES25100	—

---

# Chapter 4

## PLC Communications and Switchover

---

## Section 4.1

### Database Transfer Between Hot Standby PLCs

---

#### Purpose of this Section

This section details the cyclical data transfer between controllers that occurs in a Premium Hot Standby system.

#### What Is in This Section?

This section contains the following topics:

Topic	Page
Understanding the Premium Hot Standby Database Transfer Process	105
Understanding System Scan Time in Premium Hot Standby	107

## Understanding the Premium Hot Standby Database Transfer Process

### Overview

A Premium Hot Standby System requires two backplanes, one for the Primary controller and one for the Standby. These backplanes and their modules must be configured with identical hardware, software, and firmware. One of the controllers (PLCs) functions as the Primary controller and the other as a Standby controller.

- The Primary updates the Standby at the beginning of every scan.
- The Primary monitors the health of the system according to its programming, and regularly communicates this status information to the Standby. The Standby provides health information to the Primary by means of the Reverse Transfer System Words.
- If the Primary becomes inoperative, the Standby takes control within one scan.

### Data Transfer and User Data

The database that is cyclically transferred from the Primary controller to the Standby controller (via the Copros and the CPU-sync link) includes both system data and user application data. In both cases, some of this data is located (addressable) data, and some is unlocated. The data transferred includes:

- **System Information**

- Located (a subset of the System Bits and Words)

- Exchanged during every MAST Task:

- System Bits: %S30, %S31, %S38, %S50, %S59, %S94

- System Words: %SW0, %SW1, %SW8, %SW9, %SW49...%SW53, %SW59, %SW60, %SW70, %SW108

- Exchanged only during Switchover

- %SD18 and %SD20

- Unlocated

- A subset of the system data managed by the Primary PLC's operating system. This subset includes system counters used by function blocks such as TON, TOFF, and others.

- **User Application Data**

- Located

- All %M, %MW, %MD, and %MF data from address 100 up to the maximum number of global address fields configured in Unity Pro's Configuration tab, but no more than 128 KB. The range below 100 (for example, %MW0 - %MW99) is not transferred.

- The output (%Q) objects and any output forcing settings.

- EDT / DDT when they are located by the user.

- Sequential Function Chart (SFC) data types.

- Unlocated

- EDT / DDT when they are located by the user.

- Function Block (EFB / DFB) data types.

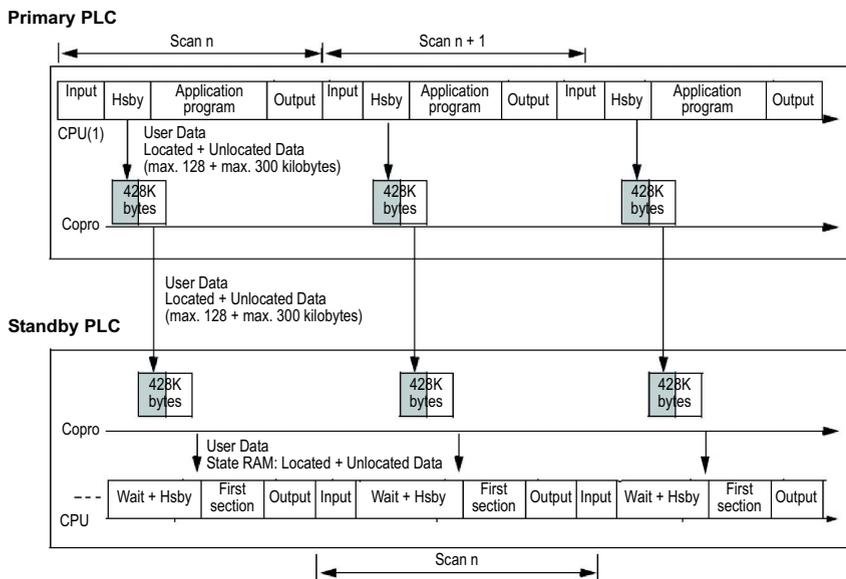
**NOTE:** In addition to the above, the Primary controller sends the values of all Forced Bits to the Standby as part of the regular database exchange.

**NOTE:** The maximum amount of located data that can be transferred in the database is 128 KB for both the TSX H57 24M and the TSX H57 44M. The maximum unlocated data is 120 KB for the TSX H57 24M and 300 KB for the TSX H57 44M.

**NOTE:** The maximum size of the entire database is approximately 165 KB for the TSX H57 24M and 405 KB for the TSX H57 44M.

### Hot Standby Transfer Diagram

The following illustrates the transfer of data from the Primary to the Standby on the TSX H57 44M:



---

## Understanding System Scan Time in Premium Hot Standby

### Effect on System Scan Time

The scan time of any Premium Hot Standby System depends on the amount of data transferred. Because data must be transferred from Primary to Standby, any Premium Hot Standby System always has a higher scan time than a comparable standalone system.

**NOTE:** In Premium PL7 Warm Standby systems, the CPU performed both:

- application program (project) processing
- communication transfer

In a Premium Hot standby, the workload is divided between the CPU and the Copro:

- CPU performs application program processing
- Copro performs communication transfer

The use of the Copro to perform the database transfer mitigates the longer scan times noted above, and significantly reduces communications latencies in the Hot Standby as compared to legacy Warm Standby systems.

### Performance Considerations

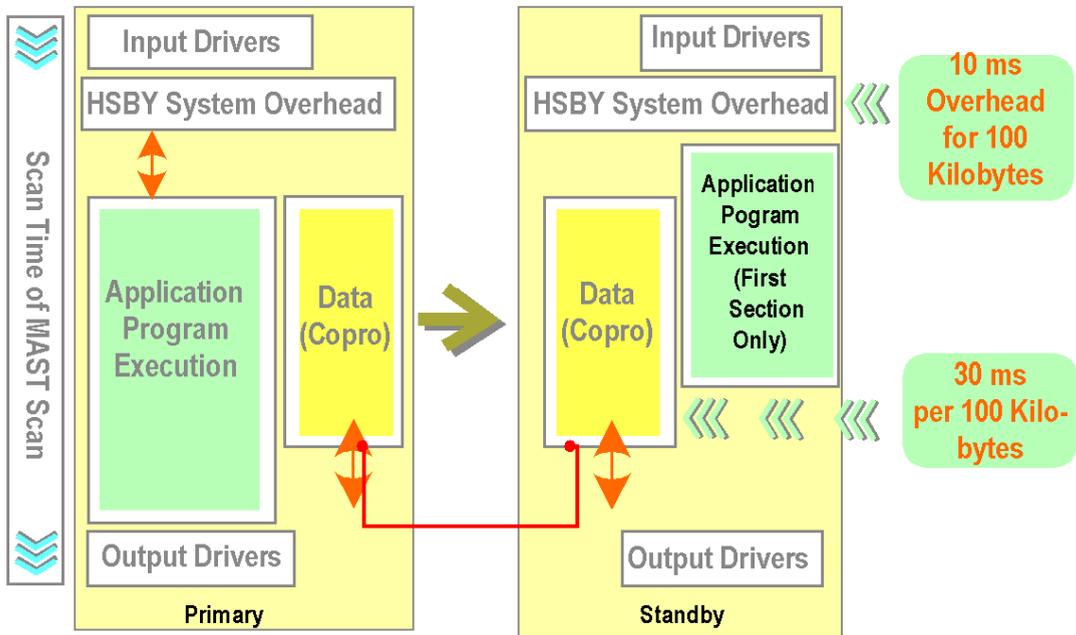
A Premium Hot Standby increases the length of MAST task scan times, creating system overhead.

**NOTE:** It takes a short but finite amount of time for the system to create the database and copy it from the PLC's internal memory into the Copro's shared memory. We refer to this period as the System Overhead.

System overhead is the time required to copy the application data to the communication link layer.

The network scan (communication between Primary and Standby Copros)

1. exchanges data between both controllers
2. runs in parallel with the application program



In most circumstances, as pictured on this page, the time required to transmit the database between the Copros will not affect the MAST task period.

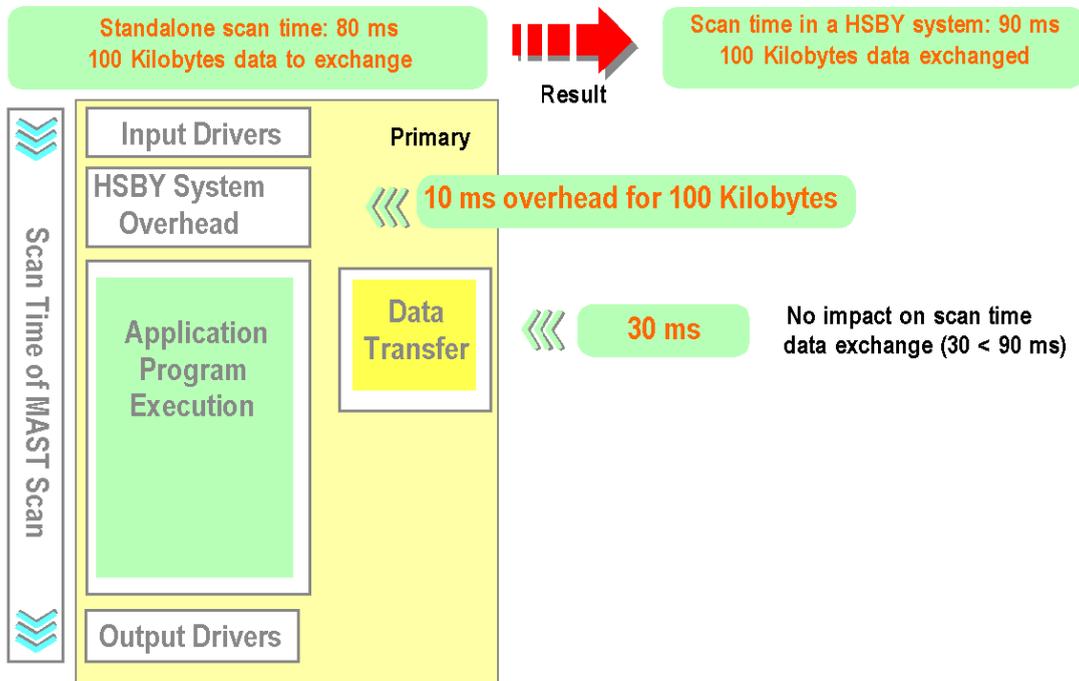
However, when processing large or intensive application programs, the additional system overhead and transmission times can affect the MAST task duration.

### Examples

#### Example #1

- Application program execution time in standalone Premium PLC: 80 ms
- Database size: 100 Kilobytes

The following illustration displays the example #1:

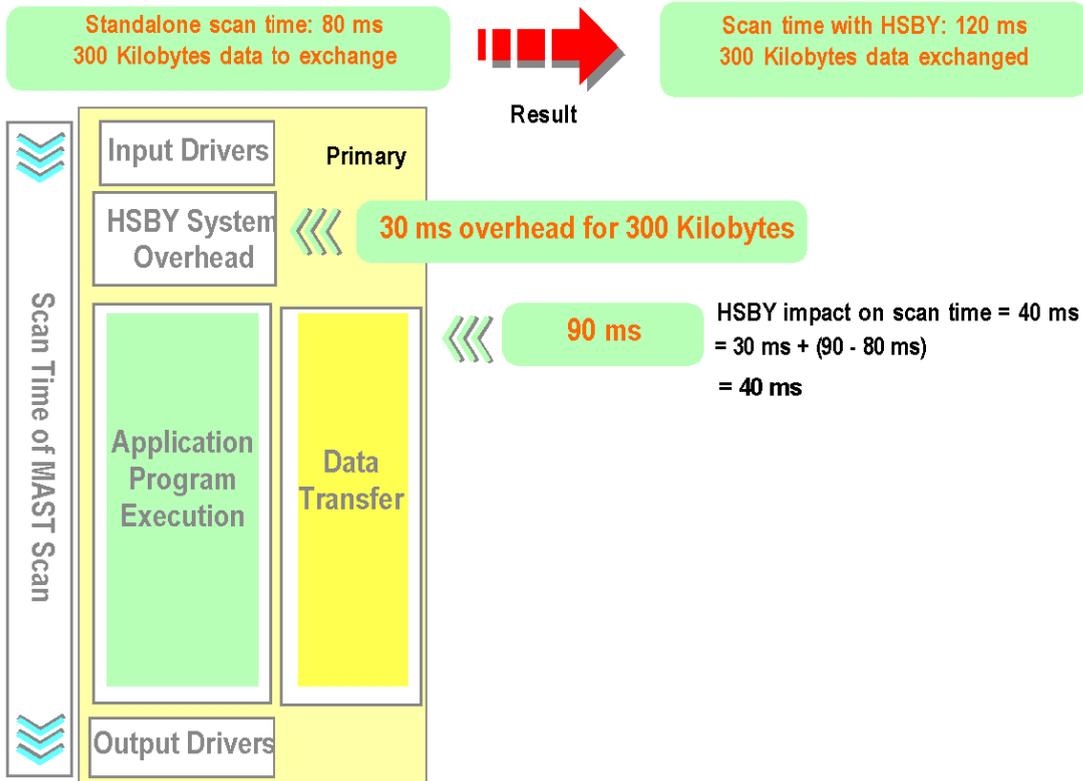


**NOTE:** Input and Output driver scan time depends on type of I/O and number of I/O. It's immaterial compared to the total scan time.

Example #2

- Application program execution time in standalone Premium PLC: 80 ms
- Database size: 300 Kilobytes

The following illustration displays the example #2:



---

# Chapter 5

## Switchover and Swap in Premium Hot Standby

---

### Purpose

This chapter describes the performance, the latencies of a Switchover event and a presentation of a Swap event in a Premium Hot Standby system.

### What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Ethernet Service Switchover Latencies	112
In-rack I/O Switchover Latencies	113
Presentation for Swapping in Premium Hot Standby	114
Operating recommendations for Swapping in a Premium Hot Standby system	115

## Ethernet Service Switchover Latencies

### Description

The following table details the typical and maximum delays likely to be encountered in reestablishing Ethernet services during a Switchover event:

Service	Maximum Swap Time
Swap IP Addresses	500 ms
I/O Scanning	500 ms + 1 initial cycle of I/O scanning
Client Messaging	Max swap over = 1.2 seconds (1.2 seconds includes 500 ms swap IP address, plus reestablishing network connections and device response time)Excludes Gateway and Bridges
Server Messaging	500 ms + the time required for the client to reestablish its connection with the server <sup>(1)</sup> (After switch over delay blocks 500 ms)
HTTP Server	500 ms + the time required for the client to reestablish its connection with the server <sup>(1)</sup>
<sup>(1)</sup> The time the client requires to reconnect with the server depends on the client's communication loss timeout settings.	

**NOTE:** The first scan of the new primary is based on the Inputs and Outputs transferred from the old Primary, and not on the current values of the IO.

## In-rack I/O Switchover Latencies

### Description

The Switchover time is the time between the last update of an output by the old Primary and the first update of the same output by the new Primary.

The following table shows the Switchover time for In-rack I/O:

Switchover event on the Primary	Average time to switchover on X-Bus I/O
Stop, Halt, or disconnection of the ETY-sync link if I/O Scanning is enabled on the Monitored ETYs	1.5 MAST time
Interruption of supply power or CPU-sync link disconnection, ETY-sync link if no I/O Scanning is configured	Watchdog time + 1.5 MAST time

**NOTE:** The Watchdog value that you configure in your Premium Hot Standby application has a direct impact on the Switchover time (in case of an interruption of the supply power to the Primary CPU or disconnection of the CPU-sync link).

**NOTE:** Depending on the Watchdog time you set, the maximum swap times for Ethernet services (see previous page) can be exceeded.

## Presentation for Swapping in Premium Hot Standby

### Definition

In a redundant configuration:

- a Switchover happens in reaction to an internal or external error detected in the PLC. It results in the loss of the formerly Primary PLC and the change of state of the Standby PLC to the Primary PLC.
- a Swap is a voluntarily action (usually part of the Unity Pro application) that consists in exchanging roles of the Primary and the Standby PLCs. For example, if PLC A is Primary and PLC B is Standby at time 0, then, at time 1, after a Swap has been performed, PLC A has become Standby and PLC B has become Primary.

### Conditions of use of swaps

Performing periodic swaps in a Premium Hot Standby system is a particular functionality that must be used only for very specific needs (testing or validation for instance) and under a very limited period of time.

Schneider Electric does not recommend using Swaps at production time, as the repetitive use of this feature may lead to PLC instability.

In any case, the minimum amount of time between two consecutive swaps must never be less than 15s.

## WARNING

### UNEXPECTED EQUIPMENT OPERATION

The HSBY\_SWAP ([see page 194](#)) DFB must be called for the purpose of testing the application only.

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

---

## Operating recommendations for Swapping in a Premium Hot Standby system

### Overview

The communication Function Block (Com FB) manager is not transferred when a Swap or a Switchover occurs. A consequence of this functionality (Swap) is that the Com FB which have been launched in the Primary CPU just before the Swap, can no longer be managed when this CPU becomes a Standby CPU.

The communication resources are not cancelled automatically. After a time all resources can be consumed. So after several Switchovers, we have the possibility to get a system without free resources.

Follow the rules provided here to reduce the possibility of consuming all resources due to Swaps and Switchovers.

These rules are common to the following Com FB, for all protocols and all networks (serials link, modbus TCP...)Switchover

- DATA\_EXCH
- INPUT\_BYTE
- INPUT\_CHAR
- OUT\_IN\_CHAR
- OUT\_IN\_MBUS
- PRINT\_CHAR
- RCV\_TLG
- READ\_ASYN
- READ\_GDATA
- READ\_VAR
- SEND\_REQ
- SEND\_TLG
- UNITE\_SERVER
- WRITE\_ASYN
- WRITE\_GDATA
- WRITE\_VAR

### Rules for the Management of the Exchange Management Table

Define a table of 4 words which are used to manage each instance of Com FB.

The Management Parameter consists of 4 contiguous words, as described below:

	Order of the word	Most Significant Byte	Least Significant Byte
Data managed by the system	Management_Param[0]	Exchange number	Activity Bit (see below)
	Management_Param[1]	operation report	communication report
Data managed by the user	Management_Param[2]	Block timeout. Values include: <ul style="list-style-type: none"> <li>● 0 = infinite wait</li> <li>● other values = timeout x 100 ms, for example:                             <ul style="list-style-type: none"> <li>● 1 = 100 ms</li> <li>● 2 = 200 ms</li> </ul> </li> </ul>	
	Management_Param[3]	Length of data send/receive parameter: <ul style="list-style-type: none"> <li>● Input (before send): length of the sending FB parameter</li> <li>● Output (after receive): length of the receiving Data parameter</li> </ul>	

**Activity Bit:** This bit indicates the execution status of the communication function. It is set to 1 when launched, and returns to 0 when the execution is complete. It is the first bit of the first element of the table.

The following is a description of the mandatory data to take care of in the implementation of the Com FB:

- Location of the 4 managements words:
  - It is mandatory that all the Com FB located in the first section of the application must have their 4 management words localized in the %MW between 0 and 99 (Non-Transfer Area [\(see page 160\)](#)).
  - For the other sections, it is recommended that the 4 management words of the Com FB are located in the Non-Transfer Area.
- Definition of the Timeout: The timeout value of the Com FB must be reinitialized each time that a Function Block is used.

The following is a description of the Com EF during a Swap or after a Switchover:

- Management of the Com EF during a Swap: CANCEL function should be used before using the Swap function. When you perform a Swap, the MAST task is executed and the Swap is effective at the end of the running task. So the EF should be canceled before the Swap.
- Management of the Com EF after a Switchover: CANCEL function should be used after a switchover occurs. When a Switchover occurs in a Hot Standby system, the MAST task stops immediately
  - If the Com EF management words can be localized in the Non-Transfer Area of any other applications sections than the first one.  
You have to detect that a Switchover occurred in order to cancel all the Com FB which have been activated in all the sections (except the first one). This code must be localized in the first section of the application in order to be activated in the two CPU (just after a Swap). If the customers does not cancel the Com FB, the correspondent resources will be free following the timeout delay.
  - If it is not possible to localize the management words of Com EF in the Non-Transfer Area, each CPU will inherit the management words which will not match with their local FB manager. Customer has to detect that a new Switchover occurs in order to reset the Activity Bit of all Com FB which were set before the Switchover in all the sections of the application (except the first one). This code must be localized in the first section of the application in order to be activated in the two CPU (just after a Switchover).

### Rules for the Management of Serial link in case of Swap

When you cancel a Com FB dedicated to a serial link, the Com EF is immediately killed in the CPU, but the request still activated in the coupler TSXSCY21600. It means that if a Primary CPU starts a `PRINT_CHAR`, and if a Swap occurred during the transmission of the frame, the TSXSCY21601 controlled by the old Primary CPU will complete the transmission of the frame, as the CPUs will Swap in parallel. So if the new Primary CPU performed a `PRINT_CHAR` on its TSXSCY21601 coupler, a collision will appear on the medium.

To avoid a collision on the medium, it is recommended to delay the transmission of data on a serial link by the new Primary CPU for a time equal to  $2 \times T_{\text{MAST}}$  + the maximum duration of the frame you need to transmit.

Management of the detected Error Code when using ETY Modules: When you perform a Com FB to a remote device behind an Ethernet coupler, the behavior of the coupler depends if the device is present or not:

- If the device is present the ETY forward the request to the device. The device answers to the ETY that receives the answer. The response is received by the CPU after a delay and the program is ready to arm a new request.
- If the targeted device is absent, the local ETY detects this by its own ARP table and does not push the request to the network. It immediately sends an acknowledge to CPU with a code 16#12FF (no free resources on the remote devices). And the CPU is ready to re-arm a new Com FB. The CPU is able to sends a new request immediately (each Mast task) to the ETY. This can affect the traffic between the CPU and the ETY. A good practice consist to detect this specific acknowledge code (16#12FF) in order to delay (500ms) the reactivation of COM function bloc.



---

# Chapter 6

## Compatibility with PL7 Systems

---

### Purpose of this Chapter

This chapter provides information regarding the compatibility of existing PL7-based systems with the Premium Hot Standby, and introduces the PL7-Unity Pro Converter.

### What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Considerations When Upgrading From PL7	120
Using the PL7-Unity Pro Converter	121

## Considerations When Upgrading From PL7

Systems created for Premium PLCs using PL7 may be converted to the Unity Pro-Premium Hot Standby platform and gain the benefits of automatic PLC redundancy and I/O redundancy, including Ethernet address swapping on Switchover.

In the case of upgrades from a PL7 Warm Standby system to a Premium Hot Standby system, consider the implementation of existing Fipio devices, and remember that the exchange of data and state information between the Primary and Standby PLCs is now automatic. This automation means that in most cases the information transfer necessary to support redundant In-rack I/O does not require user programming. This is different from and an improvement on the PL7 Warm Standby system. The following bullets summarize some of these considerations:

- You can only connect Fipio devices to a Premium Hot Standby system through an Ethernet-to-Fipio gateway. You can implement a gateway of this type using a stand-alone Premium PLC with integrated Fipio and Ethernet ports, or an integrated Fipio port and an Ethernet communication module.
- Because the database transfer necessary to support Hot Standby redundancy is now largely automatic, many of the function blocks and expert functions used are no longer applicable, and if used, will prevent your application from compiling and running. The obsolete functions are:
  - Derived Function Blocks (DFBs) specifically used for data exchange:
    - Ha\_db\_basic
    - Ha\_db\_cycle\_opt
    - Ha\_db\_size\_opt
  - Expert Functions (EFs) used to provide data and context exchanges in Sequential Function Chart (SFC) / Grafcet programming:
    - Get\_stat\_chart
    - Set\_stat\_chartFor more information on Grafcet programming, refer to the *Startup Guide for Unity Pro*, reference 35008402.
- The following legacy function blocks are specific to Premium PL7 Warm Standby behavior, and are inconsistent with Hot Standby redundancy and must not be used:
  - PL7\_COUNTER
  - PL7\_DRUM
  - PL7\_MONOSTABLE
  - PL7\_REGISTER\_32
  - PL7\_REGISTER\_255
  - PL7\_TOF
  - PL7\_TON
  - PL7\_TP
  - PL7\_3\_TIMER

## Using the PL7-Unity Pro Converter

In addition to the general considerations noted above, you should be aware of the PL7-Unity Pro Converter tool. This application will, in some cases, be able to accomplish the conversion of a Warm Standby PL7 program to one that will function on Unity Pro and a Hot Standby PLC.

**NOTE:** The PL7-Unity Pro Converter tool will not convert the obsolete functions (DFBs and EFs) noted above. This can render your application program non-functional. The PL7-Unity Pro Converter will notify you of any features that are not successfully converted by text displayed in the Unity Pro watch window. Refer to the *Unity Pro Application Converter User Manual*, reference 35006148, for further information on this tool.

**NOTE:** While the PL7-Unity Pro Converter provides a good foundation for beginning your program conversions, it cannot be comprehensive. In all cases, a thorough code review and rigorous testing will be required, particularly to integrate with new Premium Hot Standby features, such as the handling of Ethernet I/O to provide redundancy. Refer to the *Unity Pro Program Languages and Structure Reference Manual*, reference 35006144, and the *Unity Pro 3.1 Operating Modes manual*, reference 33003101, for further details on programming your system using Unity Pro.



---

# Part II

## Configuration and Use

---

### Purpose

This Part describes installation and usage considerations specific to the Premium Hot Standby system.

This section does not describe the basic physical installation of the Premium Hot Standby CPU, rack, power supply, or associated hardware. It also does not provide related information such as operating limits, grounding, electromagnetic compatibility, or other environmental considerations. For details on these topics, please refer to *Premium and Atrium Using Unity Pro Processors, Racks and Power Supply Modules Implementation Manual*, reference 35010524, and *Grounding and Electromagnetic Compatibility of PLC Systems Basic Principles and Measures User Manual*, reference 33002439, located at [www.telemecanique.com](http://www.telemecanique.com).

The four chapters included in this Part are:

- Configuring in Unity Pro
- Programming/Debugging
- Operating
- Maintaining

**NOTE:** Read and understand the following documents before attempting the procedures in this manual:

- *Premium and Atrium Using Unity Pro Processors, Racks and Power Supply Modules Implementation Manual*, reference 35010524.
- *Grounding and Electromagnetic Compatibility of PLC Systems Basic Principles and Measures User Manual*, reference 33002439.

Both manuals can be found at [www.schneider-electric.com](http://www.schneider-electric.com).

### What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
7	Configuring in Unity Pro	125
8	Programming/Debugging	169
9	Operating	199
10	Maintaining	213



---

# Chapter 7

## Configuring in Unity Pro

---

### Overview

This chapter describes configuring the Premium Hot Standby PLCs.

### What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
7.1	Configuring a System with the Unity Pro Tabs and Dialogs	126
7.2	Configuring TSX ETY 4103/5103 Modules	146
7.3	Configuring Registers	158

# Section 7.1

## Configuring a System with the Unity Pro Tabs and Dialogs

### Purpose

This section describes configuring the specific features of the Premium Hot Standby CPUs TSX H57 24M or TSX H57 44M.

 <b>WARNING</b>
<p><b>UNINTENDED EQUIPMENT OPERATION</b></p> <p>Do not attempt to modify your Premium Hot Standby PLC or In-rack module configurations in Unity Pro while your system is operational.</p> <p><b>Failure to follow these instructions can result in death, serious injury, or equipment damage.</b></p>

For configuring other standard features, refer to the *Unity Pro 3.1 Operating Modes manual*, reference 33003101.

### What Is in This Section?

This section contains the following topics:

Topic	Page
Introducing Unity Pro	127
Accessing the Base Configuration	128
Using the Overview Tab	129
Using the Configuration Tab	130
Using the Animation Tab and PLC Screen Dialogs	132
Using the Premium Hot Standby Tab	138
Configuring In-rack I/O	139
Configuring the PCMCIA Cards	140
Swapping Network Addresses at Switchover	142

## Introducing Unity Pro

### Overview

Unity Pro is a Software package for programming Modicon Premium, Modicon Quantum, Modicon M340, and Modicon Atrium PLCs.

It provides several tools for application development including:

- Project browser
- Configuration tool
- Data editor
- Program editor

The configuration tool is used to:

- Create, modify, and save the elements used to configure the PLC station
- Set up the application-specific modules
- Diagnose the modules configured in the station
- Control the number of application-specific channels configured in relation to the capacities of the processor declared in the configuration
- Assess processor memory usage

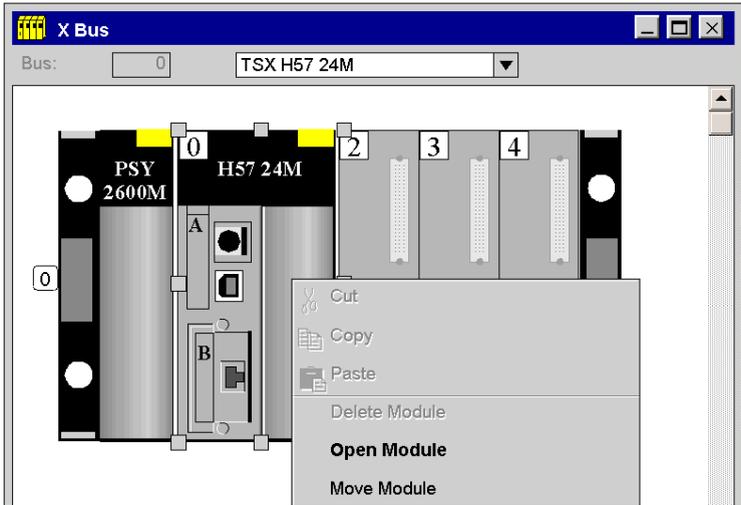
The minimum version of Unity Pro to be used to program a Hot Standby Premium Monorack application is Unity Pro V3.1 with SP1 (Service Pack 1).

The minimum version of Unity Pro to be used to program a Hot Standby Premium Multirack is Unity Pro V5.0 with HF1 (Hot Fix 1).

## Accessing the Base Configuration

### Accessing with Unity Pro

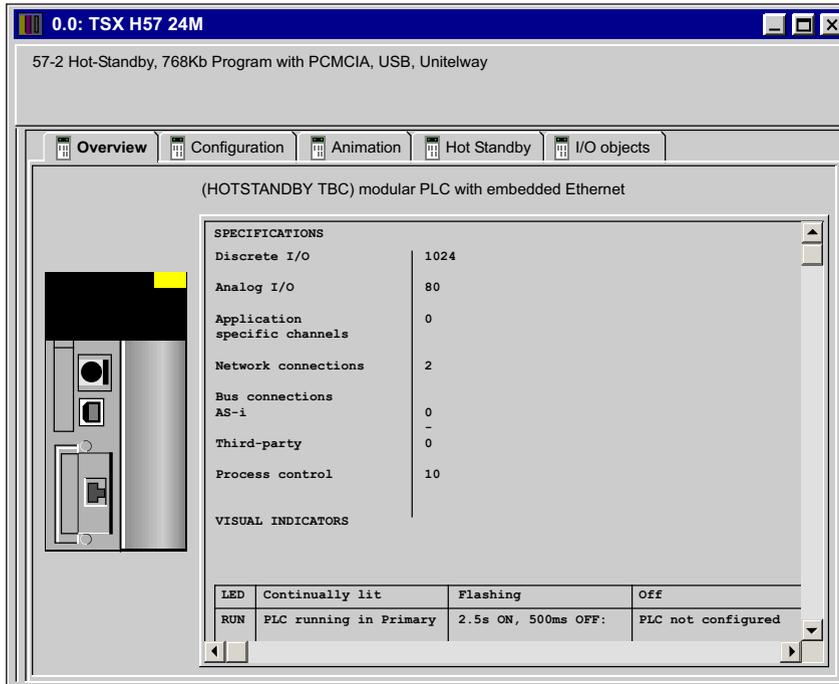
After starting Unity Pro, go to the X-Bus in the Structural View of the Project Browser.

Step	Action
1	Open the X-Bus configuration editor either by double-clicking on the X-Bus or by selecting the X-Bus and executing right-click Open. A graphical representation of the local bus appears in the configuration editor.
2	Select the Premium Hot Standby CPU module and right-click. The context menu appears.
	 <p>The screenshot shows the 'X Bus' configuration window. At the top, it displays 'Bus: 0' and 'TSX H57 24M'. The main area shows a rack of modules: a 'PSY 2600M' power supply, a 'H57 24M' CPU module (slot 0), and three empty slots (2, 3, 4). A context menu is open over the CPU module, listing options: Cut, Copy, Paste, Delete Module, <b>Open Module</b>, and Move Module.</p>
3	Select Open Module. The editor appears. The Configuration tab is default.
4	Choose one of these tabs: <ul style="list-style-type: none"> <li>● Overview</li> <li>● Configuration</li> <li>● Animation</li> <li>● Hot Standby</li> <li>● I/O Objects</li> </ul>

## Using the Overview Tab

### Viewing

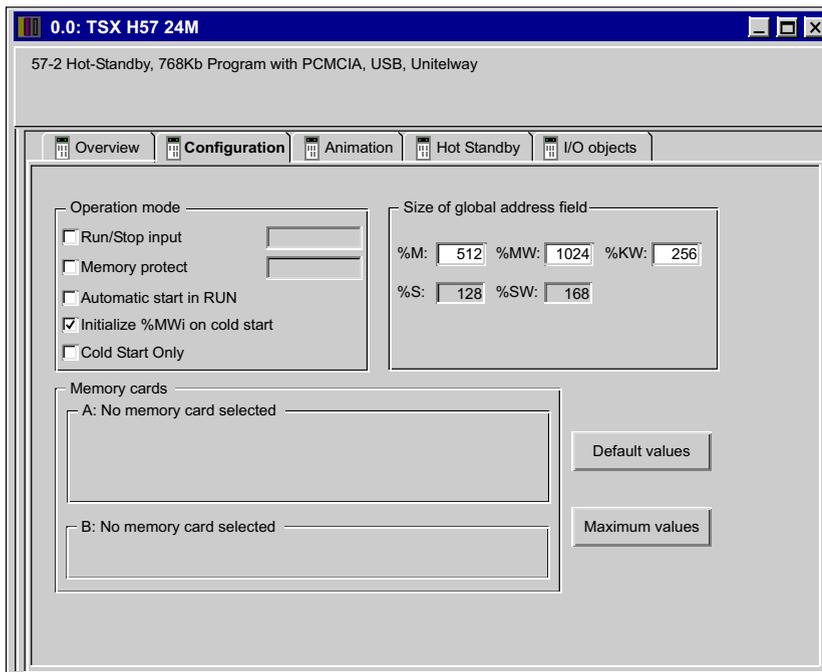
The read-only Overview tab of the editor displays detailed information about the module's specifications.



## Using the Configuration Tab

### Viewing the Configuration tab

Change values using the Configuration tab of the editor.



### Description of the Configuration tab

**NOTE:** Before determining these values, read and understand the *Unity Pro 3.1 Operating Modes manual*, reference 33003101.

Item	Option	Value	Description
Operation Mode	Run/Stop input	Note 3	Determines the operating condition during Cold Start
	Memory protect	Note 3	
	Automatic start in Run	Note 3	
	Initialize %MWi on cold start	Note 3	
	Cold Start Only	Note 3	If you wish, enable the Cold Start Only ( <a href="#">see page 131</a> ) feature.

Item	Option	Value	Description
Memory Cards	A:	N/A	Displays the configuration in the PCMCIA Slots
	B:	N/A	
Size of global address field	%M	Note 1	Size of the different memory areas <b>Note:</b> The values for %MW have to be divisible by 8, or Unity Pro will reject the value.
	%MW	Note 1	
	%KW	Note 1	
	%S	Note 2	
	%SW	Note 2	
	Default value	N/A	Permits selection of the default value: %M/%KW
	Maximum value	N/A	Permits selection of the maximum number: %M/%KW
<p><b>Note 1:</b> Enter the appropriate values based on your configuration, including the number of inputs and outputs in your system and the requirements of your application. Approximately one %M bit will normally need to be assigned for each %I and %Q object.</p> <p><b>Note 2:</b> The values cannot be selected.</p> <p><b>Note 3:</b> The user can select these options as required for the application.</p>			

### Cold Start Only

If checked, this option forces the cold start (see *Unity Pro, Program Languages and Structure, Reference Manual*) of the application, instead of the normal warm start (see *Unity Pro, Program Languages and Structure, Reference Manual*).

By default, the Cold Start Only option is unchecked.

The Cold Start Only option is only supported since PLC V2.7.

An application using this functionality will not be:

- downloadable on a PLC with a previous version,
- executable on a PLC with a previous version,
- usable with Unity Pro V4.0 or lower.

**NOTE:** The Cold Start Only check box is present only if the current selected PLC can support it.

## Using the Animation Tab and PLC Screen Dialogs

### Accessing the PLC Screen Dialogs

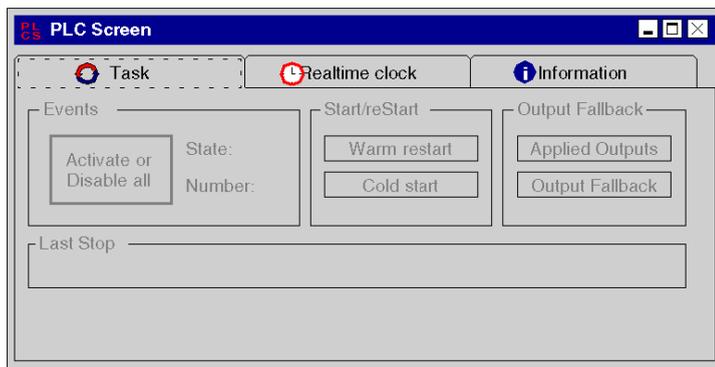
To access the Task, Realtime clock, and Information tabs of the Unity Pro Animation tab,

Step	Action
1	Select the Animation tab.
2	The PLC screen tab appears automatically.

**NOTE:** The dialogs illustrated in this section are depicted as they appear when Unity Pro is not connected to the PLC. When Unity Pro is connected to a PLC, the information displayed in these tabs changes.

### Viewing the Task Tab

Unity Pro Task tab dialog:



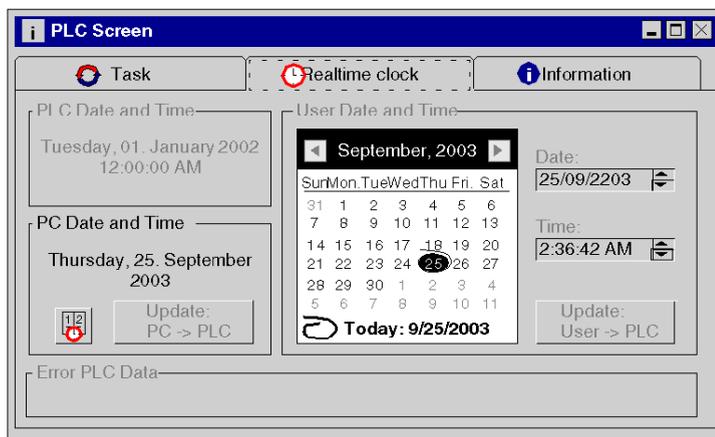
## Task Tab Description

Description of the Task tab:

Item	Option	Value	Description
Events	State:	xxx	Status information of events that may be accessed or viewed when the PLC is running and when Unity Pro is physically linked to the PLC and in "connected mode".
	Number:	xxx	Displays the number of events since the last time the application was initialized.
	Activate or Disable all	Toggle button	This button controls whether events are processed. By pushing this button you can disable all event tasks. Therefore, if your application has a task that detects a change in %I4.0 and then takes some action based on this event, you can disable that behavior by selecting this button.
Start/Restart	Warm Start	Toggle button	To initialize Warm Start
	Cold Start	Toggle button	To initialize Cold Start
Output fallback	Applied Outputs	N/A	To stop the Fallback mode
	Output Fallback	N/A	To switch the outputs into Fallback mode
Last Stop	Read only	<ul style="list-style-type: none"> <li>● Day</li> <li>● DD/MM/YY</li> <li>● Time</li> </ul>	Indicates the day, date, time, and cause of the last controller stop

## Viewing the Realtime Clock Tab

Unity Pro Realtime clock tab dialog:



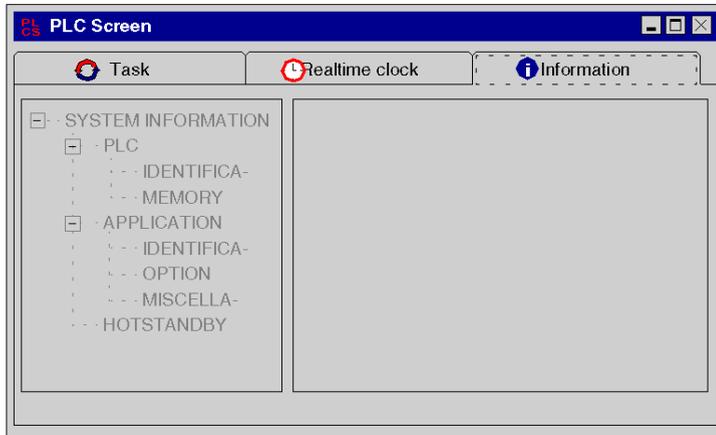
### Realtime Clock Tab Description

Description of the Realtime clock tab:

Item	Option	Description
PLC Date and Time	Read only	Indicates the current PLC date and time
PC Date and Time	Update PC->PLC	Updates the PLC with the PC system time
User Date and Time	Update User->PLC	Updates the PLC with the time set by the user
Error PLC Data	Read only	Some application events, such as the detection of a "divide by zero" error in your application program, insert a diagnostics number into a %SW System Word. When this occurs, the diagnostic number can be read here.

### Viewing the Information Tab

Unity Pro Information tab dialog:



## Information Tab Description

**NOTE:** The Hot Standby system information displayed in these tables reflects the values present in the %SW61 System Status Word.

Description of the Information tab:

Item	Option	Fields Displayed When Selected	Description
System Information	PLC / Identification	PLC Range	Only available online (when PLC is connected to PC and Unity Pro is in "Connected Mode").
		Processor name	
		Processor version	
		Hardware ID	
		Network address	
	PLC / Memory	RAM CPU size	
	Application / Identification	Name	
		Creation Product (will display "Unity Pro" followed by version number)	
		Creation Date	
		Modification Product (will display "Unity Pro" followed by version number)	
		Modification Date	
		Application Program Version (based on incrementing system counter)	
		Signature	
System Information	Application / Option	Upload Information (the information necessary to upload a binary PLC application to Unity Pro and convert it back to source code for modification)	
		Comments	
		Animation Table	
		Section Protection (lock application sections to prevent modification)	
		Application Diagnostic	
	Application / Miscellaneous	Forced Bits	

Item	Option	Fields Displayed When Selected	Description
System Information	Hot Standby	PLC Hot Standby Status	Only available online (when PLC is connected to PC and Unity Pro is in "Connected Mode".
		Peer PLC Hot Standby Status	
		Logic Mismatch between PLC and Peer PLC	
		PLC Name	
		CPU-Sync Link Not Connected	
		Main Processor OS version Mismatch	
		Coprocessor OS version Mismatch	
		One or more of the In-rack ETY modules does not have the minimum firmware version, v4.0.	
		One or more of the In-rack ETY modules has a firmware version that does not match that of its counterpart ETY module. This diagnostic result is possible even if all ETY modules have firmware version v4.0 and up.	
		TCP/IP and Modbus Addresses	
Hot Standby Entire System State			

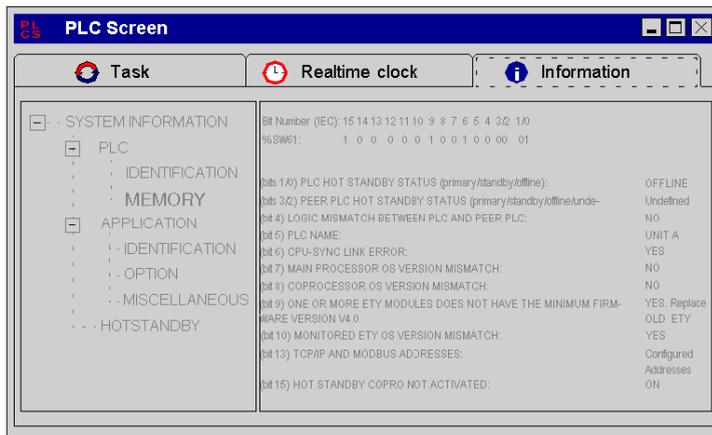
The following table presents the values that may appear in the Information Tab for the "HOTSTANDBY" category:

Bits	Line Title	String Displayed
1 and 0	PLC Hot Standby Status	Values= (0 and 1): Offline mode
1 and 0	PLC Hot Standby Status	Values= (1 and 0): Primary mode
1 and 0	PLC Hot Standby Status	Values= (1 and 1): Standby mode
3 and 2	Peer PLC Hot Standby Status	Values= (0 and 0): Undefined mode
3 and 2	Peer PLC Hot Standby Status	Values= (0 and 1): Offline mode
3 and 2	Peer PLC Hot Standby Status	Values= (1 and 0): Primary mode
3 and 2	Peer PLC Hot Standby Status	Values= (1 and 1): Standby mode
4	Logic Mismatch between PLC and Peer PLC	<ul style="list-style-type: none"> <li>● Value=0: No</li> <li>● Value=1: Yes</li> </ul>
5	PLC Name	<ul style="list-style-type: none"> <li>● Value=0: Unit A (The string "Unit A" is assigned to the Hot Standby controller with the lowest MAC address).</li> <li>● Value=1: Unit B</li> </ul>
6	CPU-sync link Not Connected	<ul style="list-style-type: none"> <li>● Value=0: No</li> <li>● Value=1: Yes</li> </ul>
7	Main Processor OS version Mismatch	<ul style="list-style-type: none"> <li>● Value=0: No</li> <li>● Value=1: Yes</li> </ul>

Bits	Line Title	String Displayed
8	Coprocessor OS version Mismatch	<ul style="list-style-type: none"> <li>● Value=0: No</li> <li>● Value=1: Yes</li> </ul>
9	One or more ETY modules does not have the minimum firmware version, v4.0.	<ul style="list-style-type: none"> <li>● Value=0: No. All ETY have the minimum required version.</li> <li>● Value=1: Yes. Replace old ETY.</li> </ul>
10	Monitored ETY OS version Mismatch	<ul style="list-style-type: none"> <li>● Value=0: No</li> <li>● Value=1: Yes</li> </ul>
11	Not Used	N/A
12	Not Used	N/A
13	TCP/IP and Modbus Addresses	<ul style="list-style-type: none"> <li>● Value=0: This controller's modules are using the IP and Modbus addresses that were configured in Unity Pro ( n ).</li> <li>● Value=1: This controller's modules are using the IP and Modbus addresses that were configured in Unity Pro + 1 ( n+1 ).</li> </ul>
14	Not Used	N/A
15	HOT STANDBY COPRO NOT ACTIVATED	<ul style="list-style-type: none"> <li>● Value=0: Off. Coprocessor has not started correctly and no CPU-sync link has been established.</li> <li>● Value=1: On. The Copro is active and the CPU-sync link has been established.</li> </ul>

### Viewing the Information Tab in connected mode

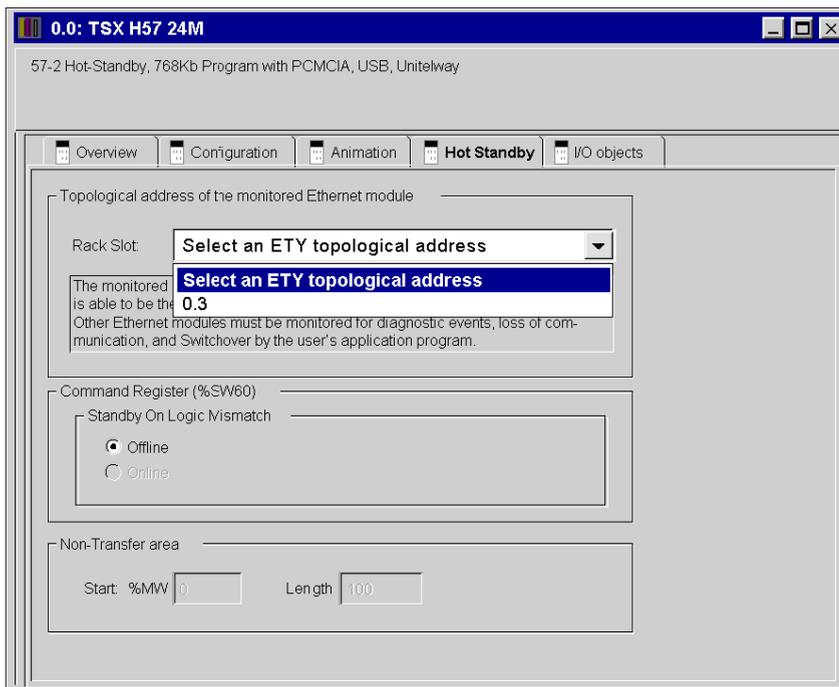
The following illustration displays Unity Pro Information tab dialog in connected mode:



## Using the Premium Hot Standby Tab

### Viewing the Hot Standby Tab

Configure Hot Standby values in the Hot Standby tab of the Unity Pro editor:



### Hot Standby Tab Description

Description of the Hot Standby tab:

Item	Option	Description
Topological address of the monitored Ethernet module	Rack Slot	This combo box is filled by the existing addresses of ETY cards
Command Register	Standby On Logic Mismatch	When Unity Pro is connected to a Premium Hot Standby PLC, the Online option is not available. Unlike the Quantum Hot Standby controller, a Premium Hot Standby controller cannot act as the Standby if there is an application program mismatch.
Non-transfer area	Start: %MW	These are read-only fields, indicating that the values in the memory range %MW0 - %MW99 are never transferred between the Hot Standby PLCs.
	Length	

## Configuring In-rack I/O

### How to configure In-rack I/O

Before attempting to configure In-rack I/O (Discrete or Analog), read and understand the User Manuals for those modules.

Output fallback modes need to be configured properly in order to maintain high availability in a Hot Standby system. Outputs should usually be configured to fall back to their present state to help avoid unintended equipment operation during the switchover period.

More specifically, in the case of discrete outputs, improper configuration can result in these outputs becoming locked in the state they last held when one of the PLCs becomes inoperative. To prevent discrete outputs from freezing when one of the controllers enters an inoperative state, the output modules using positive logic should use fallback mode 0 while those using negative logic should use fallback mode 1.

### **WARNING**

#### **UNINTENDED EQUIPMENT OPERATION**

- Configure your output module fallback modes to prevent changes in output states during Switchover.
- Use fallback mode 0 for all positive logic discrete output modules.
- Use fallback mode 0 when output modules are cabled in parallel using ABE7 ACC1 connection blocks.
- Use fallback mode 1 for all negative logic discrete output modules.

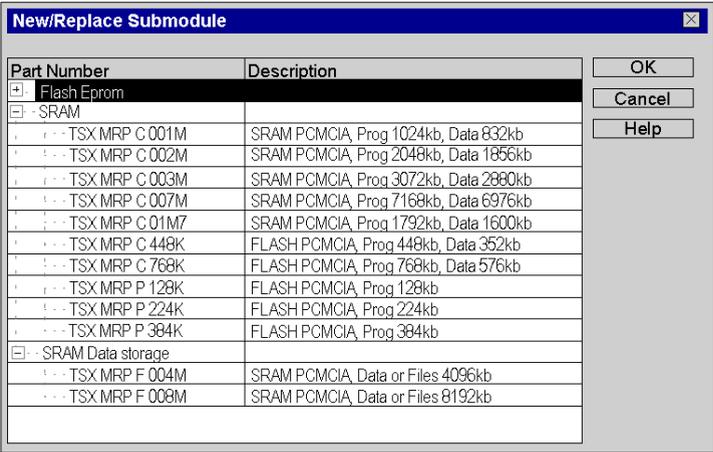
**Failure to follow these instructions can result in death, serious injury, or equipment damage.**

## Configuring the PCMCIA Cards

### Configuring with Unity Pro

Allocating memory to the memory card:

Step	Action
1	If not opened, open the X-Bus configuration editor.
2	Go to the local bus in the Structural View of the Project Browser.
3	Open the local bus either by double-clicking on the X-Bus or by selecting the X-Bus and executing right-click Open. A graphical representation of the local bus appears.
4	Point to and select either PC Card A (slot 1) or PC Card B (slot 2). <div style="text-align: center;"> <p>1 Memory configuration of the PCMCIA card 1 2 Memory configuration of the PCMCIA card 2</p> </div>

Step	Action																																
5	<p>Double-click or right-click either PCMCIA card. The New/Replace Submodule dialog appears.</p>  <p>The screenshot shows the 'New/Replace Submodule' dialog box. It features a table with the following data:</p> <table border="1" data-bbox="367 326 943 711"> <thead> <tr> <th>Part Number</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Flash Eprom</td> <td></td> </tr> <tr> <td>-SRAM</td> <td></td> </tr> <tr> <td>TSX MRP C 001M</td> <td>SRAM PCMCIA, Prog 1024kb, Data 832kb</td> </tr> <tr> <td>TSX MRP C 002M</td> <td>SRAM PCMCIA, Prog 2048kb, Data 1856kb</td> </tr> <tr> <td>TSX MRP C 003M</td> <td>SRAM PCMCIA, Prog 3072kb, Data 2880kb</td> </tr> <tr> <td>TSX MRP C 007M</td> <td>SRAM PCMCIA, Prog 7168kb, Data 6976kb</td> </tr> <tr> <td>TSX MRP C 01M7</td> <td>SRAM PCMCIA, Prog 1792kb, Data 1600kb</td> </tr> <tr> <td>TSX MRP C 448K</td> <td>FLASH PCMCIA, Prog 448kb, Data 352kb</td> </tr> <tr> <td>TSX MRP C 768K</td> <td>FLASH PCMCIA, Prog 768kb, Data 576kb</td> </tr> <tr> <td>TSX MRP P 128K</td> <td>FLASH PCMCIA, Prog 128kb</td> </tr> <tr> <td>TSX MRP P 224K</td> <td>FLASH PCMCIA, Prog 224kb</td> </tr> <tr> <td>TSX MRP P 384K</td> <td>FLASH PCMCIA, Prog 384kb</td> </tr> <tr> <td>-SRAM Data storage</td> <td></td> </tr> <tr> <td>TSX MRP F 004M</td> <td>SRAM PCMCIA, Data or Files 4096kb</td> </tr> <tr> <td>TSX MRP F 008M</td> <td>SRAM PCMCIA, Data or Files 8192kb</td> </tr> </tbody> </table>	Part Number	Description	Flash Eprom		-SRAM		TSX MRP C 001M	SRAM PCMCIA, Prog 1024kb, Data 832kb	TSX MRP C 002M	SRAM PCMCIA, Prog 2048kb, Data 1856kb	TSX MRP C 003M	SRAM PCMCIA, Prog 3072kb, Data 2880kb	TSX MRP C 007M	SRAM PCMCIA, Prog 7168kb, Data 6976kb	TSX MRP C 01M7	SRAM PCMCIA, Prog 1792kb, Data 1600kb	TSX MRP C 448K	FLASH PCMCIA, Prog 448kb, Data 352kb	TSX MRP C 768K	FLASH PCMCIA, Prog 768kb, Data 576kb	TSX MRP P 128K	FLASH PCMCIA, Prog 128kb	TSX MRP P 224K	FLASH PCMCIA, Prog 224kb	TSX MRP P 384K	FLASH PCMCIA, Prog 384kb	-SRAM Data storage		TSX MRP F 004M	SRAM PCMCIA, Data or Files 4096kb	TSX MRP F 008M	SRAM PCMCIA, Data or Files 8192kb
Part Number	Description																																
Flash Eprom																																	
-SRAM																																	
TSX MRP C 001M	SRAM PCMCIA, Prog 1024kb, Data 832kb																																
TSX MRP C 002M	SRAM PCMCIA, Prog 2048kb, Data 1856kb																																
TSX MRP C 003M	SRAM PCMCIA, Prog 3072kb, Data 2880kb																																
TSX MRP C 007M	SRAM PCMCIA, Prog 7168kb, Data 6976kb																																
TSX MRP C 01M7	SRAM PCMCIA, Prog 1792kb, Data 1600kb																																
TSX MRP C 448K	FLASH PCMCIA, Prog 448kb, Data 352kb																																
TSX MRP C 768K	FLASH PCMCIA, Prog 768kb, Data 576kb																																
TSX MRP P 128K	FLASH PCMCIA, Prog 128kb																																
TSX MRP P 224K	FLASH PCMCIA, Prog 224kb																																
TSX MRP P 384K	FLASH PCMCIA, Prog 384kb																																
-SRAM Data storage																																	
TSX MRP F 004M	SRAM PCMCIA, Data or Files 4096kb																																
TSX MRP F 008M	SRAM PCMCIA, Data or Files 8192kb																																
6	Add or replace the desired memory.																																

## Swapping Network Addresses at Switchover

### Overview

The following material describes the handling of network addresses at Switchover.

### Handling TCP/IP Addresses at Switchover

When used in a Premium Hot Standby System, the Ethernet TCP/IP network modules TSX ETY 4103/5103 support address swapping at Switchover.

The HSBY ETY module, configured to I/O scan shared Ethernet I/O, supports IP Address swapping of SCADA/HMI systems, Ethernet I/O read/write, diagnostics, and PLC Switchover.

Prior to a Switchover event the Primary and Standby HSBY ETYs must be represented by one unique IP Address.

The following table presents the unique IP Address:

IP address for	System A in Primary mode	System B in Standby mode	System A in Standby mode	System B in Primary mode
	Before Switchover		After Switchover	
HSBY ETY 1	IP1	IP1 + 1	IP1 + 1	IP1
HSBY ETY 2	IP2	IP2 + 1	IP2 + 1	IP2
HSBY ETY 3	IP3	IP3 + 1	IP3 + 1	IP3
HSBY ETY 4	IP4	IP4 + 1	IP4 + 1	IP4

**NOTE:** All the ETY modules that are present in a Hot Standby PLC will swap the IP address at Switchover.

When used in a Premium Hot Standby System, the Ethernet TCP/IP network modules TSX ETY 4103 / 5103 support address swapping at Switchover. If a Premium Hot Standby ETY module initially has IP address xxx.xxx.xxn, then the PLC automatically assigns its counterpart ETY on the Standby rack an address of xxx.xxx.xxn+1. Therefore, to avoid IP address conflicts, do not assign consecutive IP addresses to the ETY modules on the Primary PLC's rack.

Also, do not assign any ETY module on the Primary rack the addresses xxx.xxx.254 or xxx.xxx.255. The latter address is reserved for TCP/IP broadcast messages, and is not managed at Switchover. The reason why xxx.xxx.254 is not assigned is that the counterpart ETY module on the Standby rack would automatically assume an IP address of xxx.xxx.255, and begin broadcasting to all other Ethernet devices on the same subnet.

## WARNING

### RISK OF UNINTENDED EQUIPMENT OPERATION

- Do not assign consecutive IP addresses to ETY modules in the Primary rack. This will cause duplicate IP addresses to be assigned to the ETY modules on the Standby rack, leading to unintended equipment operation.
- Do not assign the IP addresses xxx.xxx.254 or xxx.xxx.255 to any ETY module in a Premium Hot Standby system.

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

**NOTE:** If an address of xxx.xxx.254 or xxx.xxx.255 is assigned to the Monitored ETY, The system will not come online when the Run command is issued, and diagnostic indications will appear on the PLC and ETY LEDs. However, these addresses are not checked in the case of additional non-Monitored ETYs on the rack, so there is a potential for network conflicts with these modules.

**NOTE:** For more information on this topic, see *Ethernet Service Switchover Latencies*, [page 112](#).

### Handling Modbus Addresses at Switchover

When a Premium Hot Standby system is operating normally, and a TSX SCP 114 module has been configured in Unity Pro, the Modbus addresses will be:

- The TSX SCP 114 card associated with PLC A (the Primary controller) will be n. The value of n is set by the user in Unity Pro, and must be in the range 1 to 98.
- The TSX SCP 114 card associated with PLC B (the Standby controller) will automatically be assigned a Modbus address of n+1.

If a Switchover occurs, and the Standby PLC becomes Primary, the reassigned Modbus addresses are:

- The TSX SCP 114 card associated with PLC A (the old Primary / new Standby controller) will be n+1.
- The TSX SCP 114 card associated with PLC B (the new Primary controller) now has an address of n. The value of n is that set by the user in Unity Pro.

**NOTE:** Only the TSX SCP 114 Modbus card in the TSX SCY 21601 base supports Hot Standby address swapping in the event of a Switchover. The integrated Modbus master ports (Channel 0) on the TSX SCY 11601 and 12601 modules are not managed redundantly, and will not swap addresses if a Switchover occurs.

When a Switchover event occurs, there is a possibility that some Modbus messages will not reach the intended devices. Program your application to confirm that the station addressed on the Modbus link has received and responded correctly to a message before sending the next message. For example, this may involve buffering all outgoing Modbus commands, trapping the occurrence of a Switchover event, and using the buffered commands to check the values of the commanded registers for compliance. This programming must occur in the first section (Section 0) of your application program.

## **WARNING**

### **UNINTENDED EQUIPMENT OPERATION**

Create your application program to confirm the receipt and effect of Modbus commands issued just before and during a Switchover event.

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

## **WARNING**

### **UNINTENDED EQUIPMENT OPERATION**

- Do not assign consecutive slave addresses to Modbus modules in the Primary rack. This will cause duplicate IP addresses to be assigned to the corresponding Modbus modules on the Standby rack, leading to unintended equipment operation.
- Do not assign the slave address 98 to any Modbus module in a Premium Hot Standby system. The automatic assignment of the n+1 address to the Standby Modbus slave will not work in this event.

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

**NOTE:** Unlike the Monitored ETY's IP addresses, there is no error detection preventing the use of Modbus slave address 98.

Do not change the communications mode (Modbus, Uni-Telway, Character) of the TSX SCP 114 module while the Hot Standby system is operational. The Primary controller does not update the Standby controller when this TSX SCP 114 configuration information changes. If this information is changed when the Hot Standby system is operating, and then a Switchover event occurs, the communications mode can change unexpectedly.

 **WARNING****UNINTENDED EQUIPMENT OPERATION**

Do not change the communications mode (Modbus, Uni-Telway, or Character) of the TSX SCP 114 module while your Hot Standby system is operating.

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

**NOTE:** The T\_COM\_MB IODDT (I/O Derived Data Type) may cause unpredictable behavior when used to query the high byte of the communications PROTOCOL variable. Only the low byte should be queried using this function.

 **WARNING****UNINTENDED EQUIPMENT OPERATION**

When using the T\_COM\_MB IODDT function to determine the Modbus protocol in use, do not query the high byte of the PROTOCOL variable.

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

## Section 7.2

### Configuring TSX ETY 4103/5103 Modules

---

#### Purpose

This material describes configuring TSX ETY 4103/5103, Premium Ethernet modules, using Unity Pro. For a more thorough description of the two ETY modules (hardware installation, functions, configuration, programming, Ethernet language objects), see the *Premium and Atrium using Unity Pro Ethernet Network User Manual 35006192*.

**NOTE:** The Global Data Publish/Subscribe Service is disabled in Unity Pro when configuring Premium Hot Standby systems.

#### What Is in This Section?

This section contains the following topics:

Topic	Page
Overview of Premium Hot Standby TSX ETY	147
ETY Operating Modes and Premium Hot Standby	150
IP Address Assignment	153
Network Effects of Premium Hot Standby	155

---

## Overview of Premium Hot Standby TSX ETY

### Please note

Because the user can configure several ETY modules in each PLC, the Monitored ETY modules that are dedicated to the ETY-sync link (only one ETY module in each PLC) must be designated using the "Topological address of the Monitored ETY module" option in Unity Pro.

The Monitored ETY is the ETY module that manages the ETY-sync link.

### Description of the Hot Standby Solution

Hot Standby-ready ETY modules (with firmware version 4.0 or above) allow automatic IP address swapping.

The TSX ETYs coordinate the swapping of IP addresses. After closing both the client and the server connections, each TSX ETY sends a swap UDP message to its peer TSX ETY unless a loss of power or a similar event prevents this from happening. The sending TSX ETY then waits for a specified time-out for the peer swap of UDP messages. Either after receiving the messages or after a time-out, the TSX ETY changes its IP address.

**NOTE:** In general, network switches offer increased performance over hubs, particularly under heavy communications loads, due to their traffic routing and filtering capabilities. It is recommended that these switches (not hubs) be used to connect the TSX ETYs to each other or to the network. Schneider Electric offers the ConneXium range of Industrial Ethernet switches; please contact a local sales office for more information.

The TSX ETY waits for either a change in the local controller's Hot Standby state or the swap of UDP messages. Then the TSX ETY performs one of two Hot Standby actions.

If the TSX ETY:

1. Detects that the new Hot Standby state is either Primary or Standby:  
The TSX ETY changes the IP address
2. Receives a swap UDP message:  
The TSX ETY transmits a Swap UDP message and swaps the IP address

All client/server services (I/O Scanner, Messaging, FTP, SNMP, and HTTP) continue to run after the Switchover once the transition to the new Primary TSX ETY is complete.

**NOTE:** Take care that HTTP connections are not closed by system on swap. They are closed after a 2 hours timeout.

**NOTE:** Redundant but non-Monitored ETYs cannot trigger an automatic Switchover. If you would like the Primary to detect errors on redundant but non-Monitored Ethernet I/O, and trigger a Switchover on this basis, you must still accomplish this in your application program.

## Monitored ETY Module

The TSXETYx103 module allows communication over Ethernet and Modbus TCP. They offer 10/100 Mbit/s throughput. The only authorized Ethernet services on the Monitored ETY are I/O Scanner and HTTP.

For more information, please refer to the "Automation platform Modicon Premium" catalog available from Schneider Electric.

The monitored ETY module enables the switching of Ethernet services and automatic IP Address swapping between the Primary and Standby TSX ETY controllers.

Although the rack position and configuration of a Monitored ETY is unrestricted in a normal Premium system, in a Hot Standby system these factors must be identical to the corresponding module and meet all other requirements of this manual. ETY modules are linked either through Ethernet switches (two switches minimum) or by Ethernet crossover cable. Using an Ethernet transceiver, an optical connection can be used for long distance connections.

To configure the Monitored ETY module in Unity Pro, the rack (topological) address of the Monitored ETY module should be set in the Hot Standby Tab of the PLC's base configuration (*see page 128*). The user selects in the combo box from a list of existing ETY module addresses.

One purpose of the Monitored ETY modules is to diagnose the status of the Premium Hot Standby system. This is achieved via the ETY-sync link. The Monitored ETYs can also be used to manage Ethernet I/O devices by configuring an Ethernet I/O scanning utility.

Configure the Ethernet I/O Scanning service in the Monitored ETY so that switchover is triggered when the primary PLC detects an inoperable ETY-sync link. If the service is not configured in the monitored ETY, a disconnected monitored link will generate a switchover. Ethernet I/O Scanning services configured for this Monitored ETY will be services after switchover.

If your system uses multiple Ethernet services, and has multiple pairs of ETY modules, the time required for Switchover can be shortened and made more predictable by configuring each Ethernet service to run on a different ETY module. For example, if you configure I/O scanning from the Monitored ETY, we advise you to distribute the other desired Ethernet services evenly among the other available ETY modules.

If a condition in the Monitored ETY leads to the initiation of a Switchover event, the CPU sends a state change command to all configured ETY modules present on the X-Bus.

All ETY modules in the Hot Standby PLC then swap IP addresses.

### **WARNING**

#### **UNINTENDED EQUIPMENT OPERATION**

Do not configure the I/O Scanner service when you make a point-to-point ETY-sync link connection with a crossover cable.

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

 **WARNING****UNINTENDED EQUIPMENT OPERATION**

If the Primary PLC will be addressed by HMI or SCADA terminals over the ETY-sync link for the purposes of system control:

- Always run the I/O Scanning service on the Monitored ETY modules.
- Use a minimum of two network switches on the ETY-sync link.

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

If the ETY-sync link will be used to support Monitored Ethernet I/O, then it must be connected through at least two network switches. If it is not connected through at least two switches, a loss of communications on the ETY-sync link will result in both PLCs entering the Offline operating mode.

 **WARNING****UNINTENDED EQUIPMENT OPERATION**

Always connect the ETY-sync link through at least two approved network switches if the ETY-sync link will be used to provide Monitored (redundant) Ethernet I/O capabilities.

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

## ETY Operating Modes and Premium Hot Standby

### Operating Modes

The ETY modes are:

- **Primary Mode**  
The associated PLC (the PLC in the same rack as this ETY) is acting as the Hot Standby Primary. All client/server services configured to run from this ETY module are active.
- **Standby Mode**  
The associated PLC is operating as the Standby. All client/server services configured to run from this ETY module are active, with the exception of I/O Scanning.
- **Offline Mode**  
The associated PLC is in either the Stop or Offline mode. The client/server services operate identically to the Standby state.

The Premium Hot Standby and the ETY operating modes are reported by Unity Pro according to the following table.

CPU Module Status	HSBY State	ETY Operating Mode
Not present or completely inoperative	N/A (PLC does not show as being installed)	Unassigned
Present and Healthy	Primary	Primary
Present and Healthy	Standby	Standby
Present and Healthy	Offline	Offline

Any one of four events will affect the ETY operating mode. These four events occur when the ETY is powered-up, when an ETY executes a Hot Standby Switchover, when an ETY goes to offline mode, or when a new application is downloaded to the PLC associated with the ETY.

### Power on and IP Address Assignment

An ETY obtains its IP Address assignment at power-up as follows:

If the HSBY state is...	Then the IP Address assigned is...
Primary	Configured IP address from the ETY configuration table
Standby	Configured IP address + 1 from the ETY configuration table

The Primary and Standby states are assigned on system startup. See *Start/Stop System*, [page 200](#) for details.

When its associated CPU stops or enters its Offline mode, the HSBY ETY goes to the Offline mode. The IP address is determined by whether or not the other controller is in transition to the Primary state.

Offline event table:

HSBY ETY Mode	IP address
Primary to Offline	If this transition occurs, and the other PLC becomes Primary (Switchover occurs), then this ETY adopts IP address + 1. If this transition occurs, and the other PLC does not become Primary (no Switchover), then this ETY keeps its existing IP address from the Unity Pro ETY configuration table.
Standby to Offline	Configured IP address + 1 from the ETY configuration table

### Power on and Ethernet Services

The following table shows how the status of an ETY service is affected by the Premium Hot Standby state:

HSBY State	Status of ETY services				
	Client Services	Client/Server Services	Server Services		
	I/O Scanner	Modbus Messaging	FTP	SNMP	HTTP
Standalone	Run	Run	Run	Run	Run
Primary	Run	Run	Run	Run	Run
Standby	Stop	Run	Run	Run	Run
Offline	Stop	Run	Run	Run	Run

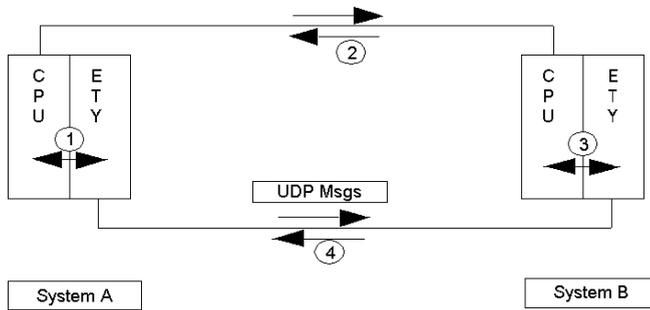
### Hot Standby Switchover

The following steps provide an example of how the ETYs would negotiate a Switchover if an ETY-sync link disconnection was detected by the Primary (PLC/ETY A is the Primary and the PLC/ETY B is the Standby):

Step	Action
1	A Switchover event occurs. System A CPU sends Offline command to HSBY ETY, ETY resets all server/client connections.
2	System A CPU informs System B CPU that a Switchover event has occurred and it is to become the Primary.
3	System B CPU commands HSBY ETY B to begin acting as the ETY associated with the new Primary PLC.
4	System A HSBY ETY initiates an exchange of UDP messages with System B HSBY ETY to coordinate the IP address Switchover.

### Hot Standby Switchover Illustration

The following illustration displays a Switchover event:



## IP Address Assignment

### Configuring the ETY

The ETY TCP/IP addresses of all ETY modules used in a Hot Standby system have to be configured manually in Unity Pro and not from a remote device acting as a BOOTP / DHCP server. Because the Primary and Standby controllers have identical physical configurations, the configured IP addresses are the same. The ETY's IP address is either the configured IP address or the configured IP address +1. The IP address is determined by the current local Hot Standby state.

In the Offline state, the IP Address is determined by whether or not the other controller is in transition to the Primary state.

The following table shows the IP Address assignments.

Hot Standby State	IP Address
Primary	Configured IP Address
Standby	Configured IP Address + 1
Transition from Primary to Offline	If other PLC successfully transitions to the Primary state, or is currently attempting such a transition: Configured IP Address + 1. If other PLC does not become Primary: Configured IP Address.
Transition from Standby to Offline	Configured IP Address + 1

### IP Address Restriction

#### **WARNING**

##### **UNINTENDED EQUIPMENT OPERATION**

- Do not assign consecutive IP addresses to ETY modules in the Primary rack. This will cause duplicate IP addresses to be assigned to the ETY modules on the Standby rack, leading to unintended equipment operation.
- Do not assign the IP addresses xxx.xxx.254 or xxx.xxx.255 to any ETY module in a Premium Hot Standby system.
- The Primary and Standby ETYs must reside in the same network and subnetwork.

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

### Duplicate ID Address Checking

When power is first applied, the ETY modules begin by checking the ETY configuration tables to see which IP addresses have been configured. The ETY modules then check for duplicate addresses on the network. If a duplicate IP address is found, the affected ETY module will remain at its default (MAC-based) IP address and display a duplicate IP address error on its front panel.

**NOTE:** The ETY modules only check for duplicate IP addresses when power is first applied. They do not check for duplicate IP addresses when a Switchover occurs or after replacement of an ETY Ethernet cable.

### IP Address Transparency

For continued Ethernet communication, the new Primary ETY must have the same IP Address as the former Primary ETY. The IP Address in the Standby ETY (an ETY in the Standby state) is the configured IP Address + 1.

The ETYs integrated into the Premium Hot Standby configuration coordinate this IP Address swapping with the management of any Ethernet services such as FTP or HTTP that are configured on these ETYs.

## Network Effects of Premium Hot Standby

### Browsers

The Premium Hot Standby's ability to automatically switch over Ethernet I/O and configured Ethernet services affects the performance of the Ethernet services. These affects occur immediately before, during, and after a Switchover. You will need to consider these effects when you design your Hot Standby system.

If a browser has requested a page served by one of the ETY modules whose IP address is controlled by the Hot Standby system and a switchover event triggers an IP address swap during the download, the browser times out. Click the **Refresh** or **Reload** button.

### Remote and Local Clients

When a Hot Standby PLC undergoes a Switchover event, the state of its In-rack ETY modules and their configured servers also changes. Any remote or local clients already connected to the affected ETY modules will experience the following:

- Outstanding requests are ignored.
- If the client is a Modbus client, the ETY issues a Modbus logout command.
- The ETY closes the connections using the TCP/IP reset command.

If a new client connection is requested during a Switchover, the connection attempt will either be rejected or immediately closed, depending on the timing of its arrival and the timing of the Switchover. This reject/close operation is again accomplished using the TCP/IP reset command.

### I/O Scanning Service

The I/O Scanning service, which is based on Modbus commands delivered over TCP/IP, allows for the repetitive exchange of I/O data with remote Ethernet I/O devices. The I/O Scanner runs on the Primary PLC's ETY module, and sends Modbus Read/Write commands to the scanned I/O devices to read and modify their input and output states. In the Standby controller, the I/O scanning service is stopped.

When the Hot Standby Switchover occurs, the Primary ETY closes all connections with I/O devices by sending a TCP/IP reset. The I/O Scanning service in this ETY is then set to Standby or Offline depending on the precipitating event.

After the Switchover, the new Primary ETY restarts the I/O Scanning service and thereby resumes its connection to each I/O device.

The TSX ETY 4103/5103 provides the I/O scanning feature, which can be configured using Unity Pro.

**NOTE:** When the I/O Scanning service is configured in the Monitored ETY, an interruption of the ETY-sync link on the Primary side will generate a Switchover. The ETY-sync link status bit can be read in the ETY module by using an explicit exchange initiated using the READ\_STS function block. In this function block, you would use the T\_GEN\_MOD I/O derived data type (IODDT) to read the %MWr.m.MOD.2.2 bit, which is set to 1 when the ETY-sync link becomes inoperative.

During a Switchover, in the period after the Primary has closed the I/O Scanner, and before the Standby has assumed the Primary role and restored the Scanner, the input and output values of all scanned Ethernet devices must be controlled to prevent unintended equipment operation.

For Monitored Ethernet inputs, this is normally accomplished using the "Last Value" option available in the I/O Scanning configuration table of the ETY module in Unity Pro. This value must be set to "Hold Last" to prevent changes in the input values of scanned Ethernet devices.

For Monitored Ethernet outputs, control of the output states during Switchover is usually asserted by selecting the "Hold Last Value" option (or equivalent) available in the configuration tool provided with the Ethernet device. If an Ethernet device does not have a "Hold last value" option, then the associated outputs may transition states briefly during a Switchover.

## WARNING

### UNINTENDED EQUIPMENT OPERATION

- Use Unity Pro to program all scanned Ethernet I/O inputs to use a "Last Value" option of "Hold Last".
- Use the Ethernet configuration tool that came with your Ethernet output device to select the "Hold Last Value" option, if available.
- If your scanned Ethernet outputs do not support the "Hold Last Value" option, only use these outputs to control systems and processes that can sustain a momentary value change without adverse effects.

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

It is possible to exchange information over I/O Scanning services using communication Function Blocks. In the period immediately before, during, and after a Switchover event, it is possible that a communication Function block will not successfully begin or conclude a transaction with the I/O Scanner. Program your application to avoid any behaviors that might arise from this.

## WARNING

### UNINTENDED EQUIPMENT OPERATION

Write your application program to confirm and if necessary resend messages passed to and from the I/O Scanner using communication Function Blocks.

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

 **WARNING****UNINTENDED EQUIPMENT OPERATION**

Do not configure the I/O Scanning services on multiple ETY modules to scan the same I/O device or IP address.

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

**FTP/TFTP Server**

The File Transfer Protocol/Trivial File Transfer Protocol (FTP/TFTP) server is available as soon as the module receives an IP address. Any FTP/TFTP client can log on to the module. Access requires the correct user name and password. Premium Hot Standby allows only one active FTP/TFTP client session per ETY module.

When the Hot Standby Switchover occurs, the Primary and Standby ETYs close the FTP/TFTP connection. If a user sends an FTP/TFTP request during the Switchover, the communication is closed.

Whenever you re-open communication, you must re-enter a user name and a password.

**HTTP Server**

**NOTE:** Take care that HTTP connections are not closed by system on swap. They are closed after a 2 hours timeout.

## Section 7.3

### Configuring Registers

---

#### Purpose

This material describes configuring a Premium Hot Standby system by selecting options that affect the Hot Standby-specific registers. You may want to use this method if your system has specific configuration needs.

#### What Is in This Section?

This section contains the following topics:

Topic	Page
Understanding System Words and Bits	159
Understanding the Non-Transfer Area and Reverse Transfer Words	160
Understanding the Unity Command Register	161
Understanding the Unity Status Register	163
Using Initialized Data	166
Synchronization of Realtime Clocks	167

## Understanding System Words and Bits

### Overview

In accordance with IEC 61131-3 standards, Unity Pro uses global objects called System Bits and System Words. These Bits and Words are used to manage the states of the two PLCs.

### System Word %SW60

System Word %SW60 can be used to read from and to write to the Premium Hot Standby Command Register.

**NOTE:** %SW60 is described and used according to the IEC 61131-3 conventions.

### System Word %SW61

System Word %SW61 can be used to read the contents of the Premium Hot Standby Status Register

**NOTE:** %SW61 is described and used according to the IEC 61131-3 conventions.

### System Words %SW62 through %SW65

The system words %SW62, %SW63, %SW64, and %SW65 are reverse registers on the standby PLC reserved for use by the Reverse Transfer process. You can use these four registers to collect information on the status of the standby PLC and its modules. The information in these System Words is then transferred to the Primary controller during each MAST task. Decide what information to make available to the Primary PLC using these System Words, and then write your application program to populate these Reverse Transfer Registers accordingly. Program your application to do this work in section 0 of your application program. (Only section 0 runs on the standby controller.)

## Understanding the Non-Transfer Area and Reverse Transfer Words

### A Non-Transfer Area

The Non-Transfer Area is the block of %MW that is not transferred from Primary to Standby during the update of the standby CPU controller.

This block is from %MW0 to %MW99. The size of this block can not be changed.

This is a major difference compared to Modicon Quantum Hot Standby products where the size of this Non-Transfer area is defined by the user (%MW1 to %MWx).

More information can be found in the "System Technical Guide - High Availability solutions" available from Schneider.

### Reverse Transfer Words

Four system words, %SW62 to %SW65, are dedicated to transfer data from the Standby controller to the Primary during the update of the standby CPU controller.

These System Words are automatically transferred from the Standby to the Primary during each MAST task, and can be used by your application program (in the first section) to store diagnostic information. It is important to remember that the Reverse Transfer Words do not have default values. Any information transferred using these registers must be inserted by Section 0 of your application program.

More information can be found in the "System Technical Guide - High Availability solutions" available from Schneider.

## Understanding the Unity Command Register

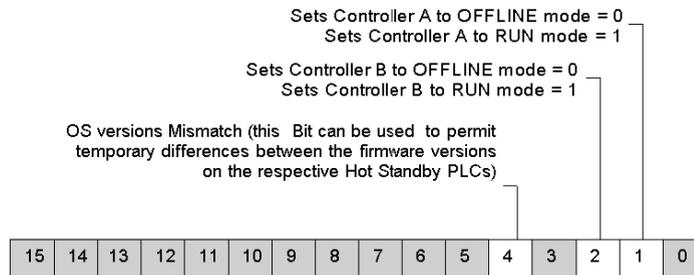
### Setting the Bits in the Command Register

The read/write Command Register defines the operating parameters of a Hot Standby application for both the Primary and Standby and is located at system word %SW60.

At each scan, the Command Register is replicated and transferred from the Primary to the Standby. Transfer occurs only from Primary to Standby. Any changes made to the Command Register on the Standby will have no effect because the values transferred from the Primary overwrite the values in the Standby.

**NOTE:** All changes to the command register %SW60 must be written to the Primary PLC. This register is copied from the Primary to the Standby PLC during each MAST task. Therefore, any changes you make directly to the Standby PLC's command register will be overwritten by the subsequent database transfer without taking effect.

The following illustration identifies the operating options provided by the Command Register.



### System Bit %SW60.1

PLC A OFFLINE/RUN mode:

- %SW60.1 = 1  
PLC A goes to Run mode
- %SW60.1 = 0  
PLC A goes to Offline mode

### System Bit %SW60.2

PLC B OFFLINE/RUN mode:

- %SW60.2 = 1  
PLC B goes to Run mode
- %SW60.2 = 0  
PLC B goes to Offline mode

### System Bit %SW60.4

This System Bit determines the behavior of the Standby PLC if the firmware (OS) versions of the two controllers do not match. The purpose of this bit is to allow upgrades to the CPU, Copro, and ETY firmware while your system is operational. The possible settings are:

- %SW60.4 = 1  
If there is an OS Versions Mismatch with the Primary PLC, the Standby PLC stays in Standby mode
- %SW60.4 = 0  
If there is an OS Versions Mismatch with the Primary PLC, the Standby PLC goes to Offline mode (default)

Reset this bit to 0 after a firmware upgrade to avoid possible unintended system behavior.

## WARNING

### UNINTENDED EQUIPMENT OPERATION

- Follow the firmware upgrade procedure (see *Executing the Firmware Upgrade Procedure*, page 237).
- Always refresh your application program after a firmware download.
- When the firmware upgrade has been completed, restore the Primary PLC's Command Register Bit 4 (%SW60.4) to a value of 0.

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

**NOTE:** Only the Standby PLC actually performs checks to determine if there is a CPU, Copro, or ETY firmware mismatch.

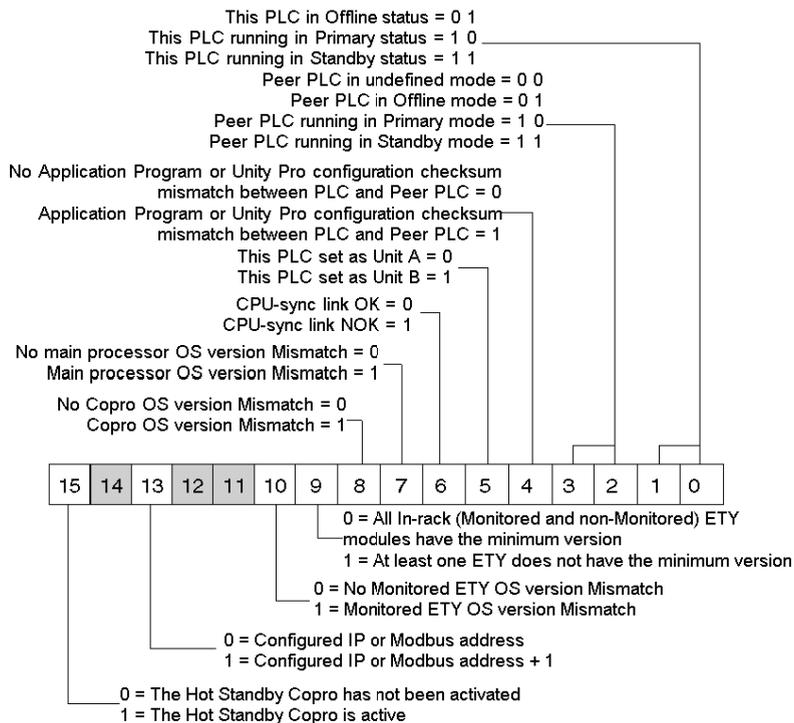
## Understanding the Unity Status Register

### Bits in the Hot Standby Status Register

The Hot Standby Status Register is a read-only register located at system word %SW61 and is used to monitor the current machine status of the Primary and Standby.

Both the Primary and the Standby have their own copy of the Status register. The Status register is not transferred from Primary to Standby. Each PLC must maintain its local Status Register based on the regular communication between the two controllers.

The following illustration identifies the operating options provided by the Status Register.



### System Bits %SW61.0 to %SW61.3

These four bits display the states of the local and remote Hot Standby controllers.

Status of local PLC

- %SW61.1 = 0 and %SW61.0 = 1 means local PLC is in OFFLINE mode
- %SW61.1 = 1 and %SW61.0 = 0 means local PLC is running in Primary mode
- %SW61.1 = 1 and %SW61.0 = 1 means local PLC is running in Standby mode

#### Status of remote PLC

- %SW61.3 = 0 and %SW61.2 = 1 means remote PLC is in OFFLINE mode
- %SW61.3 = 1 and %SW61.2 = 0 means remote PLC is running in Primary mode
- %SW61.3 = 1 and %SW61.2 = 1 means remote PLC is running in Standby mode
- %SW61.3 = 0 and %SW61.2 = 0 means remote PLC is not accessible (Power off, no communication)

#### System Bit %SW61.4

%SW61.4 is set to 1 whenever a logic mismatch (a difference between the application programs or configuration information) is detected between the Primary and Standby controllers.

#### System Bit %SW61.5

%SW61.5 is set to 0 or 1 depending on the Ethernet copro MAC address:

- %SW61.5 = 0 means that this PLC is PLC A (has the lowest MAC address).
- %SW61.5 = 1 means that this PLC is PLC B (has the highest MAC address).

**NOTE:** To perform the MAC address comparison, the two PLCs have to be connected with the CPU-sync link.

#### System Bit %SW61.6

This bit indicates if the CPU-sync link between the 2 PLCs is valid:

- %SW61.6 = 0 means the CPU-sync link is operating properly. The contents of bit 5 are significant.
- %SW61.6 = 1 means the CPU-sync link is not operating properly. In this case, the contents of bit 5 are not significant because the comparison of the 2 MAC addresses cannot be performed.

#### System Bit %SW61.7

This bit indicates if there is a Main Processor firmware (OS) version mismatch between Primary and Standby:

- %SW61.7 = 0 means no CPU firmware version mismatch.
- %SW61.7 = 1 means there is a CPU firmware version mismatch. If CPU, Copro, and ETY firmware mismatches are not allowed in the System Command register (%SW60.4 = 0), the system will not operate redundantly when a CPU firmware mismatch is detected.

#### System Bit %SW61.8

This bit indicates if there is a Copro firmware version mismatch between the Primary and Standby:

- %SW61.8 = 0 means no Copro firmware version mismatch
- %SW61.8 = 1 means there is a Copro firmware version mismatch. If CPU, Copro, and ETY firmware mismatches are not allowed in the System Command register (%SW60.4 = 0), the system will not operate redundantly when a Copro firmware mismatch is detected.

### System Bit %SW61.9

This bit indicates if at least one ETY module does not have the minimum firmware version:

- %SW61.9 = 0: All local in-rack ETY modules have the minimum version required for Premium Hot Standby.
- %SW61.9 = 1: At least one local in-rack ETY module does not have the minimum version required for Premium Hot Standby. In this case, the associated PLC goes offline.

### System Bit %SW61.10

This bit indicates if there is a Monitored ETY firmware version mismatch between Primary and Standby:

- %SW61.10 = 0 means no Monitored ETY firmware version mismatch
- %SW61.10 = 1 means there is a Monitored ETY firmware version mismatch. If CPU, Copro, and ETY firmware mismatches are not allowed in the System Command register (%SW60.4 = 0), the system will not operate redundantly when a Monitored ETY firmware mismatch is detected.

### System Bit %SW61.13

This bit indicates which IP or Modbus address is applied by each ETY or SCP module in the Local rack:

- %SW61.13 = 0 means each Local In-rack ETY or SCP module is applying its configured IP or Modbus address
- %SW61.13 = 1 means each Local In-rack ETY or SCP module is applying its configured IP or Modbus address + 1

### System Bit %SW61.15

If %SW 61.15 is set = 1, the setting indicates that the Local Ethernet Copro device is set up correctly and working. If this bit becomes 0 (zero), the %SW61.6 bit reporting the status of the CPU-sync link will go to 1.

## Using Initialized Data

### Declaring Initialized Data for Cold Starts

The Unity Premium Hot Standby supports initialized data. This initialized data is stored in the Hot Standby CPU's non-volatile memory, and is used specifically to determine the controller's behavior during a Cold Start.

Initialized data also allows you to specify custom values to be loaded and utilized when you perform a Cold Start of your Hot Standby PLC. You can declare values for the initialized data in Unity Pro.

### Online Modifications to Initialized Data

In a standalone Premium controller, it is possible to modify the values of the initialized data online (when Unity Pro is connected to the Primary PLC and in the Online connection state).

In Premium Hot Standby systems, the online modification of initialized data can create a logic mismatch that puts the Standby PLC in the offline state. This results in a loss of the system's high availability. See *Understanding Premium Hot Standby Logic Mismatch*, [page 228](#), for more information on logic mismatches.

## Synchronization of Realtime Clocks

### Synchronization of Primary and Standby Realtime Clocks

Each Premium Hot Standby PLC has an adjustable, hardware-based Realtime Clock (RTC) that manages the current date and time. This date and time information is recorded in the Primary controller's System Words (%SW49 ... %SW53). These system words are part of the database sent from the Primary to the Standby PLC during each MAST task cycle, but this information is not used to synchronize the Standby PLC's realtime clock during each MAST task. The information in these System Words is only used to synchronize the RTCs at Switchover. This is because accessing and updating the RTC takes a significant amount of time, and would extend the length of each MAST task unnecessarily if performed during each cycle.

**NOTE:** Even though the RTC system words in the Standby PLC are updated during each MAST task cycle, the actual hardware RTC in the Standby PLC is not updated until a Switchover event occurs.



---

# Chapter 8

## Programming/Debugging

---

### Overview

This chapter describes the programming and the debugging of an application intended for use on a Premium Hot Standby system.

However, it only focuses on the aspects of programming and debugging that differ in a Hot Standby system. For a more comprehensive explanation of Unity Pro programming and debugging see:

- *Premium and Atrium Using Unity Pro User Manual*, reference 35006160
- *Startup Guide for Unity Pro*, reference 35008402
- *Unity Pro 3.1 Operating Modes*, reference 33003101
- *Unity Pro Program Languages and Structure Reference Manual*, reference 35006144

### What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
8.1	Developing Your Hot Standby Application	170
8.2	Debugging Your Hot Standby Application	186

---

# Section 8.1

## Developing Your Hot Standby Application

---

### Purpose

This section describes the rules for developing an application in a Premium Hot Standby system.

### What Is in This Section?

This section contains the following topics:

Topic	Page
Programming Method	171
How to Program a Premium Hot Standby Application	176
Structure of Database	178
Transferring Your Program to the Primary and Standby PLCs	185

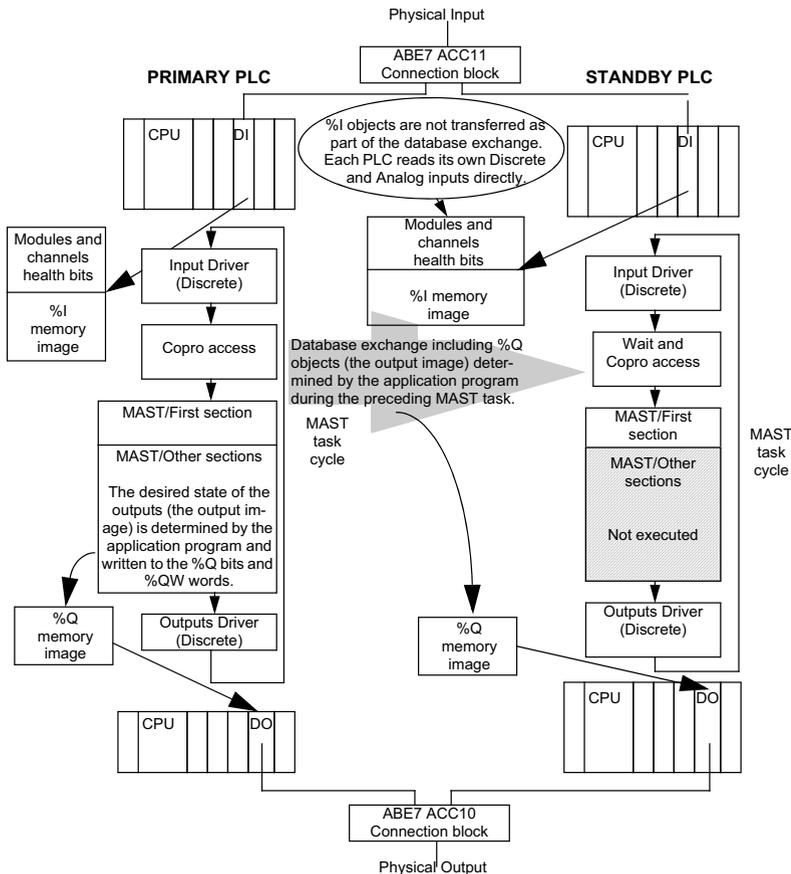
## Programming Method

### General points

For programming a Premium Hot Standby PLC, it is important to understand how the Primary PLC's processor performs reading of inputs, application program processing, updating of outputs and Copro access.

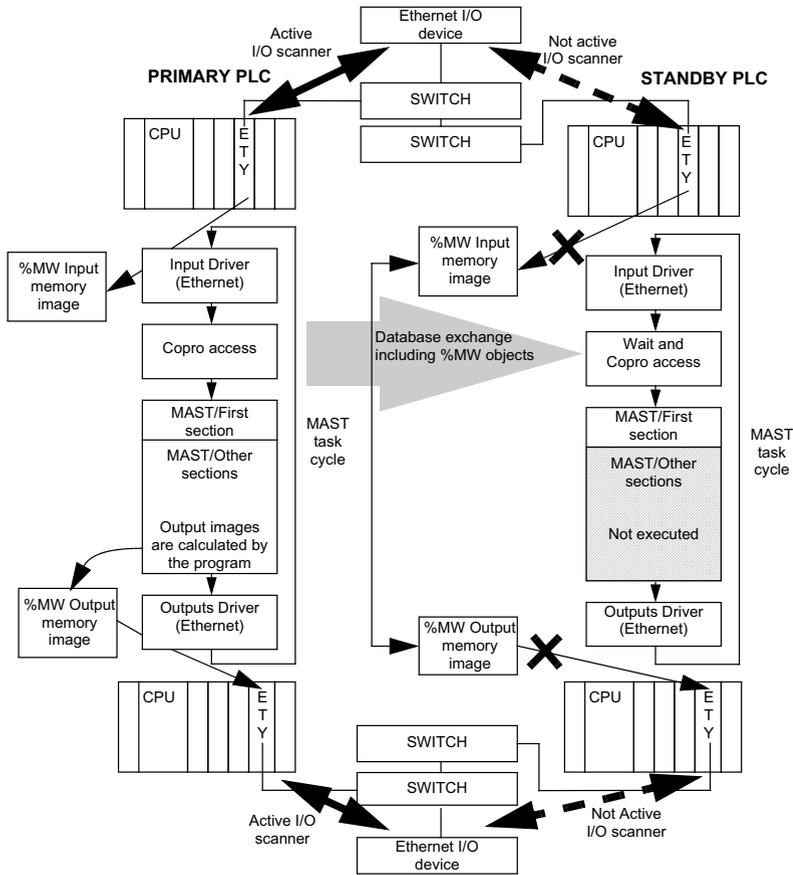
### Operation cycle with In-rack I/O (Discrete I/O Pictured)

The following graphic displays the operation cycle with In-rack I/O:



## Operation cycle with Ethernet I/O

The following graphic displays the operation cycle with Ethernet I/O:



**NOTE:** The Standby does not read the %MW Input values directly because no I/O Scanner is active on the Standby's Monitored ETY. The Standby PLC receives these %MW Input values from the Primary as part of the database transfer.

---

## Operation cycle

As described in the two above graphics, the role of each PLC is different according to the Hot Standby mode:

- PLC in Primary mode
  - Performs all the application sections
  - Determines the state of all Local In-rack inputs (discrete and analog)
  - Updates the state of all Local In-rack outputs
  - Uses its associated Monitored ETY (and the I/O Scanner configured on this ETY) to manage the input and output states of any Monitored Ethernet I/O
  - Retrieves diagnostic information from the Standby PLC
  - Manages its own diagnostic information and the information of the Hot Standby Premium system, including the health of the Local Power Supply, CPU and In-rack modules
  - Sends the database to the Standby PLC (includes the %Q / %QW discrete output images and the %MW Ethernet output states)
- PLC in Standby mode:
  - Only the first section of the application program is executed
  - Determines the state of all Local In-rack inputs (discrete and analog)
  - Applies %Q / %QW output images received from the Primary to the outputs of the Local In-rack modules
  - Receives but does not apply the Ethernet I/O (%MW) images received from the Primary
  - Receives diagnostic information from the Primary PLC
  - Manages its own diagnostic information and the information from the Hot Standby Premium system, including the health of the Local Power Supply, CPU and In-rack modules

A PLC in Offline mode does not execute any part of the application program or I/O management.

The first section of your application (section 0) is executed by both Primary and Standby PLCs. If you need to send information from the Standby to the Primary, program your application to test the state of the Local PLC by checking the %SW61 status register (bits 0 and 1) at the beginning of the first section. When the PLC is discovered to be in Standby mode, also use Section 0 programming to check the In-rack modules' health information by using implicit objects (for example %Ix.y.mod.err) and explicit objects. This health information can be written in the four reverse registers that are transferred to the Primary during each MAST task.

Carefully consider your application when determining which In-rack I/O information you will cyclically evaluate and transfer in section 0 of your program. Because the Standby PLC executes the first section (Section 0) of your application program, and then later applies the %Q information received from the Primary PLC, it is important to avoid changing the state of redundant outputs in Section 0. If you alter output values in Section 0, the output image for the Standby PLC's In-rack output modules might be changed twice in a single MAST task, and the resulting physical state might be inconsistent with that directed by the Primary PLC.

---

When developing DFBs in the first section of the MAST task in the application, there is no restriction on these variable types:

- input
- input/output
- public
- private: Private variables must be computed before they are used.
- output: Output variables are available only when a value that is computed during the current cycle does not depend on the value computed during the previous cycle.

Notice that all DFBs in the first section of an application are executed by the standby PLC in the same manner in which they were executed by the primary PLC.

To avoid this situation, we strongly recommend that you do not program DFBs in the first section of an application. If you must program DFBs in the first section, you should map the relevant variables in the nontransfer area to avoid losing variable values. Also, do not use TON, TOFF, or TP function blocks in the in the first section of an application

## WARNING

### UNINTENDED EQUIPMENT OPERATION

Do not change discrete output bit values for redundant outputs in the first section (section 0) of your application program.

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

A similar consideration applies for analog output signals:

## WARNING

### UNINTENDED EQUIPMENT OPERATION

Design your Hot Standby system so that only one analog output signal at a time is applied to an actuator.

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

It is also possible to manage actuators locally in both PLCs. In this case, actuators are not connected in parallel on two output modules but directly to one output module in each PLC. All such locally-managed outputs must be managed in Section 0 of your application program, or they will not be available on the Standby PLC, which only runs Section 0 of your application program. Also, the memory areas used to control Local I/O must not include or overlap the memory areas included in the Hot Standby database transfer.

---

## **WARNING**

### **UNINTENDED EQUIPMENT OPERATION**

- Use the first section (section 0) of your application program to control non-redundantly managed Local I/O.
- Do not use memory areas included in the Premium Hot Standby database transfer to control these same Local I/O, or the cyclical database transfer will overwrite the intended I/O states.

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

Starting with Unity Pro V6.0, Primary and Standby PLCs can send character requests via the PRINT\_CHAR function in the first section. An application using this function in the first section must be able to abort an invalid call to it, otherwise, information may be lost due to collisions.

## ***NOTICE***

### **LOSS OF INFORMATION**

Do not use the PRINT\_CHAR function in the first section unless the application can abort this function in case of an invalid call to this function.

**Failure to follow these instructions can result in equipment damage.**

---

## How to Program a Premium Hot Standby Application

### Processor configuration

There are two types of MAST task execution modes:

- **Cyclic** - the MAST task executes as rapidly as possible.
- **Periodic** - the MAST task delays execution (if necessary) to observe a user-defined minimum cycle time.

When the Periodic mode is used, the user-defined period should take into account the longer MAST task periods necessary in a redundant system.

The following table presents the characteristics of MAST tasks that may be adjusted by the user in Unity Pro:

Characteristics	Unity Pro Default Values
Max period (ms)	255
Default period (ms)	20 ( <b>Note:</b> Set 80 ms as an initial value for Hot Standby systems)
Min. period (ms)	1 (0 if the Cyclic MAST task execution mode is selected)
Period increment (ms)	1
Max Watchdog (ms)	1500
Default Watchdog (ms)	250
Min. Watchdog (ms)	10

For more details, refer to the section Adjusting MAST Task Properties ([see page 190](#)).

### Restricted Functions

Refer to the section Restricted Functions ([see page 38](#)) for an explanation of all restricted-use functions in a Premium Hot Standby system.

### Detecting Cold Start and Warm Start in a Premium Hot Standby PLC

In a Premium Hot Standby PLC, only the system word %SW10 and the system bit %S1 can be used to detect respectively a cold start and a warm start.

- **%SW10**

If the value of the bit representing the current task is set to 0, this means that the task is performing its first cycle after a cold start.

- %SW10.0: assigned to the MAST task.
- %SW10.1: assigned to the FAST task.

At the end of the first cycle of the MAST task, the system sets each bit of the word %SW10 to 1.

---

- **%S1**

The default value of %S1 is 0. This bit is set to 1 when the device power is cycled and a data save operation is performed. If this value is 1, this is an indication that the last start performed was a warm start.

It is reset to 0 by the system at the end of the first complete cycle and before the outputs are updated.

If you would like to process your application in a certain way based on the type of start, you must write your program to test whether %SW10.0 is reset to 0 (or %S1 is set to 1) at the start of first MAST task. %SW10 and %S1 can be tested by the application when it runs in either Primary or Standby mode.

---

## Structure of Database

### Principle

To take control of the process when the Primary PLC leaves the Primary mode, the Standby PLC has to know the complete status of the Hot Standby configuration. This status transfer is accomplished through the assembly and transmission of a database of all relevant information.

The database that is cyclically transferred from the Primary controller to the Standby controller (via the Copros and the CPU-sync link) includes both system data and user application data. In both cases, some of this data is located (addressable) data, and some is unlocated. The data transferred includes:

System Information:

- Located (a subset of the System Bits and Words)
  - Exchanged during every MAST Task:
    - System Bits: %S30, S31, %S38, %S50, %S59, %S94
    - System Words: %SW0, %SW1, %SW8, %SW9, %SW49...%SW53, %SW59, %SW60, %SW70, %SW108
  - Exchanged only during Switchover
    - %SD18 and %SD20
- Unlocated
  - A subset of the system data managed by the Primary PLC's operating system. This subset includes system counters used by function blocks such as TON, TOFF, and others.

User Application Data:

- Located
  - All %M, %MW, %MD, and %MF data from address 100 up to the maximum number of global address fields configured in Unity Pro's Configuration tab, but no more than 128 kB. The range below 100 (for example, %MW0 - %MW99) is not transferred.
  - The output (%Q) objects and any output forcing settings.
  - EDT / DDT when they are located by the user.
  - Sequential Function Chart (SFC) data types.
- Unlocated
  - EDT / DDT when they are located by the system.
  - Function Block (EFB / DFB) data types.

**NOTE:** In addition to the above, the Primary controller sends the values of all Forced Bits to the Standby as part of the regular database exchange.

**NOTE:** The maximum amount of located data that can be transferred in the database is 128 kB for both the TSX H57 24M and the TSX H57 44M. The maximum unlocated data is 120 kB for the TSX H57 24M and 300 Kb for the TSX H57 44M.

**NOTE:** The maximum size of the entire database is approximately 165 kB for the TSX H57 24M and 405 Kb for the TSX H57 44M.

---

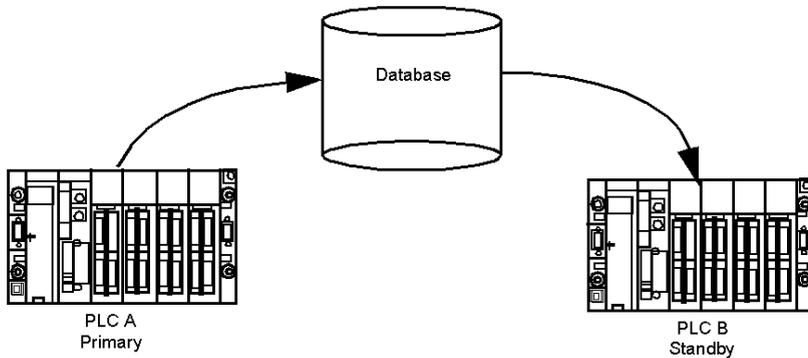
**NOTE:** To make possible a local diagnostic of I/O modules in the Standby PLC, the following objects are not transferred from Primary to Standby:

- The values of the Primary In-rack input modules (%I and %IW objects)
- The values of those %MWr.m.c objects that are related to status parameters and information

For more details on language objects and IODDTs for discrete and analog functions, refer to the *Application language objects* chapter of the "Discrete I/O Modules, Unity Pro" user manual, reference number 35010512, or to the "Analog I/O Modules, Unity Pro" manual, reference number 35010447.

### Illustration

The following illustration displays the information calculated, assembled, and transferred by the Primary PLC:



### Exchange

The database is built automatically by the Primary PLC's Operating System and is sent to the Standby PLC during each MAST task. This exchange is performed via the embedded Ethernet coprocessors of the two Hot Standby PLCs.

The approximate maximum size of database is:

- TSX H57 24M: 165 kB
- TSX H57 44M: 405 kB

---

## Storage

Three types of memory card are offered in Unity Premium range:

- Application
- Application and data storage
- Data storage

The data storage area is a memory zone that can be used to backup and restore data in the memory card using specific EF in the application program. The maximum size of this area is 8 MByte (with TSX MRP F 008M).

This memory zone is not part of the database exchange between the Primary and the Standby controllers. It's only possible to read data using two memory cards (one card in PLC A and one card in PLC B) with the same contents.

## In-rack I/O Management

The programming of a Hot Standby PLC has to take into account the fact that each sensor and probe is connected in parallel on two input or output modules.

Both PLCs read the input values simultaneously at the beginning of each MAST task.

The output values are applied by both PLCs but in a different way:

- The Primary PLC executes the full application. %Q objects are modified depending on the program execution. The discrete/analog output driver applies output values at the end of the Primary MAST cycle. The Primary PLC sends the database to the Standby PLC in the Copro access phase of the MAST cycle.
- The Standby PLC only executes the first section of the application program, mainly for diagnostic purposes and control of Local I/O. The %Q objects received from the Primary PLC are applied at the end of the Standby MAST cycle.

Because the standby PLC executes section 0 of your application program, and then later applies the %Q object image received from the primary PLC, avoid changing the state of redundant outputs in section 0. If you alter output bits in Section 0, the output image for the Standby PLC's In-rack output modules might be changed twice in a single MAST task, and the resulting physical state might be inconsistent with that directed by the Primary PLC.

### **WARNING**

#### **UNINTENDED EQUIPMENT OPERATION**

Do not change discrete output bit values for redundant outputs in the first section (section 0) of your application program.

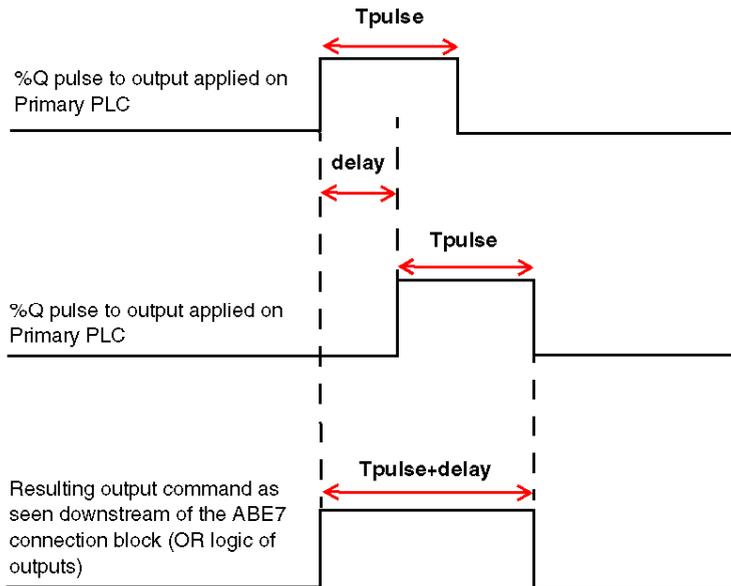
**Failure to follow these instructions can result in death, serious injury, or equipment damage.**

## Managing Pulsed Changes in Discrete Outputs Connected in Parallel

The output modules are connected in parallel to the physical output via a specific connection block. Because there is usually a delay between the application of the calculated %Q image by the Primary and the Standby, the duration of an impulse (short-duration, transient) command can have interesting effects at the parallel output values observed downstream of the connection block. The result of an impulse command is based on the time of the impulse and the delay to apply this impulse in the Standby.

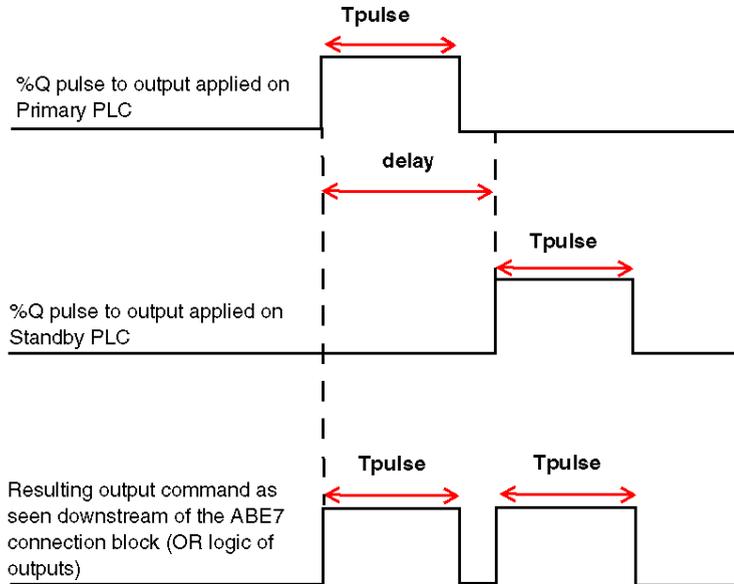
The different situations are illustrated below (the pulse is modified in the same way):

For an impulse command to positive logic outputs, with a delay between the Primary and Standby application of the output images less than  $T_{\text{pulse}}$ :



**NOTE:** The duration of the pulsed command as seen by the controlled device is slightly longer than the equivalent command received from a standalone PLC. There will be no discernible difference in behavior at the device.

For an impulse command to positive logic outputs, with a delay between the Primary and Standby application of the output images of more than  $T_{pulse}$ :



**NOTE:** Because the delay in applying the output images is longer than the commanded change of state, the actual device will execute the command twice if it is capable of reacting quickly enough.

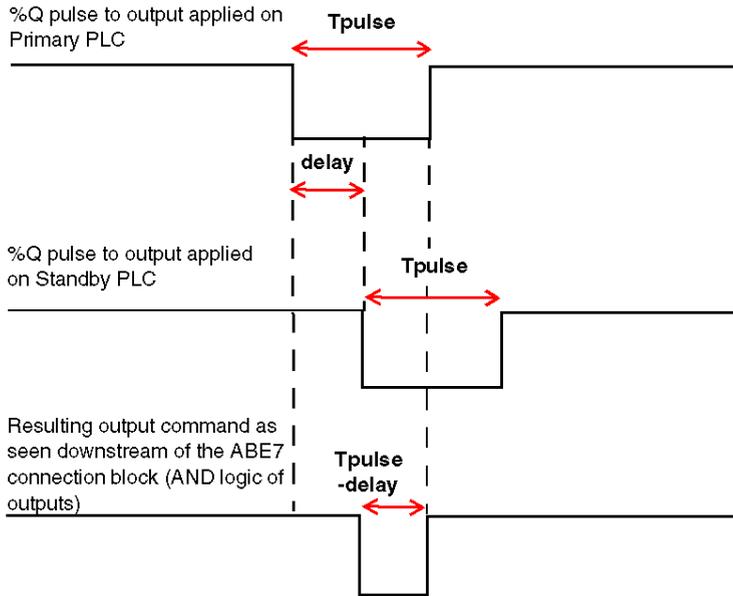
## **⚠ WARNING**

### **UNINTENDED EQUIPMENT OPERATION**

Design your system so that short-duration changes to output states have a duration greater than the maximum delay between the Primary and Standby PLC's application of their output images.

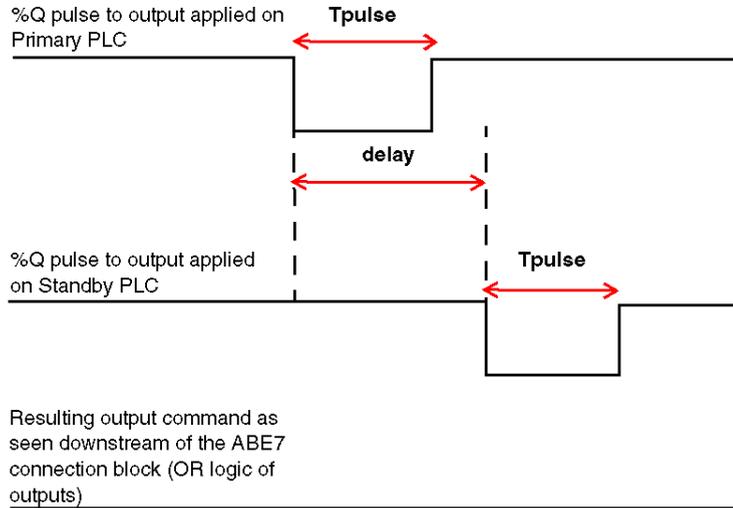
**Failure to follow these instructions can result in death, serious injury, or equipment damage.**

For an impulse command to negative logic outputs, with a delay between the Primary and Standby application of the output images of less than  $T_{pulse}$ :



**NOTE:** The duration of the pulsed command as seen by the controlled device is slightly shorter than the equivalent command received from a standalone PLC. There will be no discernible difference in behavior at the device, unless that device is incapable of responding in a timely fashion to the foreshortened pulse command.

For an impulse command to negative logic outputs, with a delay between the Primary and Standby application of the output images of more than  $T_{pulse}$ :



Resulting output command as seen downstream of the ABE7 connection block (OR logic of outputs)

**NOTE:** Because the delay in applying the output images is longer than the commanded change of state, the actual device will never see the pulse command.

## **⚠ WARNING**

### **UNINTENDED EQUIPMENT OPERATION**

Design your system so that short-duration changes to output states have a duration greater than the maximum delay between the Primary and Standby PLC's application of their output images.

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

---

## Transferring Your Program to the Primary and Standby PLCs

### Transferring Your Program

Because a Hot Standby system requires that identical application programs exist on both the Primary and Standby PLCs, you must upload your application twice, once to each PLC. The procedure is the same for both PLCs:

- Connect the PC with Unity Pro (version 3.1 or above) to the USB or Uni-Telway port on the PLC.
- Use the Unity Pro command: **PLC →Transfer program to PLC.**

---

## Section 8.2

### Debugging Your Hot Standby Application

---

#### Purpose

This section provides information that will help you debug programs created for use on your Premium Hot Standby system.

#### What Is in This Section?

This section contains the following topics:

Topic	Page
Debugging	187
Adjusting MAST Task Properties	190
Using the Hotstandby system with the HSBY_SWAP DFB	194

---

## Debugging

### Introduction

You can write an application for your Premium Hot Standby system in almost the same manner as you would for any a standalone Premium PLC. The Premium Hot Standby system does not require the use of special function blocks or user actions to support its high availability. There are some important exceptions ([see page 38](#)) to this statement.

### Debug and Diagnostic

The following table presents Debug and Diagnostic operations on Premium Hot Standby PLCs:

Diagnostic		TSX H57 24M	TSX H57 44M
Diagnostic Function Block		Yes	Yes
Diagnostic Buffer		Yes	Yes
Diag. buffer characteristics	Max buffer size	16 kB	25 kB
	Max errors	160	254
Breakpoint		1 max	1 max
Step by step (Into, over, and out)		Yes	Yes
Variable animation		<ul style="list-style-type: none"><li>● End of MAST</li><li>● Watch Point</li></ul>	<ul style="list-style-type: none"><li>● End of MAST</li><li>● Watch Point</li></ul>
Link animation		Yes	Yes

### Debug the control/command of the procedure

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

- First, you will debug the basic program operation on a standalone Hot Standby PLC. When you do this, all of the debug and diagnostic resources noted in the table above are available for your use. If a standalone Hot Standby controller is not available, you may put the Standby PLC into a Non Conf state and do this first stage of debugging on the Primary.
- Second, after the standalone debug is complete, you will debug any redundancy-specific aspects of your program on a functioning (redundant) Hot Standby system that is not actively managing your process. When accomplishing this second stage, you will not use the debug and diagnostic resources noted in the table above.

---

## WARNING

### UNINTENDED EQUIPMENT OPERATION

When debugging the redundancy-specific aspects of your application program:

- Always debug your application on a fully functioning Hot Standby system.
- Only conduct debugging on a Hot Standby system that is not actively managing your process.
- Do not use the Unity Pro debug and diagnostic features except as permitted by this manual.
- Confirm that the interaction of the MAST task mode and duration and the Watchdog values meet the needs of your application.

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

When you are conducting the second stage of debugging, confirm that you initially connect to the PLC currently acting as Primary. The Standby PLC only executes section 0 of your application program.

### Debug the First Section in Standby PLC

For debugging the first section in the Standby PLC, the following points have to be taken into account:

- only the %MW0 to %MW99 are not transferred from the Primary to the Standby. All the other application data are coming from the Primary. As a result, the value of these variables that are displayed in a Standby animation table are those coming from the Primary, except for %MW0 to %MW99.
- animation tables can be synchronized with watch points. This is the best way to animate data in synchronization with the code execution. Refer to the *Unity Pro Program Languages and Structure Reference Manual*, reference 35006144, for further details.

### System Debug

Debug and verify system performance of Hot Standby-related applications only on a Hot Standby PLC. Do not use the debug and diagnostic operations for standalone Premium PLCs on a Hot Standby system.

---

## **WARNING**

### **UNINTENDED EQUIPMENT OPERATION**

When debugging the redundancy-specific aspects of your application program:

- Always debug your application on a fully functioning Hot Standby system.
- Only conduct debugging on a Hot Standby system that is not actively managing your process.
- Do not use the Unity Pro debug and diagnostic features except as permitted by this manual.
- Confirm that the interaction of the MAST task mode and duration and the Watchdog values meet the needs of your application.

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

**NOTE:** A Switchover is not generated when the Primary application stops on a breakpoint.

The debugging that can be accomplished once your application program is loaded onto a redundant Hot Standby system is:

- Static verification  
Check that:
  - The application restrictions noted in this manual have been observed
  - The MAST task characteristics have been configured properly
- Dynamic verification  
After the application has been transferred, make sure that the two PLCs are behaving in accordance with Premium Hot Standby requirements:

Once the Hot Standby PLCs have entered either the Primary or Standby operating modes, confirm that:

- All application program sections are executed on the Primary PLC
- Only the first section is executed in the Standby PLC

---

## Adjusting MAST Task Properties

### Introduction

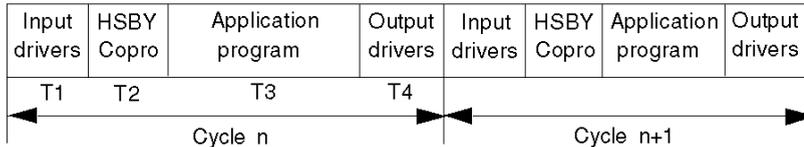
After a reminder on MAST task execution modes, this part describes the Execution time measurement method and gives the procedure to adjust the MAST task period.

### Reminder on MAST Task Execution Modes

The MAST task can be configured using one of the two following execution modes:

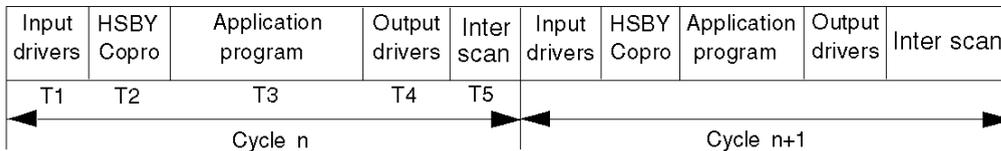
- cyclic mode,
- periodic mode.

#### Cyclic mode:



In the cyclic execution mode, MAST tasks occur in sequence with no reference to the system clock, and hence without any delay between tasks other than a very brief period of system processing. Almost immediately after one task completes, another begins. Therefore, the actual duration of a MAST task in cyclic mode can vary significantly depending on the size and activity of the application, and the number of inputs and outputs to be controlled.

#### Periodic mode:



In the periodic execution mode, MAST tasks are sequenced according to a countdown timer which is referenced to the system clock. This countdown timer may be set for a period between 1 and 255 ms. If the countdown expires before the end of the task, the task will complete normally. If this occurs regularly, the system will appear as if the cyclic MAST task execution mode had been selected. However, some applications like process control do require regular cycle times. If this is the case for your application, confirm that the task period is of sufficient length to avoid cyclic-like behavior.

## Execution Time Measurement

The execution time of the MAST task can be measured by reading system words:

- %SW30: Execution time (in ms) of the last task.
- %SW31: Execution time (in ms) of the longest task.
- %SW32: Execution time (in ms) of the shortest task.

In both cyclic and periodic mode, the MAST execution time is the sum  $T1 + T2 + T3 + T4$ .

$T5$  of the periodic mode is not taken into account.

### First step:

To measure the execution time of the MAST task in a Premium Hot Standby configuration, it is advised to measure first the execution time in standalone mode (or with one of the two PLC in STOP) with the MAST task configured in cyclic mode. In this case, there is no data exchange between the two PLCs, and the execution time of the HSBY Copro part ( $T2$ ) is reduced to its minimum.

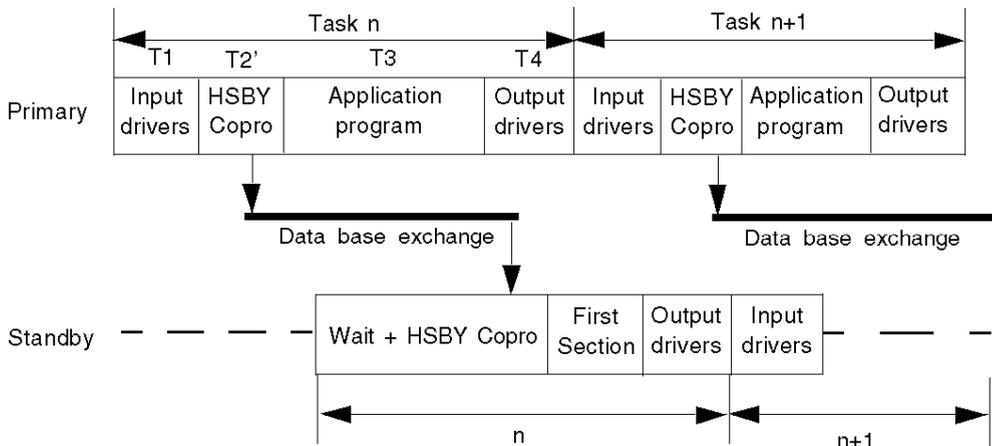
→ execution time of the last MAST cycle = %SW30 =  $T1 + T2 + T3 + T4$

### Second step:

In a second step, the execution time has to be measured with a Primary and Standby PLC.

Two cases have to be taken into account:

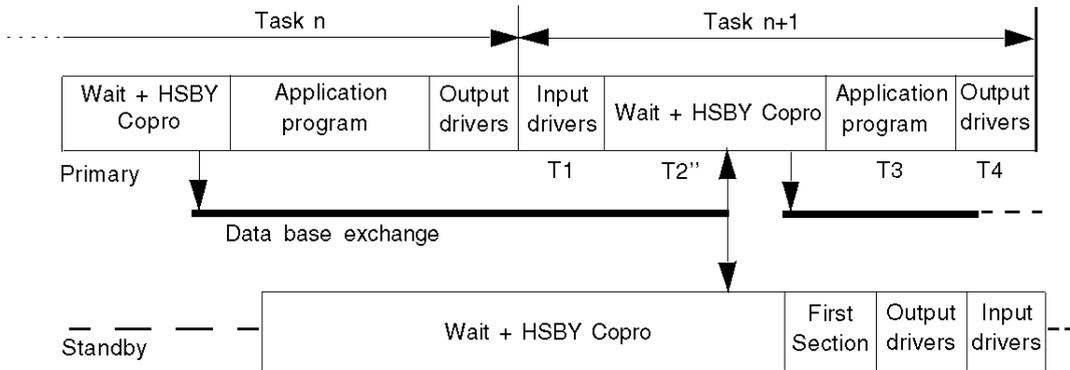
1. The data exchange has no impact on the Primary MAST task duration:



In this first case, the execution time of the HSBY part ( $T2'$ ) is increased with the time required to copy the data base from the CPU memory to the HSBY Copro shared memory.

→ execution time of the last MAST cycle = %SW30 =  $T1 + T2' + T3 + T4$  with  
 $T2' = T2 + \text{time to copy the data base from the CPU memory to the Copro shared memory.}$

2. The data exchange has an impact on the Primary MAST task duration:



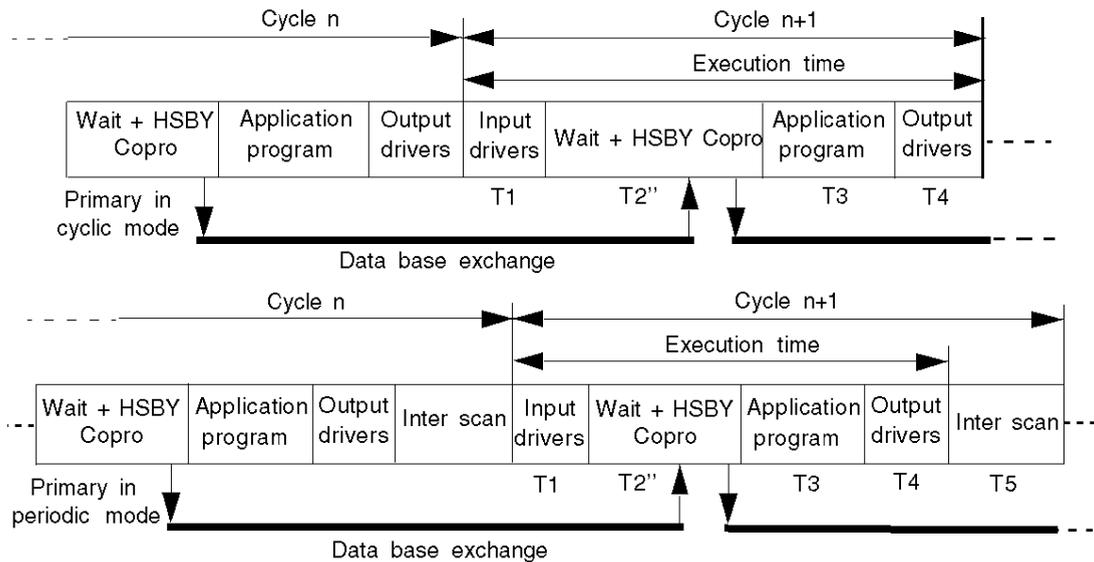
In this second case, the execution time of the HSBY part (T2'') is increased with the time to be waited until the complete transmission of the data base.

→ execution time of the last MAST cycle = %SW30 = T1 + T2'' + T3 + T4 with

T2'' = T2 + time to copy the data base from the CPU memory to the Copro + time to transmit all the data on the network and free the Copro shared memory.

### Third step:

In a third step, the execution time can be measured with the MAST task operating in periodic mode. But this mode may impact the time measurement. In the following diagram, the two applications are the same with the same size of data exchanged from the Primary to the Standby. The only one difference is the cyclic mode for the first one and the periodic mode for the second one (only the Primary time diagrams are shown):



In the periodic mode, it appears that the execution time that is measured is lower than in the cyclic mode. In certain cases, the difference between the two execution modes can be important.

### **Procedure to Adjust the MAST Task Period**

If the MAST task has to be configured in periodic mode, it is advised to:

1. measure the maximum value (%SW31) of the MAST task in cyclic mode with the Premium Hot Standby system normally running (Primary and Standby). This measure has to be done in the Primary PLC with all the configured tasks active (although only the MAST task is recommended in a Premium Hot Standby application).
2. configure the periodic mode with a period at least equal to %SW31 plus a margin of around 20%:  
Period = %SW31 + %SW31 \* 20%.

---

## Using the Hotstandby system with the HSBY\_SWAP DFB

### Function Description

The HSBY\_SWAP function block is used to initiate the swapping between primary and standby CPU.

EN and ENO can be configured as additional parameters.

This function block is convenient to be used to activate a swap by program logic. This swap between primary and standby CPU can be performed in the Hot Standby mode only.

Please refer to:

- the Modicon Premium Hot Standby with Unity - User Manual ([see page 9](#)) for information regarding Premium hot standby CPUs.  
the Modicon Quantum Hot Standby with Unity - User Manual ([see Modicon Quantum, Hot Standby System, User Manual](#)) for information regarding Quantum hot standby CPUs.

This means that when the HSBY is running, the standby PLC becomes the primary PLC, and the old primary PLC becomes the standby PLC activated by the program logic.

**NOTE:** It is not mandatory to use this function for performing a hot standby swap. Indeed, you may rely on register %SW60, as described in the Conditions for Switchover ([see page 207](#)) for Premium or in the Conditions for Switchover ([see Modicon Quantum, Hot Standby System, User Manual](#)) for Quantum.

**NOTE:** The DFB should not be used in the first section of the application.

### WARNING

#### UNEXPECTED EQUIPMENT OPERATION

The HSBY\_SWAP DFB must be called for the purpose of testing the application only.

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

### Advantages of the Swapping Function

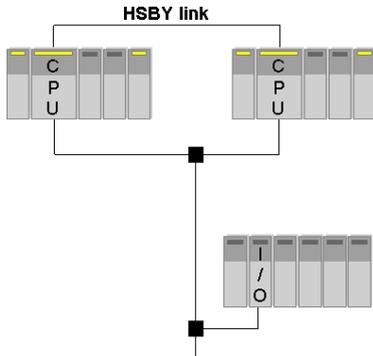
The advantages of swapping could be the following:

- The health of the standby PLC is monitored. It is checked that the standby PLC can take over.
- The switchover could be tested at regular intervals.

---

## Example of a Hot Standby Application

The illustration below shows an example of a hot standby application:



## Steps for Changing Status

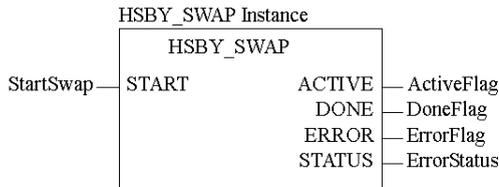
A Hot Standby swapping function is processed as follows:

Step	Action
1	<b>Status:</b> PLC-A is the primary controller, PLC-B is the standby controller. PLC-A sets itself to offline. <b>Result:</b> PLC-B becomes the primary controller.
2	<b>Status:</b> PLC-A is offline, PLC-B is the primary controller. PLC-B sets PLC-A to run-mode <b>Result:</b> PLC-A is the standby controller.
3	<b>Status:</b> PLC-A is the standby controller, PLC-B is the primary controller. EFB outputs will be set. <b>Result:</b> Hot Standby swapping function is completed.

---

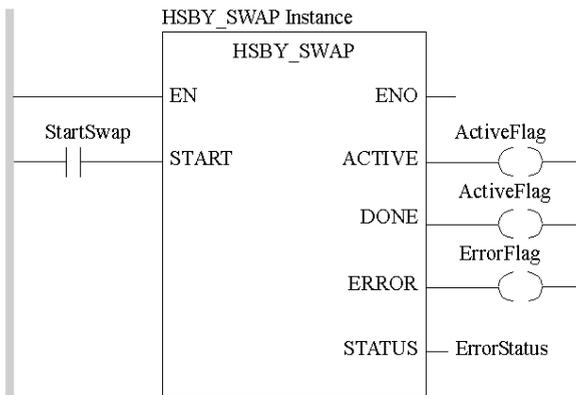
## Representation in FBD

Representation



## Representation in LD

Representation



---

## Parameter Description

Description of the input parameter

Parameter	Data Type	Meaning
START	BOOL	START = 1 starts the HSBY_SWAP operation. The value of 1 must be applied until the operation has finished or until an error has occurred.

Description of output parameters

Parameter	Data Type	Meaning
ACTIVE	BOOL	ACTIVE = 1 indicates that an HSBY_SWAP operation is in progress.
DONE	BOOL	DONE = 1 indicates that the HSBY_SWAP operation has been completed successfully.
ERROR	BOOL	ERROR = 1 indicates that an error has occurred, or that the current HSBY_SWAP operation has been aborted.
STATUS	INT	An error code (STATUS) is generated by the HSBY_SWAP block. A complete list is shown in the error code table.

## Error Status Table

The following table explains the error codes:

Error Codes	Fault Description
0	OK
1	The function HSBY_SWAP has been aborted.
2	Hot Standby not activated (%SW61.15=0).
3	Standby does not exist.
5	The swap was unsuccessful.

**NOTE:** The system words %SW60 and %SW61 reflect the status of the primary and the secondary PLC.

---

## Switchover Using Command Register System Bit %sw60.1 or %sw60.2

Another way of forcing a switchover is setting the bits in the Command Register. To achieve this, do the following:

Step	Action
1	Open file 1.
2	Connect to the primary,
3	Ensure the controller order of the primary is A or B.
4	Access <ul style="list-style-type: none"><li>● Command Register system bit %sw60.1 If the connected controller order is A.</li><li>● Command Register system bit %sw60.2 If the connected controller order is B.</li></ul>
5	Set bit to 0. <b>NOTE:</b> Ensure that the standby switched to primary.
6	Open file 2.
7	Connect to the new primary controller.
8	Access the Command Register system bit used in step 4.
9	Set bit to 1. <b>NOTE:</b> Ensure that the standby controller is now online.
10	Ensure both primary and standby controllers are in Run Primary and in Run Standby mode.

---

# Chapter 9

## Operating

---

### Overview

This chapter provides information about Operating the Premium Hot Standby System.

### What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
9.1	Start/Stop System	200
9.2	Switchover	204

---

# Section 9.1

## Start/Stop System

---

### Purpose

This section describes how to start or stop a Premium Hot Standby System.

### What Is in This Section?

This section contains the following topics:

Topic	Page
Starting the Two PLCs	201
Stopping the Premium Hot Standby	203

---

## Starting the Two PLCs

### Starting Before a Valid Application is Loaded on Both PLCs

If power is applied to the PLCs before your application has been uploaded to either, both the PLCs will start in the Non-configured (Non Conf) mode. Neither PLC can enter a Primary or Standby operating mode until a valid application has been uploaded. After the application has been uploaded to a PLC, issuing a RUN command will move it from the Stop/Offline operating mode to either the Primary or Standby mode, depending on the order in which the PLCs are started. The first PLC to be issued a RUN command will assume the Primary role.

**NOTE:** The first time a PLC is started after a full and valid application program has been uploaded, that start will be a Cold Start.

**NOTE:** To start properly after receiving the application, the two PLCs have to be linked with:

- The CPU-sync link between the two CPUs
- The ETY-sync link between the two monitored ETYs

### WARNING

#### UNINTENDED EQUIPMENT OPERATION

- Always confirm that both the CPU-sync link and ETY-sync link are physically connected before applying power.
- If communications equipment such as network switches is part of the ETY-sync link, confirm that these devices are ON, initialized, and operating properly before applying power.
- Route and protect the CPU-sync link and ETY-sync link cables so that a single accident cannot disconnect both cables.

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

### Valid applications

When identical, valid applications have already been loaded on both PLCs, the first Hot Standby PLC to which power is applied will assume the role of the Primary Controller. Therefore, the controller roles can be determined by delaying the application of power to one PLC using a time-lag relay or some related means.

When power is simultaneously applied to two Hot Standby PLCs with valid applications, the role of the Primary controller is automatically assigned based on the respective MAC addresses of the two PLCs. By default, the PLC with the lowest MAC address becomes the Primary controller.

**NOTE:** In the event of a Switchover, or if you replace one of the PLCs, the identification of PLC A and PLC B in Unity Pro may not align with the Primary and Standby operating modes in the manner you would expect, where PLC A equates to the Primary controller. The same is true for any physical labels you might apply to your PLCs to distinguish them in your system.

---

## WARNING

### UNINTENDED EQUIPMENT OPERATION

- Never assume that a PLC is in a certain operating mode before installing, operating, modifying, or servicing it.
- Before acting on a PLC, always positively confirm the operating mode of both Hot Standby PLCs by viewing their LEDs and checking their System Status Words.

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

### MAC Address

The MAC address, visible on the front panel of the PLC, is a 48-bit number written in hexadecimal notation (6 pairs of 2 digits). The digits used to represent numbers using hexadecimal notation are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F.

Rules to compare two MAC addresses:

- The two MAC addresses must be compared from left to right
- As soon as there are different digits in the same position in each MAC address, the higher MAC address is the one where the digit is higher.

### Examples of two MAC Addresses

First example:

- MAC1 = 00.80.F4.01.6E.E1
- MAC2 = 00.80.B4.01.6E.E1

The MAC1 is higher than the MAC2.

Second example:

- MAC1 = 00.80.F4.01.6E.E1
- MAC2 = 00.80.D4.01.6F.E1

The MAC1 is higher than the MAC2.

---

## Stopping the Premium Hot Standby

### Principle

Stopping a Premium Hot Standby System is identical to stopping a standalone or single PLC, but only if you stop the PLCs in the following order:

- Stop the Standby PLC
- Stop the Primary PLC

If the Standby PLC is not stopped first, a Switchover will occur when the Primary PLC is stopped.

---

# Section 9.2

## Switchover

---

### Purpose

This section describes the Switchover of the Premium Hot Standby.

### What Is in This Section?

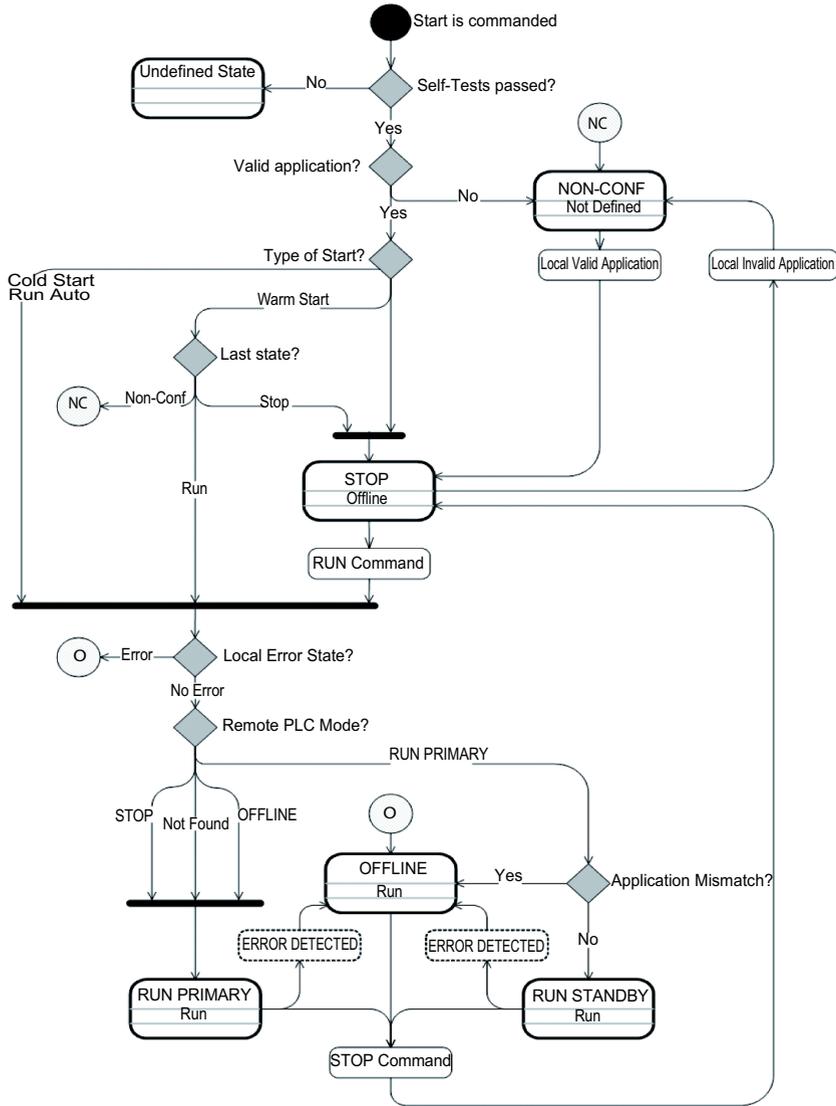
This section contains the following topics:

Topic	Page
Operating Modes Overview	205
Conditions for Switchover	207

## Operating Modes Overview

### General points

The following state diagram shows a dynamic view of the main Hot Standby states:



---

At Cold start with the "Automatic Start in Run" option configured, the PLC restarts depending on the operating mode of the other PLC, the local PLC's operability or non-operability, and on whether identical applications are present on both PLCs.

If...	Then...
The other PLC is Primary, the two applications are identical and this PLC is operating normally	The PLC restarts in Standby mode
The other PLC is Primary and the two applications are not identical or this PLC is not operating normally	The PLC restarts in Offline mode
There is no remote Primary and this PLC is operating normally	The PLC restarts in Primary mode
There is no remote Primary but this PLC is not operating normally	The PLC restarts in Offline mode

A local abnormal or inoperative operating state will be reported when:

- There is a loss of power to the CPU rack
- An application program error that generates a HALT state (for example, a blocking software error)
- The hardware or firmware of the CPU module becomes inoperative
- The CPU-sync link is disconnected

At Warm start, the PLC restarts depending on the previous PLC operating mode (Stop or Run). If the previous state was Run, the PLC restarts according to the operating mode of the other PLC, the local PLC's operability or non-operability, and on whether identical applications are present on both PLCs (refer to the above table).

---

## Conditions for Switchover

### Commanding Manual Switchovers

In addition to the system conditions that cause an automatic Switchover, a manual Switchover can be commanded by writing to bits 1 and 2 of the Unity Command Register at %SW60. This write operation can be accomplished by your application, by issuing a Modbus request from a remote HMI, or in Unity Pro's animation tables.

### Example of Switchover with PLC B Initially in Standby Mode

In this example, the initial state of the system is as follows:

- PLC A has a RUN command (%SW60.1 = 1) and is acting as the Primary
- PLC B has a RUN command (%SW60.2 = 1) and is acting as the Standby

By writing new values to bits 1 and 2 of the %SW60 command register, you can command a change in the operating modes of the Hot Standby controllers. There are four possible bit value combinations that may be written, two of which will command a manual Switchover. The following table describes the four commands and their results:

New Values Written to %SW60		Resulting PLC Operating Modes		Effects
Bit 1	Bit 2	PLC A	PLC B	
0	0	Standby	Primary	<ul style="list-style-type: none"><li>● Switchover event is immediate</li><li>● System remains redundant</li></ul>
0	1	Offline	Standby ↓ Primary	<ul style="list-style-type: none"><li>● Switchover event occurs within one MAST task<sup>1</sup></li><li>● System is no longer redundant</li></ul>
1	0	Primary	Offline	<ul style="list-style-type: none"><li>● No Switchover event</li><li>● System is no longer redundant</li></ul>
1	1	Primary	Standby	<ul style="list-style-type: none"><li>● No Switchover event</li><li>● No change from initial conditions</li></ul>

**1** In this case, we are not directly commanding a Switchover. Instead, we are commanding PLC A to enter an Offline state and we are relying on the system logic to recognize this and change PLC B from Standby to Primary during the next scan.

**NOTE:** All changes to the command register %SW60 must be written to the Primary PLC. This register is copied from the Primary to the Standby PLC during each MAST task. Therefore, any changes you make directly to the Standby PLC's command register will be overwritten by this transfer without taking effect.

**NOTE:** Bit values written to the %SW60 command register are not persistent. After the PLCs assume the new operating modes commanded by the write operation, the system automatically restores both %SW60.1 and %SW60.2 to the default value of 1.

---

### Example of Switchover with PLC B Initially in Offline Mode

In this example, the initial state of the system is as follows:

- PLC A has a RUN command (%SW60.1 = 1) and is acting as the Primary
- PLC B has a RUN command (%SW60.2 = 1) but is Offline because, for example, its local Monitored ETY is inoperative

In this example, the same four bit value combinations noted on the previous page may be written to command register %SW60, but the resulting behavior is quite different because PLC B is Offline due to a diagnostic condition. The following table describes the four commands and their results:

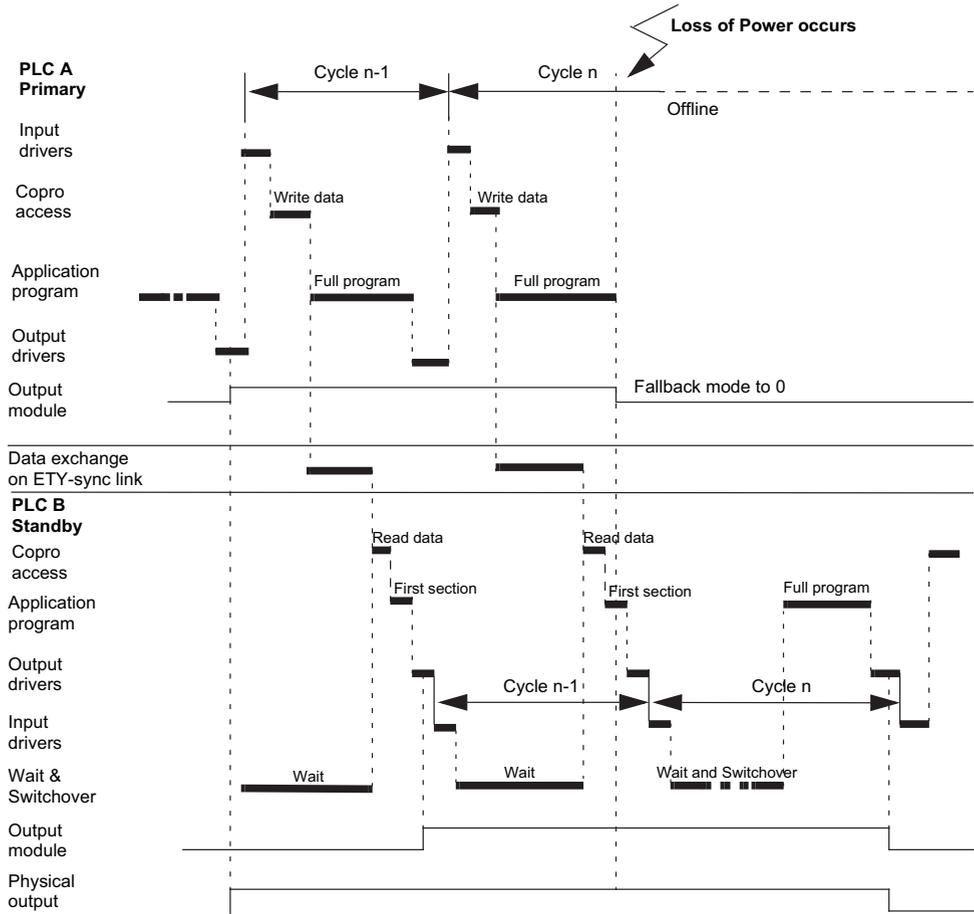
New Values Written to %SW60		Resulting PLC Operating Modes		Effects
Bit 1	Bit 2	PLC A	PLC B	
0	0	Primary	Offline	<ul style="list-style-type: none"><li>● No Switchover event</li><li>● No change from initial conditions</li></ul>
0	1	Offline	Offline	<ul style="list-style-type: none"><li>● System becomes nonoperational</li></ul>
1	0	Primary	Offline	<ul style="list-style-type: none"><li>● No Switchover event</li><li>● No change from initial conditions</li></ul>
1	1	Primary	Offline	<ul style="list-style-type: none"><li>● No Switchover event</li><li>● No change from initial conditions</li></ul>

**NOTE:** All changes to the command register %SW60 must be written to the Primary PLC. This register is copied from the Primary to the Standby PLC during each MAST task. Therefore, any changes you make directly to the Standby PLC's command register will be overwritten by this transfer without taking effect.

**NOTE:** Bit values written to the %SW60 command register are not persistent. After the PLCs assume the new operating modes commanded by the write operation, the system automatically restores both %SW60.1 and %SW60.2 to the default value of 1.

## Switchover When Primary Becomes Inoperative

The following figure displays the expected behavior when the Primary controller experiences a loss of power or a firmware error is detected. This example assumes we are using positive logic Discrete I/O and ABE7 connection blocks.



---

## WARNING

### UNINTENDED EQUIPMENT OPERATION

Do not change output bit values for redundant outputs in the first section (section 0) of your application program.

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

## WARNING

### UNINTENDED EQUIPMENT OPERATION

- Configure your output module fallback modes to prevent changes in output states during Switchover.
- Use fallback mode 0 for all positive logic discrete output modules.
- Use fallback mode 0 when output modules are cabled in parallel using ABE7 ACC1 connection blocks.
- Use fallback mode 1 for all negative logic discrete output modules.

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

**NOTE:** In a properly configured system, during a Switchover event, the physical state of the outputs is maintained at the last value received from the Primary PLC. Before updating the outputs, the new Primary controller (PLC B in the figure above) refreshes all %I objects from its local physical inputs (parallel cabling), and then runs the application program to calculate the new output values.

**NOTE:** In the case of positive logic outputs, the recommended fallback value is 0. If your application program sets an output to 1 in the MAST task immediately preceding an event which causes an automatic switchover, it is likely that this output will go to a 0 state for a short interval before the new primary PLC comes online and reasserts the 1 value. Use only your redundant Discrete in-rack I/O for applications that will not be adversely affected by this type of pulse.

## WARNING

### UNINTENDED EQUIPMENT OPERATION

Only use your redundant In-rack I/O to control systems and processes that can sustain a momentary value of 0 at the outputs without adverse effects.

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

---

## Performances on Switchover

In an operating Hot Standby system (therefore made of one Primary PLC and one Standby one), the Standby PLC is ready to take over in case the Primary one disappears or detects an error on its side.

In order to take control, the Standby PLC must make sure that:

- it has received the information that the (formerly) Primary PLC is not managing the equipment under control anymore; this may happen through its Copro link or through its ETY communication.
- and / or the Monitored ETY on the (formerly) Primary side is silent and is not managing the equipment under control anymore. This may be achieved by waiting for a watchdog used for communication on the Ethernet network on the Monitored ETY side.

Therefore, prior to taking control, the Standby PLC module may have to wait up to the sum of these durations.

In case the system performs a switchover after loss of power on the Primary system, a minimum of one minute should be given to the system to ensure its return to stability. Two consecutive switchovers due to loss of power occurring in less than one minute may lead to instability.

## Recommandations after performing a Switchover

Communication Function Blocks have to be canceled after a Switchover has been performed.

## Conditions for Swap

Recommendation :

In the Standby PLC, the first section of the program is executed, as it is done in the Primary PLC. For this reason, it is recommended to avoid programming swaps in the first section of the application as this may lead to instability of the system.

In order to be able to swap, the system must have one PLC in the Primary state AND the other one in Standby state. Both PLC modules have to be in Run mode. As for every Modicon Hot Standby systems, the hardware, firmware and software configurations have to be identical.

It is mandatory that the Hot Standby system is given some time to ensure its return to stability after swapping. This period of time shall never be less than 15 seconds.

## Performances for swaps

The swap mechanism, that involves both PLC modules, is initiated by the Primary PLC at the end of its MAST cycle. Prior to sending the command that starts a swap, the Primary PLC sends again its database to the Standby PLC, in order to swap on the latest available data. Then the Primary PLC issues the command to the Standby PLC.

The Standby PLC may process it at the beginning of its next MAST task cycle; as soon as the Standby PLC as received it, it checks that the Primary PLC has switched to another mode, and then switches to the Primary state. The new Primary PLC (previously Standby) will manage the equipment under control at the beginning of the next cycle.

---

Therefore, the time required to perform a swap may vary depending on the real-time state of the system, and may last for several MAST cycle times.

Performing periodic swaps in a Premium Hot Standby system is a particular functionality that must be used only for very specific needs (testing or validation for instance) and under a very limited period of time.

Schneider Electric does not recommend using Swaps at production time, as the repetitive use of this feature may lead to system instability.

### **Recommendations for Communication Management During Swaps**

Communication Function Blocks that may be started on the Primary PLC before a swap is performed may not be managed on a Standby PLC. Therefore, they may consume additional resources. After several swaps, too many Communication Function Blocks may have been started, which may lead to a lack of resources. In order to avoid this, we recommend that all Communication Function Blocks are cancelled before performing a swap.

It's mandatory that all the Communication Function Blocks located in the first section of the application have their four management words located in the Non Transfer Area.

We recommend for the other sections to locate the four management words of the Communication Function Blocks in the Non Transfer Area.

The timeout value of the Communication Function Blocks must be re-initialized every time a Function Bloc is used.

It's mandatory to avoid the value "0". (that corresponds to an infinite value)

### **Management of Serial Link During a Swap**

To avoid collisions of messages, we recommend to delay the transmission of data on a serial link by the new Primary for a time equal to  $2 \times \text{MAST duration} + \text{maximum duration of the frame that has to be transmitted}$ .

---

# Chapter 10

## Maintaining

---

### Overview

This chapter provides information about Maintaining a Premium Hot Standby System.

**NOTE:** Read and understand the *Premium and Atrium using Unity Pro Processors, racks and power supply modules Implementation manual*, reference 35010524, before attempting the procedures in this Part.

### What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Verifying the Health of a Premium Hot Standby	214
Premium Hot Standby Detection and Diagnostics	215
Detecting Inoperative Primary CPU- and ETY-sync Links	217
Detecting Inoperative Standby CPU- and ETY-sync Links	218
Detecting Inoperative CPU-sync Links	219
Checking for Identical Application Programs—Checksum	220
Replacing an Inoperative Module	221
Troubleshooting a Hot Standby PLC	223

---

## Verifying the Health of a Premium Hot Standby

### Generating and Sending Health Messages

Health messages are exchanged between the Primary PLC and the Standby PLC as part of the cyclic database transfer.

If the Primary reports an error, the Standby is notified and assumes the Primary role.

If the Standby reports an error, the Primary continues to operate as a standalone. The Monitored ETY modules periodically verify communication with one another.

If the Standby does not receive a message on either link, the Standby will try to determine the cause and assume control if necessary.

If the Primary does not receive a valid response from the Standby, the Primary will operate as if there was no back up available (as if the Primary were a standalone).

### Performing Automatic Diagnostic Tests

The system automatically performs two kinds of diagnostic tests on the Premium Hot Standby CPU:

- Startup tests
- Runtime tests

### Conducting Startup Tests

Startup diagnostic testing on the Premium Hot Standby PLC attempts to detect hardware errors in the module before the application is allowed to run.

If the module does not meet specifications during its startup diagnostic tests, it will remain Offline and will not communicate with the other Premium Hot Standby PLC.

### Conducting Runtime Tests

Runtime tests are related to the interface between the main processor (CPU) and the Ethernet embedded coprocessor of the Premium Hot Standby controller.

If the coprocessor does not meet these runtime test specifications, the associated Premium Hot Standby PLC remains Offline and will not communicate with the other PLC.

---

## Premium Hot Standby Detection and Diagnostics

### Important Information

If...	Then...
Component of Primary becomes inoperative	Control shifts to Standby
Component of Standby becomes inoperative	Standby goes offline
CPU-sync link becomes inoperative	Standby goes offline

### Finding Diagnostic Information with Unity Pro

Diagnostic messages and Switchover events are logged in the diagnostic buffer. To view the log,

Step	Action
1	Select <b>Tools</b> → <b>Diagnostic Viewer</b> from the main menu.

**NOTE:** The diagnostic messages that are stored in the diagnostic buffer are not transferred from the Primary to the Standby. These messages are written to persistent memory, and are usually not subject to loss due to unexpected system events. In the event of a Switchover, you may check these messages through Unity Pro by connecting to the old Primary PLC.

### Finding More Information in this Manual

Refer to the following sections:

Detected Event	Refer to section
Primary CPU and ETY-sync link becomes inoperative	See <i>Detecting Inoperative Primary CPU- and ETY-sync Links</i> , <a href="#">page 217</a>
Standby CPU and ETY-sync link becomes inoperative	See <i>Detecting Inoperative Standby CPU- and ETY-sync Links</i> , <a href="#">page 218</a>
CPU-sync link becomes inoperative	See <i>Detecting Inoperative CPU-sync Links</i> , <a href="#">page 219</a>
Application program checksum incorrect	See <i>Checking for Identical Application Programs—Checksum</i> , <a href="#">page 220</a>

For more details on diagnostic event detection, please refer to *Detailed Behavior on Interruption of Power, Communications, or Device Capabilities*, [page 251](#).

Detected Event	Refer to section
Halt or Stop Events on PLC	See <i>Halt Events or Stop Command on PLC</i> , <a href="#">page 254</a>
CPU Hardware or Firmware Becomes Inoperative	See <i>CPU Hardware or Firmware Becomes Inoperative</i> , <a href="#">page 257</a>

<b>Detected Event</b>	<b>Refer to section</b>
Interruption of Supply Power to Main Rack	See <i>Interruption of Supply Power to Main Rack</i> , <a href="#">page 260</a>
ETY Hardware or Firmware (Monitored by Hot Standby CPU) Becomes Inoperative	See <i>ETY Hardware or Firmware (Monitored by Hot Standby CPU) Becomes Inoperative</i> , <a href="#">page 263</a>
ETY Hardware or Firmware (Not Monitored by Hot Standby CPU) Becomes Inoperative	See <i>ETY Hardware or Firmware (Not Monitored by Hot Standby CPU) Becomes Inoperative</i> , <a href="#">page 266</a>
Ethernet Coprocessor Becomes Inoperative	See <i>Ethernet Copro Becomes Inoperative</i> , <a href="#">page 269</a>
Interruption of CPU-sync link between Primary and Standby PLCs	See <i>Interruption of CPU-sync link between Primary and Standby PLCs</i> , <a href="#">page 272</a>
ETY-Sync Link Cable Disconnection with I/O Scanner Active	See <i>ETY-sync Link Cable Disconnection with I/O Scanner Active</i> , <a href="#">page 274</a>
Full Monitored Ethernet I/O Link Disconnection (Both Switches for Monitored I/O Inoperative)	See <i>Full ETY I/O Link Disconnection (Both Switches for Monitored I/O Inoperative)</i> , <a href="#">page 278</a>
Discrete I/O Module Becomes Inoperative	See <i>Discrete I/O Module Becomes Inoperative</i> , <a href="#">page 280</a>
SCP Card in SCY Module Becomes Inoperative	See <i>SCP Card in SCY Module Becomes Inoperative</i> , <a href="#">page 283</a>

---

## Detecting Inoperative Primary CPU- and ETY-sync Links

### Automatic Switchover Not Commanded by the Primary PLC (Unmastered Switchover)

The following table describes the sequence of events when the Primary controller's CPU becomes inoperative and cannot command (master) the Switchover event:

Stages	Description
1	The Coprocessor in the Standby PLC detects a loss of communication on the CPU-sync link (in this example due to an inoperative Primary CPU).
2	After the user-defined Watchdog period has elapsed, the Standby Copro reports this error to the Standby CPU.
3	The Standby CPU sends a message to its local Monitored ETY to get a status of the Primary PLC through the ETY-sync link.
4	The Standby Monitored ETY attempts to check the status of the Primary PLC until its defined timeout period has elapsed. Then the Standby Monitored ETY reports an error state on the Primary PLC to the Standby CPU.
5	The Standby PLC becomes Primary.

### Automatic Switchover Commanded by the Primary PLC (Mastered Switchover)

The following table describes the sequence of events when the Primary controller experiences an event which requires it to enter the Offline state, but does not prevent it from communicating this fact to the Standby:

Stages	Description
1	Before entering the Offline mode, the Primary CPU sends a message to the Standby CPU through the CPU-sync link commanding it to assume the Primary role.
2	The Standby goes to Primary mode.

### Primary ETY-sync link Becomes Inoperative with I/O Scanner Active

The following table describes the sequence of events when the Primary PLC's ETY becomes inoperative and an I/O Scanning service was operating on that device:

Stages	Description
1	The Primary PLC determines the status of the Monitored ETY, and through it any Monitored I/O, once during every MAST task.
2	After receiving an incorrect status, the Primary CPU initiates a mastered Switchover.
3	The Standby goes to Primary mode.

---

## Detecting Inoperative Standby CPU- and ETY-sync Links

### Standby CPU Becomes Inoperative

The following table describes the sequence of events when a Standby CPU becomes inoperative:

Stage	Description
1	The Primary Copro detects that the Standby Copro is not responding normally (in this example, due to the Standby PLC's CPU becoming inoperative).
2	The Primary Copro reports this error to the Primary CPU.
3	The Primary CPU stays Primary and updates the remote station status to Offline in its status register.

### Standby ETY-sync link Becomes Inoperative

The following table describes the sequence of events when the Standby ETY-sync link becomes inoperative (Primary CPU is assumed to be operating normally):

Stage	Description
1	The Standby Monitored ETY notes a loss of communications on the ETY-sync link.
2	The Standby ETY reports the detection of this error to the Standby CPU.
3	The Standby CPU sends a message to the Primary CPU through the CPU-sync link.
4	If the Status is OK, the Primary continues to act as Primary and the Standby will go to Offline because of the disconnection on Standby side. If the status is not OK and the Primary is capable of responding, it will send a "take control" message to the Standby before entering Offline mode.

---

## Detecting Inoperative CPU-sync Links

**NOTE:** Both the Primary and Standby PLC's are capable of detecting a loss of communications on the CPU-sync link. Which PLC actually detects the loss of communications depends on the timing of the event with respect to the MAST task cycle.

### Standby is First to Detect a Loss of Communications on the CPU-sync link

At first,

Step	Action	Result
1	Standby gets no response from the Primary on the CPU-sync link within the programmed Watchdog period	<ul style="list-style-type: none"><li>• There is no more database exchange from Primary to Standby</li><li>• The system is no longer redundant as long as the Copro of the Standby PLC is reporting a loss of communications</li></ul>

---

## Checking for Identical Application Programs—Checksum

### Important Information

Fact	Result
A Hot Standby system requires that both stations must have the same application program.	This requirement prevents the Standby from executing a different application program if transfer of control occurs.

### Standby Checks for Mismatches

Checking for identical application programs

Step	Action	Result
1	At each scan, the application program, checksum (CKSM), is transferred from the Primary to the Standby as part of the database exchange.	The Standby validates the new checksum (CKSM) against its existing checksum (CKSM).
2	Standby determines if mismatch occurs.	<b>1.</b> Mismatch: Standby goes Offline <b>2.</b> No mismatch: system operates normally
3	The controller returns to Online and is the Standby as soon as the application programs are identical.	

---

## Replacing an Inoperative Module

Other than the TSX H57 \*\*\* modules themselves, most In-rack modules compatible with the Premium Hot Standby system can be replaced while the system is running. This is beneficial when one of your installed modules becomes inoperative, because an inoperative module will usually cause a Switchover event, with the result that the inoperative module is now on a rack that is in either the Standby or Offline operating mode. However, do not assume that an inoperative module you are preparing to replace is on the Standby rack. Always check the operating mode of both Premium racks before attempting to replace a module.

### WARNING

#### UNINTENDED EQUIPMENT OPERATION

- Never assume that a PLC is in a certain operating mode before installing, operating, modifying, or servicing it.
- Before acting on a PLC, always positively confirm the operating mode of both Hot Standby PLCs by viewing their LEDs and checking their System Status Words.

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

In the unlikely event that the inoperative module did not cause a Switchover, do not attempt to change the module while it still resides on the Primary rack.

### WARNING

#### UNINTENDED EQUIPMENT OPERATION

- Never attempt to replace a module residing on one of the Primary PLC backplanes. Only modules on the Standby PLC backplanes may be replaced while the system is operational.
- If the module requiring replacement is on the Primary PLC's backplane, perform a manual Switchover and confirm that the Switchover has occurred before proceeding.

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

---

Also, be aware that not all of the modules compatible with the Premium Hot Standby system can be hot swapped, that is, replaced while there is power applied to the rack. Before attempting to replace an inoperative module, observe the following:

** DANGER**

**HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH**

- Read and understand the *Premium and Atrium Using Unity Pro Processors, Racks and Power Supply Modules Implementation Manual*, reference 35010524, and the documentation for any module or accessory you are installing, removing, or maintaining to understand whether power must be removed for these operations.
- Always use a properly rated voltage sensing device to confirm that power is off.
- Replace and secure all covers and elements of the system before reapplying power.
- Confirm that all affected PLCs are loaded with the correct application program before reapplying power.
- Use only the specified voltage for your TSX PSY \*\*\* power supply when placing the system in operation.

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

Finally, observe all of the configuration requirements for identical hardware, firmware, and rack address when replacing a module, or the Standby rack will not be able to exit the Offline mode when the system is restarted.

## Troubleshooting a Hot Standby PLC

### Troubleshooting the PLC

To determine which components have become inoperative, review the LED indicators on the PLC being analyzed, and on its associated Monitored ETY:

CPU LEDs					Monitored ETY LEDs			Diagnostic Status	Description
RUN	ERR	I/O	STS	ACT	RUN	ERR	STS		
								Normal state.	Normal state. CPU in Primary mode.
								Normal state.	Normal state. CPU in Standby mode.
								CPU inoperative.	CPU hardware or firmware has become inoperative. A PLC displaying this status cannot operate as either the Primary or Standby device.
								Coprocessor inoperative.	The Coprocessor in the affected PLC did not meet its startup or runtime diagnostic test specifications.
								Application inoperative.	Multiple conditions may cause an application to cease operation. Some examples: Halt instruction, Watchdog overrun, CPU in Offline mode.
								ETY module inoperative.	Configuration-related: the module is not configured or configuration is in progress.
								ETY module inoperative.	Hardware- or firmware-related: The ETY hardware or firmware has become inoperative and as a consequence the associated CPU enters an Offline state.
								ETY module inoperative.	Software-related: This is a temporary state occurring when the ETY module experiences an event that requires it to reinitialize.

CPU LEDs					Monitored ETY LEDs			Diagnostic Status	Description
RUN	ERR	I/O	STS	ACT	RUN	ERR	STS		
								ETY module inoperative.	<p>Network-related: The STS LED will flash using the 500 ms on, 500 ms off pattern, but will do so in groups with longer pauses in between to provide additional information on the cause of the inoperative state.</p> <ul style="list-style-type: none"> <li>• 2 flashes per group: the module has no MAC address.</li> <li>• 3 flashes per group: Ethernet cable not connected to the module or switch.</li> <li>• 4 flashes per group: the module has a duplicate IP address.</li> <li>• 5 flashes per group: the module is configured as a BOOTP client and is awaiting a response from a BOOTP server.</li> <li>• 6 flashes per group: the module has an invalid IP address (set to default address).</li> </ul>

LED	Description
	Permanently ON
	Normal flashing (500 ms ON, 500 ms OFF)
	Standby flashing (2.5 s ON, 500 ms OFF)
	Offline flashing (2.5 s OFF, 500 ms ON)
	OFF
	State of this LED not significant in diagnosing this condition

---

# Part III

## Modifying and Upgrading

---

### Purpose

This part describes Modifying and Upgrading in a Premium Hot Standby System.

- Handling Application Modification
- Handling PLC Firmware Upgrades

### What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
11	Handling Application Modification	227
12	Handling PLC Firmware Upgrades	235



---

# Chapter 11

## Handling Application Modification

---

### Overview

This chapter provides information about application modification in a Premium Hot Standby system.

### What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Understanding Premium Hot Standby Logic Mismatch	228
Online/Offline Modifications to an Application Program	229

## Understanding Premium Hot Standby Logic Mismatch

### Needing Identical Application Programs

In a redundant system under normal operating conditions, both controllers must be loaded with identical application programs. When these application programs differ, the resulting state is called a "logic mismatch". The application program is updated every scan by transferring data from the Primary to the Standby. Only the Standby controller detects a logic mismatch and reports this error to the Primary.

Differences in the following will cause the Standby controller to report a logic mismatch:

- Application Program executable code
- Animation tables
- Comments (on variables and types)

**NOTE:** Animation Tables and Variable/Type Comments

Both animation tables and comments (on variables and types) may be excluded from the detection of a logic mismatch by not being included in the upload information. Exclude these values by:

- `Selecting Tools | Project Settings | Build tabs` (this is the default).
- In the Upload Information area, select "without".
- Perform an upload of the application program per the instructions in this section.

When a mismatch exists, the Standby Controller goes to Offline, and a Switchover cannot occur.

### Causing a Mismatch

In a Premium Hot Standby System, if the user does any of the following, the Standby will go into Offline mode:

- Conduct an online modification of the application program in the Standby PLC while the Primary is controlling the process in the Run / Primary mode.

**NOTE:** Unity Pro does not allow the online modification of the Standby controller until the online modification of the Primary controller has been completed. This is different than offline modifications, which are applied to the Standby PLC first.

- Conduct an online modification of the application program in the Primary while the Primary is controlling the process in the Run / Primary mode.
- Download an application program modified offline to the Standby.

**NOTE:** This is a normal occurrence during offline modifications, which require a full application download.

**NOTE:** The online modification of an application program occurs when your Hot Standby system is connected to Unity Pro and you:

- Modify the executable code by adding, suppressing or changing an instruction in the code,
- Modify a configuration parameter by changing a device or system value in the Unity Pro configuration screen.

## Online/Offline Modifications to an Application Program

### Overview

When the Primary and Standby PLCs have different application programs or system configurations, the Hot Standby configuration loses its high availability. In these cases, the Standby PLC enters Offline mode and a Switchover cannot occur. Unlike Quantum Hot Standby systems, Premium Hot Standby systems do not continue to operate when a logic mismatch exists.

The following procedures describe how the user can modify the application in the two PLCs of a Premium Hot Standby system with a minimum impact on the process. The two types of modifications possible are:

- Online Modifications
- Offline Modifications

### Online Modifications Allowed

The online modification of an application program occurs when your Hot Standby system is connected to Unity Pro and you:

- Modify the executable code of your application program by adding, suppressing or changing an instruction in the code.
- Modify a configuration parameter for any installed device or for the system by changing a value in the applicable Unity Pro configuration screen.

An online modification of your Hot Standby application program and/or configuration will normally have the least impact on your operations, because online modifications require only partial application downloads and will usually allow one PLC to remain in a Run state and continue to manage your processes.

The following table summarizes the modifications that are possible as online modifications. You may make online modifications to:

Modifications	Description
General	<ul style="list-style-type: none"> <li>● Name of station, program, section</li> <li>● Comment applied to a station, configuration, program, section</li> <li>● Documentation summary</li> <li>● Animation tables</li> <li>● Integrated operator screen</li> <li>● Functional view</li> <li>● Security information: passwords, protection attributes</li> </ul>
Program	<ul style="list-style-type: none"> <li>● Sections of program: add, delete, change execution order</li> <li>● Modify the code of the section (program section, SR, Action, DFB sections)</li> <li>● Modify the code of SFC chart</li> </ul>
Configuration / communication	Change I/O module parameters

Modifications	Description
Global variables (used in animation table or operator screen)	<ul style="list-style-type: none"> <li>● Symbol on a used variable</li> <li>● Topological address on a used variable</li> <li>● Initial value on a used variable</li> <li>● Comment on a used variable</li> <li>● Create, remove or modify unused variables (EDT, DDT)</li> <li>● Create, remove or modify unused variables (FB)</li> </ul>
Used DFB	<ul style="list-style-type: none"> <li>● All comments</li> <li>● Add a private or public variable</li> <li>● Delete or change unused private variable</li> <li>● Initial value of parameters and variables</li> <li>● Section of DFB: add, delete, change execution order</li> <li>● Modify the code of a section</li> <li>● Create a new DFB type</li> <li>● Delete an unused DFB type</li> </ul>
Used DDT	<ul style="list-style-type: none"> <li>● Create a new DDT type</li> <li>● Delete an unused DDT type</li> </ul>

### Executing the Procedure Online

To make online modifications to an application program (logic program or project) in the Primary controller, follow these steps.

Step	Action
1	Thoroughly test any modifications you plan to make on a non-operational platform before attempting an online modification of the operational system.
2	<p>Confirm the operating modes of both PLCs:</p> <ul style="list-style-type: none"> <li>● Verify that PLC A is in the Run / Primary operating mode.</li> <li>● Verify that PLC B is in the Run / Standby operating mode.</li> </ul>
3	<p>Connect Unity Pro to the Primary PLC (PLC A) and enter the "online" connection state. Make any of the allowed online modifications presented on the preceding page. If you have appropriately planned for and tested these modifications, you will observe:</p> <ul style="list-style-type: none"> <li>● The Primary PLC will continue to run as Primary</li> <li>● The Standby PLC will detect a logic mismatch and enter the Offline mode. The system is no longer operating redundantly at this point.</li> </ul> <p>If the Primary PLC did not continue to run as the Primary, then system operations can be restored by accomplishing these steps:</p> <ul style="list-style-type: none"> <li>● Connect Unity Pro to PLC B</li> <li>● Perform a Stop / Run command</li> </ul> <p>This will cause PLC B to reinitialize as the Primary PLC. At this point, you should restore the application program on PLC A to its previous working state, and conduct more testing on a non-operational test platform before proceeding. Start again at Step 1 of this procedure when you are ready.</p>
4	After the online modifications have been made to the Primary PLC, evaluate these modifications for proper operation. If the modified program is operating as expected, save (upload) the application from the PLC to your Unity Pro PC.

Step	Action
5	Connect Unity Pro to PLC B and download the application file from the PC to PLC B. <ul style="list-style-type: none"> <li>• During the download, PLC B will enter a "Non Configured" or "Non Conf" state.</li> <li>• PLC B will revert to a Stop / Offline state when the transfer is complete.</li> </ul>
6	Send a Run command to PLC B. This PLC will now start in the Run / Standby mode. This system is once again operating redundantly.

## WARNING

### UNINTENDED EQUIPMENT OPERATION

- Never assume that a PLC is in a certain operating mode before installing, operating, modifying, or servicing it.
- Before acting on a PLC, always positively confirm the operating mode of both Hot Standby PLCs by viewing their LEDs and checking their System Status Words.

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

**NOTE:** Unity Pro does not allow the online modification of the Standby controller until the online modification of the Primary controller has been completed. This is different than offline modifications, which are applied to the Standby PLC first.

**NOTE:** An online modification in an animation table or in a comment will not generate a logic mismatch if the Animation tables and Comments option is not checked in the Build Tab of Tools | Project Settings.

### Offline Modification

Make offline modifications on the Standby PLC if the modifications will require a complete download of the application.

The following table describes modifications that require a complete application download:

Modifications	Description
Program	Modify the code of EVT sections
Configuration / communication:	<ul style="list-style-type: none"> <li>• Add, move, remove an I/O module</li> <li>• Change memory sizes using the configuration screen</li> </ul>
Global variables (used in animation table or operator screen)	Remove a used variable
Used DFB	<ul style="list-style-type: none"> <li>• Change the type name of used DFB</li> <li>• Add a parameter</li> </ul>

### Executing the Procedure Offline

To make offline modifications to an application program (logic program or project) in the Standby controller, follow these steps:

Step	Action
1	Thoroughly test any modifications you plan to make on a non-operational platform before attempting an offline modification of the operational system.
2	Confirm the operating modes of both PLCs: <ul style="list-style-type: none"> <li>● Verify that PLC A is in the Run / Primary operating mode.</li> <li>● Verify that PLC B is in the Run / Standby operating mode.</li> </ul>
3	Connect Unity Pro to the Standby PLC (PLC B) and download the modified application program. <ul style="list-style-type: none"> <li>● During the download, PLC B will enter a "Non Configured" or "Non Conf" state.</li> <li>● PLC B will revert to a Stop / Offline state when the transfer is complete.</li> <li>● PLC A will remain in Run / Primary mode during the download, so the system will be active but not redundant at the conclusion of this step.</li> </ul>
4	Issue a Stop Command to the Primary PLC (PLC A). <ul style="list-style-type: none"> <li>● PLC A will enter the Stop / Offline mode.</li> <li>● The system is no longer active, nor is it redundant.</li> </ul>
5	Issue a Run Command to the old Standby PLC (PLC B). If you have appropriately planned for and tested your modified application, you will observe: <ul style="list-style-type: none"> <li>● PLC B will enter the Run / Primary mode</li> <li>● Because PLC A is still in the Stop / Offline mode, the system is now active, but it is not yet redundant.</li> </ul> <p>If PLC B does not start into the Run / Primary mode, then issue a Stop command and upload a working (unmodified) version of the application program to both PLCs. Restore the system to an operational, redundant status. Conduct further testing of the modified application program on a non-operational test platform before proceeding.</p>
6	Evaluate the performance of the modified application on PLC B (the new Primary) for proper operation.
7	Connect Unity Pro to PLC A and download the modified application file from the PC. <ul style="list-style-type: none"> <li>● During the download, PLC A will enter a "Non Configured" or "Non Conf" state.</li> <li>● PLC A will revert to a Stop / Offline state when the transfer is complete.</li> </ul>
8	Send a Run command to PLC A. This PLC will now start in the Run / Standby mode. This system is once again operating redundantly.

## ⚠ WARNING

### UNINTENDED EQUIPMENT OPERATION

- Never assume that a PLC is in a certain operating mode before installing, operating, modifying, or servicing it.
- Before acting on a PLC, always positively confirm the operating mode of both Hot Standby PLCs by viewing their LEDs and checking their System Status Words.

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

**NOTE:** Offline modifications have a greater impact on your system than online modifications because the offline method always requires system downtime. Further, offline modifications result in a Cold Start of PLC B (the PLC acting as the Standby before the procedure was started). In other words, offline modifications will result in PLC B restarting with a reinitialized data context. Plan your offline modifications of the application program and Hot Standby configuration to account for the Cold Start of PLC B.

## ⚠ WARNING

### UNINTENDED EQUIPMENT DAMAGE

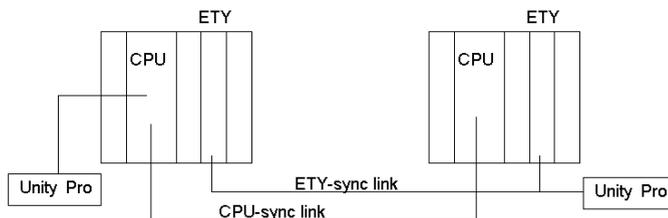
Plan your offline modifications to account for the Cold Start of PLC B.

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

### Possible Connections for Online and Offline Modifications

When conducting either type of application modification (online or offline), you may connect Unity Pro to the Hot Standby controllers through the Uni-Telway or USB terminal ports or via the ETY-sync link (Ethernet) connection.

The following illustration displays the possible connections:



**NOTE:** Schneider Electric does not recommend the use of the Ethernet connection method to perform an offline application modification. During the modification process, the PLC operating modes change frequently, resulting in multiple changes to device IP addresses. Each such change in IP address will cause a loss of communication with Unity Pro. This will increase your system downtime and required you to manually reconfigure the new IP addresses in Unity Pro before the connection can be reestablished. If you do decide to pursue an offline modification over the ETY-sync link:

- When you receive each notice that communication with the PLC has been lost, manually set the IP address in Unity Pro to the PLC's new address.
- After reconnecting, confirm that you have defined the correct IP address by observing the:
  - Unity Pro Status bar.
  - Link status (Offline, Different, Equal).
  - Hot Standby PLC Status (PLC name A/B, PLC state (Primary, Standby, Offline)).
  - Address of the connected PLC.

---

# Chapter 12

## Handling PLC Firmware Upgrades

---

### Overview

In this chapter you will find information regarding the OS upgrade method for a Premium Hot Standby System. Upgrading allows you to update the OS for the Standby controller while the process is still being controlled by the Primary controller.

### What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Overview of Premium Hot Standby Firmware Upgrades	236
Executing the Firmware Upgrade Procedure	237

## Overview of Premium Hot Standby Firmware Upgrades

### Unity OSLoader

The Unity OSLoader tool includes an "Executive Upgrade" feature that allows the Standby controller's firmware (both CPU and Copro) to be upgraded while the Primary controller continues to control the process. However, during the upgrade, the system can no longer be considered redundant. That is, there is no Standby available to assume control if the Primary should become inoperative before the Standby upgrade is complete.

### Upgrading Firmware Without Stopping

Under normal operating conditions, both controllers in a redundant system must have the same versions of firmware.

In fact, the Hot Standby PLCs have built-in tests to detect if there is a firmware mismatch.

Normally, when a firmware mismatch exists, the Standby controller detects this and goes into the Offline operating mode. If this occurs, a Switchover is no longer possible and the system is not redundant.

However, the Executive Upgrade feature, controlled by the Command Register system bit %SW60.4, allows us to conduct a firmware upgrade on the Standby PLC without causing it to enter the Stop / Offline mode. It is important to reset this bit to its default value of 0 after the firmware upgrade has been completed.

## WARNING

### UNINTENDED EQUIPMENT OPERATION

- Follow the firmware upgrade procedure (see *Executing the Firmware Upgrade Procedure, page 237*).
- Always refresh your application program after a firmware download.
- When the firmware upgrade has been completed, restore the Primary PLC's Command Register Bit 4 (%SW60.4) to a value of 0.

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

**NOTE:** Even when conducting an Executive Upgrade, this upgrade will only be possible if a Hot Standby-compatible firmware version is used. When the Unity OSLoader first attempts to connect to a CPU, it will check the CPU's Hardware ID against the allowed Hardware IDs defined in the firmware binary file. If there isn't a match, the Unity OSLoader will not establish the connection. OS upgrade is possible only with compatible firmware.

## Executing the Firmware Upgrade Procedure

### General

Perform an a PLC Firmware upgrade using the installed Unity OSLoader tool.

### How to Perform a Firmware Upgrade

Follow these steps.

Step	Action
1	Connect Unity Pro to the Primary PLC (PLC A) through the Uni-Telway terminal port
2	Access Command Register %SW60; set bit 4 to 1 (OS version mismatch allowed)
3	Stop the Primary (PLC A). Ensure the Standby (PLC B) becomes Primary
4	Disconnect Unity Pro from PLC A
5	Open the OSLoader tool
6	Download the new firmware to PLC A
7	After completing the OS download, perform application program transfer
8	Put PLC A in RUN mode. Ensure PLC A becomes Standby
9	Connect Unity Pro to the other PLC (PLC B, currently acting as the Primary) through the Uni-Telway terminal port
10	Stop the Primary (PLC B). Ensure the Standby (PLC A) becomes Primary
11	Disconnect Unity Pro from PLC B
12	Open the OSLoader tool
13	Download the new firmware to PLC B
14	After completing the OS download, perform application program transfer
15	Put PLC B in RUN mode. Ensure PLC B becomes Standby
16	Perform a Switchover. Confirm Standby becomes Primary
17	Connect to the Primary and access Command Register %SW60; set bit 4 to 0 (OS version mismatch not allowed)

 **WARNING**

**UNINTENDED EQUIPMENT OPERATION**

- Never assume that a PLC is in a certain operating mode before installing, operating, modifying, or servicing it.
- Before acting on a PLC, always positively confirm the operating mode of both Hot Standby PLCs by viewing their LEDs and checking their System Status Words.

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

---

# Appendices

---



## At a Glance

The appendices for the Premium Hot Standby are included here.

For additional information on diagnostics, refer to TSX P57/TSX H57 processors: diagnostics.

The Unity Pro online help also provides useful information on error codes and internal bits and words.

## What Is in This Appendix?

The appendix contains the following chapters:

Chapter	Chapter Name	Page
A	Additional Information	241
B	Detailed Behavior on Interruption of Power, Communications, or Device Capabilities	251



---

# Appendix A

## Additional Information

---

### Overview

This chapter describes the design specifications and error codes.

### What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Additional Premium Hot Standby Specifications	242
TextIDs	249

## Additional Premium Hot Standby Specifications

### Program and Data Memory Capacity

The following table presents the Program and Data Memory capacity of the CPUs:

Services		TSX H57 24M	TSX H57 44M
Maximum application size in Internal SRAM (Program + data + Ets (1) + symbols + OLC)		192 KB	440 KB
Maximum application size in PCMCIA	Program + Ets + symb. in PCMCIA	768 KB	2048 KB
	Max On line modif. area PLC	256 KB	512 KB
	Data in internal SRAM	192 KB	440 KB
Maximum data storage size (only in PCMCIA)	Legacy EFs	8 MB	16 MB
	DOS Files (SRAM)	Not available	Not available
Located data %MW	Max	32464 Words	
	Default	1024 Words	
	Min.	0 Words	
Located data %M	Max	8056 Bits	32634 Bits
	Default	512 Bits	
	Min.	0 Bits	
Located data %KW	Max	32760 Words	
	Default	256 Words	
	Min.	0 Words	
Located data %SW		168 Words	
Located data %S		128 Bits	
Unlocated data max size (2):		See Note 3	
<ul style="list-style-type: none"> <li>● EDT + DDT</li> <li>● EFB/DFB</li> </ul>			
<ul style="list-style-type: none"> <li>● (1): Empty Terminal Support.</li> <li>● (2) <ul style="list-style-type: none"> <li>● EDT: Elementary Data Types (bool, integers, date, real).</li> <li>● DDT: Derived Data Types (structures).</li> <li>● EFB / DFB: Function Blocks.</li> </ul> </li> <li>● (3): The memory available for unlocated data (EDT, DDT, and Function Blocks) is the size of the physical memory installed less the memory allocated to the located data.</li> </ul>			

**NOTE:** EDT and DDT are in the same memory segment. There is one memory segment per instance of EFB/DFB.

## Application Structure

The following table presents the Application Structure of the CPUs:

Services		TSX H57 24M	TSX H57 44M
MAST tasks		1 cyclic/periodic	
FAST tasks. See Note 1		1 periodic	
Auxiliary tasks		0	
Event interrupt tasks (I/O Event + Timer Event). See Note 1		0 to 63	
IO Events (Local IO). See Note 1		0 to 63 <ul style="list-style-type: none"> <li>● Priority 0 (highest priority): Assigned to %EVT0</li> <li>● Priority 1: Assigned to %EVT1 through %EVT63</li> </ul>	
Timer interrupt event		0	
Number of channels (Local I/O) per event	Discrete I/O	128	
	Analog I/O	16	
	Other	16	
<p><b>Note 1:</b> MAST tasks should be used exclusively in Premium Hot Standby systems to transfer the Primary's system and user application data to the Standby controller. Preemptive, asynchronous, or interrupt-driven tasks and programming methods, including the use of FAST tasks, events, and edge triggers, etc., should not be used. They can impact the performance of the MAST tasks and cause discrepancies between Primary and Standby output values in the event of a Switchover.</p>			

## WARNING

### UNINTENDED EQUIPMENT OPERATION

Do not use asynchronous, preemptive, or interrupt-driven tasks to program the outputs of your Premium Hot Standby System. Only MAST tasks support data synchronization between the Primary and Standby Controllers.

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

## Application Language and Embedded Communication Ports

The following table presents the Application Language and Embedded communication ports of the CPUs:

Services	TSX H57 24M	TSX H57 44M
Application Languages		
Function Block Diagram (FBD)	Yes	
Ladder Logic	Yes	
Structured Text	Yes	
Instruction List	Yes	
Sequential Function Chart (SFC)	Yes, with some restrictions ( <a href="#">see page 39</a> ).	
Derived Function Block (DFB)	Yes, with some restrictions ( <a href="#">see page 38</a> ).	
Expert Function (EF)/EFB	Yes, with restrictions for SFC / Grafset programming ( <a href="#">see page 39</a> ) and parameters runtime modification ( <a href="#">see page 39</a> ).	
PL7 Sequential Function Block (SFB)	No For more information refer to PL7 Warm Standby functions. ( <a href="#">see page 38</a> )	
Embedded communication ports		
Legacy Terminal port	Physical layer	One RS485
	Speed	19200 baud
	Protocol	Uni-Telway M/S ASCII
USB terminal port	One device connector USB V1.0 12MB/s	

## WARNING

### UNINTENDED EQUIPMENT OPERATION

Do not use asynchronous, preemptive, or interrupt-driven tasks to program the outputs of your Premium Hot Standby System. Only MAST tasks support data synchronization between the Primary and Standby Controllers.

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

 **WARNING****UNINTENDED EQUIPMENT OPERATION**

Do not use Derived Function Blocks (DFBs) or the TON, TOFF, and TP function blocks in Section 0 of your application program.

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

 **WARNING****UNINTENDED EQUIPMENT OPERATION**

- Do not program your application so that it changes Expert Function parameters unless you also program your application to transfer these changes to the Standby PLC during each MAST task.
- Do not manually modify Expert Function parameters using the Unity Pro debug screen while the system is operational.

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

 **WARNING****UNINTENDED EQUIPMENT OPERATION**

Do not use the PL7 Warm Standby function blocks ([see page 38](#)) in a Premium Hot Standby system.

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

### Memory Services and Devices

The following table presents the Memory Services and Devices of the CPUs:

Services	TSX H57 24M	TSX H57 44M
Application Backup	No	
Data storage with Legacy EF (Init, Read, Write)	Yes in memory card's data storage	
Supported SRAM PCMCIA (Max application size according to PLC characteristics)	<ul style="list-style-type: none"> <li>● TSX MRP P 128K</li> <li>● TSX MRP P 224K</li> <li>● TSX MRP P 384K</li> <li>● TSX MRP C 448K</li> <li>● TSX MRP C 768K</li> <li>● TSX MRP C 001M</li> <li>● TSX MRP C 01M7</li> <li>● TSX MRP C 002M</li> <li>● TSX MRP C 003M</li> <li>● TSX MRP C 007M</li> </ul>	
Supported FLASH PCMCIA (Max application size according to PLC characteristics)	<ul style="list-style-type: none"> <li>● TSX MFP P 128K</li> <li>● TSX MFP P 224K</li> <li>● TSX MCP C 224K</li> <li>● TSX MFP P 384K</li> <li>● TSX MFP P 512K</li> <li>● TSX MCP C 512K</li> <li>● TSX MFP P 001M</li> <li>● TSX MFP P 002M</li> <li>● TSX MCP C 002M</li> <li>● TSX MFP P 004M</li> </ul>	
Supported Data storage	<ul style="list-style-type: none"> <li>● TSX MRP F 004M</li> <li>● TSX MRP F 008M</li> </ul>	

## **WARNING**

### **UNINTENDED EQUIPMENT OPERATION**

Do not use the SAVE\_PARAM function in a Premium Hot Standby system.

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

 **WARNING****UNINTENDED EQUIPMENT OPERATION**

When using the T\_COM\_MB IODDT function to determine the Modbus protocol in use, do not query the high byte of the PROTOCOL variable.

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

 **WARNING****UNINTENDED EQUIPMENT OPERATION**

Do not change the initial values of declared variables using the System Bit %S94.

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

 **WARNING****UNINTENDED EQUIPMENT OPERATION**

Follow the suggested procedure below when using asynchronous communication function blocks.

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

The following procedure should be used to allow asynchronous communication function blocks to automatically resume operation after a Switchover:

- Program your application so that it stores the values of all function block management parameters in the Non-Transfer Memory Area (%MW0...%MW99).
- Initialize the Length parameter each time the function block is called.
- Use a separate Timer function block as a replacement for the communication function block's Timeout parameter.

 **WARNING****UNINTENDED EQUIPMENT OPERATION**

Do not change discrete output bit values for redundant outputs in the first section (section 0) of your application program.

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

### OS Download/Application Performance/System Overhead

The following table presents the OS Download, Application Performances (PCMCIA), and System Overhead of the CPUs:

Services	TSX H57 24M	TSX H57 44M
Firmware Download		
CPU firmware download	Yes Uni-Telway terminal port	
Copro firmware download	Yes through Hot Standby Ethernet (CPU-sync link) port only	
I/O modules firmware download	No	
Application performance (in Kilo-instructions per millisecond, or Kins/ms)		
100% boolean	15.5 Kins/ms	
example: 65% boolean instructions,35% numeric instructions	11.4 Kins/ms	
System overhead		
MAST task	1ms	
FAST task	0.08 ms	

### Miscellaneous Characteristics

The following table presents the Miscellaneous Characteristics of the CPUs:

Services	TSX H57 24M	TSX H57 44M
Processor format	Double width	
Microprocessor	Pentium 166 Mhz	
Processor Electrical consumption on 12V (with one memory card) 5V not used	mA typ.	1780 mA
	mA max. (1)	2492 mA
	W typ.	9.1 W
	W max.	12.7 W
Default rack	TSX RKY 6	
Default Power supply	TSX PSY 2600	
PCMCIA slots	Slot A	Type I / 5V
	Slot B	Type III / 5V
Realtime Clock	Yes	
RTC synchronization with dual CPU	No	
(1): max = typical consumption x 1.4		

## TextIDs

### TextIDs

TextIDs define the diagnostic messages written in the diagnostic buffer.

TextIDs switching from Primary to Offline

TextID	Diagnostic message
16#32C9	System halt
16#32CA	Remote IO inoperative
16#32CB	ETH device inoperative
16#32CC	ETH communication loss
16#32CD	Stop PLC command
16#32CF	Offline Command register request

TextIDs switching from Standby to Offline

TextID	Diagnostic message
16#32D0	System halt
16#32D1	Remote IO inoperative
16#32D2	ETH device inoperative
16#32D3	ETH communication loss
16#32D4	Stop PLC command
16#32D6	Offline Command register request

TextIDs switching from Standby to Primary

TextID	Diagnostic message
16#32D7	Control command over ETH
16#32D8	Control command over RIO

TextIDs switching from Offline to Primary/Standby

TextID	Diagnostic message
16#32D9	Switch from Offline to Primary
16#32DA	Switch from Offline to Standby



---

# Appendix B

## Detailed Behavior on Interruption of Power, Communications, or Device Capabilities

---

### Overview

In this chapter you will find the descriptions of Premium Hot Standby behavior when power and communication interruptions occur, or in the event a device becomes inoperative.

### What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Overview	252
Halt Events or Stop Command on PLC	254
CPU Hardware or Firmware Becomes Inoperative	257
Interruption of Supply Power to Main Rack	260
ETY Hardware or Firmware (Monitored by Hot Standby CPU) Becomes Inoperative	263
ETY Hardware or Firmware (Not Monitored by Hot Standby CPU) Becomes Inoperative	266
Ethernet Copro Becomes Inoperative	269
Interruption of CPU-sync link between Primary and Standby PLCs	272
ETY-sync Link Cable Disconnection with I/O Scanner Active	274
Full ETY I/O Link Disconnection (Both Switches for Monitored I/O Inoperative)	278
Discrete I/O Module Becomes Inoperative	280
SCP Card in SCY Module Becomes Inoperative	283

## Overview

### Introduction

A first level of Hot Standby diagnosis can be done through the %SW61 status register that is managed locally by each Hot Standby PLC.

You can obtain significantly more diagnostic information by programming your application to test the status of each In-rack module and its associated channels.

**NOTE:** If you desire that such custom diagnostic information be maintained for both the Primary and Standby racks, be sure to observe the following:

- Accomplish the collection, processing, and storage of additional diagnostic information in the section 0 of your application program. Otherwise, the standby PLC does not report current information.
- Program your application so that the additional diagnostic information is stored in nontransfer area (%MW0 – %MW99). When you do this, the primary PLC cannot overwrite the standby PLC information during the MAST task database transfer
- If any of the PLC's additional diagnostic information is reported to the primary PLC, use the reverse transfer system words (%SW62 ... %SW65). Remember that the nontransfer area (%MW0 – %MW99) also provides ample space to preprocess diagnostic information that will be sent to the primary PLC. For example, the Standby PLC might use this area to assemble %S status bits into a %SW word for insertion into the %SW62 reverse transfer word.

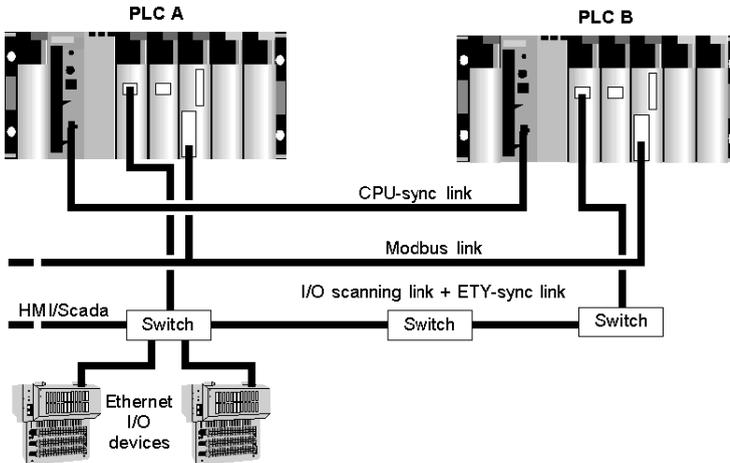
The following pages present an example Hot Standby system, and then describe the responses of this system to various service or hardware events.

### Example of Configuration

The referenced configuration is:

- PLC A and PLC B with the following modules:
  - Power supply (PS)
  - Hot standby processor (in slot 0)
  - Monitored ETY module (in slot 2)
  - Ethernet communication (in slot 3)
  - Modbus communication (SCY with SCP 114) in slot 4
  - In-rack Discrete modules (DIS IN and DIS OUT) in slots 5 and 6
- Two network switches providing a connection between the Ethernet I/O scanner and a SCADA or HMI device.
- CPU-sync link between the two CPU

The following illustration displays an example of configuration:



### Standby Reverse Register

In the example, only one %MW is used and copied in the reverse register %SW62.

The Standard rack state is:

- %MWx.0 / %SW62.0: Not used in this example
- %MWx.1 / %SW62.1: Not used in this example
- %MWx.2 / %SW62.2: Not used in this example
- %MWx.3 / %SW62.3: discrete input module state by copy of %I0.5.mod.err
- %MWx.4 / %SW62.4: discrete output module state by copy of %I0.6.mod.err
- %MWx.5 / %SW62.5: SCY state by copy of %I0.4.mod.err
- %MWx.6 / %SW62.6: SCP in SCY state by copy of %I0.4.1.err
- %MWx.7 / %SW62.7: ETY state by copy of %I0.3.mod.err

## Halt Events or Stop Command on PLC

### Halt Event or Stop Command on Primary PLC

The following table presents the effects of Halt events or a Stop command on the Primary PLC:

<p>Before the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at the end of the task cycle.</li> <li>● PLC B: PLC A output applied at the end of task cycle.</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with ethernet devices are open: I/O scanner is active.</li> <li>● PLC B: all connections with ethernet devices are closed: I/O scanner is not active.</li> </ul>	<p>The diagram shows two PLC racks connected to a central Ethernet I/O scanner + SCADA. PLC A (Primary) is on the left, and PLC B (Standby) is on the right. Each rack contains modules for PS, CPU, ETH Port, ETY, SCP 114, DIS IN, and DIS OUT. Two switches, labeled '@' and '@+1', connect the racks to the scanner. In this state, PLC A's ETH Port and SCP 114 are shaded, indicating they are active.</p>
<p>Event</p> <ul style="list-style-type: none"> <li>● HALT instruction</li> <li>● Watchdog overflow</li> <li>● Program execution error (division by 0, overflow, etc.) with %S78 = 1</li> <li>● STOP command</li> </ul> <p>Event causes an automatic Switchover: YES  <b>Note:</b> Program blocking errors may or may not result in a Halt instruction, depending on user configuration.</p>	<p>The diagram is identical to the 'Before the event' state. However, in this state, the CPU and ETH Port modules of PLC A are shaded, indicating a switchover event has occurred.</p>
<p>After the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: fallback position</li> <li>● PLC B: calculated and applied at the end of the task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are closed: I/O scanner is not active</li> <li>● PLC B: all connections with Ethernet devices are open: I/O scanner is active</li> </ul>	<p>The diagram shows the state after the event. PLC A is now 'Offline' and PLC B is 'Primary'. The ETH Port and SCP 114 modules of PLC B are shaded, indicating that the I/O scanner is now active on PLC B.</p>

Global status	Communication status	Customer diagnostic through Ethernet address @
The process is still active but the HSBY system is no longer redundant as long as the PLC A is in HALT or STOP mode.	Both PLCs are accessible through the Uni-Telway and USB ports, Modbus and Ethernet links for diagnostics	%SW61 = 1000 0000 0010 0110 ● the accessed PLC is PLC B / primary ● the other PLC is PLC A / offline  %SW62 = Not significant because one of the two PLCs is Offline or Not Responding

### Halt Event or Stop Command on Standby PLC

The following table presents the effects of Halt events or a Stop command on the Standby PLC:

Before the event	
In-rack Discrete I/O state: <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at the end of the task cycle.</li> <li>● PLC B: PLC A output applied at the end of task cycle.</li> </ul> Remote I/O state: <ul style="list-style-type: none"> <li>● PLC A: all connections with ethernet devices are open: I/O scanner is active.</li> <li>● PLC B: all connections with ethernet devices are closed: I/O scanner is not active.</li> </ul>	<p>The diagram shows two PLC racks connected to a network. The left rack is labeled 'PLC A Primary' and the right rack is 'PLC B Standby'. Each rack has modules for PS, CPU, ETY, SCY, DIS IN, and DIS OUT. Below the CPU modules are 'ETH Port' and 'SCP 114'. The racks are connected to two switches labeled '@' and '@+1'. Above the switches is an 'Ethernet I/O scanner + SCADA' connection. In the 'Before the event' state, PLC A Primary is active, and PLC B Standby is in standby mode.</p>
Event	
<ul style="list-style-type: none"> <li>● HALT instruction</li> <li>● Watch dog overflow</li> <li>● Program execution error (division by 0, overflow, etc.) with %S78 = 1</li> <li>● STOP command</li> </ul> Event causes an automatic Switchover: NO Note: Program blocking errors may or may not result in a Halt instruction, depending on user configuration.	<p>The diagram is identical to the one above, but now PLC A Primary is in a HALT or STOP state, and PLC B Standby has switched to Primary mode. Its I/O scanner and SCADA connections are now active.</p>

<p>After the event</p>		
<p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at the end of the task cycle</li> <li>● PLC B: fallback position</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>	<p>The diagram shows two PLC racks connected via Ethernet switches. The left rack is labeled 'PLC A Primary' and contains modules: PS, CPU, ETH Port, ETY, SCP 114, DIS IN, and DIS OUT. The right rack is labeled 'PLC B Offline' and contains modules: PS, CPU, ETH Port, ETY, SCP 114, DIS IN, and DIS OUT. Two switches are connected to the racks: the first switch (labeled '@') is connected to PLC A, and the second switch (labeled '@+1') is connected to PLC B. An 'Ethernet I/O scanner + SCADA' is connected to the first switch. A bracket at the bottom indicates the network connection between the two racks.</p>	
<p>Global status</p>	<p>Communication status</p>	<p>Customer diagnostic through Ethernet address @</p>
<p>The process is still active but the HSBY system is no longer redundant as long as the PLC B is in a Halted state or STOP mode.</p>	<p>Both PLCs are accessible through the Uni-Telway and USB ports, Modbus links and Ethernet links for diagnostics</p>	<p>%SW61 = 1000 0000 0000 0110</p> <ul style="list-style-type: none"> <li>● the accessed PLC is PLC A: primary</li> <li>● the other PLC is PLC B: offline</li> </ul> <p>%SW62 = Not significant because one of the two PLCs is Offline</p>

## CPU Hardware or Firmware Becomes Inoperative

### Primary CPU Hardware or Firmware Becomes Inoperative

The following table presents the effects when the Primary CPU's hardware or firmware becomes inoperative:

<p>Before the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at the end of the task cycle</li> <li>● PLC B: PLC A output applied at the end of task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>	<p>The diagram shows two PLC racks. PLC A (Primary) has its CPU module active (indicated by a grey bar) and its Ethernet Port connected to the Ethernet I/O scanner + SCADA. PLC B (Standby) has its CPU module inactive (white bar) and its Ethernet Port disconnected from the scanner. Both racks are connected to the scanner via switches.</p>
<p>Event</p> <ul style="list-style-type: none"> <li>● Processor hardware or firmware becomes inoperative.</li> </ul> <p>Event causes an automatic Switchover: YES</p>	<p>The diagram shows the state during the event. PLC A's CPU is now inactive (white bar) and its Ethernet Port is disconnected from the scanner. PLC B's CPU is now active (grey bar) and its Ethernet Port is connected to the scanner. This indicates an automatic switchover to the standby unit.</p>
<p>After the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: fallback position</li> <li>● PLC B: calculated and applied at the end of the task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are closed: I/O scanner is not active</li> <li>● PLC B: all connections with Ethernet devices are open: I/O scanner is active</li> </ul>	<p>The diagram shows the state after the event. PLC A's CPU is inactive (white bar) and its Ethernet Port is disconnected from the scanner. PLC B's CPU is active (grey bar) and its Ethernet Port is connected to the scanner. The scanner is now active and connected to PLC B.</p> <p>(1) NR: Not responding</p>

Global status	Communication status	Customer diagnostic through Ethernet address @
The process is still active but the HSBY system is no longer redundant as long as the PLC A is in an undefined/non-responsive state.	<ul style="list-style-type: none"> <li>No access to PLC A (CPU no longer running)</li> <li>Normal access to PLC B through the Uni-Telway and USB ports, Modbus and Ethernet links for diagnostics</li> </ul>	%SW61 = 1000 0000 0110 0010 <ul style="list-style-type: none"> <li>The accessed PLC is PLC B / primary</li> <li>The other PLC is PLC A / undefined</li> </ul> %SW62 = Not significant because one of the two PLCs is Not Responding.

### Standby CPU Hardware or Firmware Becomes Inoperative

The following table presents the effects when the Standby CPU's hardware or firmware becomes inoperative:

Before the event	Event
<p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>PLC A: calculated and applied at the end of the task cycle</li> <li>PLC B: PLC A output applied at the end of task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>	<p>The diagram shows two PLC racks connected to a switch network. The left rack is labeled 'PLC A Primary' and the right rack is 'PLC B Standby'. Each rack has modules for PS, CPU, ETH Port, SCY, DIG IN, and DIG OUT. The CPU and ETH Port modules are connected to a switch network. The Ethernet I/O scanner + SCADA is connected to the switch network. The address '@' is associated with the scanner and '@+1' with the standby PLC.</p>
<ul style="list-style-type: none"> <li>Processor hardware or firmware becomes inoperative.</li> </ul> <p>Event causes an automatic Switchover: NO</p>	<p>The diagram is identical to the previous one, but the CPU and ETH Port modules of the PLC B Standby rack are shaded grey, indicating they are inoperative. The Ethernet I/O scanner + SCADA remains connected to the switch network.</p>



## Interruption of Supply Power to Main Rack

### Supply Power to Primary Main Rack is Interrupted

The following table presents the effects of an interruption of supply power to the main rack of the Primary PLC:

<p>Before the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at the end of the task cycle</li> <li>● PLC B: PLC A output applied at the end of task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>	
<p>Event</p> <ul style="list-style-type: none"> <li>● Loss of supply power to the Primary standard rack.</li> </ul> <p>Event causes an automatic Switchover: YES</p>	
<p>After the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: I/O powered off</li> <li>● PLC B: calculated and applied at the end of the task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: ETY powered off</li> <li>● PLC B: all connections with Ethernet devices are open: I/O scanner is active</li> </ul>	<p>(1) NR: Not Responding</p>

Global status	Communication status	Customer diagnostic through Ethernet address @
The process is still active but the HSBY system is no longer redundant as long as the PLC A is powered off.	<ul style="list-style-type: none"> <li>No access to PLC A (CPU system no longer running)</li> <li>Normal access to PLC B through the Uni-Telway and USB ports, Modbus and Ethernet links for diagnostics</li> </ul>	%SW61 = 1000 0000 0110 0010 <ul style="list-style-type: none"> <li>The accessed PLC is PLC B / primary</li> <li>The other PLC is PLC A / undefined</li> </ul> %SW62 = Not significant because one of the two PLCs is Not Responding

### Supply Power to Standby Main Rack is Interrupted

The following table presents the effects of an interruption of supply power to the main rack of the Standby PLC:

Before the event	Event
<p>Before the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>PLC A: calculated and applied at the end of the task cycle</li> <li>PLC B: PLC A output applied at the end of task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>	
<p>Event</p> <ul style="list-style-type: none"> <li>Loss of supply power to the Standby standard rack.</li> </ul> <p>Event causes an automatic Switchover: NO</p>	

<p>After the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at the end of the task cycle</li> <li>● PLC B: I/O powered off</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: ETY powered off</li> </ul>			<p>The diagram shows an Ethernet I/O scanner + SCADA system. It consists of two PLCs, PLC A (Primary) and PLC B (NR (1)), connected to a central scanner via two switches. PLC A has modules for PS, CPU, ETY, SCY, DIG IN, and DIG OUT. PLC B has modules for PS, CPU, ETY, SCY, DIG IN, and DIG OUT. Both PLCs have an ETH Port and an SCP 114 module. The scanner is connected to both switches, and the switches are connected to each other. A bracket labeled '@' is under the ETH Port and SCP 114 modules of both PLCs. A note below the diagram states '(1) NR: Not Responding'.</p>		
Global status	Communication status	Customer diagnostic through Ethernet address @			
<p>The process is still active but the HSBY system is no longer redundant as long as the PLC B is powered off.</p>	<ul style="list-style-type: none"> <li>● Normal access to PLC A through the Uni-Telway and USB ports, Modbus link and Ethernet link for diagnostics</li> <li>● No access to PLC B (CPU system is no longer running)</li> </ul>	<p>%SW61 = 1000 0000 0100 0010</p> <ul style="list-style-type: none"> <li>● The accessed PLC is PLC A: primary</li> <li>● The other PLC is PLC B: undefined</li> </ul> <p>%SW62 = Not significant because one of the two PLCs is Not Responding</p>			

## ETY Hardware or Firmware (Monitored by Hot Standby CPU) Becomes Inoperative

### Primary Monitored ETY Hardware or Firmware Becomes Inoperative (I/O Scanning Service Active)

The following table presents the effects when the ETY on the Primary PLC's rack becomes inoperative while running an I/O Scanning service:

<p>Before the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at the end of the task cycle</li> <li>● PLC B: PLC A output applied at the end of task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>	
<p>Event</p> <ul style="list-style-type: none"> <li>● The hardware or firmware of the Primary Monitored ETY module that manages Ethernet I/O (or Ethernet I/O + SCADA / HMI) becomes inoperative.</li> </ul> <p>Event causes an automatic Switchover: YES</p>	
<p>After the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: fallback position</li> <li>● PLC B: calculated and applied at the end of the task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are closed: I/O scanner is not active</li> <li>● PLC B: all connections with Ethernet devices are open: I/O scanner is active</li> </ul>	<p>(1) NR: Not responding</p>

Global status	Communication status	Customer diagnostic through Ethernet address @
The process is still active but the system is no longer redundant as long as the ETY module of the PLC A is inoperative.	<ul style="list-style-type: none"> <li>Normal access to PLC A through the Uni-Telway and USB ports or Modbus</li> <li>No access to PLC A through Ethernet link</li> <li>Normal access to PLC B through the Uni-Telway and USB ports, Modbus or Ethernet links for diagnostics</li> </ul>	%SW61 = 1000 0000 0010 0110 <ul style="list-style-type: none"> <li>The accessed PLC is PLC B / primary</li> <li>The other PLC is PLC A / offline</li> </ul> %SW62 = Not significant because one of the two PLCs is Offline.

**Standby Monitored ETY Hardware or Firmware Becomes Inoperative (I/O Scanning Service Not Active, But Ready)**

The following table presents the effects when the ETY on the Standby PLC's rack becomes inoperative while remaining ready to activate an I/O Scanning service (service currently inactive):

Before the event	
<p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>PLC A: calculated and applied at the end of the task cycle</li> <li>PLC B: PLC A output applied at the end of task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>	
Event	
<ul style="list-style-type: none"> <li>The hardware or firmware of the Standby Monitored ETY module that can manage the Ethernet I/O and SCADA / HMI becomes inoperative.</li> </ul> <p>Event causes an automatic Switchover: NO</p>	

After the event		
<p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at the end of the task cycle</li> <li>● PLC B: fallback position</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>		<p>Ethernet I/O scanner+ SCADA (1) NR: Not Responding</p> <p>PLC A Primary: PS, CPU, ETY, SCP 114, DIG IN, DIG OUT</p> <p>PLC B Offline: PS, CPU, ETY, SCP 114, DIG IN, DIG OUT</p> <p>Labels: @, @+1</p>
Global status	Communication status	Customer diagnostic through Ethernet address @
<p>The process is still active but the system is no longer redundant as long as the PLC B is inoperative.</p>	<ul style="list-style-type: none"> <li>● Normal access to PLC A through the Uni-Telway and USB ports, Modbus, and Ethernet links for diagnostics</li> <li>● Normal access to PLC B through the Uni-Telway and USB ports or Modbus</li> <li>● No access to PLC B through Ethernet link</li> </ul>	<p>%SW61 = 1000 0000 0000 0110</p> <ul style="list-style-type: none"> <li>● The accessed PLC is PLC A: primary</li> <li>● The other PLC is PLC B: offline</li> </ul> <p>%SW62 = Not significant because one of the two PLCs is Offline.</p>

## ETY Hardware or Firmware (Not Monitored by Hot Standby CPU) Becomes Inoperative

### Non-Monitored Primary ETY (Dedicated to HMI/SCADA) Becomes Inoperative

The following table presents the effects when a non-Monitored ETY module on the Primary rack, which is being used for a dedicated HMI/SCADA connection, becomes inoperative:

<p>Before the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at end of task cycle</li> <li>● PLC B: PLC A output applied at end of task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>	
<p>Event</p> <ul style="list-style-type: none"> <li>● Hardware or firmware on the Primary ETY module that manages a dedicated SCADA / HMI connection becomes inoperative.</li> </ul> <p>Event causes an automatic Switchover: NO</p>	
<p>After the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at end of task cycle</li> <li>● PLC B: PLC A output applied at end of task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>	<p>(1) NR: Not Responding</p>

Global status	Communication status	Customer diagnostic through Ethernet address @
The process is still redundant but diagnosis is no longer possible through the HMI / SCADA link (address @ not responding). If necessary the customer can request a Switchover by setting a bit in the command register of the Primary application (if the other PLC is fully functional).	<ul style="list-style-type: none"> <li>● Normal access to PLC A through the Uni-Telway and USB ports or Modbus for diagnostics</li> <li>● No access to PLC A through Ethernet link</li> <li>● Normal access to PLC B through the Uni-Telway and USB ports, Modbus, and Ethernet links for diagnostics</li> </ul>	%SW61 = 1000 0000 0000 1110 <ul style="list-style-type: none"> <li>● The accessed PLC is PLC A / primary</li> <li>● The other PLC is PLC B / standby</li> </ul> %SW62 = 0000 000 0000 0000 <ul style="list-style-type: none"> <li>● The other PLC: fully functional</li> </ul>

If you will use HMI or SCADA devices to remotely control your Premium Hot Standby system, and you wish this control to persist through Switchover events, then you should connect the HMI / SCADA via the ETY-sync link and run the I/O Scanning service on the Monitored ETYs. However, if you do so, you must use a minimum of two network switches on the ETY-sync link to reduce the likelihood that a cable disconnection will interrupt HMI / SCADA access.

## **WARNING**

### **UNINTENDED EQUIPMENT OPERATION**

If the Primary PLC will be addressed by HMI or SCADA terminals for the purpose of system control:

- Make the HMI/SCADA connection to the ETY-sync link.
- Run the I/O Scanning service on the Monitored ETY modules.
- Use a minimum of two network switches on the ETY-sync link.

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

### Non-Monitored Standby ETY (Dedicated to HMI/SCADA) Becomes Inoperative

The following table presents the effects when a non-Monitored ETY module on the Standby rack, which is being used for a dedicated HMI/SCADA connection, becomes inoperative:

<p>Before the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at the end of the task cycle</li> <li>● PLC B: PLC A output applied at the end of task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>		
<p>Event</p> <ul style="list-style-type: none"> <li>● Hardware or firmware on the Standby ETY module that can manage the dedicated SCADA / HMI connection becomes inoperative.</li> </ul> <p>Event causes an automatic Switchover: NO</p>		
<p>After the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at end of task cycle</li> <li>● PLC B: PLC A output applied at end of task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>		
Global status	Communication status	Customer diagnostic through Ethernet address @
<p>No impact on the Hot Standby running. The process is still redundant.</p>	<ul style="list-style-type: none"> <li>● Normal access to PLC A through the Uni-Telway and USB ports, Modbus, and Ethernet links for diagnostics</li> <li>● Normal access to PLC B through the Uni-Telway and USB ports or Modbus</li> <li>● No access to PLC B through Ethernet link</li> </ul>	<p>%SW61 = 1000 0000 0000 1110</p> <ul style="list-style-type: none"> <li>● the accessed PLC is PLC A: primary</li> <li>● the other PLC is PLC B: standby</li> </ul> <p>%SW62 = 0000 0000 1000 0000</p>

## Ethernet Copro Becomes Inoperative

### Primary's Ethernet Coprocessor Becomes Inoperative

The following table presents the effects if the hardware or firmware on the Ethernet Copro of the Primary PLC becomes inoperative:

Before the event	
<p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at end of task cycle</li> <li>● PLC B: PLC A output applied at end of task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>	<p style="text-align: center;">Ethernet I/O scanner + SCADA</p> <p style="text-align: center;">@ @+1</p> <p>PLC A Primary      Switch      Switch      PLC B Standby</p> <p>PS CPU ETY ETY SCY DIG DIG PS CPU ETY ETY SCY DIG DIG      Port HMI 114 IN OUT Port HMI 114 IN OUT</p> <p style="text-align: center;">@ @+1</p>
Event	
<ul style="list-style-type: none"> <li>● The Primary PLC's Copro that manages the Hot Standby CPU-sync link becomes inoperative. The database exchange between the Primary and Standby controllers ceases.</li> </ul> <p>Event causes an automatic Switchover: NO</p>	<p style="text-align: center;">Ethernet I/O scanner + SCADA</p> <p style="text-align: center;">@ @+1</p> <p>PLC A Primary      Switch      Switch      PLC B Standby</p> <p>PS CPU ETY ETY SCY DIG DIG PS CPU ETY ETY SCY DIG DIG      Port HMI 114 IN OUT Port HMI 114 IN OUT</p> <p style="text-align: center;">@ @+1</p>
After the event	
<p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at end of the task cycle</li> <li>● PLC B: fallback position</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>	<p style="text-align: center;">Ethernet I/O scanner + SCADA</p> <p style="text-align: center;">@ @+1</p> <p>PLC A Primary      Switch      Switch      PLC B Offline</p> <p>PS CPU ETY ETY SCY DIG DIG PS CPU ETY ETY SCY DIG DIG      Port HMI 114 IN OUT Port HMI 114 IN OUT</p> <p style="text-align: center;">@ @+1</p>

Before the event		
Global status	Communication status	Customer diagnostic through Ethernet address @
The process is still active but the system is no longer redundant as long as the Ethernet copro of PLC A is inoperative.	Both PLCs are accessible through the Uni-Telway and USB ports, Modbus links and Ethernet links for diagnostics.	%SW61 = 0000 0000 0100 0110 <ul style="list-style-type: none"> <li>● The CPU-sync link is NOK</li> <li>● The accessed PLC is PLC A / Primary</li> <li>● The other PLC is PLC B / undefined</li> </ul> %SW62 = Not significant because one of the two PLCs is in an undefined state.

### Standby's Ethernet Coprocessor Becomes Inoperative

The following table presents the effects if the hardware or firmware on the Ethernet Copro of the Standby PLC becomes inoperative:

Before the event	
In-rack Discrete I/O state: <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at end of task cycle</li> <li>● PLC B: PLC A output applied at end of task cycle</li> </ul> Remote I/O state: <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>	<p>The diagram shows two PLCs, PLC A (Primary) and PLC B (Standby), connected to a network. PLC A has an active Ethernet port (ETH Port) connected to a switch labeled '@'. PLC B has an inactive Ethernet port (ETH Port) connected to a switch labeled '@+1'. Both switches are connected to an Ethernet I/O scanner + SCADA. The PLCs have various modules: PS, CPU, ETY, ETY HMI, SCY, DIG IN, and DIG OUT. SCP 114 is also shown on the network line.</p>
Event <ul style="list-style-type: none"> <li>● The Standby PLC's Copro that manages the Hot Standby CPU-sync link becomes inoperative. The database exchange between the Primary and Standby controllers ceases.</li> </ul> Event causes an automatic Switchover: NO	<p>The diagram is identical to the previous one, but the Ethernet port of PLC B (Standby) is highlighted in grey, indicating it is now inoperative. The Ethernet I/O scanner + SCADA and switches remain the same.</p>

Before the event		
After the event		
In-rack Discrete I/O state: <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at the end of the task cycle</li> <li>● PLC B: Fall back position</li> </ul> Remote I/O state: <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>		<p>The diagram shows two PLC racks connected to an Ethernet network. PLC A (Primary) has modules: PS, CPU, ETH Port, ETY, ETY HMI, SCY, SCP 114, DIG IN, and DIG OUT. PLC B (Offline) has modules: PS, CPU, ETH Port, ETY, ETY HMI, SCY, SCP 114, DIG IN, and DIG OUT. Two switches, labeled '@' and '@+1', are connected to the Ethernet ports of both PLCs. The Ethernet I/O scanner + SCADA is connected to the '@' switch.</p>
Global status	Communication status	Customer diagnostic through Ethernet address @
The process is still active but the system is no longer redundant as long as the Ethernet copro of PLC B is inoperative.	Both PLCs are accessible through the Uni-Telway and USB ports, Modbus links and Ethernet links for diagnostics.	%SW61 = 1000 0000 0100 0010 <ul style="list-style-type: none"> <li>● The accessed PLC is PLC A / primary</li> <li>● The other PLC is PLC B / undefined</li> </ul> %SW62 = Not significant because one of the two PLCs is in an undefined state.

## Interruption of CPU-sync link between Primary and Standby PLCs

### Interruption of CPU-sync link between Primary and Standby PLCs

The following table presents the effects if the CPU-sync link between the Primary and Standby PLCs is disconnected:

<p>Before the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at end of task cycle</li> <li>● PLC B: PLC A output applied at the end of task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>	
<p>Event</p> <ul style="list-style-type: none"> <li>● CPU-sync link disconnection. The database exchange between the Primary and Standby controllers ceases.</li> </ul> <p>Event causes an automatic Switchover: NO</p>	
<p>After the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at the end of the task cycle</li> <li>● PLC B: Fallback position</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>	

Global status	Communication status	Customer diagnostic through Ethernet address @
The process is still active but the system is no longer redundant as long as the CPU-sync link between the two PLCs is disconnected.	Both PLCs are accessible through the Uni-Telway and USB ports, Modbus links and Ethernet links for diagnostics.	%SW61 = 1000 0000 0100 0010 <ul style="list-style-type: none"> <li>● The accessed PLC is PLC A / primary</li> <li>● The other PLC is PLC B / undefined</li> <li>● CPU-sync link not OK</li> </ul> %SW62 = Not significant because one of the two PLCs is in an undefined state.

## ETY-sync Link Cable Disconnection with I/O Scanner Active

### Primary's Monitored ETY Cable Disconnected Between ETY and Switch

The following table presents the effects if the cable between the Primary PLC's Monitored ETY and the first network switch is disconnected. (I/O Scanning service is running on the Primary ETY):

<p>Before the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at end of task cycle</li> <li>● PLC B: PLC A output applied at end of task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>	
<p>Event</p> <ul style="list-style-type: none"> <li>● Ethernet I/O link disconnection on the Primary side. There is no more diagnostic dialog between the 2 ETY modules.</li> </ul> <p>Event causes an automatic Switchover: YES</p>	
<p>After the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: Fallback position</li> <li>● PLC B: Calculated and applied at end of task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are closed: I/O scanner is not active</li> <li>● PLC B: all connections with Ethernet devices are open: I/O scanner is active</li> </ul>	

Global status	Communication status	Customer diagnostic through Ethernet address @
The process is still active but the system is no longer redundant as long as the Ethernet I/O link is disconnected on the PLC A side.	<ul style="list-style-type: none"> <li>Normal access to PLC A through the Uni-Telway and USB ports and Modbus link for diagnostics. If an HMI/SCADA is connected to the switch, diagnosis is no longer possible through Ethernet</li> <li>Normal access to PLC B through the Uni-Telway and USB ports, Modbus, and Ethernet links for diagnostics</li> </ul>	%SW61 = 1000 0000 0010 0110 <ul style="list-style-type: none"> <li>The accessed PLC is PLC B / primary</li> <li>The other PLC is PLC A / offline</li> </ul> %SW62 = Not significant because one of the two PLCs is Offline.

### Standby's Monitored ETY Cable Disconnected Between ETY and Switch

The following table presents the effects if the cable between the Standby PLC's Monitored ETY and the first network switch is disconnected. (I/O Scanning service is running on the Primary ETY):

Before the event	
<p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>PLC A: calculated and applied at the end of the task cycle</li> <li>PLC B: PLC A output applied at the end of task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>	
Event	
<ul style="list-style-type: none"> <li>Ethernet I/O link disconnection on the Standby side. There is no more diagnostic dialog between the two ETY modules.</li> </ul> <p>Event causes an automatic Switchover: NO</p>	

<p>After the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at the end of the task cycle</li> <li>● PLC B: Fall back position</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>			<p>The diagram shows two PLC racks. The left rack is labeled 'PLC A Primary' and contains modules for PS, CPU, ETY, SCY, DIG IN, and DIG OUT. An 'ETH Port' is connected to the ETY module, and an 'SCP 114' is connected to the SCY module. The right rack is labeled 'PLC B Offline' and contains similar modules. An 'ETH Port' is connected to the ETY module, and an 'SCP 114' is connected to the SCY module. Above the racks are two switches. The left switch is connected to 'Ethernet I/O scanner + SCADA' and is labeled '@'. The right switch is labeled '@+1'. A line connects the two switches, with an 'X' over the connection to the right switch, indicating a disconnection. Brackets below the racks indicate connections to '@' and '@+1'.</p>		
<p>Global status</p> <p>The process is still active but the system is no longer redundant as long as the Ethernet I/O link is disconnected on the PLC B side.</p>	<p>Communication status</p> <ul style="list-style-type: none"> <li>● Normal access to PLC A through the Uni-Telway and USB ports and Modbus link for diagnostics.</li> <li>● Normal access to PLC B through the Uni-Telway and USB ports, Modbus, and Ethernet links for diagnostics. If an HMI/SCADA is connected to the switch, diagnosis is no longer possible through Ethernet</li> </ul>	<p>Customer diagnostic through Ethernet address @</p> <p>%SW61 = 1000 0000 0000 0110</p> <ul style="list-style-type: none"> <li>● The accessed PLC is PLC A / primary</li> <li>● The other PLC is PLC B / offline</li> </ul> <p>%SW62 = Not significant because one of the two PLCs is Offline.</p>			

### I/O Scanner Disconnection on I/O link

The following table presents the effects if the cable between a switch on the ETY-sync link and the Monitored I/O is disconnected (assuming that an I/O Scanning service is running on the Primary Monitored ETY):

<p>Before the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at the end of the task cycle</li> <li>● PLC B: PLC A output applied at the end of task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>			<p>The diagram is similar to the previous one, but the right rack is labeled 'PLC B Standby'. The connection between the two switches is now solid, indicating that the I/O link is restored.</p>		
---	--	--	---	--	--

<p>Event</p> <ul style="list-style-type: none"> <li>I/O scanner disconnection on the I/O link. The remote I/O are no longer visible from both PLCs but the diagnostic dialog between the 2 PLCs is still active.</li> </ul> <p>Event causes an automatic Switchover: NO</p>			<p>The diagram shows two PLC racks connected to a central Ethernet I/O scanner + SCADA unit. PLC A (Primary) and PLC B (Standby) are connected via switches. The scanner is connected to the switches, with a connection point marked with an 'X' and '@'. The PLC racks show various modules: PS, CPU, ETY, SCY, DIG IN, and DIG OUT. Ethernet ports are also indicated.</p>		
<p>After the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>PLC A: calculated and applied at the end of the task cycle</li> <li>PLC B: PLC A output applied at the end of the task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>PLC A: The connections with redundant Ethernet I/O devices are disconnected. The I/O Scanning service remains active on the Primary PLC, but will be unable to connect.</li> <li>PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>			<p>This diagram is identical to the one above, but the connection point between the scanner and the switches is now marked with '@+1', indicating a change in the system's state after the event.</p>		
Global status	Communication status	Customer diagnostic through Ethernet address @			
<p>The process is still active on In-rack I/O but the system is no longer redundant as long as the Ethernet I/O link is disconnected on the I/O link side.</p>	<p>The 2 PLCs are accessible through the Uni-Telway and USB ports, Modbus links and Ethernet links for diagnostics.</p>	<p>%SW61 = 1000 0000 0000 0110</p> <ul style="list-style-type: none"> <li>The accessed PLC is PLC A / primary</li> <li>The other PLC is PLC B / standby</li> </ul> <p>%SW62 = 0000 0000 0000 0000</p> <ul style="list-style-type: none"> <li>The other PLC: functioning normally</li> </ul>			

**NOTE:** When redundant Ethernet I/O is disconnected in this manner, upstream of the actual ETY-sync link switches, no automatic Switchover occurs. If you would like to cause a Switchover or some other behavior in this situation, you must program it in your application program.

## Full ETY I/O Link Disconnection (Both Switches for Monitored I/O Inoperative)

### Full Ethernet I/O Link Disconnection

The following table presents the effects if both network switches on an ETY-sync link with an active I/O Scanning service become inoperative:

Before the event	
<p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at nd of task cycle</li> <li>● PLC B: PLC A output applied at end of task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>	
Event	
<ul style="list-style-type: none"> <li>● Full Ethernet I/O link disconnection. The remote I/O are no longer visible from both PLCs and the dialog between the two PLCs is no longer active.</li> </ul> <p>This event is important because both PLCs will be unable to communicate with the Monitored I/O and both will go to the Offline operating mode as a result.</p>	
After the event	
<p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: Fallback position</li> <li>● PLC B: Fallback position</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are closed: I/O scanner is not active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>	

Global status	Communication status	Customer diagnostic through Ethernet address @
The process is no longer active and the Hot Standby system is no longer redundant as long as the switches remain inoperative.	The 2 PLCs are accessible through the Uni-Telway and USB ports and Modbus links for diagnostics. If an HMI/SCADA is connected to either inoperative switch, diagnosis is no longer possible through Ethernet.	%SW61 = 1000 0000 0000 0101 <ul style="list-style-type: none"> <li>● The accessed PLC is PLC A / offline</li> <li>● The other PLC is PLC B / offline</li> </ul> %SW62 = Not significant because the two PLCs are Offline.

## **WARNING**

### **UNINTENDED EQUIPMENT OPERATION**

Always connect the ETY-sync link through at least two approved network switches if you plan to use the ETY-sync link to provide Monitored (redundant) Ethernet I/O capabilities.

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

**NOTE:** If both network switches have become inoperative, follow this procedure to restart your Hot Standby system:

- Replace the network switches with functional equipment.
- Perform a Stop command followed by a Run command on the PLC you wish to use as the Primary.

## Discrete I/O Module Becomes Inoperative

### Discrete I/O Module on Primary PLC's Rack Inoperative or Removed

The following table presents the effects when a Discrete I/O module on the Primary PLC's rack becomes inoperative or is removed:

<p>Before the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at end of task cycle.</li> <li>● PLC B: PLC A output applied at end of task cycle.</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active.</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active.</li> </ul>	
<p>Event</p> <ul style="list-style-type: none"> <li>● The Discrete I/O module becomes inoperative or is removed from the Primary's X-Bus rack.</li> </ul> <p>Event causes an automatic Switchover: NO</p>	
<p>After the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated but cannot be applied to the inoperative module(s)</li> <li>● PLC B: PLC A output applied at end of task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>	

Global status	Communication status	Customer diagnostic through Ethernet address @
No impact on the Hot Standby system. The process is still redundant because PLC B also applies the output image through the ABE7 connection blocks during each MAST task. If needed, the customer can request a Switchover by setting a bit in the command register of the Primary application (if the other PLC is functioning normally).	The 2 PLCs are accessible through the Uni-Telway and USB ports, Modbus links and Ethernet links for diagnostics.	%SW61 = 1000 0000 0000 1110 <ul style="list-style-type: none"> <li>● The accessed PLC is PLC A / primary</li> <li>● The other PLC is PLC B / standby</li> </ul> %SW62 = 0000 0000 0001 1000 <ul style="list-style-type: none"> <li>● Other PLC: functioning normally</li> </ul> Primary PLC: error bit (%I0.x.mod.err) of the 2 discrete modules set to 1.

### Discrete I/O Module on Standby PLC's Rack Inoperative or Removed

The following table presents the effects when a Discrete I/O module on the Standby PLC's rack becomes inoperative or is removed:

Before the event	
<p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at end of task cycle</li> <li>● PLC B: PLC A output applied at end of task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>	
Event	
<ul style="list-style-type: none"> <li>● The Discrete I/O module becomes inoperative or is removed from the Standby's X-Bus rack.</li> </ul> <p>Event causes an automatic Switchover: NO</p>	

Before the event		
After the event		
<p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at end of task cycle</li> <li>● PLC B: PLC A output is received but cannot be applied to the inoperative module(s)</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>		<p>The diagram shows two PLC racks connected via switches. The left rack is labeled 'PLC A Primary' and contains modules: PS, CPU, ETH Port, ETY, SCP 114, SCY, DIG IN, and DIG OUT. The right rack is labeled 'PLC B Standby' and contains modules: PS, CPU, ETH Port, SCP 114, SCY, DIG IN, and DIG OUT. Two switches are shown above the racks, labeled '@' and '@+1'. An 'Ethernet I/O scanner + SCADA' is connected to switch '@'. A bracket below the racks indicates they are accessible through Ethernet address '@'.</p>
Global status	Communication status	Customer diagnostic through Ethernet address @
No impact on the Hot Standby system. The process is still redundant.	The 2 PLCs are accessible through terminal ports, Modbus links and Ethernet links for diagnostics.	<p>%SW61 = 1000 0000 0000 1110</p> <ul style="list-style-type: none"> <li>● The accessed PLC is PLC A: primary</li> <li>● The other PLC is PLC B: standby</li> </ul> <p>%SW62 = 0000 0000 0001 1000</p> <ul style="list-style-type: none"> <li>● Other PLC: 2 discrete modules not operating normally</li> </ul>

## SCP Card in SCY Module Becomes Inoperative

### The Modbus SCP Card in the Primary SCY Module Becomes Inoperative

The following table presents the effects when the SCP card in the Primary SCY becomes inoperative or is removed:

<p>Before the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied end of task cycle</li> <li>● PLC B: PLC A output applied end of task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>	
<p>Event</p> <ul style="list-style-type: none"> <li>● The Primary Modbus SCP card becomes inoperative or is removed.</li> </ul> <p>Event causes an automatic Switchover: NO</p>	
<p>After the event</p> <p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: calculated and applied end of task cycle</li> <li>● PLC B: PLC A output applied at end of task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>	<p>(1) NR: Not Responding</p>

Global status	Communication status	Customer diagnostic through Ethernet address @
No impact on the Hot Standby system. The process is still redundant. If needed, the customer can request a Switchover by setting a bit in the command register of the Primary application (if the other PLC is functioning normally).	<ul style="list-style-type: none"> <li>Normal access to PLC A through the Uni-Telway and USB ports and Ethernet link for diagnostics. No access through Modbus link</li> <li>Normal access to PLC B through the Uni-Telway and USB ports, Ethernet link and Modbus link for diagnostics</li> </ul>	%SW61 = 1000 0000 0000 1110 <ul style="list-style-type: none"> <li>The accessed PLC is PLC A / primary</li> <li>The other PLC is PLC B / standby</li> </ul> %SW62 = 0000 0000 0000 0000 <ul style="list-style-type: none"> <li>Other PLC: functioning normally</li> </ul> Primary PLC: error bit of SCP / SCY modules (%I0.4.mod.err,%I0.4.1.err) set to 1

**The Modbus SCP Card in the Standby SCY Module Becomes Inoperative**

The following table presents the effects when the SCP card in the Standby SCY becomes inoperative or is removed:

Before the event	Event
<p>In-rack Discrete I/O state:</p> <ul style="list-style-type: none"> <li>PLC A: calculated and applied at end of task cycle</li> <li>PLC B: PLC A output applied at end of task cycle</li> </ul> <p>Remote I/O state:</p> <ul style="list-style-type: none"> <li>PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>	<p>The diagram shows two PLC racks connected to an Ethernet I/O scanner + SCADA system. PLC A (Primary) has an active connection to the scanner, while PLC B (Standby) has a closed connection. The scanner is labeled with address @ and @+1.</p>
<ul style="list-style-type: none"> <li>The Standby Modbus SCP card becomes inoperative or is removed.</li> </ul> <p>Event causes an automatic Switchover: NO</p>	<p>The diagram shows the same setup as before, but now the connection to PLC B is active, and the scanner is active for both PLC A and PLC B.</p>

After the event		
In-rack Discrete I/O state: <ul style="list-style-type: none"> <li>● PLC A: calculated and applied at end of task cycle</li> <li>● PLC B: PLC A output applies at end of task cycle</li> </ul> Remote I/O state: <ul style="list-style-type: none"> <li>● PLC A: all connections with Ethernet devices are open: I/O scanner is active</li> <li>● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</li> </ul>		<p>(1) NR: Not Responding</p>
Global status	Communication status	Customer diagnostic through Ethernet address @
No impact on the Hot Standby system. The process is still redundant.	<ul style="list-style-type: none"> <li>● Normal access to PLC A through the Uni-Telway and USB ports, Modbus, and Ethernet links for diagnostics</li> <li>● Normal access to PLC B through the Uni-Telway and USB portst and Ethernet link for diagnostics. No access through Modbus link</li> </ul>	%SW61 = 1000 0000 0000 1110 <ul style="list-style-type: none"> <li>● The accessed PLC is PLC A: primary</li> <li>● The other PLC is PLC B: standby</li> </ul> %SW62 = 0000 0000 0110 0000 <ul style="list-style-type: none"> <li>● Other PLC: SCP/SCY inoperative</li> </ul>





## B

### BOOL

**BOOL** is the abbreviation of Boolean type. This is the elementary data item in computing. A **BOOL** type variable has a value of either: 0 (**FALSE**) or 1 (**TRUE**).

A **BOOL** type word extract bit, for example: `%MW10 . 4`.

## E

### EN

**EN** means **EN**able, this is an optional block input. When **EN** is activated, an **ENO** output is automatically drafted.

If **EN** = 0, the block is not activated, its internal program is not executed and **ENO** its set to 0.

If **EN** = 1, the internal program of the block is executed, and **ENO** is set to 1 by the system. If an error occurs, **ENO** is set to 0.

### ENO

**ENO** means **Error NO**tification, this is the output associated to the optional input **EN**.

If **ENO** is set to 0 (caused by **EN**=0 or in case of an execution error),

- the outputs of function blocks remain in the status they were in for the last correct executed scanning cycle and
- the output(s) of functions and procedures are set to "0".

## I

### INT

**INT** is the abbreviation of single integer format (coded on 16 bits).

The lower and upper limits are as follows:  $-(2 \text{ to the power of } 15)+1$  to  $(2 \text{ to the power of } 15) - 1$ .

Example:

`-32768, 32767, 2#11111110001001001, 16#9FA4`.



---

# Index

---



## A

Allowed Devices, *90, 91*  
Analog I/O (Inputs Only), *60*  
Analog I/O (Outputs Only), *62*

## C

Certifications and Standards, *53*  
checksums, *220*  
confidence tests, *214*  
configuring, *125*  
    Ethernet modules, *146*  
    registers, *158*  
configuring processors, *126*  
Connection Devices, *88, 89*  
controller failures, *223*  
controller troubleshooting, *223*

## D

Database Transfer, *104*  
diagnosing processors, *223*  
diagnostics  
    buffers, *215, 249*  
Discrete I/O, *57*  
distribution groups, *155*

## E

error detection, *217*  
Ethernet I/O, *66*  
Ethernet Network Devices, *92*  
ETY-sync link, *76*

## F

FTP servers, *155*

## H

hot standby network effects, *155*  
Hot Standby systems, *97*  
Hot Standby Systems, *55*

## I

I/O scanning, *155*  
I/O Type, *56*  
identical applications, *228*  
In-rack Communication Modules, *83, 84*  
In-rack I/O Modules, *85, 86, 87*  
IP addresses  
    restriction, *153*

## L

local clients, *155*  
logic mismatches, *227*

## M

maintenance, *213*  
Minimum configurations, *56*  
Mixed Ethernet and Modbus, *102*  
Modbus Network Devices, *93*  
modes, *150*  
Multiple ETYs Running I/O Scanning Services, *98*

## O

offsets, *142*  
overhead, *107*

## P

programming, *169*

## R

Redundant, *57, 60, 62, 66*

Redundant I/O and SCADA Network Services, *100*

registers

    command, *161*

    status, *163*

remote clients, *155*

## S

scan times, *107*

swapping addresses, *142*

Switchover, *30*

switchovers

    cold start, *166*

    logic mismatches, *227*

    swapping addresses, *142*

System, *98, 100, 102*

system errors, *251*

## T

TFTP servers, *155*

transfer time, *112*

TSX H57 24M, *19*

TSX H57 44M, *19*

## U

upgrading, *235*