Premium Hot Standby with Unity User Manual

35012068 02 september 2007 eng

35012068_02

a brand of Schneider Electric

Telemecanique

www.telemecanique.com

Table of Contents

Safety Information

Important Information

NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, 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 an imminently hazardous situation, which, if not avoided, **will result** in death or serious injury.

WARNING

WARNING indicates a potentially hazardous situation, which, if not avoided, **can result** in death, serious injury, or equipment damage.

CAUTION

CAUTION indicates a potentially hazardous situation, which, if not avoided, **can result** in injury or equipment damage.

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.

About the Book

At a Glance

Document Scope

This guide describes the Premium Hot Standby System consisting of the Unity Pro software, the Premium Hot Standby processor TSX H57 24M or TSX H57 44M, power supplies, Ethernet I/O and TCP/IP Ethernet communication module TSX ETY 4103/5103.

This guide describes how to build a Premium Hot Standby System. Users of PL7 Warm Standby Premium systems should note that significant differences exist between Unity and PL7 systems, and where important, this guide identifies those differences.

Note: Software Requirements
Required to use a Premium Hot Standby:
- Unity Pro 3.0 or higher version
- ETY 4103/5103 V4.0 or higher version

Note: Who should use this document?
Anyone who uses a Hot Standby system or needs fault-tolerant availability through redundancy in an automation system.
You should have knowledge of programmable logic controllers (PLCs).
You should possess a working knowledge of the Unity Pro software.

Note: Terminology
This guide uses the following terminology.

- Application program = a project or logic program
- Controller = a Unity Programmable Logic Controller (PLC) module, which contains both
  1. A CPU
  2. A Copro
- CPU = (Central Processing Unit) a microprocessor in the controller, which processes the application program
- Copro = a microprocessor in the controller, which communicates between two controllers
- Modify = to edit or to change an application program
- Module = any unit either a controller, ETY, DEY, DSY, AEY, ASY, SCY
- Scan = program cycle

Because Premium Hot Standby delivers fault-tolerant availability through redundancy, use a Premium Hot Standby when downtime cannot be tolerated. Redundancy means that two backplanes are configured identically. A Premium Hot Standby must have identical configurations:

- Identical Hot Standby processor TSX H57 24M or TSX H57 44M
- Identical TCP/IP Ethernet communication module TSX ETY 4103/5103
- Identical versions of the CPU, Copro, and ETY firmware
- Identical power supplies
- Identical In-rack I/O (if they are used)
- Identical cabling and cabling systems
- Identical sequential placement on the backplane
- Identical application
- Identical cartridge

Validity Note

Related Documents

Title of Documentation	Reference Number
Premium and Atrium Using Unity Pro User Manual	Available on Unity Pro documentation CD Telemecanique.com web site.
Grounding and Electromagnetic Compatibility of PLC System	Available on Unity Pro documentation CD Telemecanique.com web site.

User Comments

We welcome your comments about this document. You can reach us by e-mail at techpub@schneider-electric.com

Introduction

I

At a Glance

Purpose

This part introduces the Premium Hot Standby System. The content describes the hardware available, the compatibility of Premium Hot Standby with PL7 systems, and using IEC logic and Unity.

What's in this Part?

This part contains the following chapters:

Overview

1

Introduction

Overview

In this chapter you will find a brief overview of the Premium Hot Standby System, the module, the CPUs, and the indicators.

What's in this Chapter?

This chapter contains the following topics:

Overview of the Premium Hot Standby System

Purpose of a Hot Standby System

Use a Premium Hot Standby System when downtime cannot be tolerated. Hot Standby Systems deliver high availability through redundancy. A hot standby PLC system consists of single or multi-rack configuration.

The mandatory redundant components are:

- Premium rack with line terminators
- Hot Standby processor TSX H57 24M or TSX H57 44M
- Power Supply Module
- One TCP/IP Ethernet communication module TSX ETY 4103/5103, minimum firmware version 4.0

The optional redundant components are:

- Extension racks with power supply
- Other TCP/IP Ethernet communication module TSX ETY 4103/5103, minimum firmware version 4.0
- Modbus communication module TSX SCP 114 in TSX SCY 21601
- Discretes/Analog input module
- Discretes/Analog output module

The two Hot Standby PLCs are configured with identical hardware and software.

One of the Hot Standby processors TSX H57 24M or TSX H57 44M's acts as the Primary controller, and the other acts as the Standby controller.

Primary and Standby Controllers

The Primary controller executes the application program, controls the Ethernet I/O and In-rack I/O, and updates the Standby controller at the beginning of every scan (program cycle). If the Primary controller fails, the Standby controller takes control within one scan. To determine if the Primary controller failed, note controller's status displayed in the Display block with indicator lamp.

The Standby controller does not execute the full application program but only the first section, and the Standby controller does not control the redundant In-rack I/O and Ethernet I/O but checks the Primary health.

Note:
- Redundant In-rack I/Os are those that are connected in parallel between the 2 PLCs via specific connection blocks
- Local In-rack I/Os are not connected in parallel

The Primary and the Standby controllers can manage local In-rack I/O with some restrictions.

Switchover Capability

Either of the two controllers may function as the Primary controller and the other as the Standby controller.

Primary and Standby states are switchable.

Therefore, if one of the two controllers is functioning as the Primary controller, the other must be in Standby mode. Otherwise, the second controller is in the default mode, which is offline.

The Ethernet I/O and the redundant In-rack I/O are always controlled by the Primary controller.

Monitoring the System

The Primary and the Standby controllers communicate with each other constantly to monitor the functionality of the system.

- If the Primary controller fails, the state of the controllers is switched.
  The Standby controller becomes the Primary, executes the application program, and controls the Ethernet I/O and the redundant in-rack I/O.
- If the Standby controller fails, the Primary controller continues to run without redundancy and acts as a stand alone system.

Power Cycle

On power cycle, the controller that has the lowest MAC address will become the Primary. The second system automatically becomes the Standby.

Handling In-rack I/O

In-rack I/O are supported in a Premium Hot Standby system.

Software Requirements

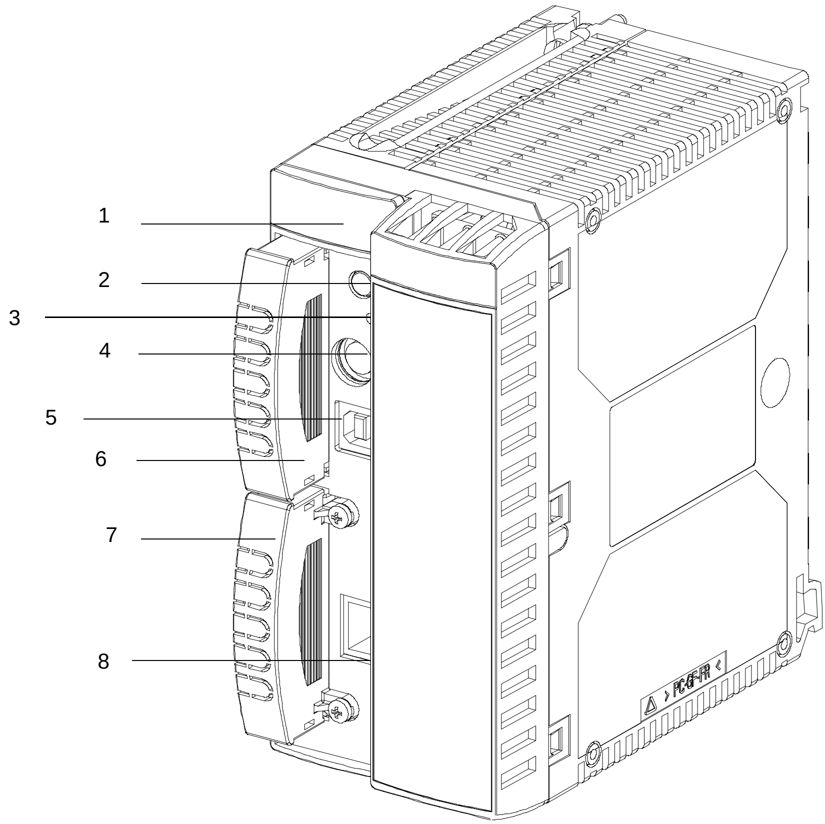
Required to use for a Premium Hot Standby System:

- Unity Pro 3.0 or higher

Premium Hot Standby CPUs Overview

Illustration

The following figure shows the Premium Hot Standby CPU TSX H57 24M and its components (same description for TSX H57 44M).



1 Display block with indicator lamps
2 DOS File Memory extract button (not used)
3 Cold start reset button
4 Uni-Telway Terminal port (programming tool connection, HMI)
5 USB Terminal port (programming tool connection)
6 PCMCIA slot for application memory card extension (Slot A)
7 PCMIA slot for data storage card (Slot B)
8 Dedicated port for CPU-sync link connection

> **Note:** Unity Premium Standby CPUs are equipped with two receptacles (A and B) in which to install PCMCIA cards. PCMCIA is a standard type of memory card.

Norms and company standards

The TSX H57 24M and TSX H57 44M are compliant with the following classifications:

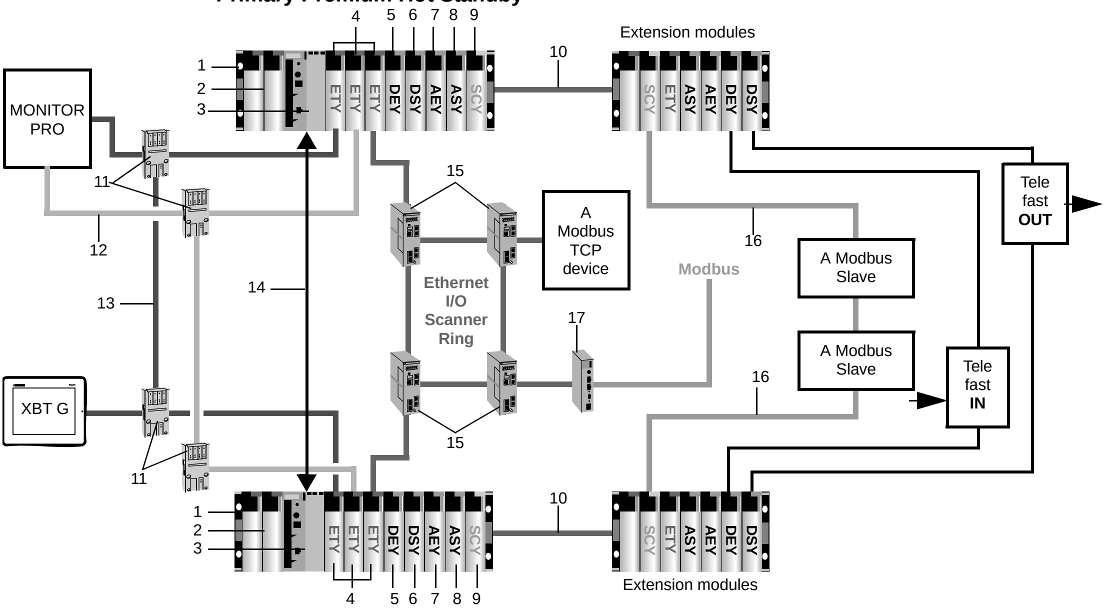
- Non Maritime:
  - CE
  - ICE
  - UL
  - CSA
  - Hazardous location by CSA
- Maritime:
  - BV
  - DNV
  - Lloyd's
  - GL
  - RINA
  - ABS

Premium Hot Standby System Overview

System Components

The following graphic shows a typical architecture example for a Premium Hot Standby System.

Primary Premium Hot Standby



Standby Premium Hot Standby

The following table describes the items of typical architecture example for a Premium Hot standby:

Items	Description
1	Main rack
2	Power supply
3	PLC processor (TSX H57 22M or TSX H57 44M)
4	Ethernet modules (TSX ETY 4103/5103) with Monitored ETY that manages an I/O scanner ring
5	Discrete Input module (example: TSX DEY 64D2K)
6	Discrete Output module (example: TSX DSY 64T2K)
7	Analog Input module (example: Low level isolated Inputs, termocouples, temperature probes TSX AEY 414)
8	Analog Output module (example: Isolated Output s TSX ASY 410)

Items	Description
9	Communication module (TSX SCY 21601 with Modbus PCMCIA TSX SCP 114)
10	XBus
11	Ethernet Switch
12	Ethernet and SCADA Bus #2
13	Ethernet and SCADA Bus #1
14	CPU-sync Link
15	Ethernet Ring Switch
16	Modbus RS485 cable
17	Modbus Gateway (example: TSX ETG 1000)

Modbus components

A Modbus TCP device can be:

- STB
- OTB
- Momentum I/O
- ATV61
- XBT G
- XBT GT
- Premium

A Modbus slave can be:

- STB
- OTB
- ATV31
- TEsysU

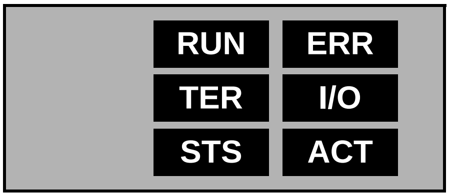
Premium Hot Standby CPUs TSX H57 24M and TSX H57 44M Components

Display Block

The display Block provides the following informations:

- ERR: faults relating to the processor module.
- RUN: program execution states and Hot Standby mode.
- I/O: faults on another station module or configuration fault
- TER: activity on the Terminal port

The following illustration presents the Display block:



Memory extract button

This button is not used.

Cold start Reset Button

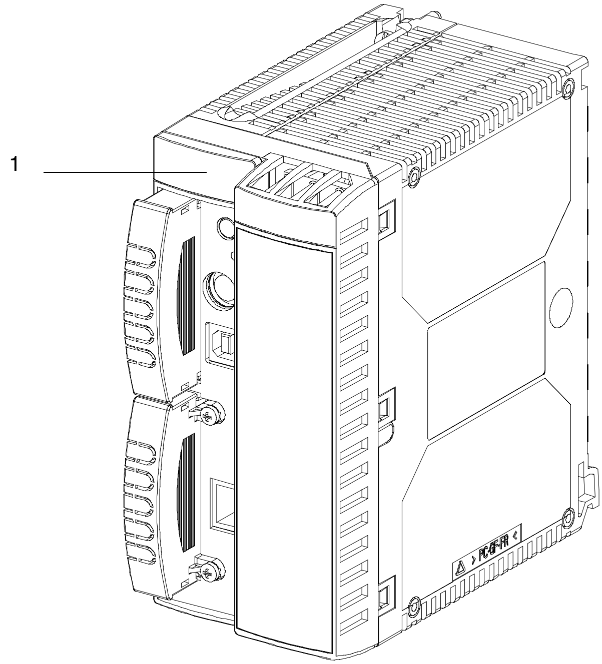
This button forces a cold start of the PLC.

Using Premium Hot Standby CPUs LED indicators

Overview

The LED indicators are positioned on the Display Block.

Position of indicators on Premium Hot Standby CPUs TSX H57 24M and TSX H57 44M:



1 Display Block with LED indicators

Interpreting the LED Indicators

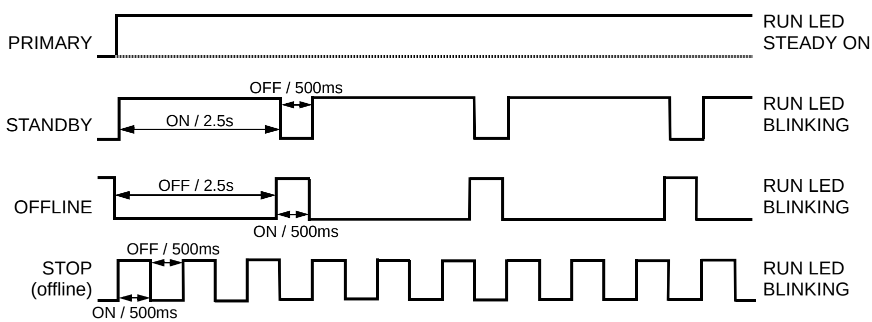
The LEDs provide information.

CPUs TSX H57 24M and TSX H57 44M		
LEDs	**Color**	**Indicates**
ACT	Yellow	Blinking: communication activity between Primary and standby controllers
STS	Yellow	• Blinking: the system is redundant and data are exchanged from the Primary to Standby controller • steady on: the system is not redundant or the Copro is booting from power-on to end of self-tests • Steady off: Copro auto tests failed
Note: No activity returns the LEDs to the default. The Premium Hot Standby CPU uses an embedded coprocessor (Copro) to provide a dedicated communications link, which transfers data between the Primary and Standby controllers.		

The state of the RUN LED depends of the HSBY mode:

- STOP
- Primary
- Standby
- Offline

The following illustration displays the CPU status with the LEDs:



Compatibility, Differences, and Restrictions

2

Introduction

Overview

In this chapter you will find an overview of compatibilities, for a system that has already been installed, differences from a PL7 Warm Standby Premium system, and restrictions for the Premium Hot Standby Unity system.

What's in this Chapter?

This chapter contains the following topics:

Compatibility with Installed PL7 System

Unity Premium Legacy Systems

The Unity Premium HSBY functionality is partially compatible with the PL7 one because:

- Compatible: FIPIO devices can only be connected to a HSBY Premium system through an Ethernet-to-Fipio gateway. Such a gateway can be programmed using a standalone Premium PLC with a Fipio integrated port and an Ethernet port
- Not compatible: use of specific DFB for the data exchange: Ha-db_basic, Ha_db_cycle_opt, Ha_db_size_opt
- Not compatible: use of specific EF for Grafcet (SFC in Unity) context exchange

PL7 Warm Standby conversion

In most cases, a PL7 Warm Standby application will be accepted by the PL7 Unity Pro converter.

The features that are not supported by the Premium HSBY PLC will not be converted (errors signaled by the converter).

After conversion, the new Unity Pro application will require important modifications to fit to the Ethernet I/O and new Premium HSBY features.

Understanding System Words and System Bits

Overview

In accordance with IEC standards, Unity 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 using the IEC convention.

System Word %SW61

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

Note: %SW61 is described using the IEC convention.

System Word %SW62/63/64/65

System Words %SW62/63/64/65 are reverse registers reserved by the Reverse Transfer process. These four reverse registers can be written by the application program (first section) of the Standby controller and are transferred at each scan to the Primary controller.

Understanding Multitasking Restrictions

General

In a Premium Hot Standby, the Standby controller is ready to assume the role of the Primary controller by having the same application loaded (in the Standby) and by receiving from the Primary—once per scan—a copy of the Primary's data. During the scan, there is a tight synchronization between the Primary and Standby.

MAST TASK

Schneider Electric recommends using only MAST task to execute the application Program. Using MAST task is consistent with the fact that data transfer is synchronized with the MAST task.

Asynchronous Events

Using a Premium Hot Standby in a multitasking environment may cause data to change between scans. Because in a multi-tasking system, events may occur asynchronously to the normal scan. Those events may happen at a faster rate, the same rate, or at a slower rate. The result is that data modified by these events can be changed during a transfer.

FAST TASK

⚠ CAUTION

RISK OF EQUIPMENT DAMAGE

The use of a Fast task driving dedicated outputs is not recommended because the output values are transmitted from the Primary to the Standby at the Mast task frequency.
Ensure that you both analyze your system needs and account for problems that may arise if you use Fast.

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

In-rack I/O and Ethernet I/O Restrictions

General

Note the two following restrictions:

- Only In-rack discrete I/O and Analog I/O can be used with a Premium Hot Standby System. These I/O are a part of the redundant system.
- Ethernet I/O are not considered part of the redundant system. They are shared between the two PLCs.
- Only the Primary PLC manages the redundant In-rack I/O and the Ethernet I/O.

Allowed Module in Premium Hot Standby

General

The following table presents the redundant modules supported by the Premium Hot Standby:

Designation	Reference	Function	Quantity
Communication			
Ethernet TCP/IP communication module for redundant applications.	TSX ETY 4103/5103 Version min. 4.0	Ethernet TCP/IP module with transparency of addressing for third-party devices (SCADA/HMI)	2xn
Modbus communication module	TSX SCY 21601 Version min.: 2.1	Communication Modbus master and support of PCMCIA TSX SCP 114	2xn
Modbus communication module	TSX SCY 11601	Communication Modbus Master	2xn
Multi protocol card	TSX SCP 114 (RS 485)Version min.: 1.7	Modbus slave communication with transparency of addressing for third-party Master devices (1)	2xn
Discrete inputs/outputs modules			
Discrete inputs modules	TSX DEY...K	Discrete input modules with HE10 connectors	2xn
Discrete outputs modules	TSX DSY...K	Discrete output modules with HE10 connectors	2xn
Discrete inputs/outputs modules	TSX DMY...K	Discrete event / reflex input/output modules with HE10 connectors	2xn
Discrete inputs modules	TSX DEY...	Discrete input modules with screw terminal block	2xn
Discrete outputs modules	TSX DSY...	Discrete output modules with screw terminal block	2xn
Preventa Safety modules	TSX PAY...	Safety modules with screw terminal block and SUB-D	2xn
16 channel modularity input connection bases	ABE7 ACC11	Facilitate the wiring for the redundant discrete input modules	1xn
16 channel modularity output connection bases	ABE7 ACC10	Facilitate the wiring for the redundant discrete output modules	1xn
Analog inputs/outputs modules			
Analog inputs modules	TSX AEY...	Analog inputs modules with screw terminal block or SUB-D	2xn
Analog outputs modules	TSX ASY...	Analog outputs modules with screw terminal block or SUB-D	2xn

⚠ CAUTION

RISK OF EQUIPMENT DAMAGE

The HSBY system operation is not guaranteed if other in rack redundant modules than the listed ones are used.

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

(1): This card is accepted in the SCY PCMCIA slot and not in the CPU PCMCIA slot.

The following table presents the shared modules supported by the Premium Hot Standby:

Designation	Reference	Function	Quantity
Advantys STB modules			
Advantys STB NIM	STB NIP 2212	Ethernet TCP/IP communicator for Advantys STB	1xn
Input/output modules	STB...	STB input/output modules (discrete/analog)	1xn
Counting module	STB EHC 3020	40Khz counter module	1xn
Advantys FTB/FTM modules			
Input/output modules	FTB..., FTM...	FTB/FTM input/output modules	1xn
Momentum modules			
Ethernet communicator	170 ENT 110 0x	Ethernet communicator for Momentum I/O	1xn
Input/output modules	170 A...	Momentum Input/output modules	1xn
Advantys OTB and Twido modules			
Ethernet communicator	OTB,1E0, DM9LP	Ethernet communicator with embedded I/O	1xn
Twido I/O	TWD...	Twido I/O modules	1xn

Designation	Reference	Function	Quantity
Altivar			
Variable Speed drives	Altivar xx	Altivar with Ethernet interface.	1xn
TesysU motor starters over Modbus are compatible with Premium Hot Standby system. They have to be used with Telemecanique Ethernet/Modbus Gateway one of the following: ● TSX ETG 100 gateway ● TSX ETG 1000 gateway ● 174 CEV 30020 gateway			
ConneXium-Ethernet			
All products of the ConneXium family that are compatible with standard TSX ETY 4103/5103 Ethernet modules in a non Hot Standby configuration are also compatible with the new Hot Standby ETY modules in a Hot Standby configuration. These ConneXium products can be used in different Ethernet topology: tree, ring,... With the 499NxS27100 or TCSESM0x3F2CU0 switches, it is possible to share Ethernet devices on a redundant optical ring or a redundant copper ring			
RTU modules			
TSX ETW 320/330 Wade RTU modules			

Understanding USB and Uni-Telway Link Restrictions

No address swapping on USB and Uni-Telway link

The USB and Uni-Telway terminal ports are only point to point connections that cannot be used for transparent access to the Primary controller:

- In Master mode (default mode), the Uni-Telway terminal port is a point to point connection allowing Unity Pro to communicate with its local controller.
- In Slave mode, the Uni-Telway terminal port does not support address swapping at switch over.

Understanding Application Restrictions

Application restrictions

The application restrictions are:

- The use of events tasks is not recommended. An event can be lost if it occurs just before or during a switch over.
- The use of a FAST tasks driving dedicated outputs is not recommended. Some change of state on the outputs can be lost at switch over.
- The use of counting modules is not recommended. Depending on the frequency, a certain amount of pulses can be lost at switch over.
- The use of edges is not recommended. It is not possible to guarantee that they are taken into account during a switch over.
- The use of the SAVE_PARAM function is not recommended 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 being not transferred from the primary to the standby. More generally, the explicit instructions like WRITE_CMD and WRITE_PARAM have to be used carefully.
  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. In case of switch over, the new Primary will restart with the Modbus mode rather than the Character mode.
- It is not possible to replace the initial values of the declared variables with a save attribute (,e.g.: DFB variables) with the current values: no use of %S94.
- The following Legacy function blocks are forbidden:
  - PL7_COUNTER
  - PL7_DRUM
  - PL7_MONOSTABLE
  - PL7_REGISTER_32
  - PL7_REGISTER_255
  - PL7_TOF, PL7_TON, PL7_TP
  - PL7_3_TIMER
- The use of the TON / TOFF / TP function blocks is forbidden in the first section.
- The use of DFB is not recommended in the first section.

⚠ CAUTION
RISK OF UNINTENDED EQUIPMENT OPERATION AND EQUIPMENT DAMAGE The online modification of an expert function parameter (e.g. control process parameter) is not transferred from the Primary to the Standby. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

Behavior and Performances

3

Introduction

Overview

This chapter provides information about behavior and performances of a Premium Hot Standby System.

What's in this Chapter?

This chapter contains the following sections:

3.1 Behavior of Premium Hot Standby

At a Glance

Purpose

This section describes the Behavior of the Premium Hot Standby system.

What's in this Section?

This section contains the following topics:

Premium Hot Standby with IEC Logic

Overview

A Premium Hot Standby System requires two backplanes configured with identical hardware, software, and firmware. One of the controllers (PLC) 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 and Standby communicate constantly monitoring the health of the system.
- If the Primary fails, the Standby takes control within one scan.

Data Transfer and User Data

In a Premium Hot Standby System, data is transferred from Primary to Standby at the beginning of every scan.

The following data transfers at the beginning of every scan:

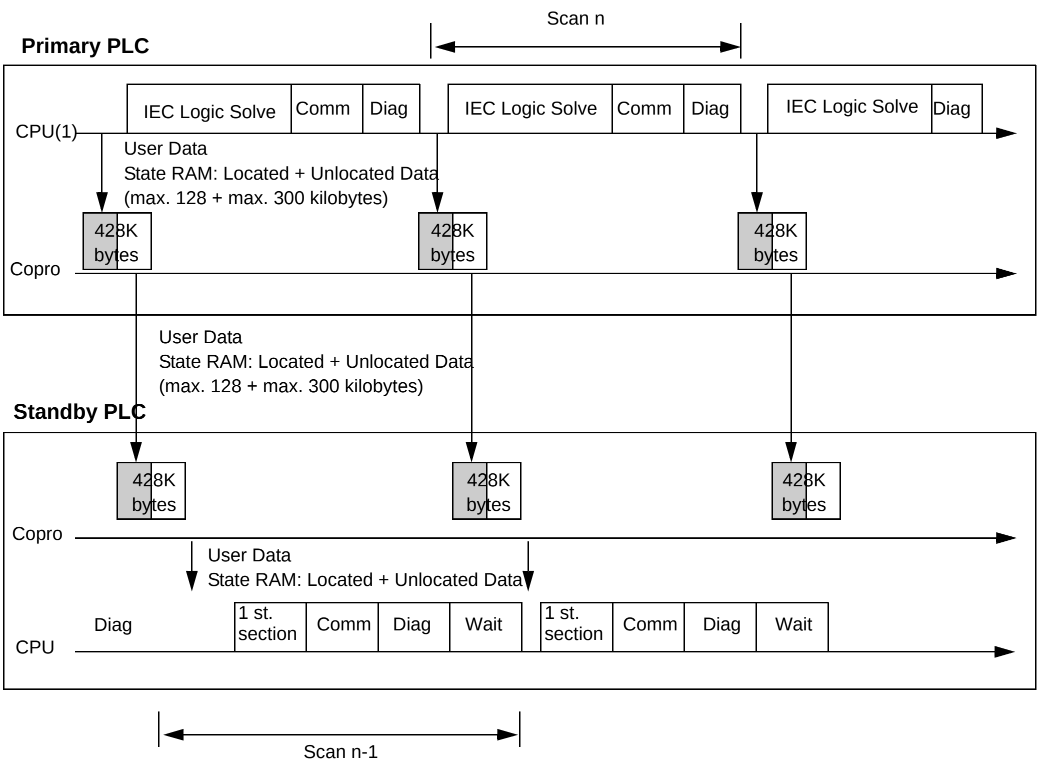
- Output objects and command / adjustment parameters
- Located Variables (maximum 128 Kilobytes)
- All Unlocated variables up to 300 Kilobytes on TSX H57 44M
- All instances of the DFB and EFB type
- SFC variable area
- A part of the System Bits and Words.

Note: Forced Bits at Transfer
At each scan, all forced bits are transferred from the Primary to the Standby.

Understanding the Premium Hot Standby Data Base Transfer Process

Hot Standby Transfer Diagram

The following illustrates the transfer of data from the Primary to the Standby:



Item	CPU model	Max Data size
(1)	TSX H57 24M	192 Kilobytes
	TSX H57 44M	440 Kilobytes

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: A CHANGE FROM LEGACY
In legacy systems (PL7 Warm Standby Premium), the CPU performed both:
- application program (project) processing
- communication transfer

In a Premium Hot standby, in parallel:
- CPU performs application program processing
- Copro performs communication transfer

Result: Greatly reduced transfer time with Unity

Performance Considerations

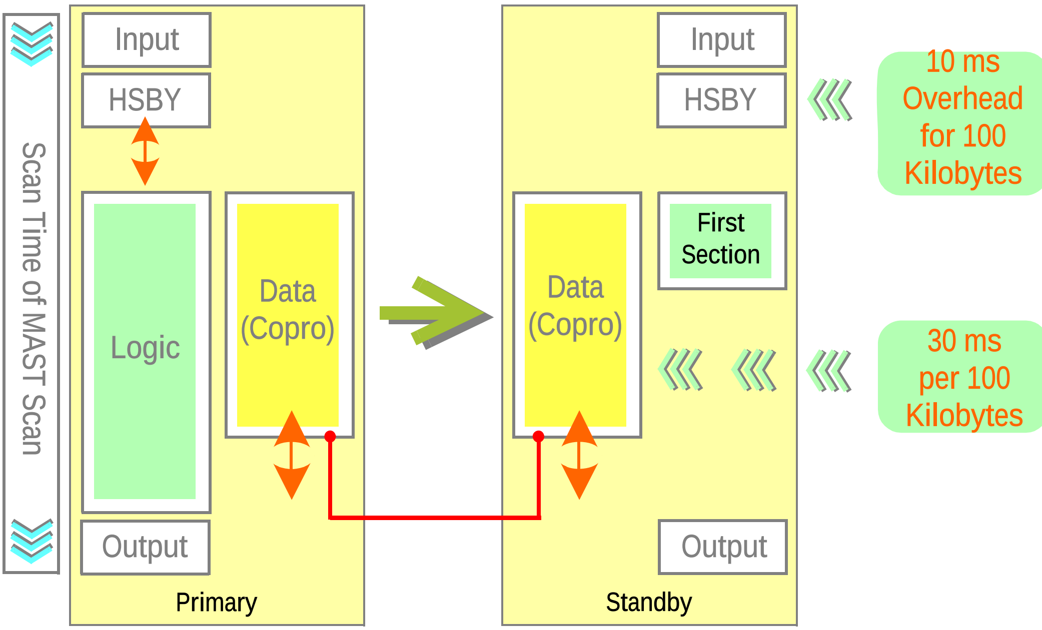
A Premium Hot Standby increases the length of a MAST task scan time, creating system overhead.

> **Note:** 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

A Hot Standby system



Most of the time, the network scan time is included in the MAST scan time.

Examples

However, when processing some application programs, additional system overhead may occur.

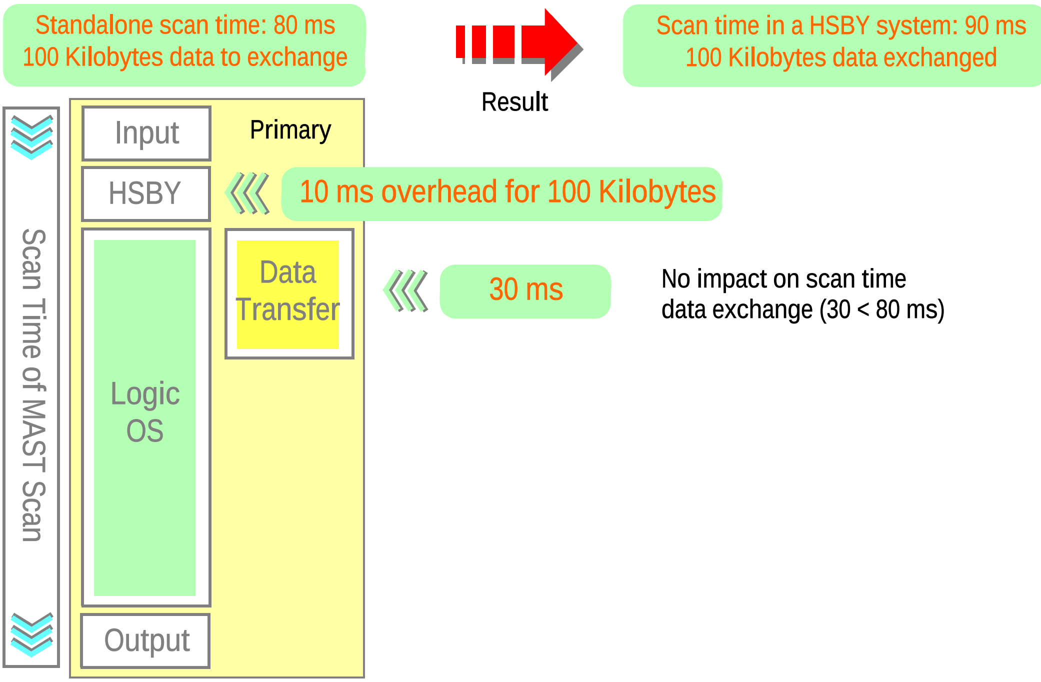
Example #1

- Standalone application scan time: 80 ms
- Data (state RAM + unlocated variables): 100 Kilobytes

Example #2

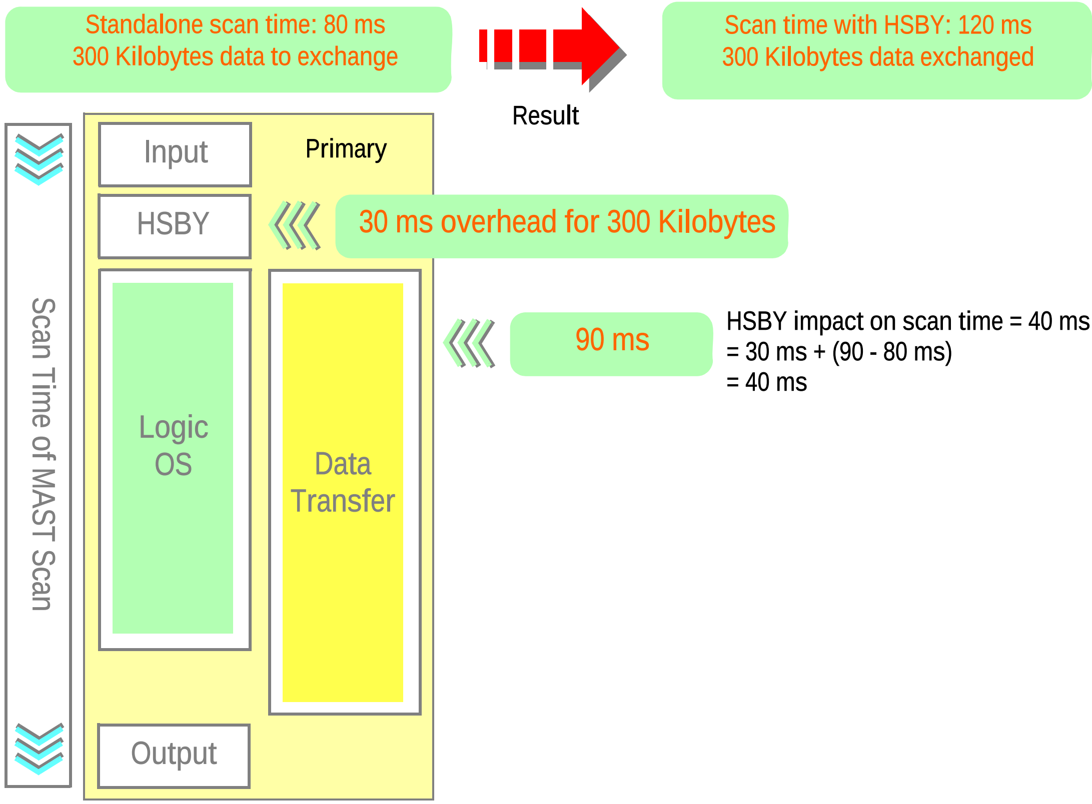
- Standalone application scan time: 80 ms
- Data (state RAM + unlocated variables): 300 Kilobytes

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.

The following illustration displays the example #2:



3.2 Performances of Premium Hot Standby

At a Glance

Purpose

This section describes the Performance of Premium Hot Standby system.

What's in this Section?

This section contains the following topics:

Address Swap Times

Description

The following table details what the "time for an Address swap" comprises, such as the time to close connections, time to swap IP addresses, or time to establish connections.

The following table shows the swap time for each of the Ethernet services.

Service	Typical Swap Time	Maximum Swap Time
Swap IP Addresses	6 ms	500 ms
I/O Scanning	1 initial cycle of I/O scanning	500 ms + 1 initial cycle of I/O scanning
Client Messaging	1 CPU scan	500 ms + 1 CPU scan
Server Messaging	1 CPU scan + the time of the client reestablishment connection	500 ms + the time of the client reestablishment connection
FTP/TFTP Server	The time of the client reestablishment connection	500 ms + the time of the client reestablishment connection
SNMP	1 CPU scan	500 ms + 1 CPU scan
HTTP Server	The time of the client reestablishment connection	500 ms + the time of the client reestablishment connection

X-Bus I/O switchover time

Definition

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 X-Bus I/O:

Switchover event on the Primary	Average time to switchover on X-Bus I/O
Stop, Halt, Cable disconnection	1.5 Mast time
Power-cut	Watch Dog time + 1.5 Mast time

Note: The Watch Dog value that is configured in a Premium Hot Standby application has a direct impact on the switchover time (in case of power-cut on the Primary CPU).

Maintaining

II

At a Glance

Purpose

This part describes five important processes in using a Premium Hot Standby System.

- Setting up, Installing, and Cabling
- Configuring
- Programming/Debugging
- Operating
- Maintaining

What's in this Part?

This part contains the following chapters:

Setting up, Installing, and Cabling

4

Introduction

Overview

This chapter provides an overview of setting up, installing, and cabling a Premium Hot Standby System.

What's in this Chapter?

This chapter contains the following topics:

Setting Up the Premium Hot Standby

Overview

Schneider Electric is a leader in fault-tolerant, redundant systems, Hot Standby. Setting up a Premium Hot Standby System involves a number of processes, summarized in the following paragraphs here and explained in detail in other chapters of this document.

Mapping the Backplane Extensions

A Premium Hot Standby System requires two backplanes.

You must map the two backplanes in an identical manner with:

- Mandatory module:
  - Premium rack with line terminators
  - Hot Standby processor TSX H57 24M or TSX H57 44M
  - Power Supply Module
  - One TCP/IP Ethernet communication module TSX ETY 4103/5103 (configured as Monitored ETY)
- Optional module:
  - Extension racks with power supply
  - Other TCP/IP Ethernet communication module TSX ETY 4103/5103
  - Modbus communication module TSX SCP 114 in TSX SCY 21601
  - Discrete/Analog input module
  - Discrete/Analog output module

Note: The sequence of the modules on the backplane is not predefined, but the sequence of the modules on the backplanes of the Primary and the Standby must be identical. Otherwise, a Premium Hot Standby System will not be redundant because the standby will go to Offline.

Connecting Two Standby CPUs

The link between the two Premium Hot Standby CPUs is called CPU-sync link.
It can be:

- A Twisted Pair/Copper crossover cable
- Fiber cable with optical switches for long distance connections

CAUTION

RISK OF EQUIPMENT DAMAGE

The CPU-sync link is a point to point link dedicated to exchange application data from the Primary PLC to the Standby PLC and to provide information on the Hot Standby system status.
Do not, in any case, connect other Ethernet devices on this link. This may impact the database exchange between the two PLCs and the switchover time.

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

Establishing the Primary and Standby Controllers

The system determines that one of the two Premium Hot Standby CPUs will be the Primary controller and the second controller as the Standby.

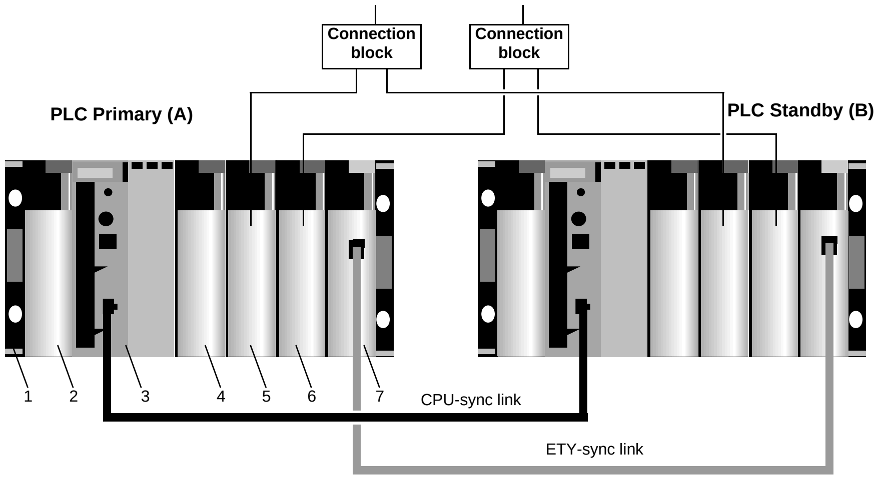
The CPU with the lowest MAC address becomes PLC (A) Primary. The other CPU becomes PLC (B) Standby.

To guarantee which PLC will become the Primary when the two PLCs are powered-up simultaneously, it is possible to use a time-lag relay on the supply of the main rack of one of the two PLCs. During this process, the PLC that has the time-lag relay in its supply cabling will be the Standby PLC.

Connecting the ETY modules

Because it is not possible to have a non-ambiguous diagnostic of the Premium Hot Standby system with only one link between the two PLCs (CPU-sync link), it is mandatory to configure one Ethernet module in each PLC, the two ETY modules must to be linked with an Ethernet cable (with or without switches).

The following illustration displays a very simple Premium Hot Standby configuration:



1 Premium rack with line terminators
2 Power supply
3 Hot Standby processor (TSX H57 24M or TSX H57 44M)
4 Communication module (TSX SCY 21601 with Modbus PCMCIA TSX SCP 114)
5 Discrete output module (example: TSX DSY 64T2K)
6 Discrete input module (example: TSX DEY 64D2K)
7 Hot Standby Ethernet module (TSX ETY 4103/5103)

Example:

In case of power failure on the Primary PLC, the Standby PLC will identify a communication error on the CPU-sync link. But this same communication error will also occur in the case of CPU-sync link disconnection. To distinguish between these two cases, the Standby CPU requests from its local ETY module the status of the counterpart ETY module. In case of fault, the Standby diagnoses that the Primary is offline and becomes Primary.

The link between the two ETYs modules is called ETY-sync link. The two ETYs are called monitored ETYs.

The Monitored ETY modules can manage:

- Only diagnostic information in case of exclusive Bus-X configuration
- Diagnostic information and I/O scanning service if Ethernet I/O devices are connected on the link
- Diagnostic information, I/O scanning service and other Ethernet services

In the above Premium Hot Standby configuration, the two monitored ETYs are linked with a crossover cable. There is no Ethernet device connected to the ETY-sync link. A failure on this link is not a condition to generate a switch over because the ETY-sync link is not part of the I/O or messaging process.

On the contrary, when Ethernet I/O devices or other equipment are connected to the ETY-sync link, it is necessary to generate a switch over if a failure appears on the Primary side.

For more details, refer to *Configuring TSX ETY 4103/5103 Modules, p. 94*.

Mapping the Backplane Extension

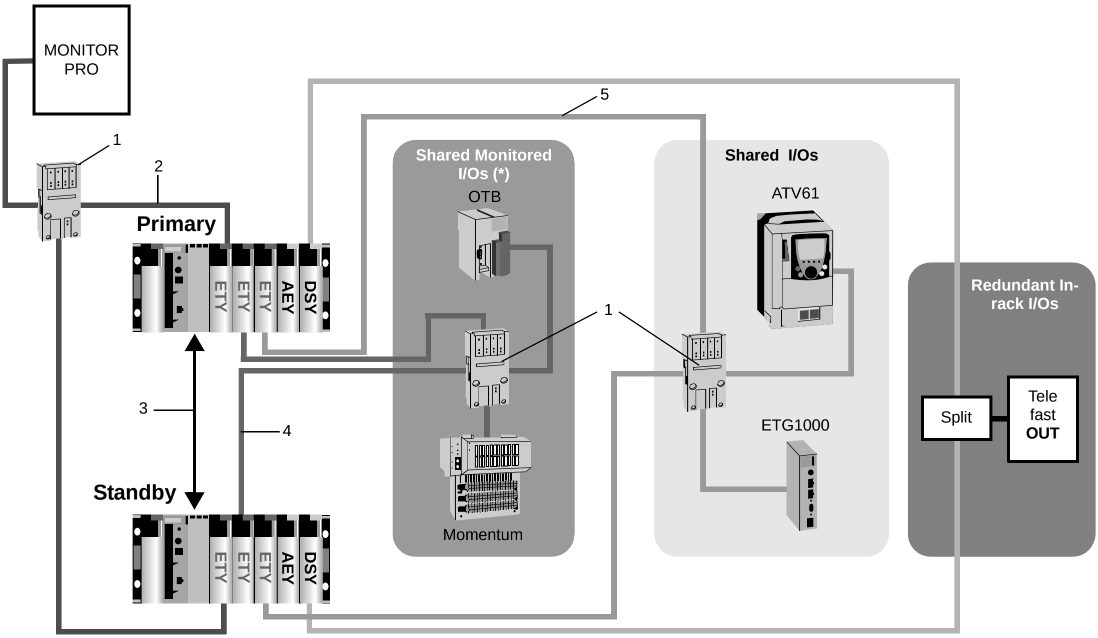
Requiring Identical Backplanes

Two backplanes must be configured with identical hardware, software, and firmware in identical order. Then, both controllers may function either as a Primary controller or as a Standby controller.

> **Note:** INSTALLING CONTROLLERS
> Schneider Electric recommends referring to Schneider Electric planning and installation guidelines. You will find more information in the ***Premium and Atrium Using Unity Pro User Manual 35006160*** and in ***Grounding and Electromagnetic Comptabilty of PLC System 33002439***.

Architecture example with Multiple I/O scanning ETY

The following graphic shows an architecture example with Multiple I/O scanning ETY:



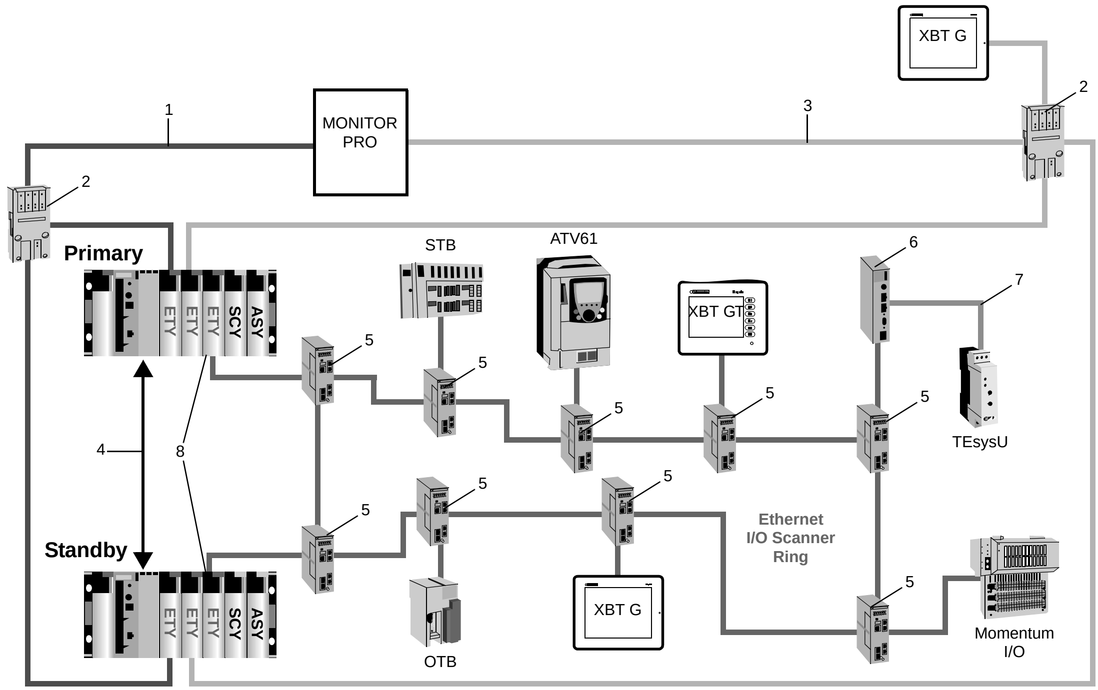
(*) "Monitored" means a failure in the ETY or in the link to the first switch/hub will cause an automatic switch over

The following table describes the items of an architecture example with Multiple I/O scanning ETY:

Items	Description
1	Ethernet Switch
2	Ethernet TCP/IP
3	CPU-sync Link
4	Ethernet I/O Scanner #1
5	Ethernet I/O Scanner #2

Architecture example with Redundant I/O and SCADA network

The following graphic shows an architecture example with Redundant I/O and SCADA network:

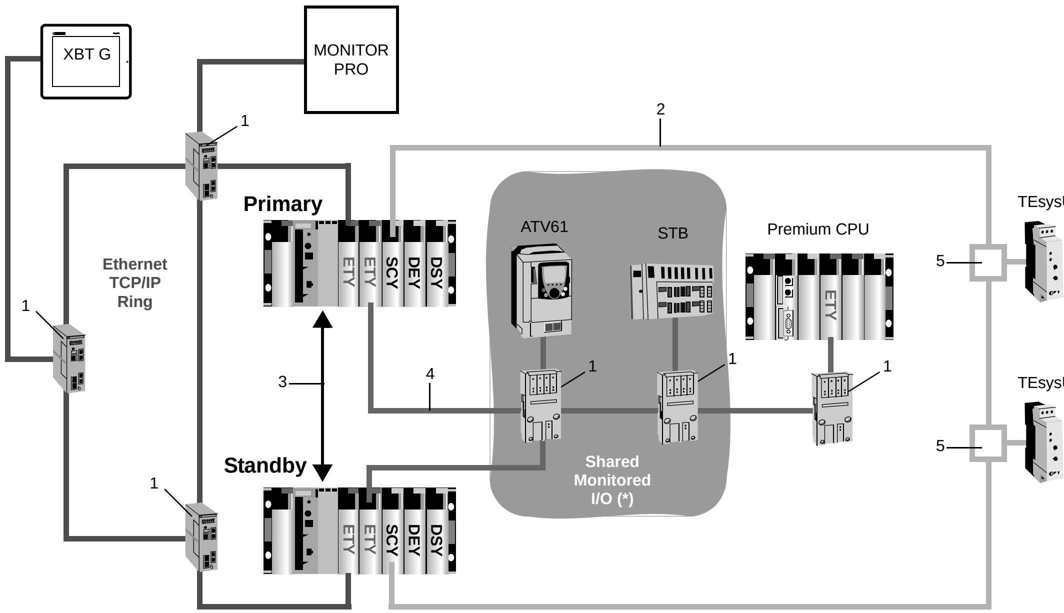


The following table describes the items of an architecture example with Redundant I/O and SCADA network:

Items	Description
1	Ethernet TCP/IP network #1
2	Ethernet Switch
3	Ethernet TCP/IP network #2 and #3
4	CPU-sync Link
5	ConneXium Ethernet Switch with Ring capability
6	Modbus Gateway (example: TSX ETG 1000)
7	Modbus
8	Monitored ETY

Architecture example with Mixed Ethernet and Modbus

The following graphic shows an architecture example with Mixed Ethernet and Modbus:



(*) “Monitored” means a failure in the ETY or in the link to the first switch/hub will cause an automatic switch over

The following table describes the items of an architecture example with Mixed Ethernet and Modbus:

Items	Description
1	ConneXium Ethernet Switch with Ring capability
2	Modbus RS485 cable
3	CPU-sync Link
4	Ethernet I/O scanner communications
5	Junction box

Connecting Two Premium Hot Standby PLCs

Required cable connections

To work properly, the Primary and Standby PLCs have to be linked with:

- The CPU-sync link between the two CPUs
- The ETY-sync link between the two monitored ETY modules

If these two links do not work properly, the two PLCs will start as standalone PLCs.

CAUTION

RISK OF EQUIPMENT DAMAGE

You must route the two cables as far away as possible to one another to prevent double Primary PLC when the two links are broken.

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

The CPU-sync link is a point to point connection dedicated to application data exchange and Hot Standby system diagnostic.

CAUTION

RISK OF EQUIPMENT DAMAGE

Do not, connect other Ethernet devices on this link.
This may impact the database exchange between the two PLCs and the switch over time.

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

The following cables can be used:

- A Twisted Pair/Copper cable
- Fiber cable with optical switches for long distance connections

Twisted Pair/ Copper crossover cable

All products of the ConneXium family that are compatible with standard TSX ETY 4103/5103 modules in a non Hot Standby configuration are also compatible with the new Hot Standby ETY (version min. 4.0) used in a Hot Standby configuration.

For more details on twisted pair cables, refer to the ConneXium catalog and technical publications.

Fiber cable

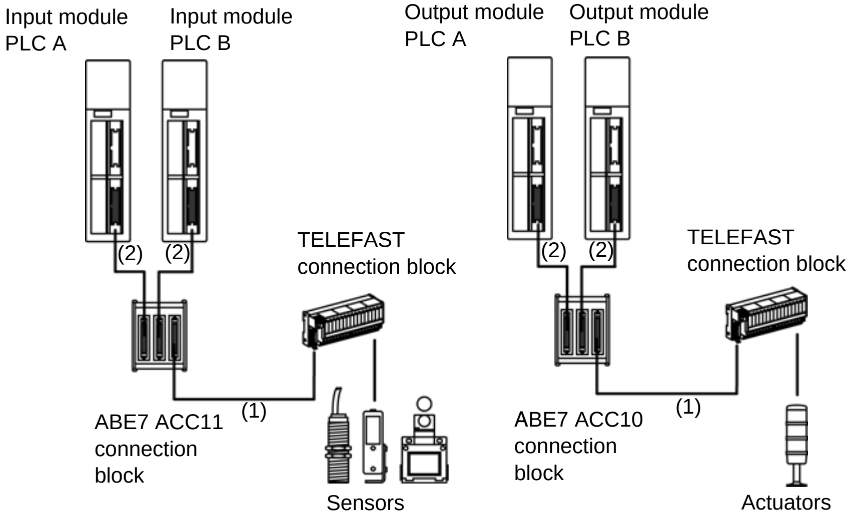
For more details on fiber optic cables, refer to the ConneXium catalog and technical publications.

Connecting In-rack I/O

Sensor/ Actuators cabled to modules in the rack

Each sensor and actuator is connected in parallel on two input or output modules:

The following illustration displays the Sensor/Actuators cabled:

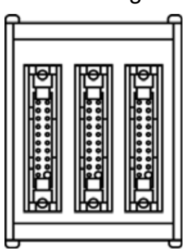


(1): ABF-H20H008 (0.08 m, 3.15 in)
(2): TSX CDP ••3

The cabling for the sensor or actuator is standard and is used according to the TELEFAST terminal block selected.

The terminal blocks ABE7 ACC10 and ABE7 ACC11 have a modularity of 16 channels. They are completely passive and equipped with anti-return diodes on each of the channels.

The following illustration displays the terminal block ABE7 ACC1X:



The authorized input/output modules are modules with positive logic equipped with HE 10 connectors.

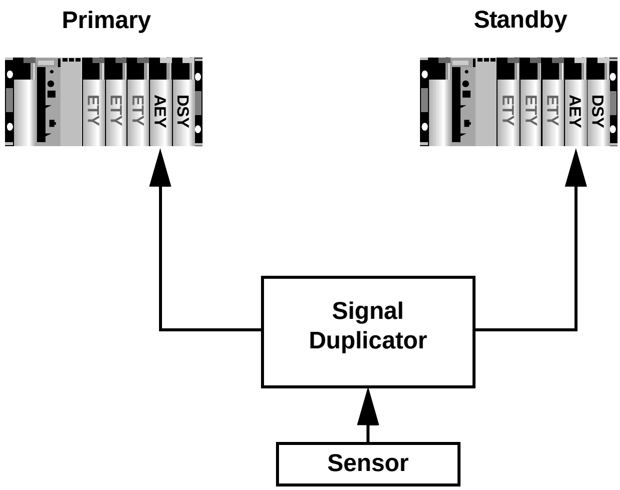
Other Assemblies

It is possible to use other input / output modules with a screwed terminal or negative logic. In this case, the ABE7 ACC10 and ABE7 ACC11 cannot be used any more and it is important to guarantee the independance of the channels by using anti-return diodes.

Analog Input module cabling

For a analog input, a signal duplicator can be used (e.g. JM Concept JK3000N2).

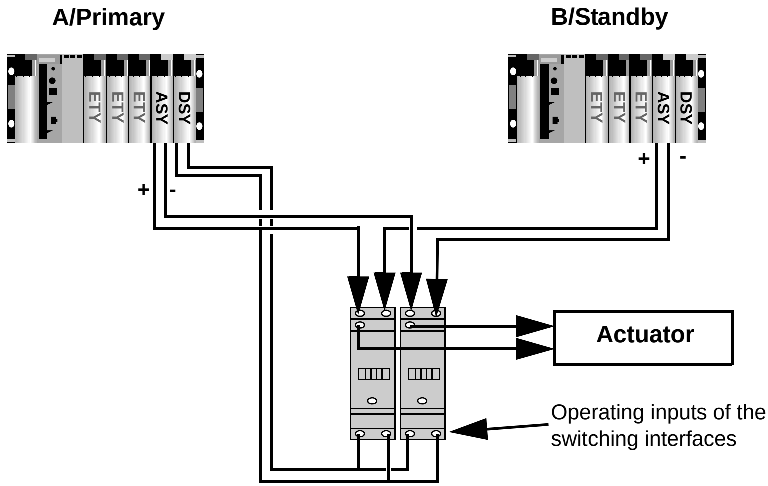
The following illustration displays an example of sensor cabling:



Analog Output module cabling

For analog output, two low level switching interfaces can be used (Telemecanique ABR-2EB312B or JM Concept GK3000D1).

The following illustration displays an example of actuator cabling:



Only one PLC acts on the operating input of the two low level switching interfaces (PLC A in the above illustration). In Primary mode, the output bit is set to 1. In Standby mode, the output bit is reset to 0.

The output bit must be managed in the section 0 of both PLC in the following way:

if bits 1 and 0 of%SW61 are set to 1 and 0 (this PLC running in Primary status):

- Then: Output bit on DSY module set to 1 (Analog Output of PLC B switched on actuator)
- Else: Output bit on DSY module reset to 0 (Analog Output of PLC A switched on actuator)

Note: The DSY module must be configured in fallback to 0.

⚠ WARNING
RISK OF UNINTENDED EQUIPMENT OPERATION AND EQUIPMENT DAMAGE Because the same application is running in both PLCs, the above sequence is the same in PLC A and PLC B: You must execute at each PLC cycle in Standby mode (first section). If not, the Output bit of the Standby PLC (reset to 0 in the above example) will be forced to 1, that is the value coming from the Primary PLC. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

Connecting Ethernet I/O

Ethernet I/O cabling

As described before, the link between the two monitored ETY modules (ETY-sync link) is used to transfer information to diagnose the Hot Standby system. It can also be used to manage Ethernet I/O devices by configuring an Ethernet I/O scanner in each monitored ETY.

The following architectures can be used:

- Low level architecture: two standard Ethernet switches connected to each monitored ETY
- High level architecture: several Ethernet ring switches connected to the Ethernet devices

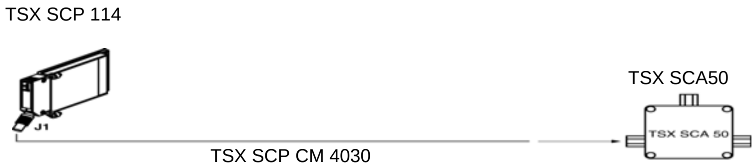
For using hubs or switches in different network topologies like star, tree or ring, refer to ConneXium catalog and Transparent Ready technical publications.

Connecting Modbus

Modbus Slave link on RS485, two wires

The Modbus Slave function is used from the card PCMCIA (TSX SCP 114). This may be located only in the module TSX SCY 21601. It is preferable for the network polarization to be implemented by the Master Modbus equipment.

The following illustration displays a Modbus Slave link on RS485, two wires:



Modbus Master link on RS485, two wires

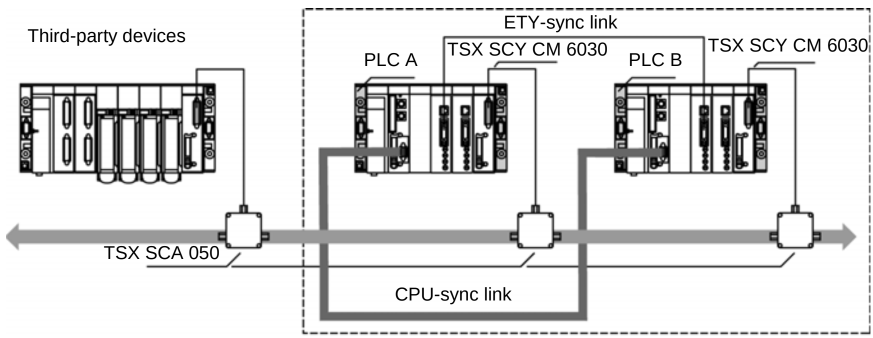
The Modbus Master function is used from the integrated channel of the module TSX SCY 21601/TSX SCY 11601. The link is type RS485, 2 wires.

When the modules are redundant (one in each PLC), the polarization of the network must be carried out starting from the two channels. Because of this, changing a module will be possible without disturbing communication.

The cord to use is the TSX SCY CM 6030. The line can be adapted by positioning the corresponding connector on **ON** in the TSX SCA 50 boxes at the end of the RS 485 line.

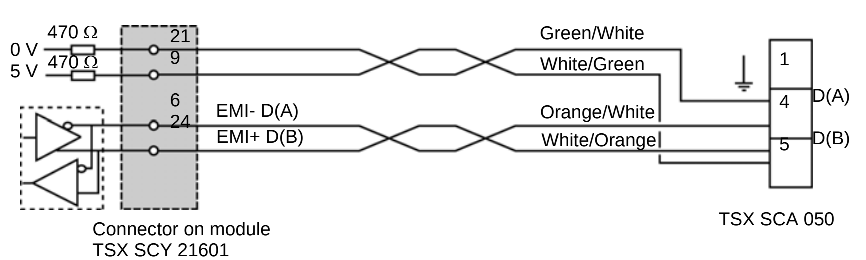
The following illustration displays the Modbus Master link on RS485:

Premium Hot Standby



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

J1



Configuring

5

Introduction

Overview

This chapter describes configuring the Premium Hot Standby PLCs.

What's in this Chapter?

This chapter contains the following sections:

5.1 Configuring a System with the Unity Pro Tabs and Dialogs

At a Glance

Purpose

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

For configuring other standard features, refer to the Unity Pro Operating Modes manual.

What's in this Section?

This section contains the following topics:

Introducing Unity Pro

Overview

Unity Pro is a Software package for programming Telemecanique 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

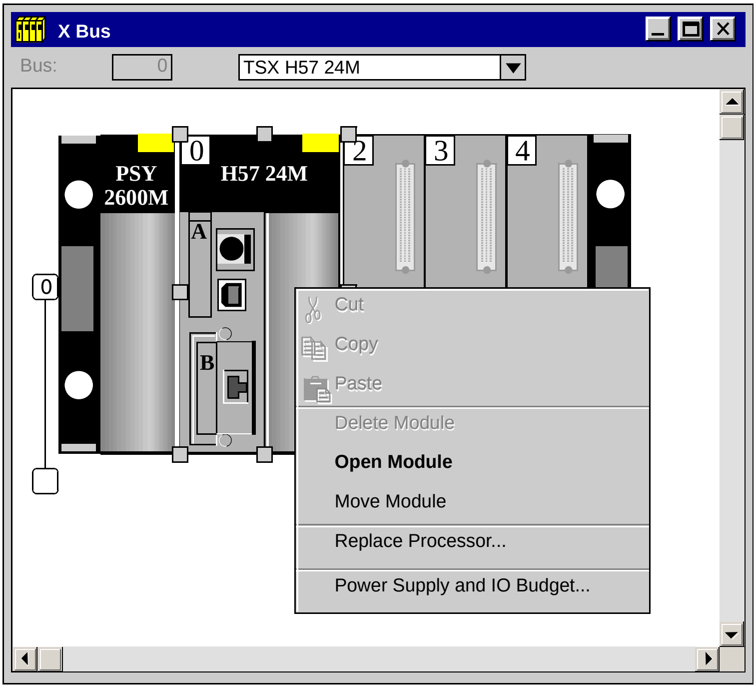
The configuration tool is used to:

- Create, modify, and save the elements used to configure the PLC station
- Set up the application-specific modules including the station
- 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

Accessing the Base Configuration

Accessing with Unity Pro

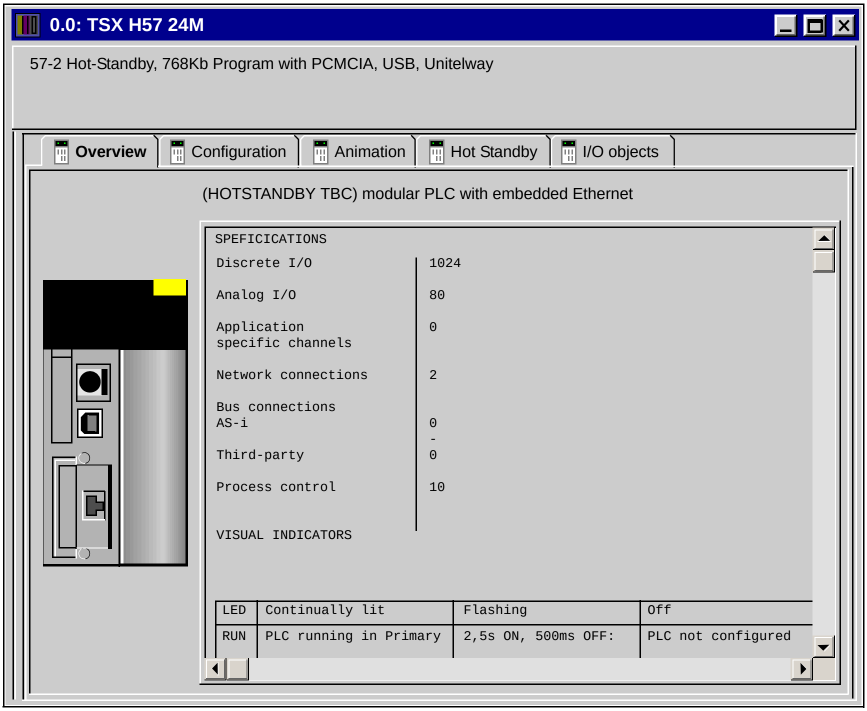
After starting Unity Pro, go to the X Bus in the Structural View of the Project Browser.

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. 
3	Select Open Module. The editor appears. The Configuration tab is default.
4	Choose one of these tabs: ● Overview ● Configuration ● Animation ● Hot Standby ● I/O Objects

Using the Overview Tab

Viewing

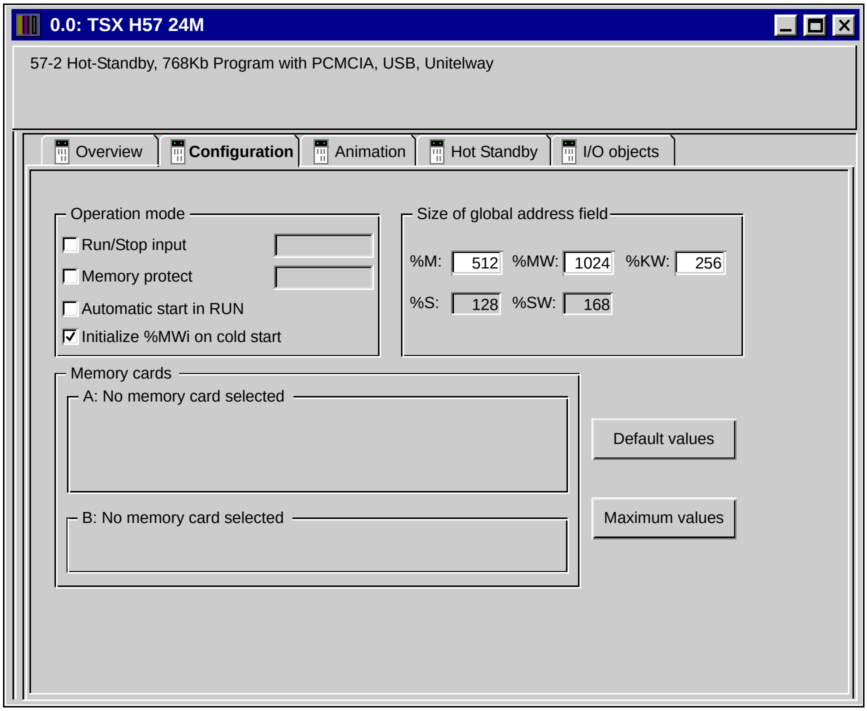
The read only Overview tab of the editor displays detailed information about the module’s specifications.



Using the Configuration Tab

Viewing the Configuration tab

Change values using the Configuration tab of the editor.



Description of the Configuration tab

Configuration tab:

Item	Option	Value	Description
Operation Mode	Run/Stop input	x	Determines the operating condition during Cold Start
	Memory protect	x	
	Automatic start in Run	x	
	Initialize %MWi on cold start	x	
Memory Cards	A:	N/A	Displays the configuration in the PCMCIA Slots
	B:	N/A	
	Default value	N/A	Permits selection of the default value: %M/%KW
	Maximum value	N/A	Permits selection of the maximum number: %M/ %KW
Size of global address field	%M	1.	Size of the different memory areas **Note:** The values for %MW has to be divisible by 8
	%MW	1.	
	%KW	1.	
	%S	2.	
	%SW	2.	
1. Enter the appropriate values. All values depend on Hot Standby configuration. 2. The values cannot be selected.			

Using the Animation Tab and PLC Screen Dialogs

Accessing the PLC Screen Dialogs

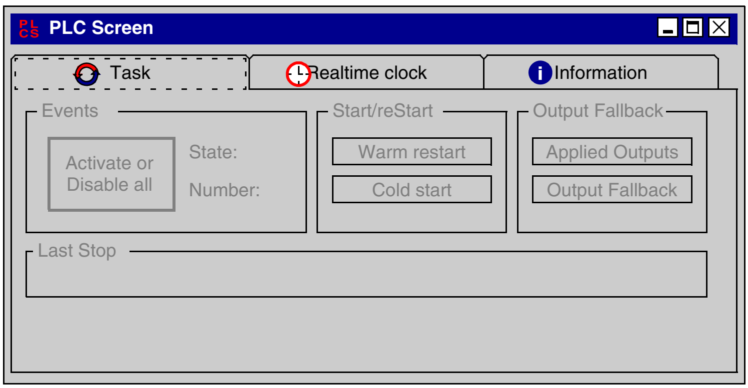
To access the Task, Realtime clock, and Information tabs of the Unity Pro Animation tab,

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

Note: The dialogs illustrated here are depicted 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:



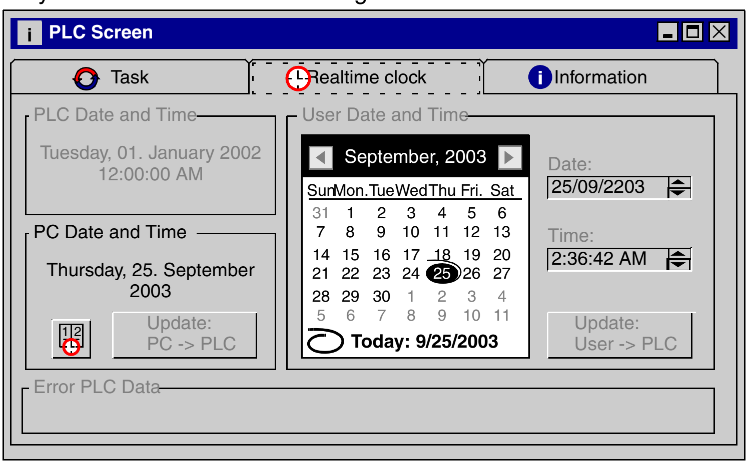
Task Tab Description

Description of the Task tab:

Item	Option	Value	Description
Events	State:	xxx	Status information of events available Online
	Number:	xxx	N/A
	Activate or Disable all	Click button	Button to control the events
Start/reStart	Warm Start	Click button	To initialize Warm Start
	Cold Start	Click 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	• Day • DD/MM/YY • Time	Indicates the day, date, time, and cause of the last controller stop

Viewing the Realtime Clock Tab

Unity Pro Realtime clock tab dialog:



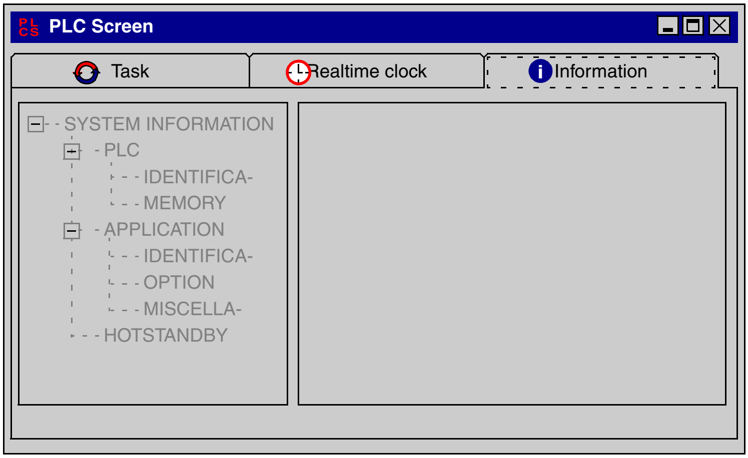
Realtime Clock Tab Description

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

Viewing the Information Tab

Unity Pro Information tab dialog:



Information Tab Description

Description of the Information tab:

Item	Option	Value	Description
System Information	PLC / Identification	PLC Range	Only Online available
		Processor name	
		Processor version	
		Hardware ID	
		Network address	
	PLC / Memory	RAM CPU size	
	Application / Identification	Name	
		Creation Product	
		Date	
		Modification Product	
		Date	
		Version	
		Signature	
	Application / Option	Upload Information	
		Comments	
		Animation Table	
		Section Protection	
		Application Diagnostic	
	Application / Miscellaneous	Forced Bits	

Item	Option	Value	Description
System Information	Hot Standby	PLC Hot Standby Status	Only Online available
		Peer PLC Hot Standby Status	
		Logic Mismatch between PLC and Peer PLC	
		PLC Name	
		CPU-Sync Link Error	
		Main Processor OS version Mismatch	
		Co Processor OS version Mismatch	
		At least One ETY do not have the minimum version V4	
		Monitored ETY OS version Mismatch	
		TCP/IP and MODBUS Addresses	
		Hot Standby Entire System State	

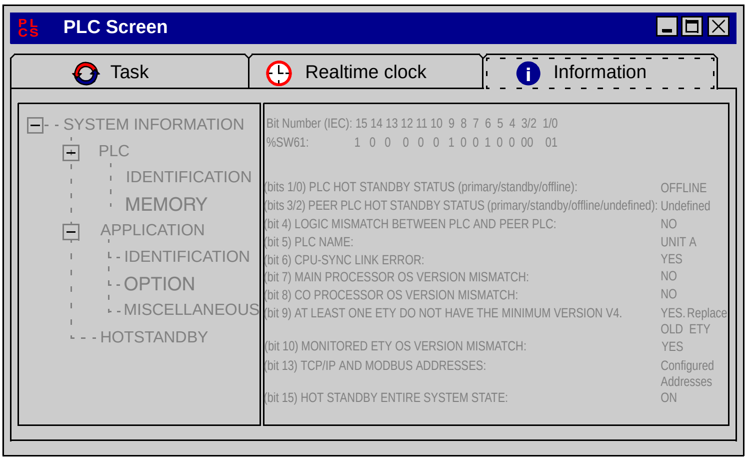
The following table presents the values in the Information Tab:

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= (1and 1): Standby mode
4	Logic Mismatch between PLC and Peer PLC	• Value=0: No • IValue=1: Yes
5	PLC Name	• Value=0: Unit A • Value=1: Unit B
6	CPU-sync link Error	• Value=0: No • Value=1: Yes
7	Main Processor OS version Mismatch	• Value=0: No • Value=1: Yes
8	Co Processor OS version Mismatch	• Value=0: No • Value=1: Yes

Bits	Line Title	String Displayed
9	At least One ETY do not have the minimum version V4	• Value=0: No. All ETY have the minimum required version. • Value=1: Yes. Replace old ETY.
10	Monitored ETY OS version Mismatch	• Value=0: No • Value=1: Yes
13	TCP/IP and MODBUS Addresses	• Value=0: Configured addresses • Value=1: Configured addresses + 1
15	Hot Standby Entire System State	• Value=0: Off • Value=1: On

Viewing the Information Tab in connected mode

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

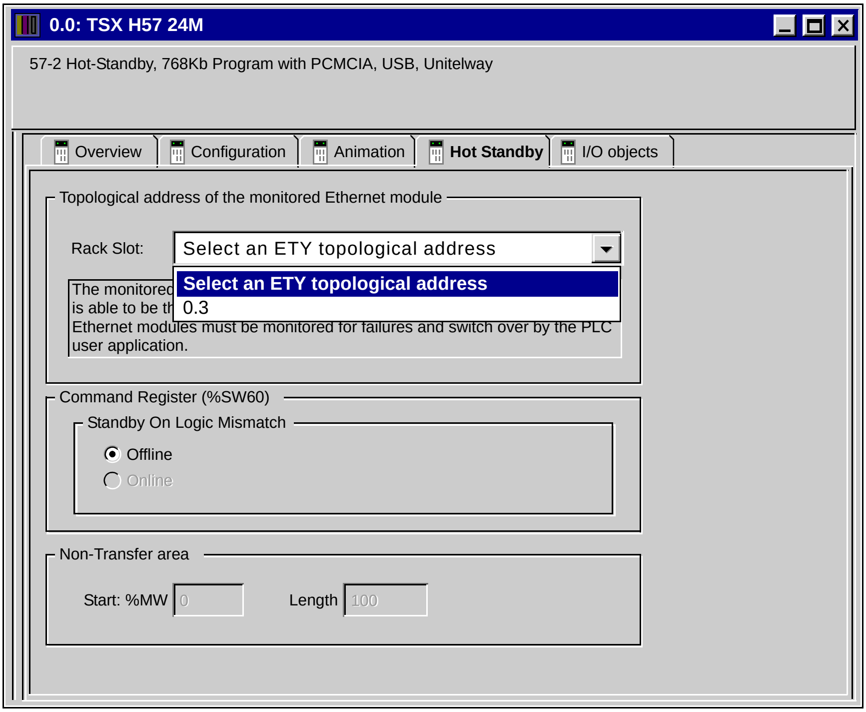


Using the Premium Hot Standby Tab

Viewing the Hot Standby Tab

Note: All the ETY modules should be configured.

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 is filled by the existing addresses of ETY cards
Command Register	Standby On Logic Mismatch	The Standby On Logic Mismatch is only in Offline
Non-transfer area	Start: %MW	%MW0 to 99: Data are not transferred
	Length	

Configuring In-rack I/O

How to configure In-rack I/O

For configuring In-rack I/O (discretes and analog), refer to the following Unity Pro user manuals:

- Premium and Atrium using Unity Pro, Discretes I/O modules user manual
- Premium and Atrium using Unity Pro, Analog Input / Output user manual

CAUTION

RISK OF EQUIPMENT DAMAGE

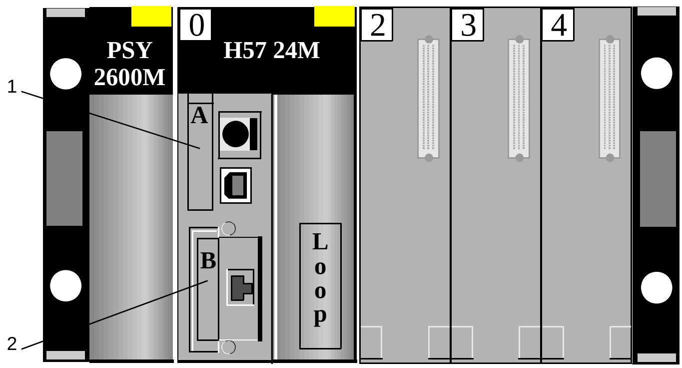
To prevent the freeze of discrete output bits when one on the two PLCs fails, you must configure output modules in fallback mode to 0. This configuration mode is mandatory when output modules are cabled in parallel with ABE7 ACC1x connection blocks. In case of negative logic, you must configure output modules in fallback mode to 1.

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

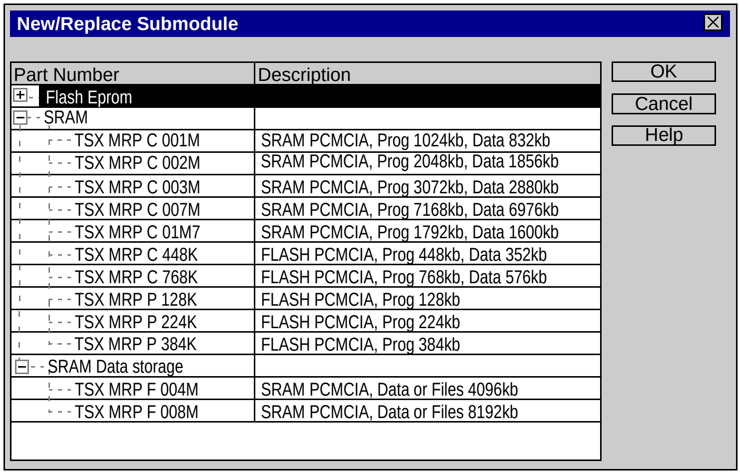
Configuring the PCMCIA Cards

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).  **1** Memory configuration of the PCMCIA card 1 **2** Memory configuration of the PCMCIA card 2

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>5</td><td>Double-click or right-click either PCMCIA card.
The New/Replace Submodule dialog appears.



</td></tr>
<tr><td>6</td><td>Add or replace the desired memory.</td></tr>
</table>

Swapping Network Addresses at Switch over

Overview

The following material describes handling network addresses at Switch over.

Handling TCP/IP address at switch over

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

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 switch over.

> **Note:** IP Address nnn.nnn.nnn.255 reserved to broadcast messages
> The user must not configure the Primary address as: nnn.nnn.nnn.254, which would cause Standby IP address to be: nnn.nnn.nnn.255. If this occurs the ETY will return the diagnostic code: Bad IP configuration.

Prior to a switch over 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 Switch over		After Switch over	
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 switch over.

WARNING

RISK OF UNINTENDED EQUIPMENT OPERATION

To prevent duplicate IP address error when several ETY modules are present in a Hot Standby PLC, the user must not configure these ETY modules with consecutive IP addresses.

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

Handling Modbus address at switch over

When a Premium Hot Standby configuration is in a nominal mode, the TSX SCP 114 module Modbus addresses are (Primary is PLC A/Standby is PLC B):

- Primary TSX SCP 114 module (A): "n"
- Standby TSX SCP 114 module (B): "n+1"

If the Standby PLC becomes Primary, the TSX SCP 114 module Modbus addresses become:

- TSX SCP 114 module B (new Primary): "n"
- TSX SCP 114 module A (old Primary): "n+1"

Note: There is no swap for Channel 0 of TSX SCY21601 and TSX SCP1160.

For testing the protocol with the T_COM_MB IODDT, only the low byte of the PROTOCOL variable has to be tested. The high byte is not significant.

⚠ CAUTION

RISK OF EQUIPMENT DAMAGE

At switchover time, it may be possible to lose a message (question or answer). To prevent this kind of communication fault, you must check by application that a station addressed on the modbus link has correctly received a message before sending a new one.

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

⚠ CAUTION

RISK OF EQUIPMENT DAMAGE

The possible value for Modbus slave number lie between 1 and 98.
If the Primary slave address is configured as 98, the Standby slave address must be configured as 1 (address 99 doesn't exist).

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

⚠ WARNING

RISK OF UNINTENDED EQUIPMENT OPERATION

To prevent duplicate Modbus address when the main rack is powered-off, it is advised to configure the SCY / SCP module in the main rack.

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

⚠ WARNING

RISK OF UNINTENDED EQUIPMENT OPERATION

Because the module configuration is not transferred from Primary to Standby, the protocol that is configured in an SCP114 module (Modbus, Uni-Telway, Character mode) must not be changed when the application is running.

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

5.2 Configuring TSX ETY 4103/5103 Modules

At a Glance

Purpose

This material describes configuring TSX ETY 4103/5103, Premium Ethernet modules, using Unity Pro. For a complete 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.*

⚠ WARNING

RISK OF UNINTENDED EQUIPMENT OPERATION AND EQUIPMENT DAMAGE

The Global Data service must not be used in a Premium Hot Standby TSX ETY.

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

What's in this Section?

This section contains the following topics:

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) have to be configured in Unity Pro.

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

Description of the Hot Standby Solution

ETY Hot Standby allows 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. The sending TSX ETY then waits for a specified time-out (50 ms) 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:** Schneider Electric recommends that a switch (not a hub) is 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 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 switch over from the old to the new Primary TSX ETY.

> **Note:** Failure of the Monitored ETY is a condition for the Primary system to leave the Primary state. Failure of a non Monitored ETY is not a condition for the Primary system to leave the Primary state.

⚠ CAUTION

RISK OF EQUIPMENT DAMAGE

Failure of a non Monitored ETY has to be managed by the application program.

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

Monitored ETY Module

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

The position of the monitored ETY is unrestricted in the Premium configuration (in terms of firmware, configuration, and position), both PLCs must be configured identically in terms of material and module position. ETY modules are linked either through Ethernet switches (one switch per ETY) or a Ethernet crossover cable. By using an Ethernet transceiver, an optical connection can be used for long distance.

To configure the Monitored ETY module in Unity Pro, the topology address of the Monitored ETY module should be set in the Hot Standby TAB of the CPU screen The user selects in the combo box from a list of existing ETY card addresses.

The Monitored ETY Module is used to diagnose the status of the complete Premium Hot Standby configuration. This is achieved via the ETY-sync link. It can also be used to manage Ethernet I/O devices by configuring an Ethernet I/O scanning utility.

To perform a switch over when an ETY-sync link failure appears on Primary side, the Ethernet I/O scanning service must be configured in the monitored ETY. On the contrary, if this service is not configured in the monitored ETY, an ETY-sync link failure will not generate a switch over.

For better performance and more predictable time at switch over, the different Ethernet services should be split between the different ETYs of the configuration. For example, if you configure an I/O scanning in the monitored ETY, we advise to configure other Ethernet services (if needed) in another ETY module.

In case of failure in the Monitored ETY module, the CPU sends a state change command to all configured ETY modules present on the X-BUS (main and extended rack).

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

⚠ WARNING

RISK OF UNINTENDED EQUIPMENT OPERATION

If you use a Cross over cable between the Monitored ETYs, make sure that the I/O scanning service is not configured in the ETY modules.

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

⚠ WARNING

RISK OF UNINTENDED EQUIPMENT OPERATION

We do not advise using the Monitored ETY without I/O Scanning setup unless the Primary PLC is never addressed by an external equipment over the ETY-sync link.

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

⚠ WARNING

RISK OF UNINTENDED EQUIPMENT OPERATION

When the I/O Scanning service is used in the Monitored ETY, we advise using one switch on each ETY.

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 Hot Standby state is primary, and all client/server services are active.
- **Standby Mode**
  The Hot Standby state is standby, and all server services are active except I/O Scanning.
- **Standalone Mode**
  Occurs when ETY is in a non redundant system, or if the CPU module is not present or is not healthy.
- **Offline Mode**
  CPU is stopped.
  CPU module is in Offline mode.

The Premium Hot Standby and the ETY operating modes are synchronized by the conditions described in the following table.

CPU Module Status	HSBY State	ETY Operating Mode
Not present or unhealthy	N/A	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 switch over, when an ETY goes to offline mode, or when a new application is downloaded to 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...
Standalone	ETY configuration table
Primary	Configured IP address from the ETY configuration table
Standby	Configured IP address + 1 from the ETY configuration table
Power off to power on	The IP address is determined by which controller powers up first (after check remote, the second ETY takes IP Address + 1), or if powered up at the same time, by a "resolution algorithm": ● Lower Copro MAC address: IP address / Primary state ● Higher Copro MAC address: IP address + 1 / Standby state

Offline event table:

HSBY ETY Mode	IP address
Primary to Offline	Configured IP address from the ETY configuration table, if the peer controller does not go to Primary state
Standby to Offline	Configured IP address + 1 from the ETY configuration table

When the CPU stops, 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.

Power on and Ethernet Services

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

HSBY State	Status of ETY services				
	Client Services	**Client/Server Services**	**Server Services**		
	I/O Scanner	**Modbus Messaging**	**FTP**	**SNMP**	**HTTP**
Power off to power on	Run	Run	Run	Run	Run
Primary	Run	Run	Run	Run	Run
Standby	Stop	Run	Run	Run	Run
Offline	Stop	Run	Run	Run	Run

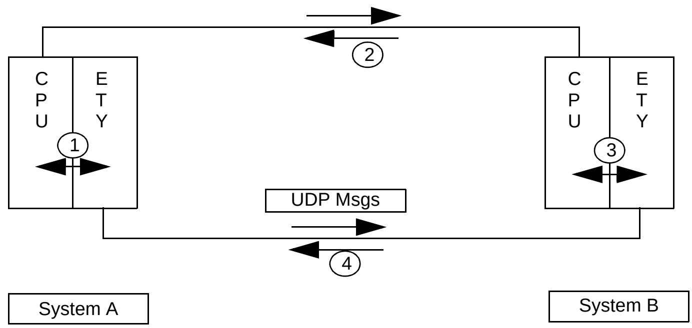
Hot Standby Switch over

The following steps describe how ETYs coordinate the Hot Standby switch over (PLC/ETY A is the Primary and the PLC/ETY B is the Standby):

Step	Action
1	A switch over event occurs. System A CPU commands HSBY ETY A to switch to the Offline mode.
2	System A CPU informs System B CPU that a switch over event has occurred and it is to become the Primary.
3	System B CPU commands HSBY ETY B to become the new Primary.
4	System A HSBY ETY initiates an exchange of UDP messages with System B HSBY ETY to coordinate the IP address switch over.

Hot Standby Switch over Illustration

The following illustration displays a switch over event:



IP Address Assignment

Configuring the ETY

The ETY TCP/IP address has to be configured manually in Unity Pro and not from a remote device acting as a BOOTP / DHCP server. Since the Primary and Standby controllers must have an identical configuration, the configured IP Addresses will be 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.

Note: For a Premium Hot Standby, the two IP Addresses will be consecutive.

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	Configured IP Address, if peer controller **does not** go to Primary
Transition from Standby to Offline	Configured IP Address + 1

Note: Offline - Results depend on whether or not the other controller is detected to be in transition into the primary state. If current IP is the configured IP address and the other PLC is in transition to Primary, then IP address changes to IP address + 1.

IP Address Restriction

Note: Configuring ETY
Do not use either broadcast IP Address or broadcast IP Address - 1 to configure a ETY.
The Primary ETY and the Standby ETY IP addresses must be in the same network and subnetwork.

Duplicate ID Address Checking

> **Note:** The duplicate IP address checking is only performed at power-up of the Hot Standby PLC. It is not performed during a switch over or after a removal / replacement of the 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 IP Address + 1.

The ETYs integrated into the Premium Hot Standby configuration coordinate this IP Address swapping with the management of Ethernet services used.

⚠ CAUTION

RISK OF EQUIPMENT DAMAGE

Do not use the address IP + 1. For a Premium Hot Standby configuration do not use consecutive IP addresses for consecutive ETY modules configured. Do not configure the Primary address as: nnn.nnn.nnn.254, which would cause Standby IP address to be: nnn.nnn.nnn.255. Doing that: the ETY would then return the diagnostic code: Bad IP configuration.

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

Network Effects of Premium Hot Standby

Overview

Premium Hot Standby is a powerful feature of the ETYs, a feature that increases the reliability of your installation. Hot Standby uses a network, and using the Hot Standby feature over a network can affect the behavior of:

- Browsers
- Remote and Local clients
- I/O Scanning service
- FTP/TFTP server

The following are factors you may encounter while using the Premium Hot Standby solution.

Browsers

If a browser requests a page and during the process of downloading that page an IP Address swap occurs, the browser will either hang or time out. Click the **Refresh** or **Reload** button.

Remote Clients

Hot Standby swaps affect remote clients.

An ETY will reset under the following conditions:

- **Remote Connection Request during Hot Standby Swap**
  If a remote client establishes a TCP/IP connection during a Hot Standby swap, the server closes the connection using a TCP/IP reset.
- **Hot Standby Swap during Remote Connection Request**
  If a remote client makes a connection request and a Hot Standby swap occurs during the connection request, the Server rejects the TCP/IP connection by sending a reset.
- **Outstanding Requests**
  If there is an outstanding request, the ETY will not respond to the request, but the ETY will reset the connection.

The ETY will do a Modbus logout if any connection has logged in.

Local Clients

During a swap, the ETY will reset all client connections using a TCP/IP reset.

I/O Scanning Service

I/O Scanning provides the repetitive exchange of data with remote Ethernet I/O devices. While the PLC is running the Primary ETY sends Modbus Read/Write, requests to remote I/O devices, and transfers data to and from the PLC memory. In the Standby controller, the I/O scanning service is stopped.

When the Hot Standby swap 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 Standby.

After the swap, the new Primary ETY re-establishes the connection with each I/O devices. It restarts the repeat exchange of data with these re-connections.

The TSX ETY 4103/5103 provides the I/O scanning feature. Configure using Unity Pro software.

Note: When the I/O Scanning service is configured in the Monitored ETY, an ETY-sync link failure on the Primary side will generate a switch over. The ETY-sync link failure bit can be read in the ETY module by using an explicit exchange (READ_STS) and the IODDT T_GEN_MOD. The bit %MWr.m.MOD.2.2 is set to 1 in case of failure.

⚠ CAUTION

I/O SCANNING AND SWITCH OVER WITH CRITICAL APPLICATIONS

The following Ethernet I/O scanning considerations have been taken during a switch over:

- If a communication function block is used for TCP/IP, the block will not complete its transaction
- While the ETY is in the process of performing the transaction, a new communication function block may become active.
- The input states of the scanned Ethernet I/O devices will follow the state defined in the last value option configured in the I/O scanning table of the ETY module (in Unity Pro software)
  These two states are either:
  - Set to 0
  - Hold last (will be set in the I/O scanner)

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

⚠ CAUTION

RISK OF EQUIPMENT DAMAGE

To guarantee a proper operation in the system, do not configure multiple ETY module to I/O scan the same I/O device or IP address.

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

⚠ CAUTION

RISK OF EQUIPMENT DAMAGE

To prevent a pulse on Scanned I/Os when one of the two PLCs fails, the user must configure output Ethernet devices with the Hold last value mode. This configuration has to be done with the configuration tool that is provided with the Ethernet device. For the Ethernet devices that only support the fallback to 0 position, a pulse may appear during a switchover.

Failure to follow these instructions can result in 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 swap occurs, the Primary and Standby ETYs close the FTP/TFTP connection. If a user sends an FTP/TFTP request during the swap, the communication is closed.

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

5.3 Configuring Registers

At a Glance

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's in this Section?

This section contains the following topics:

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.

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

Reverse Transfer Words

Four system words,%SW62 to %SW65, are dedicated to transfer data from the Standby controller to the Primary.

These system words can be used by the application program (in the first section) to register diagnostic information.

The data coming from the Standby is transferred at each scan and is available to the Primary.

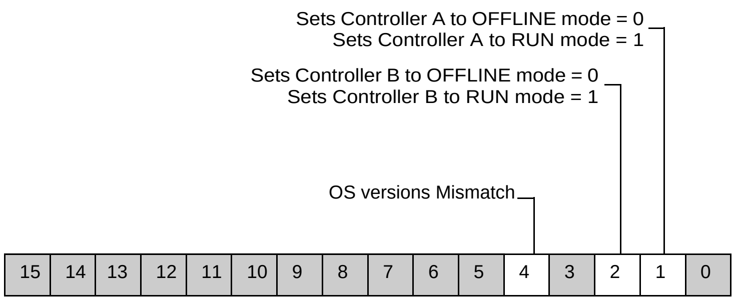
Understanding the Unity Command Register

Setting the Bits in the Command Register

The 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.

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



System Word %SW60.1

Controller A OFFLINE/RUN mode:

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

System Word %SW60.2

Controller B OFFLINE/RUN mode:

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

System Word %SW60.4

Standby behavior if OS Versions Mismatch:

- %SW60.4 = 1
  If OS Versions Mismatch with Primary PLC, Standby stays in standby mode
- %SW60.4 = 0
  If OS Versions Mismatch with Primary, Standby goes to Offline mode

Firmware OS Mismatch.This relates to main processor OS version, embedded copro OS version, monitored ETY OS version and enables a Hot Standby system to operate with different versions of the OS running on the Primary and Standby.

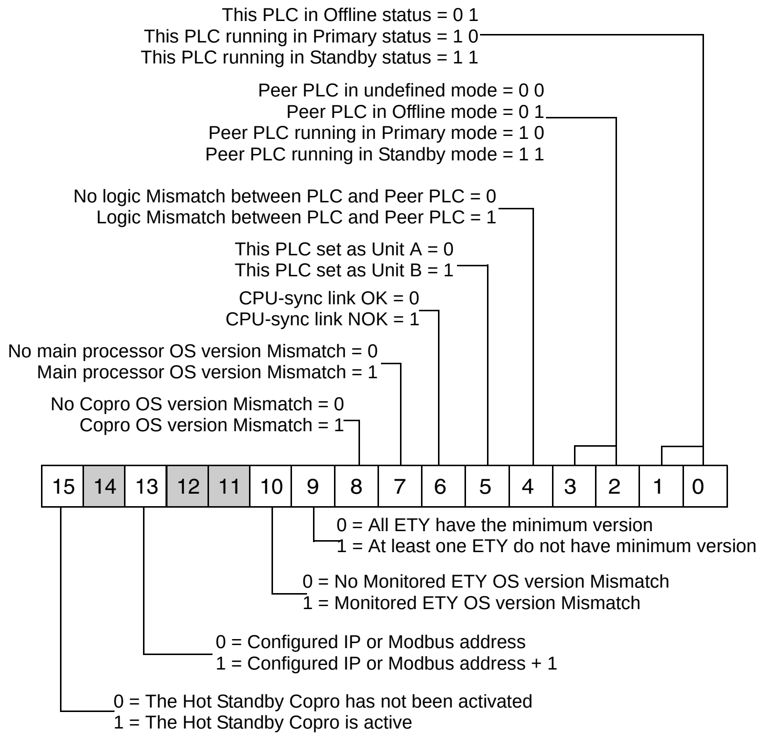
Understanding the Unity Status Register

Bits in the Hot Standby Status Register

The Hot Standby Status Register is a readable 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/Offline 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 Words %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 = 1means 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 = 1means 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 Word %SW61.4

%SW61.4 is set to 1 whenever a logic mismatch is detected between the Primary and Standby controllers.

System Word %SW61.5

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

- %SW61.5 = 0 means the PLC with the lowest MAC address becomes PLC A.
- %SW61.5 = 1 means the PLC with the highest MAC address becomes PLC B

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

System Word %SW61.6

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

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

System Word %SW61.7

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

- %SW61.7 = 0 means no OS version firmware mismatch
- %SW61.7 = 1 means OS version mismatch. If OS version mismatch is not allowed in the command register (bit 4 = 0), the system will not work as redundant as soon as the fault is signaled

System Word %SW61.8

This bit indicates if there is a COPRO OS version mismatch between Primary and Standby:

- %SW61.8 = 0 means no COPRO OS version mismatch
- %SW61.8 = 1 means COPRO OS version mismatch. If OS version mismatch is not allowed in the command register (bit 4 = 0), the system will not work as redundant as soon as the fault is signaled

System Word %SW61.9

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

- %SW61.9 = 0 means all the ETY modules have the minimum version
- %SW61.9 = 1 means at least one ETY module doesn't have the minimum version. In this case, the PLC will go to Offline mode.

System Word %SW61.10

This bit indicates if there is a monitored ETY OS version mismatch between Primary and Standby:

- %SW61.10 = 0 means no monitored ETY OS version mismatch
- %SW61.10 = 1 means monitored ETY OS version mismatch. If OS version mismatch is not allowed in the command register (bit 4 = 0), the system will not work as redundant as soon as the fault is signaled

System Word %SW61.13

This bit indicates which IP or Modbus address is applied by each ETY or SCP module of the configuration:

- %SW61.13 = 0 means each ETY or SCP module applies its configured IP or Modbus address
- %SW61.13 = 1 means each ETY or SCP module applies its configured IP or Modbus address + 1

System Word %SW61.15

If %SW 61.15 is set = 1, the setting indicates that Ethernet Copro device is set up correctly and working.

Transferring User Data

General

To enable the Standby to take over control from the Primary, the Hot Standby configuration status is sent from the Primary to the Standby via a database.

Transferred Hot Standby Status Information

The Hot Standby status information that will be transferred includes:

- The values of the Primary In-rack output modules (%Q and %QW objects)
- The values of command words and adjustment parameters (%MW.r.m.c objects)
- The values of discrete input and output forcing
- User application data (located and unlocated)
- System data of the Primary PLC
- All instances of DFB and EFB data
- SFC states
- A part of System Bits and Words
  List of System Bits and Words that are exchanged permanently:
  - %S30, %S31, %S38, %S50, %S59, %S93, %S94,
  - %SW0, %SW1, %SW8, %SW9, %SW49... %SW53, %SW59, %SW60, %SW70, %SW108.
  %SD18 and %SD20 are only exchanged at switch over.

Database

The Database is built automatically by the Primary PLC Operating system (transparent to the customer application, no use of specific language instruction for database exchange) and sent at each Primary PLC cycle to the Standby PLC.

This exchange is performed via the embedded Ethernet coprocessor of the two Hot Standby PLCs and the CPU-sync link.

The size of the database is approximately:

- 180 kilobytes on TSX H57 24M
- 428 kilobytes on TSX H57 44M

Data storage

The Unity Premium range offers three types of memory card:

- Application
- Application and data storage
- Data storage

The data storage area is a memory zone that can be used to backup/restore data in the memory card using specific EF in the application program.

The maximum size of this data storage area is 8 Mb and cannot be used to store Hot Standby Status information. It is thus not part of the database exchange between Primary and Standby.

It is only possible to read data using two memory cards (1 card in each PLC) having the same contents.

Using Initialized Data

Loading at Cold-start Time

The Unity Premium Hot Standby supports initialized data.

Initialized data allows you to specify initial values for the data that are to be loaded at cold-start time. Declare the variables before a cold start.

Updating Online

In addition to declaring values before a cold start, you can update the initial values Online.

Updating the initial values online creates a mismatch situation in a redundant system, in this case the Standby goes to Offline mode.

Synchronization of Real Time Clocks

Synchronization of Primary and Standby Real Time Clocks

Each processor in a Unity Premium Hot Standby configuration has a savable **R**eal **T**ime **C**lock hardware component which manages the current Date and Time. This Date and Time is part of the database that is sent at each Primary PLC cycle to the Standby PLC, but the synchronization of the new Primary RTC is only done at switchover time.

Prior to switchover, only the Primary and Standby date and time system words (%SW49 ... %SW53) are synchronized, because they are part of the database.

Programming/Debugging

6

Presentation

Overview

This chapter describes the Programming and the Debugging of a Premium Hot Standby system.

What's in this Chapter?

This chapter contains the following sections:

6.1 Development of an Application

At a Glance

Purpose

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

What's in this Section?

This section contains the following topics:

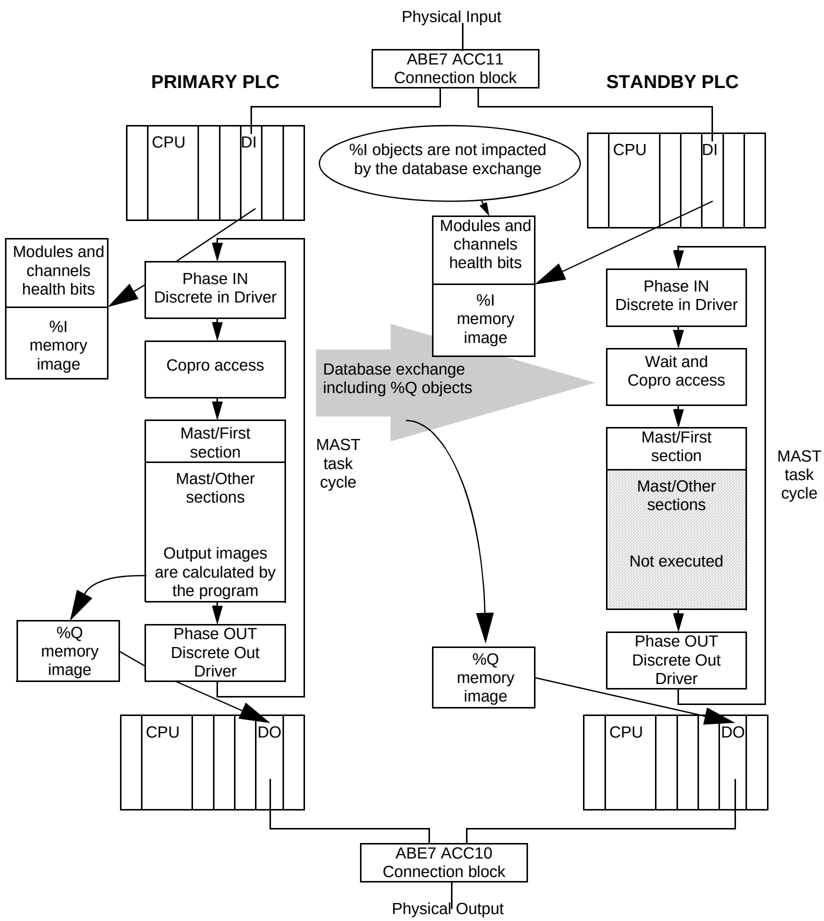
Programming Method

General points

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

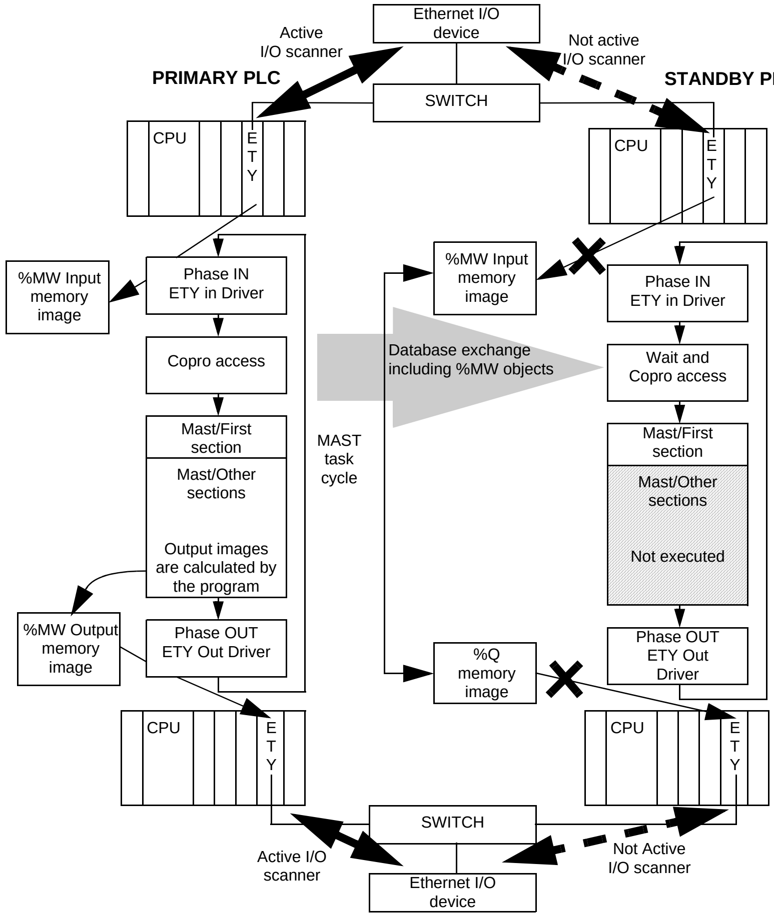
Operation cycle with In-rack I/O

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:



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 (comprising the first section)
  - Acquires the local input for the in rack modules
  - Updates the local output of the in rack modules
  - Sends the database to the Standby PLC
  - Manage the Ethernet I/Os of the dedicated ETY I/O scanner table
  - Retrieves diagnostic information from the Standby PLC
  - Manages its' own diagnostic information and the information of the Hot Standby Premium system
  - Monitor health of Power Supply, CPU and In-rack modules
- PLC in Standby mode:
  - Only the first section of the application program is executed
  - Acquires the local input for the in rack modules
  - Applies the output images received from the Primary to the output of the local in rack modules
  - Receives from the Primary the Ethernet I/O images
  - Retrieves diagnostic information from the Primary PLC
  - Manages own diagnostic information and the information from the Hot Standby Premium system
  - Monitors health of Power Supply, CPU and In-rack modules

The PLC in Offline mode does not perform application program and I/O management. Offline is mainly a fault state when the PLC can't be neither a Standby nor a Primary PLC.

The first section (section 0) is executed by both Primary and Standby PLC. If you need to send information from the Standby to the Primary, it is recommended to test the state of the PLC by checking the %SW61 status register (bits 0 and 1) at the beginning of the first section. When the PLC is in Standby mode, it is recommended to check the In-rack modules health informations 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 at each scan to the Primary.

CAUTION
RISK OF EQUIPMENT DAMAGE Actuators that are connected in parallel on two output modules are only managed by the Primary PLC (refer to the Programming Method section for more details) They must not be written in the section 0 of the Standby PLC. **Failure to follow these instructions can result in injury or equipment damage.**

It is also possible to manage actuators locally in both PLC. In this case, actuators are not connected in parallel on two output modules but directly to one output module in each PLC.

⚠ CAUTION
RISK OF EQUIPMENT DAMAGE When actuators are managed locally in each PLC, the output values must be evaluated in the section 0 at each PLC scan. If this is not done, the Standby output value will be erased by the value coming from the Primary PLC. **Failure to follow these instructions can result in injury or equipment damage.**

How to Program a Premium Hot Standby Application

Processor configuration

The MAST task can be defined in cyclic or periodic cycle.

The period should take into account the requirement time for redundancy.

The following table presents the characteristic MAST tasks on Processor:

Characteristics	TSX H57 24M/TSX H57 44M
Max period (ms)	255
Default period (ms)	20
Min. period (ms)	1 (0 to cyclic)
Period increment (ms)	1
Period Time Base (1/10ms)	10
Period Time Unit	20
Max Watchdog (ms)	1500
Default Watchdog (ms)	250
Min. Watchdog (ms)	10
Watchdog increment (ms)	10
WD Time Base (ms)	1
WD Time Unit	250

For more details, see *Adjusting Mast Task Properties in a Premium Hot Standby PLC, p. 143*.

Event and counting restrictions

Ethernet I/O counting modules are compatible with a Hot Standby configuration, but they have to be used carefully. It is impossible to guarantee that counting pulses are taken into account mainly at switch over time. More generally, events management is not recommended in a Hot standby application (event tasks, fast inputs,...).

Language restrictions

- The use of edges is not recommended. It is not possible to guarantee that they are taken into account during a switch over
- The use of the SAVE_PARAM function is not recommended 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 being not transferred from the primary to the standby. More generally, the explicit instructions like WRITE_CMD and WRITE_PARAM have to be used carefully.
  Example: If the WRITE_CMD is related to a "Modbus change to character mode" command in TSX SCP 114 module, this change will only be done in the Primary PLC. In case of switch over, the new Primary will restart with the Modbus mode rather than the Character mode.
- It is not possible to replace the initial values of the declared variables with a save attribute (,e.g.: DFB variables) with the current values -> no use of %S94
- The use of DFB is not recommended in the first section.

Forbidden Legacy function blocks

The following Legacy function blocks are Not Allowed:

- PL7 Counter
- PL7 Drum
- PL7 Monostable
- PL7 Register 32
- PL7 Register 255
- PL7 TOF, PL7 TON, and PL7 TP
- PL7 3 Timer

Forbidden Standard Function Blocks

The use of the TON / TOFF / TP function blocks is forbidden in the first section.

Using Communication Function Blocks

For using a communication function block (i.e. WRITE_VAR), you are advised to:

- locate the management parameters in the %MW from 0 to 99 (those that are not transferred from Primary to Standby),
- initialize the Length parameter each time the function block is started,
- use an external Timer function block as a replacement of the Timeout parameter.

If the management parameters cannot be located in the %MW from 0 to 99, and in the case of a switchover when a function block is active, then the activity bit must be reset to 0 by the application before restarting the function block in the new Primary.

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 current task bit 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**
  Normally at 0, this bit is set to 1 by a power restoral with data save.
  It is reset to 0 by the system at the end of the first complete cycle and before the outputs are updated.

In the event of cold start (or warm restart), if you want the application to be processed in a particular way, you must write the corresponding program conditional on the test that %SW10.0 is reset to 0 (or %S1 is set to 1) at the start of the master task program. %SW10 and %S1 are significant in Primary and 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 is given by:

- The values of the Primary In-rack output modules (%Q and %QW objects)
- The values of command words and adjustment parameters (%MWr.m.c objects)
- The values of discrete input and output forcing
- The input/output values of all the remote devices
- The user application data (located and unlocated) and system data of the Primary PLC
- All instances of DFB and EFB data
- SFC states
- Some system bits and words:
  - %S30, %S31, %S38, %S50, %S59, %S93, %S94
  - %SW0, %SW1, %SW8, %SW9, %SW49... %SW53, %SW59, %SW60, %SW70, %SW108
  - %SD18 and %SD20 are only exchanged at switch over

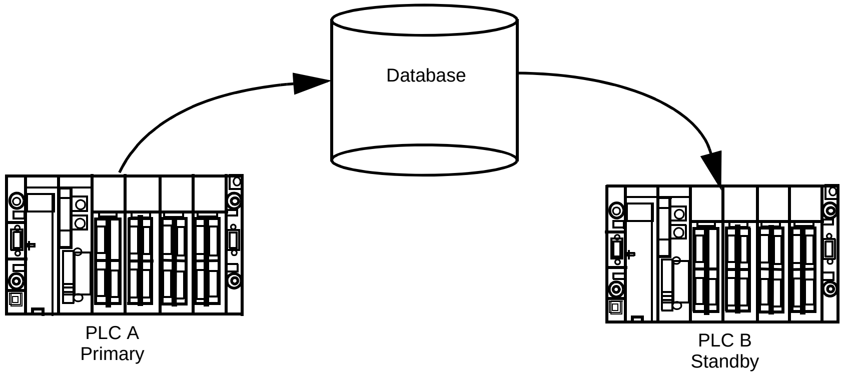
To do this, the two PLCs have to share a Database that is built automatically by the Primary PLC.

Note: To make possible a local diagnostic of I/O modules in the standard 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 status parameters (%MWr.m.c objects)

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" and "Analog I/O modules" documentations.

Illustration

The following illustration displays information worked out by the Primary PLC:



Exchange

The Database is built automatically by the Primary PLC Operating System and sent at each Primary PLC cycle to the Standby PLC. This exchange is performed via the embedded Ethernet coprocessor of the two HSBY PLCs.

The size of database is:

- TSX H57 24M: 180 kilobytes
- TSX H57 44M: 428 kilobytes

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 (A) and the Standby (B). 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 in the Phase IN of the Mast cycle at the same time.

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 purpose. The %Q objects received from the Primary PLC are applied at the end of the Standby Mast cycle.

CAUTION

RISK OF EQUIPMENT DAMAGE

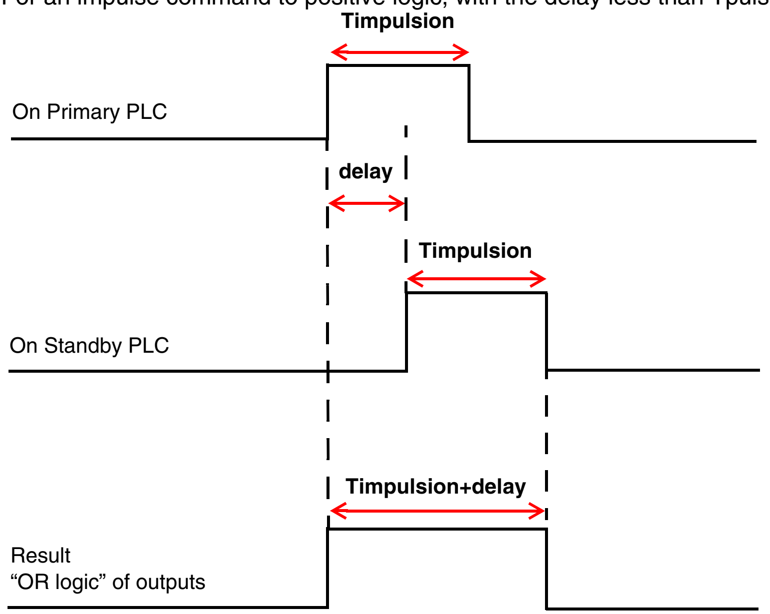
The output bits that are connected in parallel between the 2 PLCs must not be written in the section 0 of the Standby PLC.
This leads to affect the output bit values that are sent by the Primary.

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

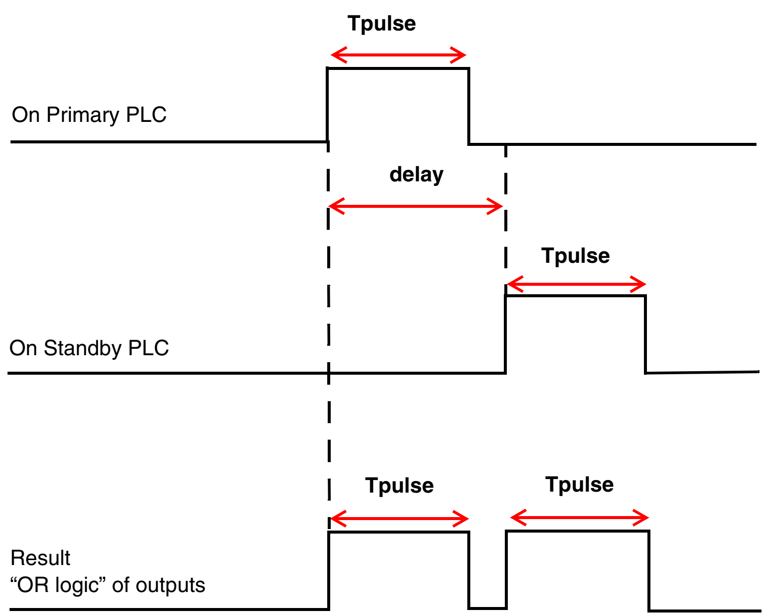
The output modules are connected in parallel to the physical output via a specific connection block. The result of an impulse command is based on the time of the impulse and the delay to apply this impulse in the Standby.

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

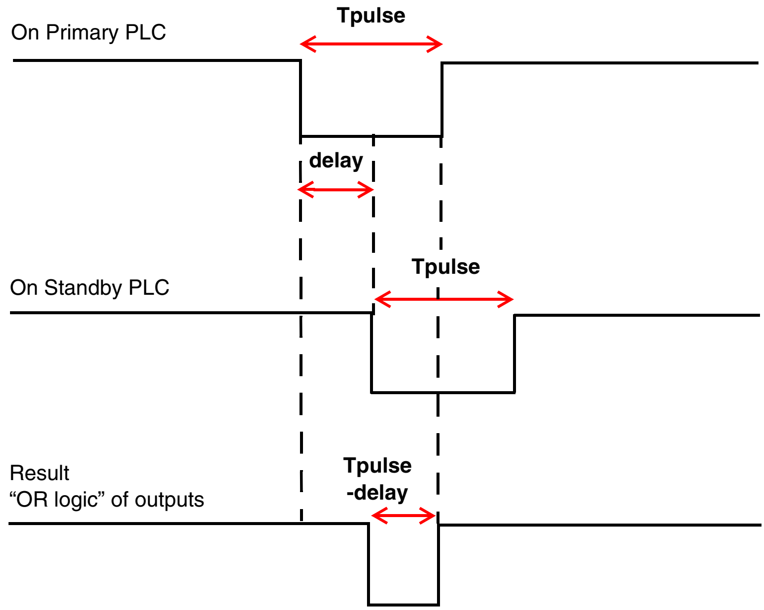
For an impulse command to positive logic, with the delay less than Tpulse:



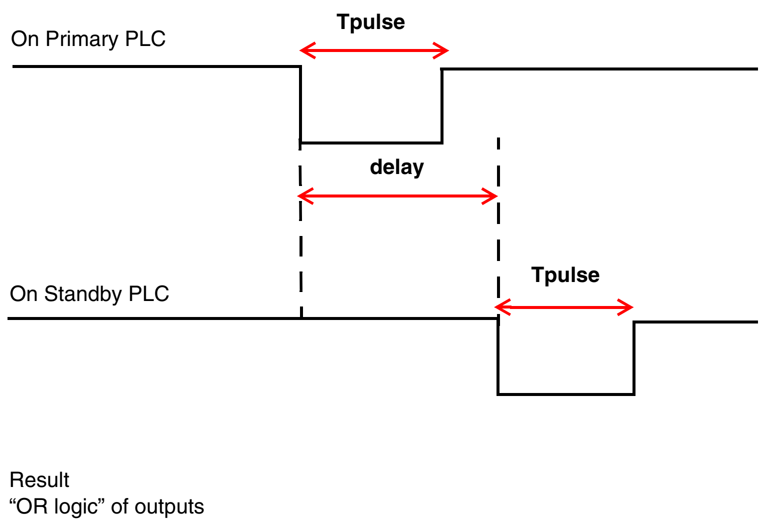
For an impulse command to positive logic, with the delay more than Tpulse:



For an impulse command to negative logic, with the delay less than Tpulse:



For an impulse command to negative logic, with the delay more than Tpulse:



Local I/O management

It is possible to manage actuators locally in both PLC. In this case, actuators are not connected in parallel on two output modules but directly to one output module in each PLC. They may be written with different values at the same time depending on the application program processing.

⚠ CAUTION

RISK OF EQUIPMENT DAMAGE

When actuators are managed locally in each PLC, the output values must be evaluated in the section 0 at each PLC scan.
If not, the Standby output value will be erased by the value coming from the Primary PLC.

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

Transferring the program in the Primary and the Standby

Transferring the program

Transfer the program to the Primary CPU:

- Connect the PC to USB plug or Uni-Telway
- Use the Unity Pro command: **PLC** → **Transfer program to PLC**.

Transfer the program to the Standby CPU:

- Connect the PC to USB plug or Uni-Telway
- Use the Unity Pro command: **PLC** → **Transfer program to PLC**.

6.2 Debug Program

At a Glance

Purpose

This section describes the Debug Program of the Premium Hot Standby.

What's in this Section?

This section contains the following topics:

Debugging

Introduction

An application for a Premium Hot Standby PLC integrates the control/command part of the procedure like a non Hot Standby PLC. It doesn't integrate any specific function blocks relating to the redundancy.

Debug and Diagnostic

The following table presents Debug and Diagnostic operations on CPUs:

Diagnostic		**TSX H57 24M**	**TSX H57 44M**
Diagnostic Function Block		Yes	Yes
Diagnostic Buffer		Yes	Yes
Diag. buffer characteristics	Max buffer size	16K8	25K8
	Max errors	160	254
Breakpoint		One single Bkpt	One single Bkpt
Step by step (Into, over, and out)		Yes	Yes
Variable animation		• End of Mast • Watch Point	• End of Mast • Watch Point
Link animation		Yes	Yes

Debug the control/command of the procedure

Debugging the application must be carried out on one PLC. This PLC is automatically Primary.

Note: For programming / debugging an application in a Hot Standby PLC, it is recommended:
- To use a Standalone PLC,
- To use PLC A (if 2 PLCs are connected) with PLC B in Non Conf state.

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 all variables that are displayed in a Standby animation table are those coming from the Primary, excepted 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.

Debugging the redundancy part

Debugging the application must be carried out with the Premium Hot Standby System.

The debugging tools proposed by Unity Pro must not be used (example: the "step by step"). They may introduce malfunctions into the Premium Hot Standby architecture.

We advise you to proceed as follows:

- Static verification
  Check that:
  - The application restrictions have been applied
  - The MAST task characteristics have been configured properly
- Dynamic verification
  After each PLC has been made live (application already transferred), check that the redundancy function is correctly performed in each PLC: the bit %SW61:X15 is equal to 1 and the bit %SW61:X6 is equal to 0.

The Hot Standby Premium being in the nominal functioning mode, confirm that:

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

Note: A switchover is not generated when the Primary application stops on a breakpoint.

Non Transfer Area

A fixed size of %MW is not transferred from the Primary to the Standby. These %MW are from offset 0 to offset 99.

Adjusting Mast Task Properties in a Premium Hot Standby PLC

Introduction

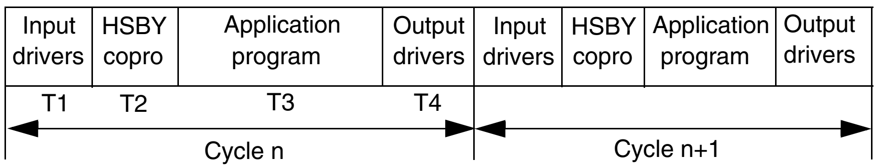
After a reminder on Mast task execution modes, this part describes the Execution time measurement method and gives advices to adjust the Mast task period.

Reminder on Mast Task Execution Modes

The Mast task can be configured using one of the two following execution mode:

- cyclic mode,
- periodic mode.

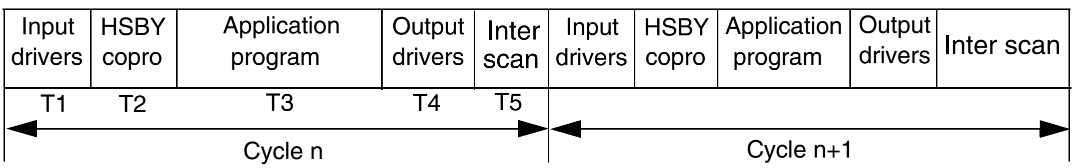
Cyclic mode:



This type of operation consists on sequencing the task cycles, one after another.

After having updated the outputs, the system performs its own specific processing then starts another task cycle, without pausing.

Periodic mode:



In this operating mode, input acquisition, application program processing and outputs update are all carried out periodically over a defined period set between 1 and 255 ms.

At the start of the PLC cycle, a time out whose current value is initialized to the defined period starts the countdown.

The PLC cycle must be completed before this time out expires and launches a new cycle.

> **Note:** If a Fast task is configured (although multitasking is not recommended in a Premium Hot Standby application), it interrupts the execution of the Mast task which has a lower priority. As a results, the execution time of the Mast task is increased.

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

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.

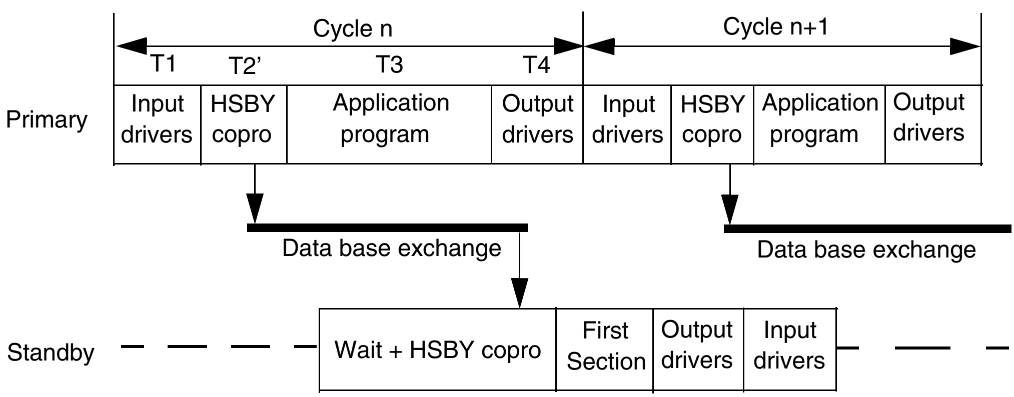
→ execution time of the last Mast cycle = %SW30 = T1 + T2 + T3 + T4

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:

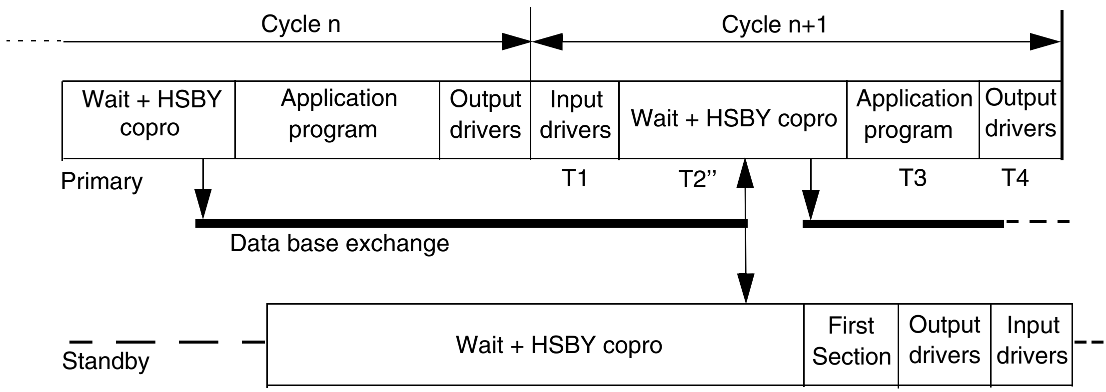
1. The data exchange has no impact on the Primary cycle time:



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 + 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 cycle time:

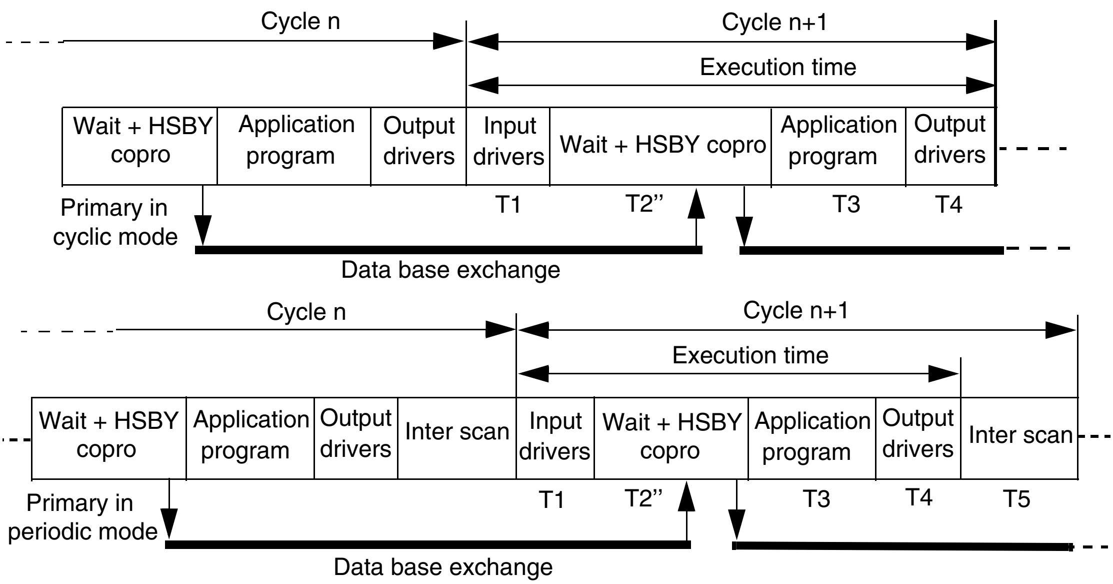


In this second case, the execution time of the HSBY part (T2'') is increased with the time to be waited until the complete transmission of the data base.

→ 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.

Advices 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%

⚠ WARNING

RISK OF UNINTENDED EQUIPMENT OPERATION AND EQUIPMENT DAMAGE

The Mast task period and the Watch dog value have to be adjusted very carefully taking into account the data base exchange and the Fast task interruptions. In case of permanent period overrun, the correct functioning of the Premium Hot Standby system is not guaranteed.
More generally, the Mast task period and the watch dog value must be evaluated in the Primary PLC when the Premium Hot Standby system is normally running (it means when there is a Primary and a Standby PLC). This evaluation must never be done in a Standalone system or in an Offline PLC.

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

Operating

7

Introduction

Overview

This chapter provides information about Operating the Premium Hot Standby System.

What's in this Chapter?

This chapter contains the following sections:

7.1 Start/Stop System

At a Glance

Purpose

This section describes how to start or stop a Premium Hot Standby System.

What's in this Section?

This section contains the following topics:

Starting the two PLCs

Invalid applications

The PLCs do not have a valid application. When the PLCs are made live and they are waiting for an application transfer, there is no Primary A/Standby B selection. The first PLC receiving the application will become the Primary PLC after a RUN command, the other will be the Standby PLC after receiving the same application and a RUN command.

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

Valid applications

The use of a time-lag relay on the main rack supply of one of the PLCs makes it possible to guarantee which PLC will be the Primary PLC when the two PLCs are made live simultaneously. During this process, the PLC, which has the time-lag relay in its supply cabling, will be the Standby PLC.

If there is no time-lag relay, the choice of Primary/Standby depends on the copro MAC address. The PLC with the lowest MAC address becomes the Primary PLC (A). The other one becomes the Standby PLC (B).

CAUTION

RISK OF EQUIPMENT DAMAGE

In case of CPU replacement, the identification A/B of the 2 PLCs can be inverted. Respect this, specially if the application requires a strong link between the geographical position of each PLC and its identification.

Failure to follow these instructions can result in 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.**F**4.01.6E.E1
- MAC2 = 00.80.**B**4.01.6E.E1

The MAC1 is higher than the MAC2.

Second example:

- MAC1 = 00.80.**F**4.01.6**E**.E1
- MAC2 = 00.80.**D**4.01.6**F**.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 simple PLC but respecting the following stop order:

- Stop the Standby PLC
- Stop the Primary PLC

If the Standby PLC is not stopped first, a switch over would occur when the Primary PLC is stopped.

7.2 Switchover

At a Glance

Purpose

This section describes the Switchover of the Premium Hot Standby.

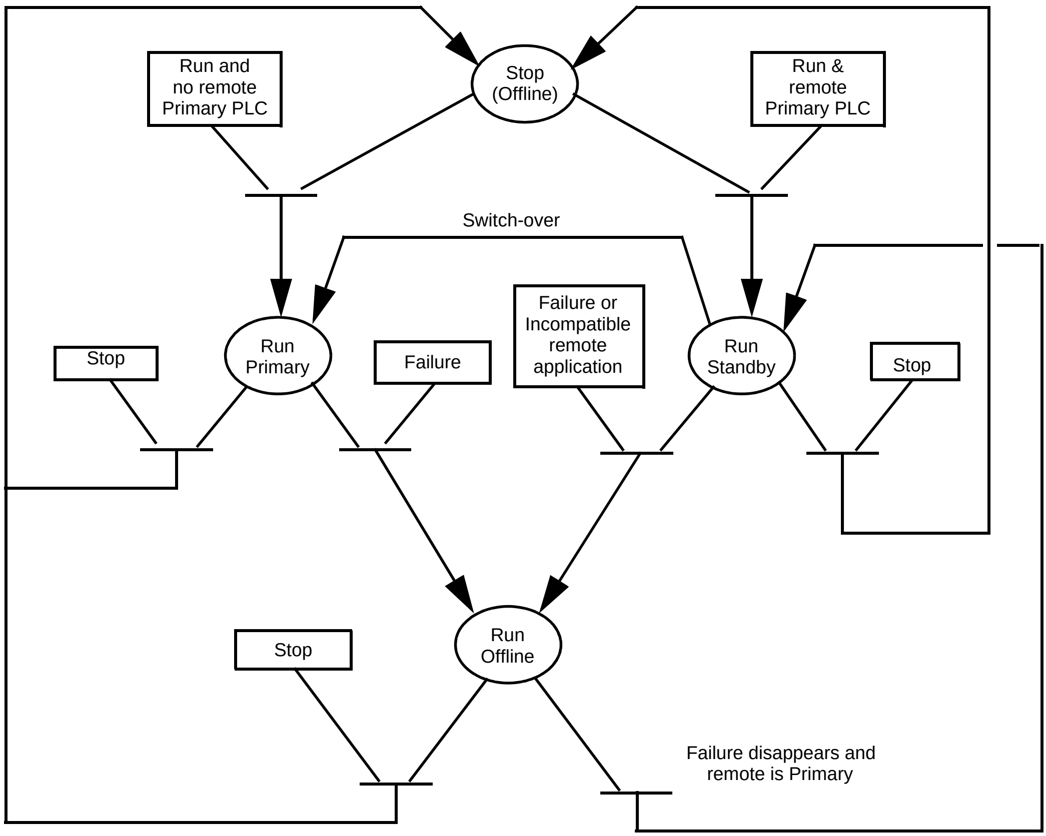
What's in this Section?

This section contains the following topics:

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 remote PLC state, local failure state, application mismatch state:

If...	Then...
The remote PLC is Primary, the two applications are identical and no local failure	The PLC restarts in Standby mode
The remote PLC is Primary and the two applications are not identical or there is a local failure	The PLC restarts in Offline mode

If...	Then...
There is no remote Primary and no local failure	The PLC restarts in Primary mode
There is no remote Primary but there is a local failure	The PLC restarts in Offline mode

A local failure is mainly:

- A power supply failure on the CPU rack
- An application program fault that generates a HALT state
- An hardware or firmware failure on the CPU module
- An hardware or firmware failure on the monitored ETY module
- A cable disconnection between the monitored ETY and the first hub/switch
- A CPU-sync link failure (only when PLC is Standby)

At Warm start, the PLC restarts depending on the previous PLC state, Stop or Run. If Run, the PLC restarts depending on the remote PLC state, local failure state, application mismatch state (refer to the above table).

Note: When a cable failure appears between the Monitored ETY and the first switch, the Hot Standby PLC reacts depending on the I/O Scanning configuration.

ETY-sync link cabling	**Failure**	**Monitored ETY configuration**	
		No I/O scanning configured	**I/O scanning configured**
Cross over cable	Cable failure or disconnection	Primary stays Primary Standby goes Offline	Primary goes Offline Standby goes Primary
Double switch	Cable failure or disconnection on Primary side	Primary stays Primary Standby goes Offline	Primary goes Offline Standby goes Primary
	Cable failure or disconnection on Standby side	Primary stays Primary Standby goes Offline	Primary stays Primary Standby goes Offline

Conditions for Switch over

Overview

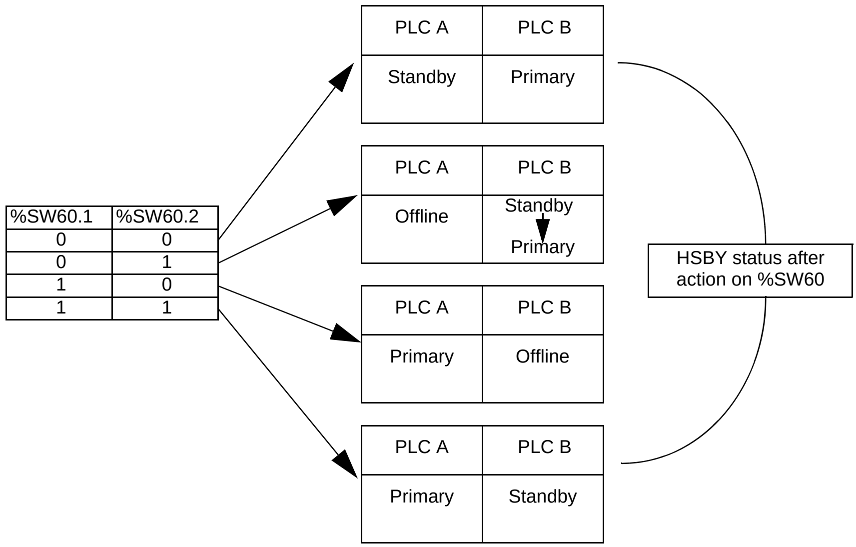
The manual Switch over is commendable from application program or requests.

Example of Switch over with PLC B in Standby mode

Before the action on %SW60, the status are:

- The two Bits are at 1 (default value set by the system)
- The PLC A is Primary
- The PLC B is Standby

When one of the following actions is done on the command register %SW60 in the Primary PLC (bits 1 and 2), it generates a change of state of the two PLCs as it is shown in the right part of the following illustration:



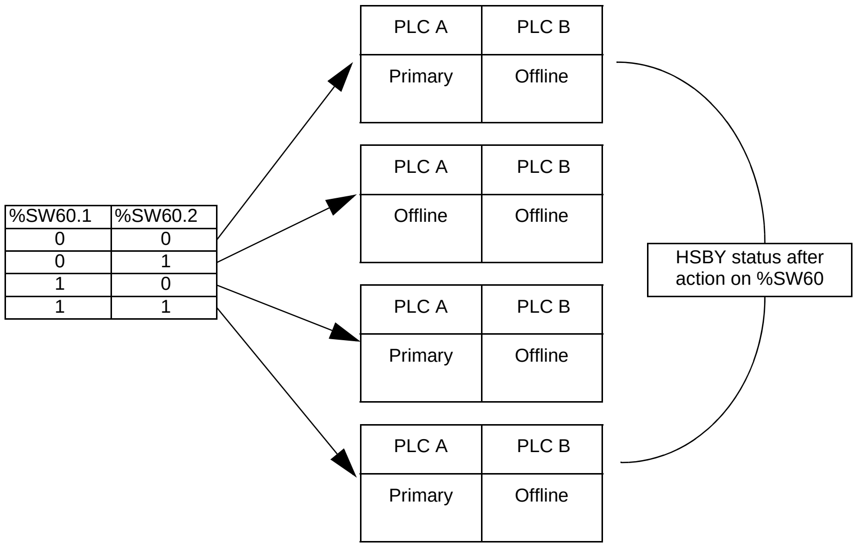
When the action is done, the two bits are automatically set to 1 by the system.

Example of Switch over with PLC B in Offline mode

Before the action on %SW60, the status are:

- The two Bits are at 1 (default value set by the system)
- The PLC A is Primary
- The PLC B is Offline due, for example, to a hardware or firmware failure in the monitored ETY

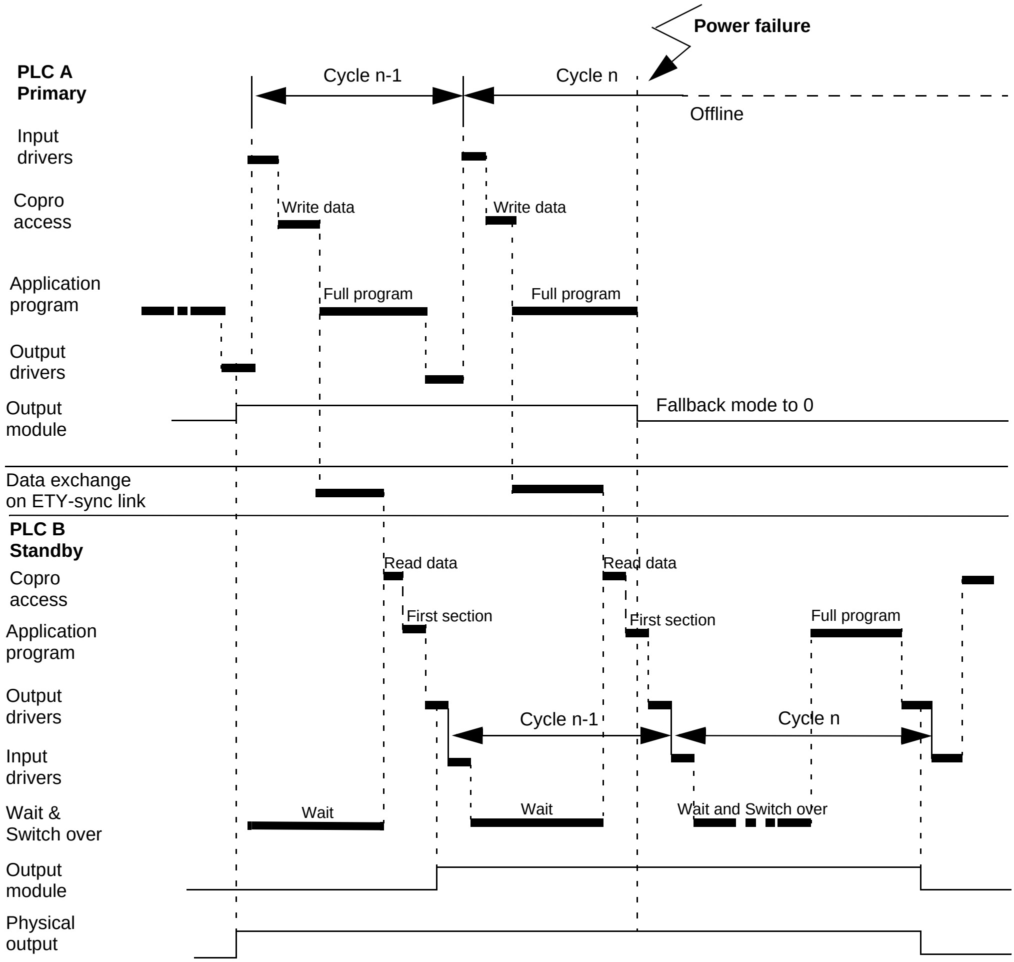
The following illustration is an example of Switch over with the PLC B in Offline mode:



When the action is done, the two bits are automatically set to 1 by the system.

Switch over on Primary failure

The following figure displays the behavior when a power supply failure or a main processor crash occur on the Primary PLC:



Note: During the switch over, the physical output is maintained at the last value received from the Primary PLC. When the PLC B starts in Primary mode, the %I object are refreshed from the physical input (parallel cabling). The application program calculates the new output values and applies these values on the output module.

⚠ WARNING

RISK OF UNINTENDED EQUIPMENT OPERATION

When an output is set to 1 in the cycle preceding the Event (example: Power Failure), there is a risk of having a pulse to 0 on the Probe.
To avoid that, use in-rack I/O for applications that can support this kind of pulse.

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

Maintaining

8

Introduction

Overview

This chapter provides information about Maintaining a Premium Hot Standby System.

What's in this Chapter?

This chapter contains the following topics:

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.

If the Primary has an error, the Standby is notified and assumes the Primary role.

If the Standby has 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 of the failure and assumes 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 Confidence Tests

The system automatically performs two kinds of confidence tests on the Premium Hot Standby CPU:

- Startup tests
- Run time tests

Conducting Startup Tests

Startup confidence testing on the Premium Hot Standby PLC with Unity Copro attempt to detect hardware errors in the module before the application is allowed to run.

If the module fails any of its tests, it will remain offline and will not communicate with the other Premium Hot Standby PLC.

Conducting Run Time Tests

Run time tests are related to the interface between the main processor and the Ethernet embedded coprocessor of the Premium Hot Standby CPU.

If the coprocessor fails, the Premium Hot Standby CPU remains Offline and will not communicate with the other CPU.

Detecting and Diagnosing Failures in a Premium Hot Standby

Important Information

Please note.

If...	Then...
Component of Primary fails	Control shifts to Standby
Component of Standby fails	Standby goes offline
CPU-sync link fails	Standby goes offline

Finding Diagnostic Information with Unity Pro

Errors and switch overs are logged in the diagnostic buffer. To view the log,

Step	Action
1	Select **Tools** → **Diagnostic Viewer** from the main menu.

Note: The diagnostic messages that are stored in the diagnostic buffer are not transferred from the Primary to the Standby.

Finding More Information in this Manual

Refer to the following sections

Type of failure	Refer to section
Primary CPU and ETY-sync link failures	See *Detecting Primary CPU and ETY-sync link failures, p. 167*
Standby CPU and ETY-sync link failures	See *Detecting Standby CPU and ETY-sync link failures, p. 168*
CPU-sync link failures	See *Detecting CPU-sync Link Failures, p. 169*
Application program checksum failures	See *Checking for Identical Application Programs—Checksum, p. 170*

For more details on failure detection, please refer. *System Detailed Behavior upon Failures, p. 199*

Type of failure	Refer to section
Halt or Stop Events on PLC	See *Halt or Stop Events on PLC, p. 203*
Hardware or Firmware CPU Failure	See *Hardware or Firmware CPU Failure , p. 206*
Power Failure on the Main Rack	See *Power Failure on the Main Rack , p. 209*
Power Failure on an Extendable Rack	See *Power Failure on an Extendable Rack, p. 213*
Hardware or Firmware ETY failure	See *Hardware or Firmware ETY failure , p. 217*
Hardware or Firmware Failure on ETY Dedicated to HMI and SCADA	See *Hardware or Firmware Failure on ETY Dedicated to HMI and SCADA , p. 220*
Failure on the Ethernet Copro	See *Failure on the Ethernet Copro , p. 223*
CPU-sync link failure between Primary and Standby PLCs	See *CPU-sync link failure between Primary and Standby PLCs , p. 226*
I/O Scanner Disconnection	See *Monitored ETY and I/O Scanner Disconnection , p. 228*
Full Ethernet I/O Link Disconnection	See *Full Ethernet I/O Link Disconnection, p. 234*
Hardware Failure of a Digital Module	See *Hardware Failure of a Digital Module , p. 236*
Hardware Failure of the SCP card in CPU or SCY	See *Hardware Failure of the SCP card in SCY , p. 239*

Detecting Primary CPU and ETY-sync link failures

Non mastered Primary CPU failure

The following table presents a Non mastered Primary CPU failure:

Stages	Description
1	A communication error occurs in the Standby Copro that manages the CPU-sync link.
2	Standby Copro reports this error to the Standby CPU.
3	Standby CPU sends a message to its local Monitored ETY to get a status of the ETY-sync link.
4	Because the Primary PLC is not responding, the Standby CPU gets a wrong status from its local Monitored ETY.
5	The Standby PLC becomes Primary.

Mastered Primary CPU failure

The following table presents a Mastered Primary CPU failure:

Stages	Description
1	The Primary CPU sends a take control message to the Standby CPU through the CPU-sync link before entering the Offline mode.
2	Standby goes to Primary mode.

Primary ETY-sync link failure

The following table presents a Primary ETY-sync link failure:

Stages	Description
1	The Primary CPU checks every scan the Monitored ETY status.
2	After receiving a wrong status, the Primary CPU sends a take control message to the Standby CPU through the CPU-sync link before entering the Offline mode.
3	Standby goes to Primary mode.

Detecting Standby CPU and ETY-sync link failures

Standby CPU failure

The following table presents a Standby CPU failure:

Stage	Description
1	A communication error occurs in the Primary Copro that manages the CPU-sync link.
2	The Primary Copro reports this error to the Primary CPU.
3	The Primary CPU stays Primary and update the remote station status to Offline or Undefined into its status register.

Standby ETY-sync link failure

The following table presents a Standby CPU failure (Primary CPU is assumed to work fine):

Stage	Description
1	A communication error occurs in the Standby monitored ETY that manages the ETY-sync link.
2	The Standby ETY reports 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, Primary stays acting as Primary and the Standby will go to Offline because a disconnection on Standby side.If the status is not OK, it will send a take control to the Standby before entering Offline mode.

Detecting CPU-sync Link Failures

Important Information

Facts

1	CPU-sync link connects the two Copros.
2	Using the CPU-sync link, the Primary controller communicates with the Standby on every Mast cycle.
3	Primary sends either **1.** Data message **2.** Health message

> **Note:** If both the Primary and Standby do not hear from each other, either station can detect a CPU-sync link failure.

Standby Detects a Failure

At first,

Step	Action	Result
1	Standby gets no response from the Primary on the CPU-sync link	• There is no more data base exchange from primary to standby • The system is no longer redundant as long as the Ethernet copro of the PLC is in failure mode

Standby Assumes Control

The Standby becomes Primary

Step	Action	Result
1	After the Primary controller goes offline or disappeared	Health message or no answer from the Primary
2	Standby controller scans the ETY-sync link once	-
3	If Standby controller gets no response	Standby knows that the failure must be on both the Primary Copro and Primary CPU
4	Standby assumes control	-

Checking for Identical Application Programs—Checksum

Important Information

Please note

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's instruction, checksum (CKSM), is transferred from the Primary to the Standby along with any other necessary data.	The Standby validates the new checksum (CKSM) against its existing checksum (CKSM).
2	Standby determines if mismatch occurs.	**1.** Mismatch: Standby goes Offline **2.** No mismatch: system operates normally
3	The controller returns to Online and is the Standby as soon as the application programs are identical.	

Replacing a Faulty Module

Important

You may replace a faulty module while a system is running.

Ensure that the replacement module:

1. Installs in the Standby backplane
2. Resides in the same position in both backplanes
3. Is same type of module

Same type of module means ETY4103 replaces ETY4103.

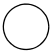
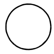
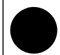
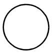
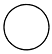
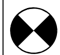
⚠ WARNING
RISK OF UNINTENDED EQUIPMENT OPERATION Follow this informations: **1.** Perform a switch over if replacing a Primary. **2.** Do NOT remove a Primary controller with under powerer (Hot Swap). **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

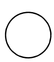
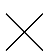
Troubleshooting a Hot Standby PLC

Troubleshooting the PLC

To determine which components have failed, note PLC's status on CPU LED display and Monitored ETY LED display:

CPU LEDs					Monitored ETY LEDs			Failure type	Description
RUN	**ERR**	**I/O**	**STS**	**ACT**	**RUN**	**ERR**	**STS**		
●	○	×	⧓	●	●	○	●	No failure.	Normal state. CPU in Primary mode.
⧓	○	×	⧓	●	●	○	●	No failure.	Normal state. CPU in Standby mode.
×	●	×	×	×	×	×	×	CPU faults.	Serious hardware or firmware fault. Correct operation of the CPU is no longer assured.CPU no more Primary nor Standby.
×	×	×	○	×	×	×	×	Copro fault.	Copro auto tests failed.
⧓	⧓	×	×	×	×	×	×	Application fault.	Halt instruction, watchdog overrun,...CPU in Offline mode.
×	×	●	×	×	×	⧓	×	ETY fault.	Module not configured or configuration in progress.
⧗	○	●	×	×	○	●	○	ETY fault.	Serious hardware or firmware fault. Correct operation of the ETY is no longer assured.CPU is Offline.

CPU LEDs					Monitored ETY LEDs			Failure type	Description
RUN	ERR	I/O	STS	ACT	RUN	ERR	STS		
								ETY fault.	Software operation error. Temporary state causing module re initialization.
								ETY fault.	• 2 flashes on ETY STS LED: module has no MAC address. • 3 flashes on ETY STS LED: Ethernet cable not connected on the module or Hub side. • 4 flashes on ETY STS LED: the module IP address is duplicated. • 5 flashes on ETY STS LED: module configured as a BOOTP client and is waiting for a BOOTP server response. • 6 flashes on ETY STS LED: invalid IP address. Module is set to its default IP address.

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
	No significant

Modifying and Upgrading

III

At a Glance

Purpose

This part describes Modifying and Upgrading in a Premium Hot Standby System.

- Handling application Modification
- Handling CPU OS Upgrade

What's in this Part?

This part contains the following chapters:

Handling Application Modification

9

Introduction

Overview

This chapter provides information about application modification in a Premium Hot Standby system.

What's in this Chapter?

This chapter contains the following topics:

Understanding Premium Hot Standby Logic Mismatch

Needing Identical Application Programs

In a fault-tolerant redundant system and under normal operating conditions, both controllers must load the identical application program (also called a logic program). The application program is updated every scan by transferring data from the Primary to the Standby. Only the Standby by controller detects a logic mismatch and reports error on Primary.

The following conditions cause a mismatch in the application program: a difference between:

- Programs
- Animation tables
- Comments (on variables and types)

> **Note:** Animation Tables and Comments
> Both animation tables and comments (on variables and types) may be excluded from the mismatch by not being included in the upload information.
> - Exclude by selecting `Tools | Project Settings | Build tabs` (default). In the Upload Information area, select `without`.
> - Inclusion requires downloading the application program

When a mismatch exists, the Standby Controller goes to Offline, and switch over 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:

- Modify (edit) online an application program in the Standby while the Primary controls the process
- Modify online an application program in the Primary while the Primary controls the process
- Download an offline-modified application program to the Standby.

> **Note:** Modify online an application program means:
> - modify the executable code (whatever the task) by adding, suppressing or changing an instruction in the code,
> - modify a configuration parameter by changing a value in a configuration screen.

Online/Offline Modifications to an Application Program

Overview

A Hot Standby configuration is no longer redundant when there are different applications (executable program or hardware/logical configuration) in the Primary and Standby PLCs. In this case, the Standby PLC is Offline and so the switch over cannot occur. Logic mismatch is not supported by a Premium Hot Standby system.

The following procedure describes how the user can modify the application in the two PLCs of a Premium Hot Standby system with a minimum impact on the process.

Online Mollifications allowed

The following table describes the modifications allowed in on-line mode:

Modifications	**Description**
General	• Name of station, program, section • Comment of station, configuration, program, section • Documentation summary • Animation table • Integrated operator screen • Functional view • Security informations: passwords, protection attributes
Program	• Sections of program: add, delete, change execution order • Modify the code of the section (task section, SR, transition, Action, DFB sections • Modify the code of SFC chart
Configuration / communication	Change I/O module parameters
Global variables (used in animation table or operator screen)	• Symbol on a used variable • Topologic address on a used variable • Initial value on a used variable • Comment on a used variable • Create, remove or modify unused variables (EDT, DDT) • Create, remove or modify unused variables (FB)
Used DFB	• All comments • Add a private or public variable • Delete or change unused private variable • Initial value of parameters and variables • Section of DFB: add, delete, change execution order • Modify the code of a section • Create a new DFB type • Delete an unused DFB type
Used DDT	• Create a new DDT type • Delete an unused DDT type

This kind of modification leads to a partial application download and the PLC doesn't change its execution mode (RUN / STOP).

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	Ensure both Primary (A) and Standby (B) controllers are in Run Primary and Run Standby mode.
2	Modify online the application on the Primary PLC Results: • The Standby PLC (B) goes to Offline mode (Logic Mismatch) • The Primary PLC (A) is active on the process The system is no longer operating in redundant mode
3	After tests, save the application in the PC
4	Download the saved application to the Standby PLC (B) Result: • During the transfer, the PLC (B) is in Non Configuration state • At the end of transfer, the PLC goes to the Stop/Offline mode
5	Initiate RUN command on the Standby PLC (B) Result: The PLC B goes to Standby mode Note: If the Primary PLC A failed during the Online modification, the user has to connect Unity Pro to the PLC B and perform a STOP/RUN command. The PLC B will go in Run/Primary mode.

Note: The online modification in the Standby controller first is not allowed by Unity Pro.

Note: An online modification in an animation table or in a comment will not generate a logic mismatch if the Animation tables and Comments options are 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 requires an application download:

Modifications	Description
Program	Modify the code of EVT sections
Configuration / communication:	• Add, move, remove an I/O module • Changing memory sizes in configuration screen,
Global variables (used in animation table or operator screen)	Remove a used variable
Used DFB	• Type name of used DFB • Add a parameter

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	Ensure both Primary (A) and Standby (B) controllers are in Run Primary and Run Standby mode.
2	Download of the new application in the standby PLC (B) Results: ● The Standby PLC (B) goes to Non Configuration state ● At the end of the download, the PLC B goes in Stop/Offline mode
3	Stop on the PLC A Result: The PLC A goes in Stop Offline mode The system is neither more active nor redundant
4	Run on the PLC B Result: The PLC B goes in Run Primary mode The system is active again, but not redundant
4	Download of the new application in the PLC A Result: ● The PLC A goes to the Non Configuration state ● At the end of the download, PLC A goes in Stop/Offline mode
5	RUN command on the PLC A Result: The PLC A goes to the Run/Standby mode The system is active and redundant

⚠ WARNING

RISK OF UNINTENDED EQUIPMENT DAMAGE

The Offline method has more impact on the process than the Online method:

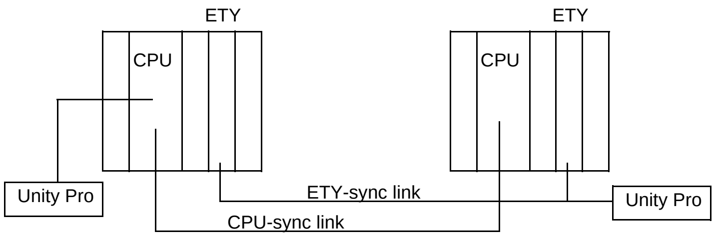
- There is no PLC active on the system during few seconds,
- When the PLC B restarts in Run / Primary mode, this is done on a data context that has been re initialized.

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

Online/Offline application of Modifications

For these two kinds of application modifications, Unity Pro can be connected to Ethernet or a local terminal port of one of the two PLCs (routing capabilities).

The following illustration displays the connection:



CAUTION

RISK OF UNINTENDED EQUIPMENT OPERATION

When executing an Offline modification, some changes of PLC state will generate a change of IP address. If Unity Pro is connected to Ethernet, the change of IP address will impact the connection with the PLC (PLC communication failed message).
After reconnecting Unity Pro to the PLC, be sure you have defined the right IP address by taking into account the Unity Pro Status bar and more especially the following information:

- The link Status (Offline, Different, Equal),
- The Hot Standby PLC Status (PLC name A/B, PLC state Offline/Standby/Primary),
- The address of the connected PLC.

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

Handling CPU OS Upgrade

10

Introduction

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's in this Chapter?

This chapter contains the following topics:

Overview of Premium Hot Standby OS Upgrade

Upgrading while Process is Running

The Executive Upgrade feature allows the Standby controller OS 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 fail before the Standby upgrade is complete.

Upgrading OS without Stopping

Under normal operating conditions, both controllers in a redundant system must have the same versions of firmware.

In fact, there are checks by the controllers to detect if there is a mismatch in firmware.

Normally, when a mismatch exists, performing a switchover would not be possible because the Standby controller would not be allowed to go online.

However, to allow an OS Upgrade without stopping the application, overriding is possible by setting the Command Register system bit %SW60.4.

Note: IMPORTANT INFORMATION
OS upgrade is possible only with compatible firmware.

⚠ CAUTION

RISK OF EQUIPMENT DAMAGE

Enabling OS upgrade without stopping the application overrides the process of checking whether the Primary and Standby are configured identically.
Disable the upgrade without stopping bit as soon as the OS upgrade is finished.

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

Executing the OS Upgrade Procedure

General

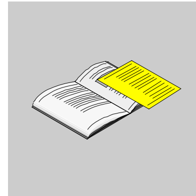
Perform an OS upgrade using the installed OSLoader tool.

How to perform an OS Upgrade

Follow these steps.

Step	Action
1	Connect Unity Pro to the Primary PLC through Uni-Telway terminal port
2	Access Command Register %SW60; set bit 4 to 1 (OS version mismatch allowed)
3	Stop the Primary. Ensure Standby becomes Primary
4	Disconnect Unity Pro
5	Open the OSLoader tool
6	Download the new OS
7	After completing the OS download, perform application program transfer
8	Put the PLC in RUN mode. Ensure PLC becomes Standby
9	Connect Unity Pro to the other PLC (that is the Primary) through Uni-Telway terminal port
10	Stop the Primary. Ensure Standby becomes Primary
11	Disconnect Unity Pro
12	Open the OSLoader tool
13	Download the new OS
14	After completing the OS download, perform application program transfer
15	Put the PLC in RUN mode. Ensure PLC becomes Standby
16	Perform a switchover (or connect Unity Pro to the Primary). Ensure Standby becomes Primary
17	Access Command Register %SW60; set bit 4 to 0 (OS version mismatch not allowed)

Appendices



Appendices for Premium Hot Standby

At a Glance

The appendices for the Premium Hot Standby are included here.

What's in this Appendix?

The appendix contains the following chapters:

Additional Information

A

Introduction

Overview

This chapter describes the design specifications and error codes.

What's in this Chapter?

This chapter contains the following topics:

CPUs TSX H57 24M/TSX H57 44M Specifications for Premium Hot Standby

Maximum configuration

The following table presents the maximum configuration of the CPUs:

Services	TSX H57 24M	TSX H57 44M
Local racks (12EX/4-6-8EX)	8/16	
Discrete I/Os channels	1024	2048
Analog I/Os channels	80	256
Experts modules (1)	0	
Ethernet modules	2	4
Other Networks modules (2)	0	
Open Field Bus modules (Interbus/Profibus)	0	
Sensor Bus modules (As-i)	0	
Process channels	10	20
Process loops	30	60

- 1): motion, weighing, counting, stepper.
- (2): Modbus+, Fipway. For Premium/Atrium this is the maximum number of channel supported.

Note: The Ethernet port for the CPU-sync link is a point to point connection dedicated to the Premium Hot Standby database exchange.

Program and Data Memory capacity

The following table presents the Programme 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 kilobytes	440 kilobytes
Maximum application size in PCMCIA	Program + Ets + symb. in PCMCIA	768 kilobytes	2048 kilobytes
	Max On line modif. area PLC	256 kilobytes	512 kilobytes
	Data in internal SRAM	192 kilobytes	440 kilobytes
Maximum data storage size (only in PCMCIA)	Legacy EFs	8 Mbytes	16 Mbytes
	DOS Files (SRAM)	Not available	Not available
Located data %MW	Max	32464	
	Default	1024	
	Min.	0	
Located data %M	Max	8056	32634
	Default	512	
	Min.	0	
Located data %KW	Max	32760	
	Default	256	
	Min.	0	
Located data %SW		168	
Located data %S		128	
Unlocated data max size (2): ● EDT + DDT ● EFB/DFB		No limit (3)	

- (1): Empty Terminal Support.
- (2)
  - EDT: Elementary Data Types (bool, integers, date, real).
  - DDT: Derived Data Types (structures).
  - EFB / DFB: Function Blocks
- (3): No limit means that the amount of Function Blocks is only dependant on the memory size.

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:

<table>
<tr><th colspan="2">Services</th><th>TSX H57 24M</th><th>TSX H57 44M</th></tr>
<tr><td colspan="2">Mast task</td><td colspan="2">1 cyclic/periodic</td></tr>
<tr><td colspan="2">Fast task</td><td colspan="2">1 periodic</td></tr>
<tr><td colspan="2">Auxiliary tasks</td><td colspan="2">0</td></tr>
<tr><td colspan="2">Event interrupt tasks (IO Event + Timer Event)</td><td colspan="2">64</td></tr>
<tr><td colspan="2">IO Events (Local IO)</td><td colspan="2">0 to 63
● Prior 0: %evt0
● Prior 1: %evt1 to %evt63</td></tr>
<tr><td colspan="2">Timer interrupt event</td><td colspan="2">0</td></tr>
<tr><td rowspan="3">Number of channels (Local I/O) per event</td><td>Discrete I/O</td><td colspan="2">128</td></tr>
<tr><td>Analog I/O</td><td colspan="2">16</td></tr>
<tr><td>Other</td><td colspan="2">16</td></tr>
</table>

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 (FBD)		Yes	
Ladder Logic		Yes	
Structured Text		Yes	
Instruction List		Yes	
SFC		Yes	
DFB		Yes	
EF/EFB		Yes	
PL7 SFB		Not recommended	
Embedded communication ports			
Legacy Terminal port	Physical layer	One RS 485	
	Speed	19200 baud	
	Protocol	Uni-Telway M/S ASCII	
USB terminal port		One device connector USB V1.0 12Mbytes	

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 cards Data storage	
Supported SRAM PCMCIA (Max application size according to PLC characteristics)	• TSX MRP P 128K • TSX MRP P 224K • TSX MRP P 384K • TSX MRP C 448K • TSX MRP C 768K • TSX MRP C 001M • TSX MRP C 01M7 • TSX MRP C 002M • TSX MRP C 003M • TSX MRP C 007M	
Supported FLASH PCMCIA (Max application size according to PLC characteristics)	• TSX MFP P 128K • TSX MFP P 224K • TSX MCP C 224K • TSX MFP P 384K • TSX MFP P 512K • TSX MCP C 512K • TSX MFP P 001M • TSX MFP P 002M • TSX MCP C 002M • TSX MFP P 004M	
Supported Data storage	• TSX MRP F 004M • TSX MRP F 008M	

OS Download/ Application Performances/ 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**
OS Download		
CPU OS download	Yes Uni-Telway terminal port	
HSBY OS download	Yes through Ethernet port only	
I/O modules OS download	No	
Application performances (PCMCIA)		
100% boolean	15.5 Kins/ms	
65% boolean,35% num.	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 6EX	
Default Power supply		TSX PSY 2600	
PCMCIA slots	Slot A	Type I / 5V	
	Slot B	Type III / 5V	
Real Time Clock		Yes	
RTC synchronization with dual CPU		No	
(1): max = typical consumption x 1.4			

TextIDs

TextIDs

TextIds define the warning messages written in the diagnostic buffer.

TextIDs switching from Primary to Offline

TextID	Warning message
13001	System halt
13002	Remote IO failure
13003	ETH device failure
13004	ETH communication problem
13005	Stop PLC command
13007	Offline Command register request

TextIDs switching from Standby to Offline

TextID	Warning message
13008	System halt
13009	Remote IO failure
13010	ETH device failure
13011	ETH communication problem
13012	Stop PLC command
13014	Offline Command register request

TextIDs switching from Standby to Primary

TextID	Warning message
13015	Control command over ETH
13016	Control command over RIO

TextIDs switching from Offline to Primary/Standby

TextID	Warning message
13017	Switch from Offline to Primary
13018	Switch from Offline to Standby BY

System Detailed Behavior upon Failures

B

Introduction

Overview

In this chapter you will find the failures that can occur in Premium Hot Standby system.

What's in this Chapter?

This chapter contains the following topics:

Overview of Failures

Introduction

A first level of Hot Standby diagnosis can be done through the status register that is managed locally by each Hot Standby PLC. The user can obtain more diagnostic information by managing PLC states, module bits in the first section of his application depending on the process requirements.

This diagnostic information can be stored in non transfer %MW area. To report this diagnostic information from the Standby to the Primary PLC it can be copied to the reverse transfer registers %SW62 - %SW65.

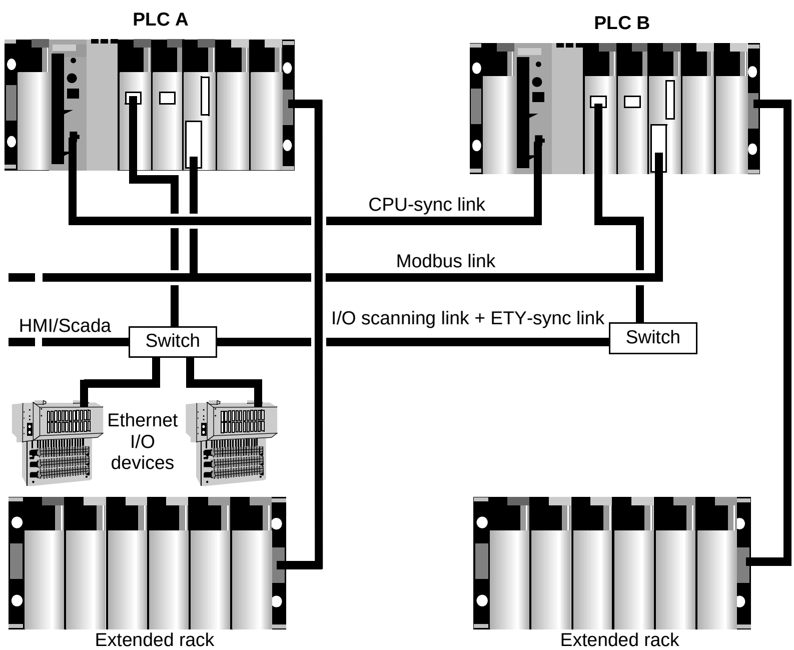
The following pages describe different cases of failures that can occur in a Hot Standby system with an example of configuration.

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 module (DIS IN and DIS OUT) in slot 5 and 6
- Only one switch (for simplified schema) to insure connection between Ethernet I/O scanner and SCADA or HMI.
- 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 Main rack state is:

- %MWx.0 / %SW62.0: reserved
- %MWx.1 / %SW62.1: reserved
- %MWx.2 / %SW62.2: reserved
- %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

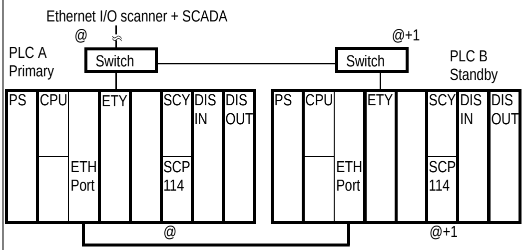
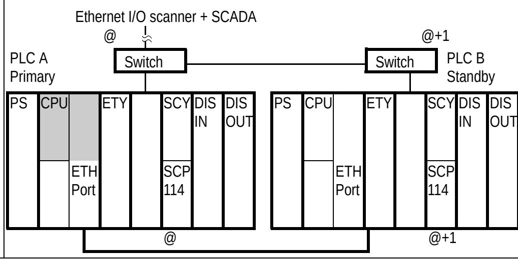
The Extended rack state is:

- %MWx.8 / %SW62.8: discrete module state by copy of %I1.0.mod.err
- %MWx.9 / %SW62.9: discrete module state by copy of %I1.1.mod.err
- %MWx.10 / %SW62.10: discrete module state by copy of %I1.2.mod.err
- %MWx.11 / %SW62.11: discrete module state by copy of %I1.3.mod.err
- %MWx.12 / %SW62.12: discrete module state by copy of %I1.4.mod.err
- %MWx.13 / %SW62.13: discrete module state by copy of %I1.5.mod.err
- %MWx.14 / %SW62.14: discrete module state by copy of %I1.6.mod.err

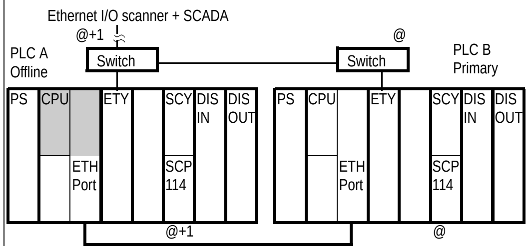
Halt or Stop Events on PLC

Halt or Stop on Primary PLC

The following table presents Halt or Stop events on Primary PLC:

Before the event	
In rack Discrete I/O state: • PLC A: calculated and applied at the end of the task cycle. • PLC B: PLC A output applied at the end of task cycle. Remote I/O state: • PLC A: all connections with ethernet devices are open: I/O scanner is active. • PLC B: all connections with ethernet devices are closed: I/O scanner is not active.	
Event	
• HALT instruction • Watch dog overflow • Program execution error (division by 0, overflow, etc.) with %S78 = 1 • STOP command This is a critical event because an automatic switch over occurs.	

<table>
<tr><td colspan="3">After the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
● PLC A: fallback position
● PLC B: calculated and applied at the end of the task cycle

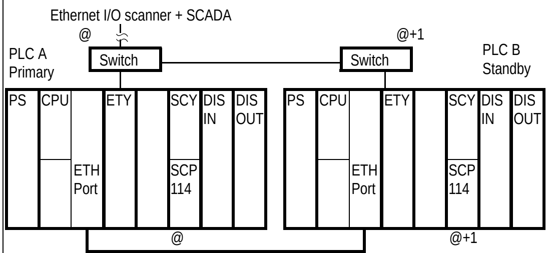
Remote I/O state:
● PLC A: all connections with Ethernet devices are closed: I/O scanner is not active
● PLC B: all connections with Ethernet devices are open: I/O scanner is active</td><td>
</td></tr>
<tr><td>Global status</td><td>Communication status</td><td>Customer diagnostic through Ethernet address @</td></tr>
<tr><td>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.</td><td>Both PLCs are accessible through terminal ports, Modbus and Ethernet links for diagnostics</td><td>%SW61 = 1000 0000 0010 0110
● the accessed PLC is PLC B / primary
● the other PLC is PLC A / offline

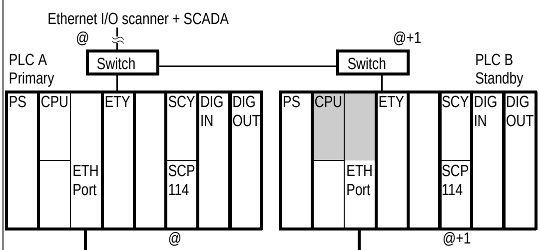
%SW62 = Not significant because one of the two PLC is Offline or Not Responding</td></tr>
</table>

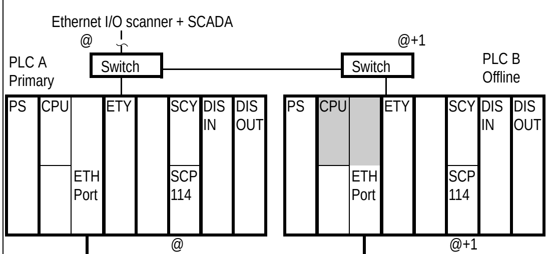
Halt or Stop on Standby PLC

The following table presents Halt or Stop events on Standby PLC:

<table>
<tr><td colspan="3">Before the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
● PLC A: calculated and applied at the end of the task cycle.
● PLC B: PLC A output applied at the end of task cycle.

Remote I/O state:
● PLC A: all connections with ethernet devices are open: I/O scanner is active.
● PLC B: all connections with ethernet devices are closed: I/O scanner is not active.</td><td>
</td></tr>
<tr><td colspan="3">Event</td></tr>
<tr><td colspan="2">● HALT instruction
● Watch dog overflow
● Program execution error (division by 0, overflow, etc.) with %S78 = 1
● STOP command

This is not a critical event because there is not switch over.</td><td>
</td></tr>
<tr><td colspan="3">After the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
● PLC A: calculated and applied at the end of the task cycle
● PLC B: fallback position

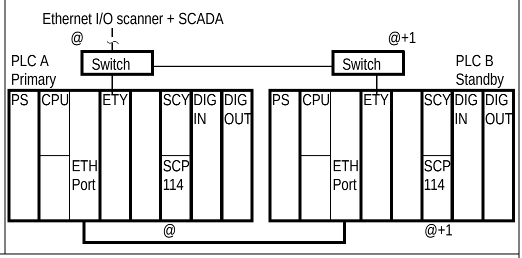
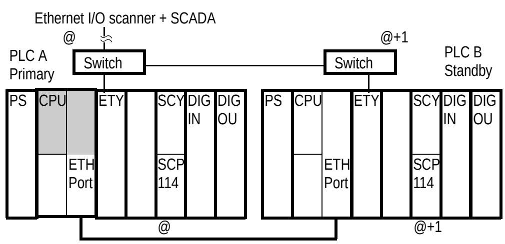
Remote I/O state:
● PLC A: all connections with Ethernet devices are open: I/O scanner is active
● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>
</td></tr>
<tr><td>Global status</td><td>Communication status</td><td>Customer diagnostic through Ethernet address @</td></tr>
<tr><td>The process is still active but the HSBY system is no longer redundant as long as the PLC B is STOP mode.</td><td>Both PLCs are accessible through terminal ports, Modbus links and Ethernet links for diagnostic</td><td>%SW61 = 1000 0000 0000 0110
● the accessed PLC is PLC A: primary
● the other PLC is PLC B: offline

%SW62 = Not significant because one of the two PLC is Offline</td></tr>
</table>

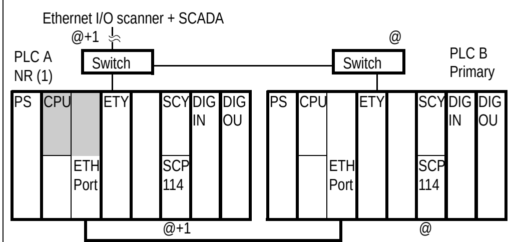
Hardware or Firmware CPU Failure

CPU Failure on Primary

The following table presents CPU failure on Primary:

Before the event	
In rack Discrete I/O state: ● PLC A: calculated and applied at the end of the task cycle ● PLC B: PLC A output applied at the end of task cycle Remote I/O state: ● PLC A: all connections with Ethernet devices are open: I/O scanner is active ● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active	
Event	
Hardware or firmware failure on the processor. This is a critical event because an automatic switch over occurs.	

<table>
<tr><td colspan="3">After the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
• PLC A: fallback position
• PLC B: calculated and applied at the end of the task cycle

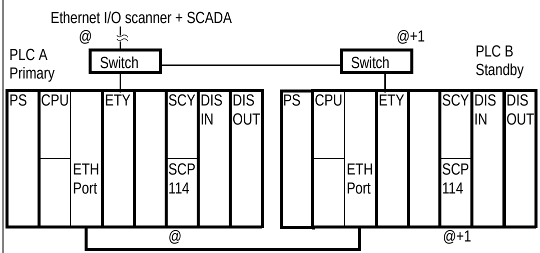
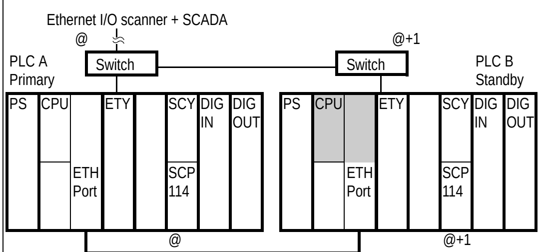
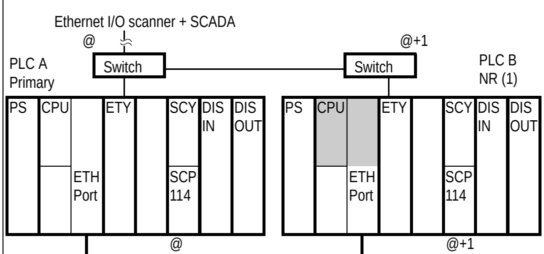
Remote I/O state:
• PLC A: all connections with Ethernet devices are closed: I/O scanner is not active
• PLC B: all connections with Ethernet devices are open: I/O scanner is active</td><td>

(1) NR: Not responding</td></tr>
<tr><td>Global status</td><td>Communication status</td><td>Customer diagnostic through Ethernet address @</td></tr>
<tr><td>The process is still active but the HSBY system is no longer redundant as long as the PLC A is in ERROR mode.</td><td>• No access to PLC A (CPU no longer running)
• Normal access to PLC B accessible through terminal port, Modbus and Ethernet links for diagnostics</td><td>%SW61 = 1000 0000 0110 0010
• The accessed PLC is PLC B / primary
• The other PLC is PLC A / undefined

%SW62 = Not significant because one of the two PLC is Not Responding.</td></tr>
</table>

CPU Failure on Standby

The following table presents CPU failure on Standby PLC:

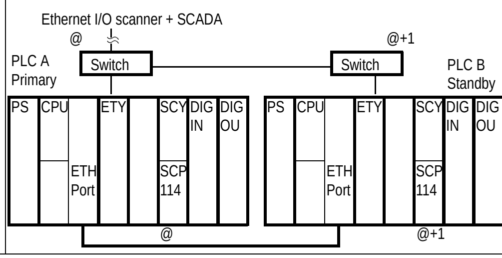
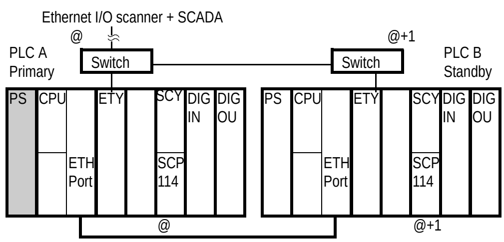
<table>
<tr><td colspan="3">Before the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
• PLC A: calculated and applied at the end of the task cycle
• PLC B: PLC A output applied at the end of task cycle
Remote I/O state:
• PLC A: all connections with Ethernet devices are open: I/O scanner is active
• PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>
</td></tr>
<tr><td colspan="3">Event</td></tr>
<tr><td colspan="2">Hardware or firmware failure on the processor.
This is not a critical event because there is no switchover.</td><td>
</td></tr>
<tr><td colspan="3">After the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
• PLC A: calculated and applied at the end of the task cycle
• PLC B: fallback position
Remote I/O state:
• PLC A: all connections with Ethernet devices are open: I/O scanner is active
• PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>

(1) NR: Not responding</td></tr>
<tr><td>Global status</td><td>Communication status</td><td>Customer diagnostic through Ethernet address @</td></tr>
<tr><td>The process is still active but the HSBY system is no longer redundant as long as the PLC B is in ERROR mode.</td><td>• Normal access to PLC A through terminal port
• No access to PLC B (CPU no longer running)</td><td>%SW61 = 1000 0000 0100 0010
• The accessed PLC is PLC A: primary
• The other PLC is PLC B: undefined
%SW62 = Not significant because one of the two PLC is Not Responding.</td></tr>
</table>

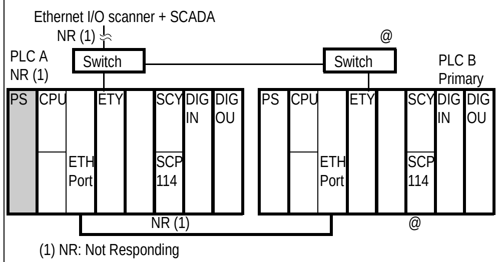
Power Failure on the Main Rack

Power Failure on Primary Main Rack

The following table presents power failure on the main rack of the Primary PLC:

Before the event	
In rack Discrete I/O state: • PLC A: calculated and applied at the end of the task cycle • PLC B: PLC A output applied at the end of task cycle Remote I/O state: • PLC A: all connections with Ethernet devices are open: I/O scanner is active • PLC B: all connections with Ethernet devices are closed: I/O scanner is not active	
Event	
Power failure on the Primary main rack. This is a critical event because an automatic switch over occurs.	

<table>
<tr><td colspan="3">After the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
• PLC A: I/O powered off
• PLC B: calculated and applied at the end of the task cycle

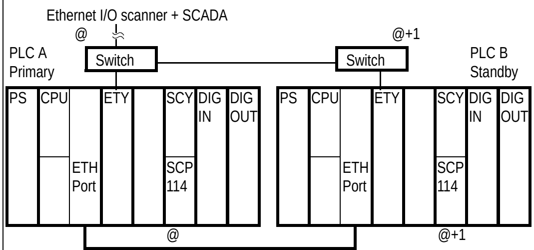
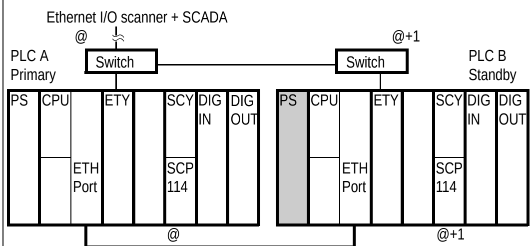
Remote I/O state:
• PLC A: ETY powered off
• PLC B: all connections with Ethernet devices are open: I/O scanner is active</td><td>
</td></tr>
<tr><td>Global status</td><td>Communication status</td><td>Customer diagnostic through Ethernet address @</td></tr>
<tr><td>The process is still active but the HSBY system is no longer redundant as long as the PLC A is powered off.</td><td>• No access to PLC A (CPU system no longer running)
• Normal access to PLC B accessible through terminal port, Modbus and Ethernet links for diagnostics</td><td>%SW61 = 1000 0000 0110 0010
• The accessed PLC is PLC B / primary
• The other PLC is PLC A / undefined

%SW62 = Not significant because one of the two PLC is Not Responding</td></tr>
</table>

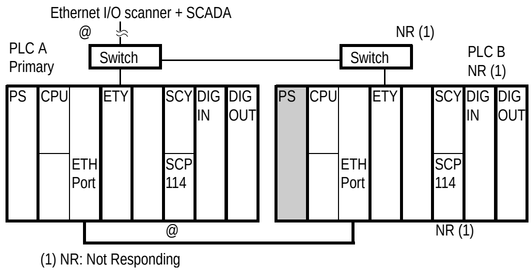
Power Failure on Standby Main Rack

The following table presents power failure on the main rack of the Standby PLC:

<table>
<tr><td colspan="2">Before the event</td></tr>
<tr><td>In rack Discrete I/O state:
• PLC A: calculated and applied at the end of the task cycle
• PLC B: PLC A output applied at the end of task cycle

Remote I/O state:
• PLC A: all connections with Ethernet devices are open: I/O scanner is active
• PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>
</td></tr>
<tr><td colspan="2">Event</td></tr>
<tr><td>Power failure on the Standby main rack.
This is not a critical event because there is no switch over.</td><td>
</td></tr>
</table>

<table>
<tr><td colspan="3">After the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
● PLC A: calculated and applied at the end of the task cycle
● PLC B: I/O powered off

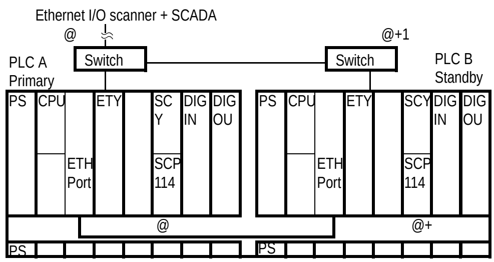
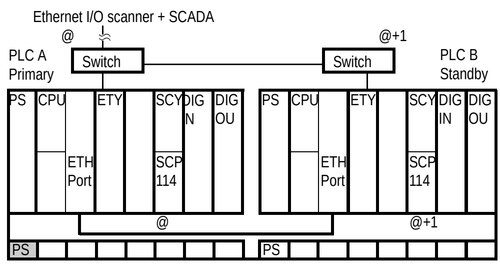
Remote I/O state:
● PLC A: all connections with Ethernet devices are open: I/O scanner is active
● PLC B: ETY powered off</td><td>
</td></tr>
<tr><td>Global status</td><td>Communication status</td><td>Customer diagnostic through Ethernet address @</td></tr>
<tr><td>The process is still active but the HSBY system is no longer redundant as long as the PLC B is powered off.</td><td>● Normal access to PLC A through terminal port, Modbus link and Ethernet link for diagnostics
● No access to PLC B (CPU system is no longer running)</td><td>%SW61 = 1000 0000 0100 0010
● The accessed PLC is PLC A: primary
● The other PLC is PLC B: undefined

%SW62 = Not significant because one of the two PLC is Not Responding</td></tr>
</table>

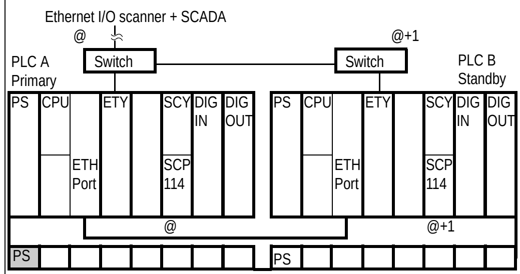
Power Failure on an Extendable Rack

Power Failure on Primary Extendable Rack

The following table presents power failure on an extendable rack of the Primary PLC:

<table>
<tr><td colspan="2">Before the event</td></tr>
<tr><td>In rack Discrete I/O state:
● PLC A: calculated and applied at end of task cycle
● PLC B: PLC A output applied at end of task cycle
Remote I/O state:
● PLC A: all connections with Ethernet devices are open: I/O scanner is active
● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>
</td></tr>
<tr><td colspan="2">Event</td></tr>
<tr><td>Power failure on an extendable rack.
The status of the Hot Standby system does not change.</td><td>
</td></tr>
</table>

<table>
<tr><td colspan="3">After the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
● PLC A main rack: processed normally
● PLC A ext rack: powered off
● PLC B: PLC A output applied

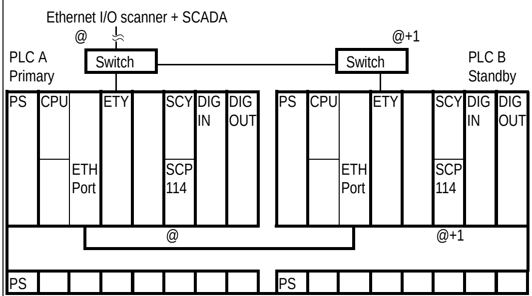
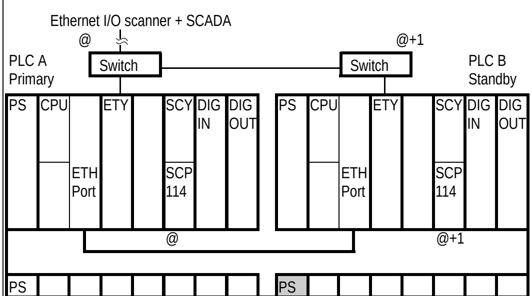
Remote I/O state:
● PLC A: all connections with Ethernet devices are open: I/O scanner is active
● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>
</td></tr>
<tr><td>Global status</td><td>Communication status</td><td>Customer diagnostic through Ethernet address @</td></tr>
<tr><td>The process is still active but with some Discrete and Analog I/Os that are not processed. If needed the customer can request a switchover by setting a bit in the primary application command register (if there is no fault in the other PLC).</td><td>Both PLCs are accessible through terminal ports, Modbus links and Ethernet links for diagnostics.</td><td>%SW61 = 1000 0000 0000 1110
● The accessed PLC is PLC A / primary
● The other PLC is PLC B / standby

%SW62 = 0000 0000 0000 0000:
● The other PLC: no fault

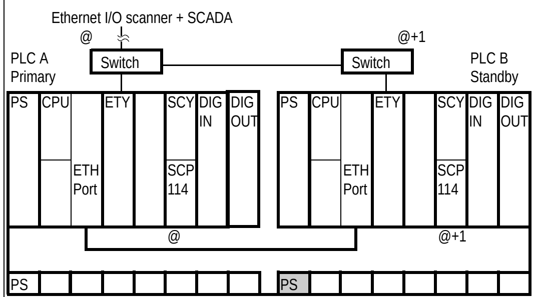
Primary PLC: error bit (%I1.x.mod.err) of all the modules in the extended rack set to 1</td></tr>
</table>

Power Failure on Standby Extendable Rack

The following table presents power failure on an extendable rack of the Standby PLC:

Before the event	
In rack Discrete I/O state: • PLC A: calculated and applied at the end of the task cycle • PLC B: PLC A output applied at the end of task cycle Remote I/O state: • PLC A: all connections with Ethernet devices are open: I/O scanner is active • PLC B: all connections with Ethernet devices are closed: I/O scanner is not active	
Event	
Power failure on an extendable rack. The status of the Hot Standby system does not change.	

<table>
<tr><td colspan="3">After the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
● PLC A: processed normally
● PLC B: PLC A output applied at the end of the task cycle

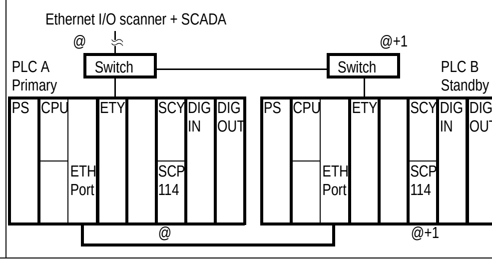
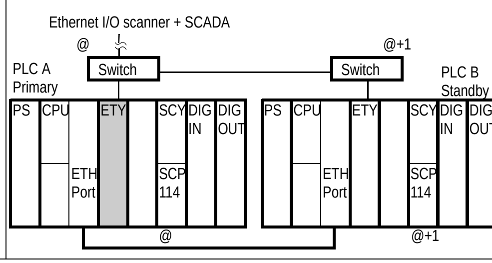
Remote I/O state:
● PLC A: all connections with Ethernet devices are open: I/O scanner is active
● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>
</td></tr>
<tr><td>Global status</td><td>Communication status</td><td>Customer diagnostic through Ethernet address @</td></tr>
<tr><td>The process is still active and the HSBY system is still redundant.In case of switchover PLC B will become Primary with some Discrete / Analog I/Os in failed mode.</td><td>Both PLCs are accessible through terminal ports, Modbus links and Ethernet links for diagnostics.</td><td>%SW61 = 1000 0000 0000 1110
● The accessed PLC is PLC A / primary
● The other PLC is PLC B / standby

%SW62 = 0111 1111 0000 0000:
● The other PLC: all discrete modules of extended rack in fault</td></tr>
</table>

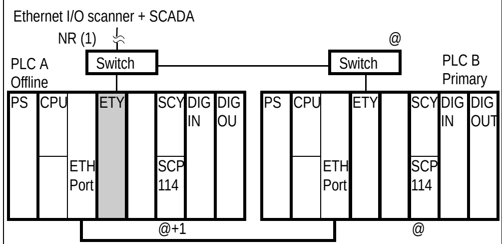
Hardware or Firmware ETY failure

ETY Failure on Primary

The following table presents ETY failure (hardware or firmware) on the Primary PLC:

<table>
<tr><td colspan="2">Before the event</td></tr>
<tr><td>In rack Discrete I/O state:
• PLC A: calculated and applied at the end of the task cycle
• PLC B: PLC A output applied at the end of task cycle
Remote I/O state:
• PLC A: all connections with Ethernet devices are open: I/O scanner is active
• PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td></td></tr>
<tr><td colspan="2">Event</td></tr>
<tr><td>Hardware or firmware failure on the Monitored ETY module that manages Ethernet I/O (or Ethernet I/O + SCADA / HMI).
This is a critical event because an automatic switch over occurs.</td><td></td></tr>
</table>

<table>
<tr><td colspan="3">After the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
● PLC A: fallback position
● PLC B: calculated and applied at the end of the task cycle

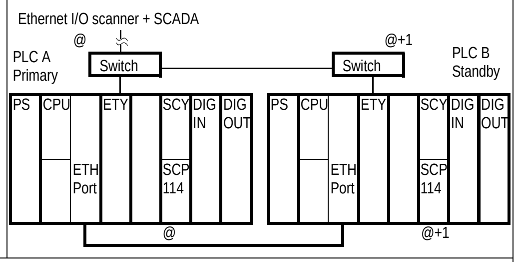
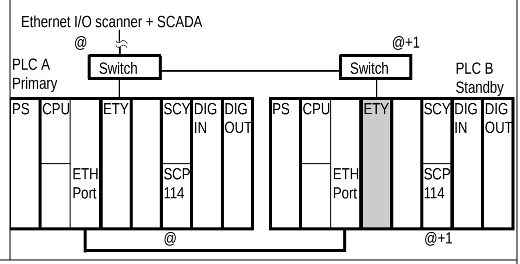
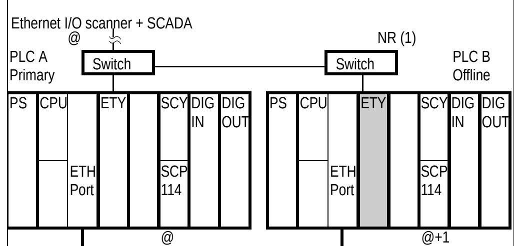
Remote I/O state:
● PLC A: all connections with Ethernet devices are closed: I/O scanner is not active
● PLC B: all connections with Ethernet devices are open: I/O scanner is active</td><td>

(1) NR: Not responding</td></tr>
<tr><td>Global status</td><td>Communication status</td><td>Customer diagnostic through Ethernet address @</td></tr>
<tr><td>The process is still active but the system is no longer redundant as long as the ETY module of the PLC A is in failed mode.</td><td>● Normal access to PLC A through terminal port or Modbus
● No access to PLC A through Ethernet link
● Normal access to PLC B through terminal port, Modbus or Ethernet links for diagnostics</td><td>%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 PLC is Offline.</td></tr>
</table>

ETY Failure on Standby

The following table presents ETY failure (hardware or firmware) on the Standby PLC:

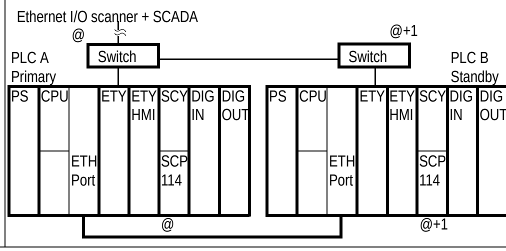
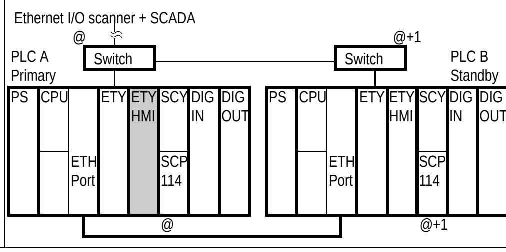
<table>
<tr><td colspan="3">Before the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
• PLC A: calculated and applied at the end of the task cycle
• PLC B: PLC A output applied at the end of task cycle
Remote I/O state:
• PLC A: all connections with Ethernet devices are open: I/O scanner is active
• PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>
</td></tr>
<tr><td colspan="3">Event</td></tr>
<tr><td colspan="2">Hardware or firmware failure on the Monitored ETY module that can manage Ethernet I/O (or Ethernet I/O + SCADA / HMI).
This is not a critical event because there is no switch over.</td><td>
</td></tr>
<tr><td colspan="3">After the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
• PLC A: calculated and applied at the end of the task cycle
• PLC B: fallback position
Remote I/O state:
• PLC A: all connections with Ethernet devices are open: I/O scanner is active
• PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>

(1) NR: Not Responding</td></tr>
<tr><td>Global status</td><td>Communication status</td><td>Customer diagnostic through Ethernet address @</td></tr>
<tr><td>The process is still active but the system is no longer redundant as long as the PLC B is in failed mode.</td><td>• Normal access to PLC A through terminal port or Modbus or ethernet link for diagnostics
• Normal access to PLC B through terminal port or Modbus
• No access to PLC B through Ethernet link</td><td>%SW61 = 1000 0000 0000 0110
• The accessed PLC is PLC A: primary
• The other PLC is PLC B: offline
%SW62 = Not significant because one of the two PLC is Offline.</td></tr>
</table>

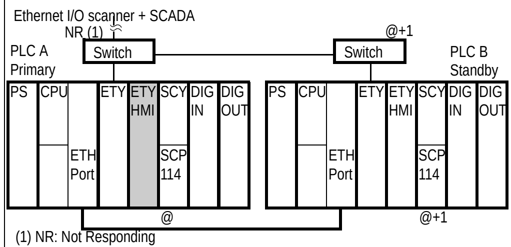
Hardware or Firmware Failure on ETY Dedicated to HMI and SCADA

Failure on Primary ETY (HMI & SCADA)

The following table presents failure (hardware or firmware) on the Primary ETY dedicated to HMI and SCADA (ETY is not the HSBY Monitored ETY):

Before the event	
In rack Discrete I/O state: • PLC A: calculated and applied at end of task cycle • PLC B: PLC A output applied at end of task cycle Remote I/O state: • PLC A: all connections with Ethernet devices are open: I/O scanner is active • PLC B: all connections with Ethernet devices are closed: I/O scanner is not active	
Event	
Hardware or firmware failure on the ETY module that manages SCADA / HMI. This is a not a critical event because there is no automatic switchover.	

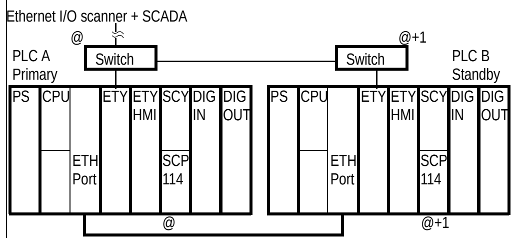
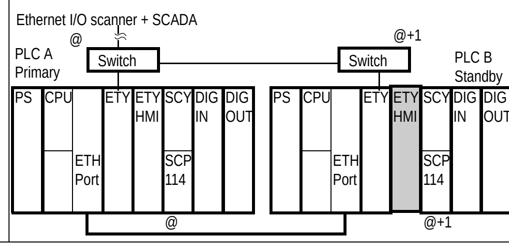
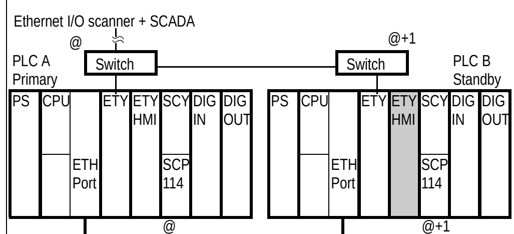
<table>
<tr><td colspan="3">After the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
● PLC A: calculated and applied at end of task cycle
● PLC B: PLC A output applied at end of task cycle

Remote I/O state:
● PLC A: all connections with Ethernet devices are open: I/O scanner is active
● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>
</td></tr>
<tr><td>Global status</td><td>Communication status</td><td>Customer diagnostic through Ethernet address @</td></tr>
<tr><td>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 there is no fault in the other PLC).</td><td>● Normal access to PLC A through terminal port or Modbus for diagnostics
● No access to PLC A through Ethernet link
● Normal access to PLC B through terminal port, Modbus or Ethernet links for diagnostics</td><td>%SW61 = 1000 0000 0000 1110
● The accessed PLC is PLC A / primary
● The other PLC is PLC B / standby

%SW62 = 0000 000 0000 0000
● The other PLC: no fault</td></tr>
</table>

Failure on Standby ETY (HMI & SCADA)

The following table presents failure (hardware or firmware) on the Standby ETY dedicated to HMI and SCADA (ETY is not the HSBY Monitored ETY):

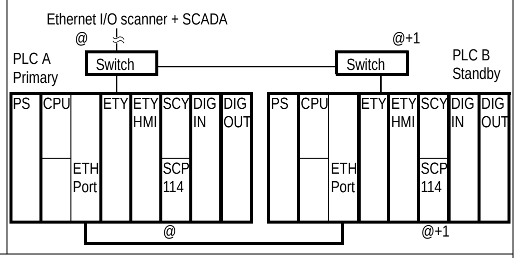
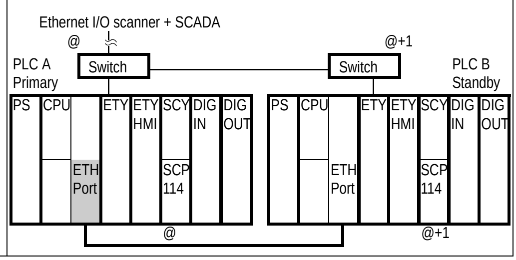
Before the event	
In rack Discrete I/O state: • PLC A: calculated and applied at the end of the task cycle • PLC B: PLC A output applied at the end of task cycle Remote I/O state: • PLC A: all connections with Ethernet devices are open: I/O scanner is active • PLC B: all connections with Ethernet devices are closed: I/O scanner is not active	
Event	
Hardware or firmware failure on the ETY module that manages SCADA / HMI. This is not a critical event because there is no switchover.	
After the event	
In rack Discrete I/O state: • PLC A: calculated and applied at end of task cycle • PLC B: PLC A output applied at end of task cycle Remote I/O state: • PLC A: all connections with Ethernet devices are open: I/O scanner is active • PLC B: all connections with Ethernet devices are closed: I/O scanner is not active	

Global status	Communication status	Customer diagnostic through Ethernet address @
No impact on the Hot Standby running. The process is still redundant.	• Normal access to PLC A through terminal port or Modbus or Ethernet link for diagnostics • Normal access to PLC B through terminal port or Modbus • No access to PLC B through Ethernet link	%SW61 = 1000 0000 0000 1110 • the accessed PLC is PLC A: primary • the other PLC is PLC B: standby %SW62 = 0000 0000 1000 0000

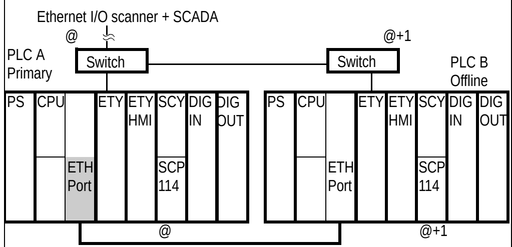
Failure on the Ethernet Copro

Failure on Primary Ethernet Copro

The following table presents failure (hardware or firmware) on the Ethernet Copro of the Primary PLC:

<table>
<tr><th colspan="2">Before the event</th></tr>
<tr><td>In rack Discrete I/O state:
• PLC A: calculated and applied at end of task cycle
• PLC B: PLC A output applied at end of task cycle
Remote I/O state:
• PLC A: all connections with Ethernet devices are open: I/O scanner is active
• PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>
</td></tr>
<tr><td colspan="2">Event</td></tr>
<tr><td>Hardware or firmware failure on the ETY copro that manages the Hot Standby CPU-sync link. There is no more database exchange from Primary to Standby.
This is a not a critical event because there is no automatic switch over.</td><td>
</td></tr>
</table>

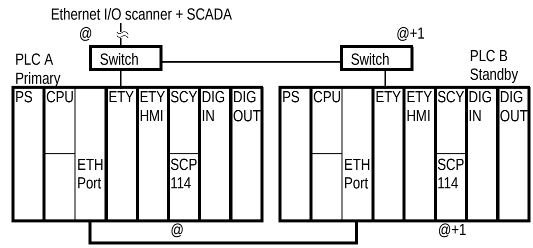
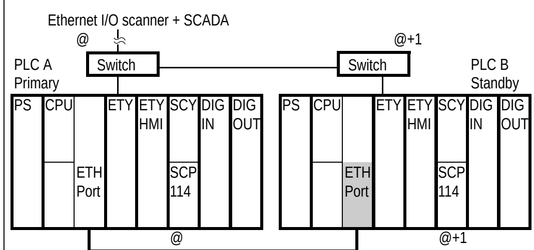
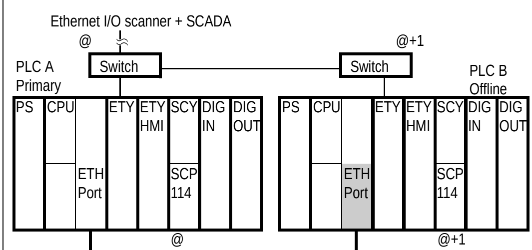
<table>
<tr><td colspan="3">Before the event</td></tr>
<tr><td colspan="3">After the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
• PLC A: calculated and applied at end of the task cycle
• PLC B: fallback position

Remote I/O state:
• PLC A: all connections with Ethernet devices are open: I/O scanner is active
• PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>
</td></tr>
<tr><td>Global status</td><td>Communication status</td><td>Customer diagnostic through Ethernet address @</td></tr>
<tr><td>The process is still active but the system is no longer redundant as long as the Ethernet copro of PLC A is in failed mode.</td><td>Both PLCs are accessible through terminal ports, Modbus links and Ethernet links for diagnostics.</td><td>%SW61 = 0000 0000 0100 0110
• The CPU-sync link is NOK
• The accessed PLC is PLC A / Primary
• The other PLC is PLC B / undefined

%SW62 = Not significant because one of the two PLC is undefined.</td></tr>
</table>

Failure on Standby Ethernet Copro

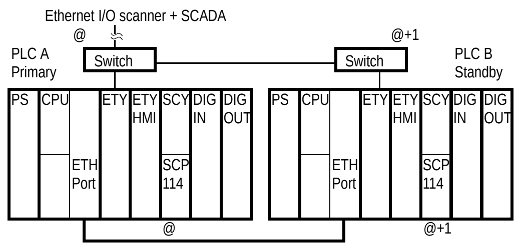
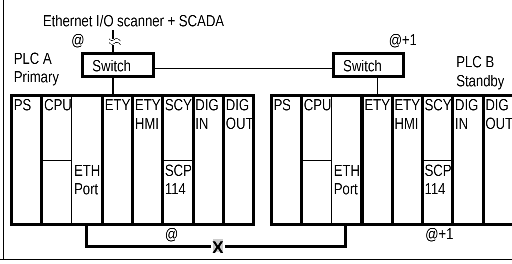
The following table presents failure (hardware or firmware) on the Ethernet Copro of the Standby PLC:

<table>
<tr><td colspan="3">Before the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
● PLC A: calculated and applied at end of task cycle
● PLC B: PLC A output applied at end of task cycle
Remote I/O state:
● PLC A: all connections with Ethernet devices are open: I/O scanner is active
● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>
</td></tr>
<tr><td colspan="3">Event</td></tr>
<tr><td colspan="2">Hardware or firmware failure on the ETY copro that manages the Hot Standby CPU-sync link. There is no more database exchange from Primary to Standby.
This is not a critical event because there is no switch over.</td><td>
</td></tr>
<tr><td colspan="3">After the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
● PLC A: calculated and applied at the end of the task cycle
● PLC B: Fall back position
Remote I/O state:
● PLC A: all connections with Ethernet devices are open: I/O scanner is active
● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>
</td></tr>
<tr><td>Global status</td><td>Communication status</td><td>Customer diagnostic through Ethernet address @</td></tr>
<tr><td>The process is still active but the system is no longer redundant as long as the Ethernet copro of PLC B is in failed mode.</td><td>Both PLCs are accessible through terminal ports, Modbus links and Ethernet links for diagnostics.</td><td>%SW61 = 1000 0000 0100 0010
● The accessed PLC is PLC A / primary
● The other PLC is PLC B / undefined
%SW62 = Not significant because one of the two PLC is undefined.</td></tr>
</table>

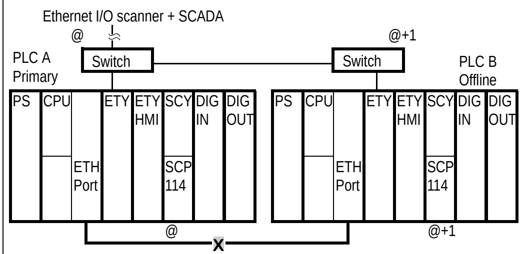
CPU-sync link failure between Primary and Standby PLCs

Primary and Standby CPU-sync Failure

The following table presents CPU-sync link failure between the Primary and Standby PLCs:

<table>
<tr><td colspan="2">Before the event</td></tr>
<tr><td>In rack Discrete I/O state:
PLC A: calculated and applied at end of task cycle
PLC B: PLC A output applied at the end of task cycle
Remote I/O state:
PLC A: all connections with Ethernet devices are open: I/O scanner is active
PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>
</td></tr>
<tr><td colspan="2">Event</td></tr>
<tr><td>CPU-sync link disconnection. There is no more database exchange from Primary to Standby.
This is a not a critical event because there is no automatic switchover.</td><td>
</td></tr>
</table>

<table>
<tr><td colspan="3">After the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
● PLC A: calculated and applied at the end of the task cycle
● PLC B: Fallback position

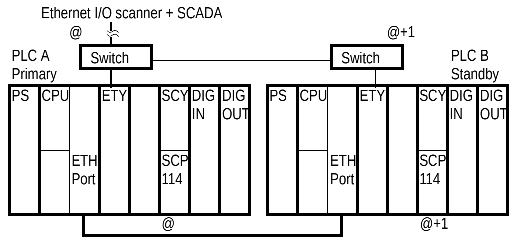
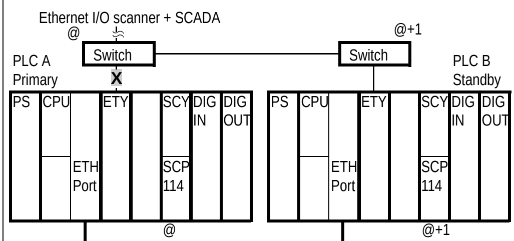
Remote I/O state:
● PLC A: all connections with Ethernet devices are open: I/O scanner is active
● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>
</td></tr>
<tr><td>Global status</td><td>Communication status</td><td>Customer diagnostic through Ethernet address @</td></tr>
<tr><td>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.</td><td>Both PLCs are accessible through terminal ports, Modbus links and Ethernet links for diagnostics.</td><td>%SW61 = 1000 0000 0100 0010
● The accessed PLC is PLC A / primary
● The other PLC is PLC B / undefined
● CPU-sync link not OK

%SW62 = Not significant because one of the two PLC is undefined.</td></tr>
</table>

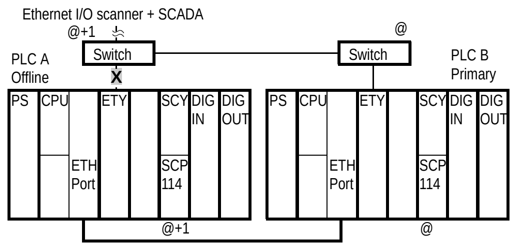
Monitored ETY and I/O Scanner Disconnection

Monitored ETY Disconnection on Primary

The following table presents Monitored ETY Disconnection on the Primary PLC side (the Monitored ETY is managing an I/O Scanner):

Before the event	
In rack Discrete I/O state: • PLC A: calculated and applied at end of task cycle • PLC B: PLC A output applied at end of task cycle Remote I/O state: • PLC A: all connections with Ethernet devices are open: I/O scanner is active • PLC B: all connections with Ethernet devices are closed: I/O scanner is not active	
Event	
Ethernet I/O link disconnection on the Primary side. There is no more diagnostic dialog between the 2 ETY modules. This is a critical event because there is an automatic switch over.	

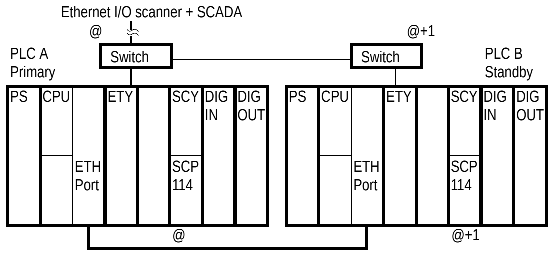
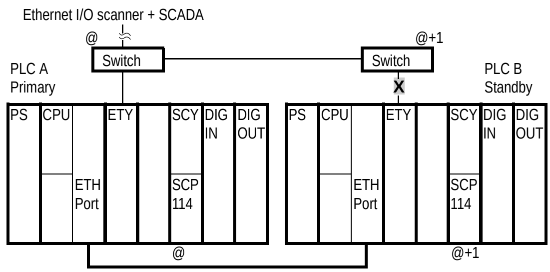
<table>
<tr><td colspan="3">After the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
● PLC A: Fallback position
● PLC B: Calculated and applied at end of task cycle

Remote I/O state:
● PLC A: all connections with Ethernet devices are closed: I/O scanner is not active
● PLC B: all connections with Ethernet devices are open: I/O scanner is active</td><td>
</td></tr>
<tr><td>Global status</td><td>Communication status</td><td>Customer diagnostic through Ethernet address @</td></tr>
<tr><td>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.</td><td>● Normal access to PLC A through terminal port and Modbus link for diagnostics. If an HMI/SCADA is connected to the switch, diagnosis is no longer possible through Ethernet
● Normal access to PLC B through terminal port, Modbus link and Ethernet link for diagnostics</td><td>%SW61 = 1000 0000 0010 0110
● The accessed PLC is PLC B / primary
● The other PLC is PLC A / offline

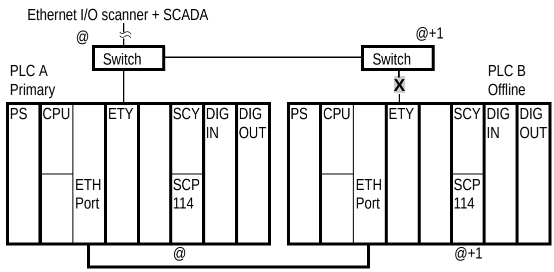
%SW62 = Not significant because one of the two PLC is Offline.</td></tr>
</table>

Monitored ETY Disconnection on Standby

The following table presents Monitored ETY Disconnection on the Standby PLC side (the Monitored ETY is managing an I/O Scanner):

Before the event	
In rack Discrete I/O state: • PLC A: calculated and applied at the end of the task cycle • PLC B: PLC A output applied at the end of task cycle Remote I/O state: • PLC A: all connections with Ethernet devices are open: I/O scanner is active • PLC B: all connections with Ethernet devices are closed: I/O scanner is not active	
Event	
Ethernet I/O link disconnection on the Standby side. There is no more diagnostic dialog between the two ETY modules. This is not a critical event because there is no switch over.	

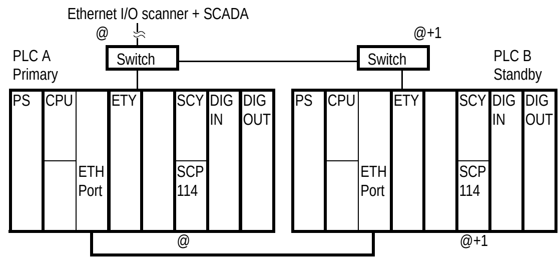
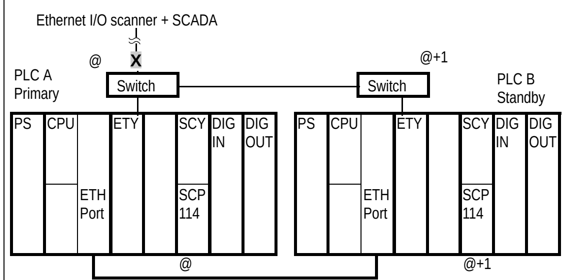
<table>
<tr><td colspan="3">After the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
● PLC A: calculated and applied at the end of the task cycle
● PLC B: Fall back position

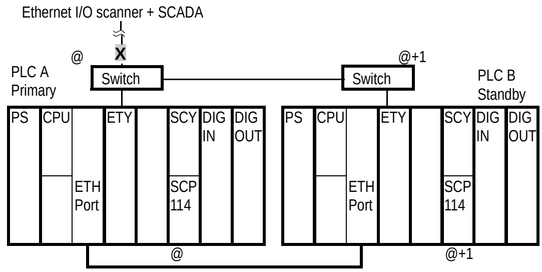
Remote I/O state:
● PLC A: all connections with Ethernet devices are open: I/O scanner is active
● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>
</td></tr>
<tr><td>Global status</td><td>Communication status</td><td>Customer diagnostic through Ethernet address @</td></tr>
<tr><td>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.</td><td>● Normal access to PLC A through terminal port and Modbus link for diagnostics.
● Normal access to PLC B through terminal port and Modbus link and Ethernet link for diagnostics. If an HMI/ SCADA is connected to the switch, diagnosis is no longer possible through Ethernet</td><td>%SW61 = 1000 0000 0000 0110
● The accessed PLC is PLC A / primary
● The other PLC is PLC B / offline

%SW62 = Not significant because one of the two PLC is Offline.</td></tr>
</table>

I/O Scanner Disconnection on I/O link

The following table presents I/O Scanner Disconnection on the I/O link side (the Monitored ETY is managing an I/O Scanner):

Before the event	
In rack Discrete I/O state: • PLC A: calculated and applied at the end of the task cycle • PLC B: PLC A output applied at the end of task cycle Remote I/O state: • PLC A: all connections with Ethernet devices are open: I/O scanner is active • PLC B: all connections with Ethernet devices are closed: I/O scanner is not active	
Event	
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. This is not a critical event because there is no switch over.	

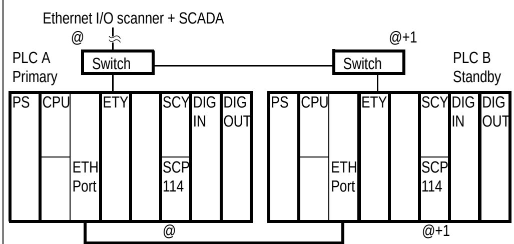
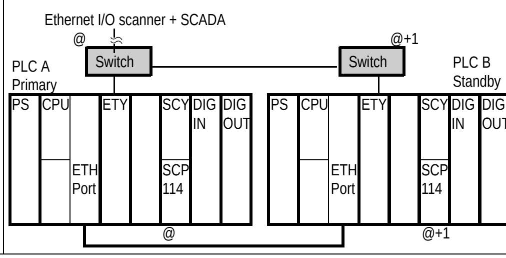
<table>
<tr><td colspan="3">After the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
● PLC A: calculated and applied at the end of the task cycle
● PLC B: PLC A output applied at the end of the task cycle
Remote I/O state:
● PLC A: all connections with Ethernet devices are open: I/O scanner is active
● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>
</td></tr>
<tr><td>Global status</td><td>Communication status</td><td>Customer diagnostic through Ethernet address @</td></tr>
<tr><td>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.</td><td>The 2 PLCs are accessible through terminal ports, Modbus links and Ethernet links for diagnostics.</td><td>%SW61 = 1000 0000 0000 0110
● The accessed PLC is PLC A / primary
● The other PLC is PLC B / standby
%SW62 = 0000 0000 0000 0000
● The other PLC: no fault</td></tr>
</table>

Note: This kind of failure has to be managed by the application program. It is equivalent to all the Ethernet I/O devices that are disconnected.

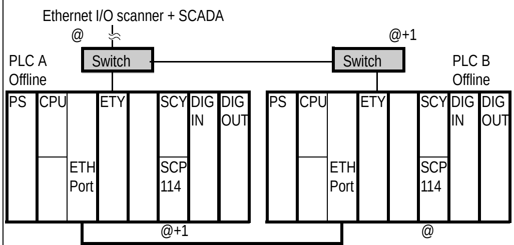
Full Ethernet I/O Link Disconnection

Full Ethernet I/O Link Disconnection

The following table presents Full Ethernet I/O Link Disconnection (for example a failure in both switches):

Before the event	
In rack Discrete I/O state: • PLC A: calculated and applied at nd of task cycle • PLC B: PLC A output applied at end of task cycle Remote I/O state: • PLC A: all connections with Ethernet devices are open: I/O scanner is active • PLC B: all connections with Ethernet devices are closed: I/O scanner is not active	
Event	
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. This is a critical event because there is no PLC active on the process.	

<table>
<tr><td colspan="3">After the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
• PLC A: Fallback position
• PLC B: Fallback position

Remote I/O state:
• PLC A: all connections with Ethernet devices are closed: I/O scanner is not active
• PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>
</td></tr>
<tr><td>Global status</td><td>Communication status</td><td>Customer diagnostic through Ethernet address @</td></tr>
<tr><td>The process is no longer active and the Hot Standby system is no longer redundant as long as the switch remains failed.</td><td>The 2 PLCs are accessible through terminal ports and Modbus links for diagnostics. If an HMI/SCADA is connected to the failed switch, diagnosis is no longer possible through Ethernet.</td><td>%SW61 = 1000 0000 0000 0101
• The accessed PLC is PLC A / offline
• The other PLC is PLC B / offline

%SW62 = Not significant because the two PLCs are Offline.</td></tr>
</table>

Note: To have a new Primary after the switch replacement, it is required to perform a Stop / Run command on one of the 2 PLCs. The other one becomes Standby.

⚠ WARNING

Risk of unintended equipment operation

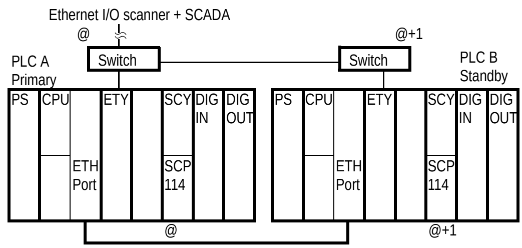
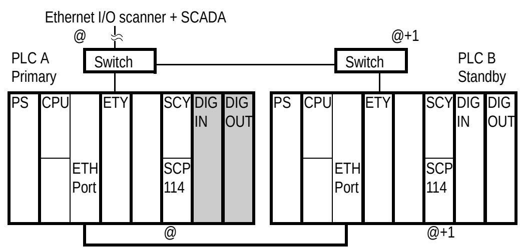
When the I/O Scanning service is used in the monitored ETY, we advise using one switch on each ETY.

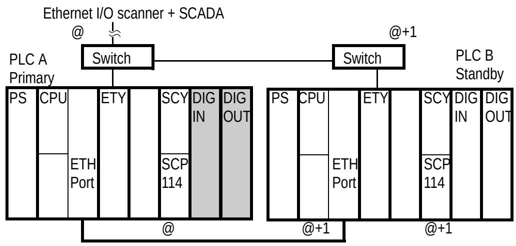
Failure to follow these instructions can result in death, serious injury, or equipment damage.

Hardware Failure of a Digital Module

Hardware Failure: Primary Digital Module

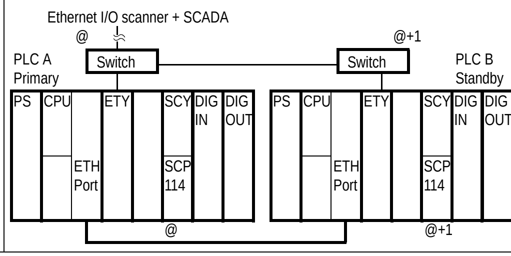
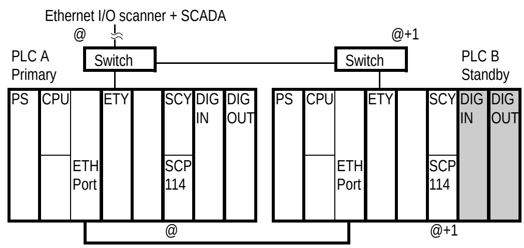
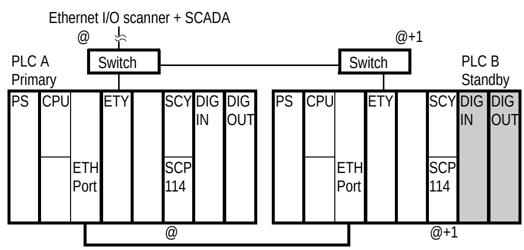
The following table presents hardware failure (or removal) of a digital module in the Primary PLC (main or extendable rack):

Before the event	
In rack Discrete I/O state: • PLC A: calculated and applied at end of task cycle. • PLC B: PLC A output applied at end of task cycle. Remote I/O state: • PLC A: all connections with Ethernet devices are open: I/O scanner is active. • PLC B: all connections with Ethernet devices are closed: I/O scanner is not active.	
Event	
Hardware failure (or the module is removed from the X-BUS rack) of a digital module. This is not a critical event because no automatic switch over occurs.	

<table>
<tr><td colspan="3">After the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
• PLC A: calculated and applied at end of task cycle
• PLC B: PLC A output applied at end of task cycle
Remote I/O state:
• PLC A: all connections with Ethernet devices are open: I/O scanner is active
• PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>
</td></tr>
<tr><td>Global status</td><td>Communication status</td><td>Customer diagnostic through Ethernet address @</td></tr>
<tr><td>No impact on the Hot Standby system. The process is still redundant. If needed, the customer can request a switch over by setting a bit in the command register of the Primary application (if there is no fault in the other PLC.</td><td>The 2 PLCs are accessible through terminal ports, Modbus links and Ethernet links for diagnostics.</td><td>%SW61 = 1000 0000 0000 1110
• The accessed PLC is PLC A / primary
• The other PLC is PLC B / standby
%SW62 = 0000 0000 0000 0000
• Other PLC: no fault
Primary PLC: error bit (%I0.x.mod.err) of the 2 discrete modules set to 1.</td></tr>
</table>

Hardware Failure: Standby Digital Module

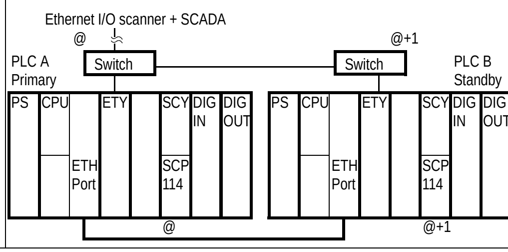
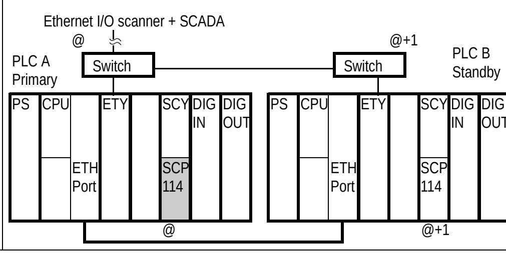
The following table presents hardware failure (or removal) of a digital module in the Standby PLC (main or extendable rack):

<table>
<tr><td colspan="3">Before the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
• PLC A: calculated and applied at end of task cycle
• PLC B: PLC A output applied at end of task cycle
Remote I/O state:
• PLC A: all connections with Ethernet devices are open: I/O scanner is active
• PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>
</td></tr>
<tr><td colspan="3">Event</td></tr>
<tr><td colspan="2">Hardware failure (or the module is removed from the X-BUS rack) of a digital module.
This is not a critical event because there is no switchover.</td><td>
</td></tr>
<tr><td colspan="3">After the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
• PLC A: calculated and applied at end of task cycle
• PLC B: PLC A output applies at end of task cycle
Remote I/O state:
• PLC A: all connections with Ethernet devices are open: I/O scanner is active
• PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>
</td></tr>
<tr><td>Global status</td><td>Communication status</td><td>Customer diagnostic through Ethernet address @</td></tr>
<tr><td>No impact on the Hot Standby system. The process is still redundant.</td><td>The 2 PLCs are accessible through terminal ports, Modbus links and Ethernet links for diagnostics.</td><td>%SW61 = 1000 0000 0000 1110
• The accessed PLC is PLC A: primary
• The other PLC is PLC B: standby
%SW62 = 0000 0000 0001 1000
• Other PLC: 2 discrete modules in fault</td></tr>
</table>

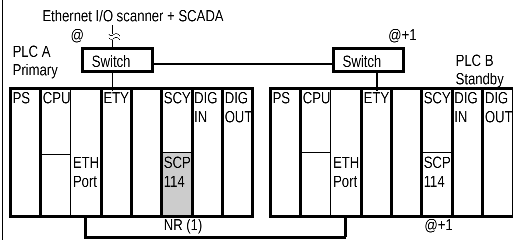
Hardware Failure of the SCP card in SCY

SCP card failure in Primary SCY

The following table presents hardware failure (or removal) of the SCP card in the Primary SCY:

Before the event	
In rack Discrete I/O state: ● PLC A: calculated and applied end of task cycle ● PLC B: PLC A output applied end of task cycle Remote I/O state: ● PLC A: all connections with Ethernet devices are open: I/O scanner is active ● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active	
Event	
Hardware failure (or the module is removed from the SCY module) of the Modbus SCP card. This is not a critical event because no automatic switch over occurs.	

<table>
<tr><td colspan="3">After the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
● PLC A: calculated and applied end of task cycle
● PLC B: PLC A output applied at end of task cycle

Remote I/O state:
● PLC A: all connections with Ethernet devices are open: I/O scanner is active
● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>

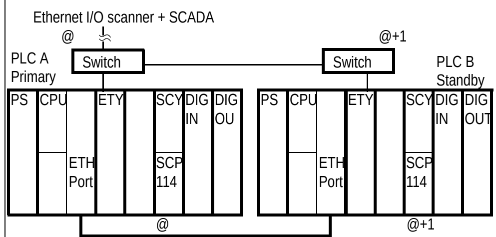
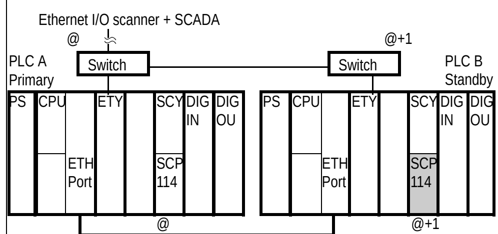
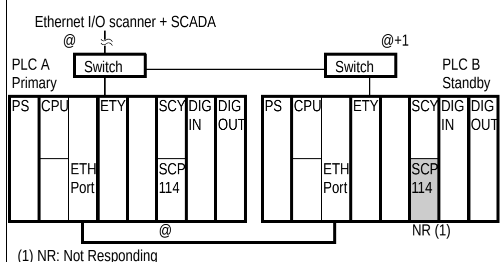
(1) NR: Not Responding</td></tr>
<tr><td>Global status</td><td>Communication status</td><td>Customer diagnostic through Ethernet address @</td></tr>
<tr><td>No impact on the Hot Standby system. The process is still redundant. If needed, the customer can request a switch over by setting a bit in the command register of the Primary application (if there is no fault in the other PLC.</td><td>● Normal access to PLC A through terminal port and Ethernet link for diagnostics. No access through Modbus link
● Normal access to PLC B through terminal port, Ethernet link and Modbus link for diagnostics</td><td>%SW61 = 1000 0000 0000 1110
● The accessed PLC is PLC A / primary
● The other PLC is PLC B / standby

%SW62 = 0000 0000 0000 0000
● Other PLC: no fault

Primary PLC: error bit of SCP / SCY modules (%I0.4.mod.err,%I0.4.1.err) set to 1</td></tr>
</table>

Hardware Failure of SCP card in SCY

The following table presents hardware failure (or removal) of the SCP card in the Standby SCY:

<table>
<tr><td colspan="3">Before the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
● PLC A: calculated and applied at end of task cycle
● PLC B: PLC A output applied at end of task cycle
Remote I/O state:
● PLC A: all connections with Ethernet devices are open: I/O scanner is active
● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>
</td></tr>
<tr><td colspan="3">Event</td></tr>
<tr><td colspan="2">Hardware failure (or the module is removed from the SCY module) of the Modbus SCP card.
This is not a critical event because there is no switch over.</td><td>
</td></tr>
<tr><td colspan="3">After the event</td></tr>
<tr><td colspan="2">In rack Discrete I/O state:
● PLC A: calculated and applied at end of task cycle
● PLC B: PLC A output applies at end of task cycle
Remote I/O state:
● PLC A: all connections with Ethernet devices are open: I/O scanner is active
● PLC B: all connections with Ethernet devices are closed: I/O scanner is not active</td><td>
</td></tr>
<tr><td>Global status</td><td>Communication status</td><td>Customer diagnostic through Ethernet address @</td></tr>
<tr><td>No impact on the Hot Standby system. The process is still redundant.</td><td>● Normal access to PLC A through terminal port and Ethernet link and Modbus link for diagnostics
● Normal access to PLC B through terminal port and Ethernet link for diagnostics. No access through Modbus link</td><td>%SW61 = 1000 0000 0000 1110
● The accessed PLC is PLC A: primary
● The other PLC is PLC B: standby
%SW62 = 0000 0000 0110 0000
● Other PLC: SCP/SCY in fault</td></tr>
</table>

Glossary

!

%I According to the IEC standard, `%I` indicates a discrete input-type language object.

%IW According to the IEC standard, `%IW` indicates an analog input -type language object.

%KW According to the IEC standard, `%KW` indicates a constant word-type language object.

%M According to the IEC standard, `%M` indicates a memory bit-type language object.

%MW According to the IEC standard, `%MW` indicates a memory word-type language object.

%Q According to the IEC standard, `%Q` indicates a discrete output-type language object.

%QW According to the IEC standard, `%QW` indicates an analog output-type language object.

A

ADDR_TYPE This predefined type is used as output for ADDR function. This type is ARRAY[0..5] OF Int. You can find it in the libset, in the same family of the EFs which use it.

ANL_IN `ANL_IN` is the abbreviation of Analog Input data type and is used when processing analog values. The `%IW` adresses for the configured analog input module, which were specified in the I/O component list, are automatically assigned data types and should therefore only be occupied with Unlocated Variables.

ANL_OUT

`ANL_OUT` is the abbreviation of Analog Output data type and is used when processing analog values. The `%MW` adresses for the configured analog input module, which were specified in the I/O component list, are automatically assigned data types and should therefore only be occupied with Unlocated Variables.

ANY

There is a hierarchy between the different types of data. In the DFB, it is sometimes possible to declare which variables can contain several types of values. Here, we use `ANY_xxx` types.
The following diagram shows the hierarchically-ordered structure:

```
ANY
  ANY_ELEMENTARY
    ANY_MAGNITUDE_OR_BIT
      ANY_MAGNITUDE
        ANY_NUM
          ANY_REAL
            REAL
          ANY_INT
            DINT, INT, UDINT, UINT
        TIME
      ANY_BIT
        DWORD, WORD, BYTE, BOOL
    ANY_STRING
      STRING
    ANY_DATE
      DATE_AND_TIME, DATE, TIME_OF_DAY
    EBOOL
  ANY_DERIVED
    ANY_ARRAY
      ANY_ARRAY_ANY_EDT
        ANY_ARRAY_ANY_MAGNITUDE
          ANY_ARRAY_ANY_NUM
            ANY_ARRAY_ANY_REAL
              ANY_ARRAY_REAL
            ANY_ARRAY_ANY_INT
              ANY_ARRAY_DINT
              ANY_ARRAY_INT
              ANY_ARRAY_UDINT
              ANNY_ARRAY_UINT
          ANY_ARRAY_TIME
        ANY_ARRAY_ANY_BIT
          ANY_ARRAY_DWORD
          ANY_ARRAY_WORD
          ANY_ARRAY_BYTE
          ANY_ARRAY_BOOL
        ANY_ARRAY_ANY_STRING
          ANY_ARRAY_STRING
        ANY_ARRAY_ANY_DATE
          ANY_ARRAY_DATE_AND_TIME
          ANY_ARRAY_DATE
          ANY_ARRAY_TIME_OF_DAY
        ANY_ARRAY_EBOOL
      ANY_ARRAY_ANY_DDT
    ANY_STRUCTURE
      ANY_DDT
      ANY_IODDT
      ANY_FFB
        ANY_EFB
        ANY_DFB
```

ARRAY

An ARRAY is a table of elements of the same type.
The syntax is as follows: ARRAY [<terminals>] OF <Type>
Example:
ARRAY [1..2] OF BOOL is a one-dimensional table made up of two BOOL-type elements.
ARRAY [1..10, 1..20] OF INT is a two-dimensional table made up of 10x20 INT-type elements.

B

Base 10 literals

A literal value in base 10 is used to represent a decimal integer value. This value can be preceded by the signs "+" and "-". If the character "_" is employed in this literal value, it is not significant.
Example:
-12, 0, 123_456, +986

Base 16 Literals

A literal value in base 16 is used to represent an integer in hexadecimal. The base is determined by the number "16" and the sign "#". The signs "+" and "-" are not allowed. For greater clarity when reading, you can use the sign "_" between bits.
Example:
16#F_F or 16#FF (in decimal 255)
16#E_0 or 16#E0 (in decimal 224)

Base 2 Literals

A literal value in base 2 is used to represent a binary integer. The base is determined by the number "2" and the sign "#". The signs "+" and "-" are not allowed. For greater clarity when reading, you can use the sign "_" between bits.
Example:
2#1111_1111 or 2#11111111 (in decimal 255)
2#1110_0000 or 2#11100000 (in decimal 224)

Base 8 Literals

A literal value in base 8 is used to represent an octal integer. The base is determined by the number "8" and the sign "#". The signs "+" and "-" are not allowed. For greater clarity when reading, you can use the sign "_" between bits.
Example:
8#3_77 or 8#377 (in decimal 255)
8#34_0 or 8#340 (in decimal 224)

BCD

BCD is the abbreviation of Binary Coded Decimal format
BCD is used to represent decimal numbers between 0 and 9 using a group of four bits (half-byte).
In this format, the four bits used to code the decimal numbers have a range of unused combinations.

Example of BCD coding:
- the number `2450`
- is coded: `0010 0100 0101 0000`

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`.

BYTE

When 8 bits are put together, this is called a `BYTE`. A `BYTE` is either entered in binary, or in base 8.
The `BYTE` type is coded in an 8 bit format, which, in hexadecimal, ranges from `16#00` to `16#FF`

D

DATE

The `DATE` type coded in BCD in 32 bit format contains the following information:
- the year coded in a 16-bit field,
- the month coded in an 8-bit field,
- the day coded in an 8-bit field.

The `DATE` type is entered as follows: **`D#`**`<Year>`**`-`**`<Month>`**`-`**`<Day>`
This table shows the lower/upper limits in each field:

Field	Limits	Comment
Year	[1990,2099]	Year
Month	[01,12]	The left 0 is always displayed, but can be omitted at the time of entry
Day	[01,31]	For the months 01\03\05\07\08\10\12
	[01,30]	For the months 04\06\09\11
	[01,29]	For the month 02 (leap years)
	[01,28]	For the month 02 (non leap years)

DATE_AND_TIME

see `DT`

DBCD

Representation of a Double BCD-format double integer.
The Binary Coded Decimal (BCD) format is used to represent decimal numbers between 0 and 9 using a group of four bits.
In this format, the four bits used to code the decimal numbers have a range of unused combinations.

Example of DBCD coding:
- the number 78993016
- is coded: 0111 1000 1001 1001 0011 0000 0001 0110

DDT

DDT is the abbreviation of Derived Data Type.
A derived data type is a set of elements of the same type (ARRAY) or of various types (structure)

DFB

DFB is the abbreviation of Derived Function Block.
DFB types are function blocks that can be programmed by the user ST, IL, LD or FBD.
By using DFB types in an application, it is possible to:
- simplify the design and input of the program,
- increase the legibility of the program,
- facilitate the debugging of the program,
- reduce the volume of the generated code.

DINT

DINT is the abbreviation of Double Integer format (coded on 32 bits).
The lower and upper limits are as follows: -(2 to the power of 31) to (2 to the power of 31) - 1.
Example:
-2147483648, 2147483647, 16#FFFFFFFF.

DT

DT is the abbreviation of Date and Time.
The DT type coded in BCD in 64 bit format contains the following information:
- The year coded in a 16-bit field,
- the month coded in an 8-bit field,
- the day coded in an 8-bit field,
- the hour coded in a 8-bit field,
- the minutes coded in an 8-bit field,
- the seconds coded in an 8-bit field.

Note: The 8 least significant bits are unused.

The DT type is entered as follows:
DT#<Year>**-**<Month>**-**<Day>**-**<Hour>**:**<Minutes>**:**<Seconds>
This table shows the lower/upper limits in each field:

Field	Limits	Comment
Year	[1990,2099]	Year
Month	[01,12]	The left 0 is always displayed, but can be omitted at the time of entry

Field	Limits	Comment
Day	[01,31]	For the months 01\03\05\07\08\10\12
	[01,30]	For the months 04\06\09\11
	[01,29]	For the month 02 (leap years)
	[01,28]	For the month 02 (non leap years)
Hour	[00,23]	The left 0 is always displayed, but can be omitted at the time of entry
Minute	[00,59]	The left 0 is always displayed, but can be omitted at the time of entry
Second	[00,59]	The left 0 is always displayed, but can be omitted at the time of entry

DWORD

DWORD is the abbreviation of Double Word.
The DWORD type is coded in 32 bit format.
This table shows the lower/upper limits of the bases which can be used:

Base	Lower limit	Upper limit
Hexadecimal	16#0	16#FFFFFFFF
Octal	8#0	8#37777777777
Binary	2#0	2#11111111111111111111111111111111

Representation examples:

Data content	Representation in one of the bases
00000000000010101101110011011110	16#ADCDE
00000000000000010000000000000000	8#200000
00000000000010101011110011011110	2#10101011110011011110

E

EBOOL

EBOOL is the abbreviation of Extended Boolean type. It can be used to manage rising or falling edges, as well as forcing.
An EBOOL type variable takes up one byte of memory.

EF

Is the abbreviation of Elementary Function.
This is a block which is used in a program, and which performs a predefined software function.

A function has no internal status information. Multiple invocations of the same function using the same input parameters always supply the same output values. Details of the graphic form of the function invocation can be found in the "[Functional block (instance)] ". In contrast to the invocation of the function blocks, function invocations only have a single unnamed output, whose name is the same as the function. In FBD each invocation is denoted by a unique [number] via the graphic block, this number is automatically generated and can not be altered.
You position and set up these functions in your program in order to carry out your application.
You can also develop other functions using the SDKC development kit.

EFB
Is the abbreviation for Elementary Function Block.
This is a block which is used in a program, and which performs a predefined software function.
EFBs have internal statuses and parameters. Even where the inputs are identical, the output values may be different. For example, a counter has an output which indicates that the preselection value has been reached. This output is set to 1 when the current value is equal to the preselection value.

Elementary Function
see EF

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 **E**rror **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".

ERP
Enterprise Resource Planning (ERP) systems.

F

FBD
FBD is the abbreviation of Function Block Diagram.

FBD is a graphic programming language that operates as a logic diagram. In addition to the simple logic blocks (AND, OR, etc.), each function or function block of the program is represented using this graphic form. For each block, the inputs are located to the left and the outputs to the right. The outputs of the blocks can be linked to the inputs of other blocks to form complex expressions.

FDR Faulty Device Replacement.

FFB Collective term for EF (Elementary Function), EFB (Elementary Function Block) and DFB (Derived Function block)

FTB Temperature base factor.

FTM Field Terminal Module.

FTP File Transfer Protocol.

Function see EF

Function Block Diagram see FBD

G

GRAY Gray or "reflected binary" code is used to code a numerical value being developed into a chain of binary configurations that can be differentiated by the change in status of one and only one bit.
This code can be used, for example, to avoid the following random event: in pure binary, the change of the value 0111 to 1000 can produce random numbers between 0 and 1000, as the bits do not change value altogether simultaneously.
Equivalence between decimal, BCD and Gray:

Decimal	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001
Gray	0000	0001	0011	0010	0110	0111	0101	0100	1100	1101

H

HMI Software based operator interface tool

HSBY Hot Standby

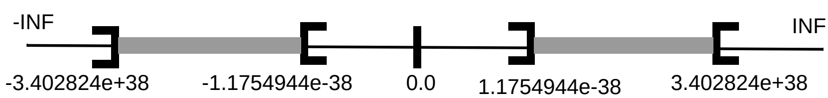
HTTP Hypertext Transfer Protocol

I

IEC 61131-3 International standard: Programmable Logic Controls
Part 3: Programming languages.

IL IL is the abbreviation of Instruction List.
This language is a series of basic instructions.
This language is very close to the assembly language used to program processors.
Each instruction is composed of an instruction code and an operand.

INF Used to indicate that a number overruns the allowed limits.
For a number of Integers, the value ranges (shown in gray) are as follows:



When a calculation result is:
- less than -3.402824e+38, the symbol `-INF` (for -infinite) is displayed,
- greater than +3.402824e+38, the symbol `INF` (for +infinite) is displayed.

INT `INT` is the abbreviation of single integer format (coded on 16 bits).
The lower and upper limits are as follows: -(2 to the power of 15)+1 to (2 to the power of 15) - 1.
Example:
`-32768`, `32767`, `2#1111110001001001`, `16#9FA4`.

Integer Literals Integer literal are used to enter integer values in the decimal system. The values can have a preceding sign (+/-). Individual underlines (_) between numbers are not significant.

Example:
`-12`, `0`, `123_456`, `+986`

IODDT IODDT is the abbreviation of Input/Output Derived Data Type.
The term IODDT designates a structured data type representing a module or a channel of a PLC module. Each application expert module possesses its own IODDTs.

K

Keyword A keyword is a unique combination of characters used as a syntactical programming language element (See annex B definition of the IEC standard 61131-3. All the key words used in Unity Pro and of this standard are listed in annex C of the IEC standard 61131-3. These keywords cannot be used as identifiers in your program (names of variables, sections, DFB types, etc.)).

L

LD LD is the abbreviation of Ladder Diagram.
LD is a programming language, representing the instructions to be carried out in the form of graphic diagrams very close to a schematic electrical diagram (contacts, coils, etc.).

Located variables A located variable is a variable for which it is possible to know its position in the PLC memory. For example, the variable `Water_pressure`, is associated with`%MW102`. `Water_pressure` is said to be localized.

M

MES Manufacturing Execution System.

Multiple Token Operating mode of an SFC. In multitoken mode, the SFC may possess several active steps at the same time.

N

Naming conventions (Identifier)

An identifier is a sequence of letters, numbers and underlines beginning with a letter or underline (e.g. name of a function block type, an instance, a variable or a section). Letters from national character sets (e.g: ö, ü, é, õ) can be used except in project and DFB names. Underlines are significant in identifiers; e.g. `A_BCD` and `AB_CD` are interpreted as different identifiers. Multiple leading underlines and consecutive underlines are invalid.
Identifiers cannot contain spaces. Not case sensitive; e.g. `ABCD` and `abcd` are interpreted as the same identifier.
According to IEC 61131-3 leading digits are not allowed in identifiers. Nevertheless, you can use them if you activate in dialog **Tools** → **Project settings** in tab **Language extensions** the check box **Leading digits**.
Identifiers cannot be keywords.

NAN

Used to indicate that a result of an operation is not a number (NAN = Not A Number). Example: calculating the square root of a negative number.

Note: The IEC 559 standard defines two classes of NAN: quiet `NAN` (`QNAN`) and signaling `NaN` (`SNaN`) `QNAN` is a `NAN` with the most significant fraction bit set and a `SNAN` is a `NAN` with the most significant fraction bit clear (Bit number 22). `QNAN`s are allowed to propagate through most arithmetic operations without signaling an exception. `SNAN` generally signal an invalid-operation exception whenever they appear as operands in arithmetic operations (See %SW17 and %S18).

Network

There are two meanings for Network.
- In LD:
  A network is a set of interconnected graphic elements. The scope of a network is local to the program organization unit (section) in which the network is located.
- With communication expert modules:
  A network is a group of stations which communicate among one another. The term network is also used to define a group of interconnected graphic elements. This group forms then a part of a program which may be composed of a group of networks.

NTP

Network Time Protocol.

O

OTB

The OTB NIM is an Input / Output module that has 12 input nodes and 8 output nodes.

P

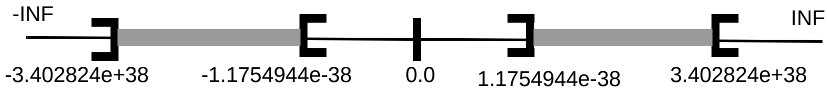
Procedure

Procedures are functions view technically. The only difference to elementary functions is that procedures can take up more than one output and they support data type `VAR_IN_OUT`. To the eye, procedures are no different than elementary functions.
Procedures are a supplement to IEC 61131-3.

R

REAL

Real type is a coded type in 32 bits.
The ranges of possible values are illustrated in gray in the following diagram:



When a calculation result is:
- between -1.175494e-38 and 1.175494e-38 it is considered as a `DEN`,
- less than -3.402824e+38, the symbol `-INF` (for - infinite) is displayed,
- greater than +3.402824e+38, the symbol `INF` (for +infinite) is displayed,
- undefined (square root of a negative number), the symbol `NAN` or `NAN` is displayed.

Note: The IEC 559 standard defines two classes of NAN: quiet `NAN` (`QNAN`) and signaling `NaN` (`SNaN`) `QNAN` is a `NAN` with the most significant fraction bit set and a `SNAN` is a `NAN` with the most significant fraction bit clear (Bit number 22). `QNAN`s are allowed to propagate through most arithmetic operations without signaling an exception. `SNAN` generally signal an invalid-operation exception whenever they appear as operands in arithmetic operations (See %SW17 and %S18).

> **Note:** when an operand is a `DEN` (Demoralizing number) the result is not significant.

Real Literals

A literal real value is a number expressed in one or more decimals.
Example:
`-12.0`, `0.0`, `+0.456`, `3.14159_26`

Real Literals with Exponent

A literal decimal value can be expressed using standard scientific notation. The representation is as follows: mantissa + exponential.
Example:
`-1.34E-12` or `-1.34e-12`
`1.0E+6` or `1.0e+6`
`1.234E6` or `1.234e6`

S

SCADA

Software based operator interface tool

SFC

SFC is the abbreviation of Sequential Function Chart.
SFC enables the operation of a sequential automation device to be represented graphically and in a structured manner. This graphic description of the sequential behavior of an automation device, and the various situations which result from it, is performed using simple graphic symbols.

Single Token

Operating mode of an SFC chart for which only a single step can be active at any one time.

SMTP

Simple Mail Transfer Protocol.

SNMP

Simple Network Management Protocol.

ST

ST is the abbreviation of Structured Text language.
Structured Text language is an elaborated language close to computer programming languages. It enables you to structure series of instructions.

STB

Standard Terminal Block.

STRING

A variable of the type `STRING` is an ASCII standard character string. A character string has a maximum length of 65534 characters.

T

TFTP Trivial File Transfer Protocol.

TIME The type TIME expresses a duration in milliseconds. Coded in 32 bits, this type makes it possible to obtain periods from 0 to 2^{32}-1 milliseconds.
The units of type TIME are the following: the days (d), the hours (h), the minutes (m), the seconds (s) and the milliseconds (ms). A literal value of the type TIME is represented by a combination of previous types preceded by T#, t#, TIME# or time#.
Examples: T#25h15m, t#14.7S, TIME#5d10h23m45s3ms

Time literals The units of type TIME are the following: the days (d), the hours (h), the minutes (m), the seconds (s) and the milliseconds (ms). A literal value of the type TIME is represented by a combination of previous types preceded by T#, t#, TIME# or time#.
Examples: T#25h15m, t#14.7S, TIME#5d10h23m45s3ms

TIME_OF_DAY see TOD

TOD TOD is the abbreviation of Time of Day.
The TOD type coded in BCD in 32 bit format contains the following information:
- the hour coded in a 8-bit field,
- the minutes coded in an 8-bit field,
- the seconds coded in an 8-bit field.

Note: The 8 least significant bits are unused.

The Time of Day type is entered as follows: **TOD#**<Hour>**:**<Minutes>**:**<Seconds>
This table shows the lower/upper limits in each field:

Field	Limits	Comment
Hour	[00,23]	The left 0 is always displayed, but can be omitted at the time of entry
Minute	[00,59]	The left 0 is always displayed, but can be omitted at the time of entry
Second	[00,59]	The left 0 is always displayed, but can be omitted at the time of entry

Example: TOD#23:59:45.

Token An active step of an SFC is known as a token.

TOPO_ADDR_TYPE This predefined type is used as output for READ_TOPO_ADDR function. This type is an ARRAY[0..4] OF Int. You can find it in the libset, in the same family than the EFs which use it.

U

UDINT `UDINT` is the abbreviation of Unsigned Double Integer format (coded on 32 bits) unsigned. The lower and upper limits are as follows: 0 to (2 to the power of 32) - 1.
Example:
`0`, `4294967295`, `2#11111111111111111111111111111111`, `8#37777777777`, `16#FFFFFFFF`.

UINT `UINT` is the abbreviation of Unsigned integer format (coded on 16 bits). The lower and upper limits are as follows: 0 to (2 to the power of 16) - 1.
Example:
`0`, `65535`, `2#1111111111111111`, `8#177777`, `16#FFFF`.

Unlocated variable An unlocated variable is a variable for which it is impossible to know its position in the PLC memory. A variable which have no address assigned is said to be unlocated.

V

Variable Memory entity of the type `BOOL`, `WORD`, `DWORD`, etc., whose contents can be modified by the program during execution.

W

WORD The `WORD` type is coded in 16 bit format and is used to carry out processing on bit strings.
This table shows the lower/upper limits of the bases which can be used:

Base	Lower limit	Upper limit
Hexadecimal	16#0	16#FFFF

Base	Lower limit	Upper limit
Octal	8#0	8#177777
Binary	2#0	2#1111111111111111

Representation examples

Data content	Representation in one of the bases
0000000011010011	16#D3
1010101010101010	8#125252
0000000011010011	2#11010011

Index