Modicon M580

Hardware
Reference Manual

10/2013

EIO0000001578.00

www.schneider-electric.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 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 or serious injury.

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

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

PLEASE NOTE

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

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

About the Book

At a Glance

Document Scope

This manual describes the Modicon M580 central processing unit (CPU), power supplies, and racks. It focuses on

- the installation of a local rack in the Modicon M580 system
- the configuration of the CPUs
- the remote and distributed I/O scanner capabilities of the CPU

Validity Note

This document is valid from Unity Pro V8.0.

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

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

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

Related Documents

Title of Documentation	Reference Number
Control Panel Technical Guide, How to protect a machine from malfunctions due to electromagnetic disturbance	CPTG003_EN (English), CPTG003_FR (French)
Grounding and Electromagnetic Compatibility of PLC Systems, Basic Principles and Measures, User Manual	33002439 (English), 33002440 (French),33002441 (German), 33003702 (Italian),33002442 (Spanish),33003703 (Chinese)
Modicon M580 System Planning Guide	HRB62666 (English), HRB65318 (French), HRB65319 (German), HRB65320 (Italian), HRB65321 (Spanish), HRB65322 (Chinese)
Modicon M580 Remote I/O Modules Installation and Configuration Guide	EIO0000001584 (English), EIO0000001585 (French), EIO0000001586 (German), EIO0000001588 (Italian), EIO0000001587 (Spanish), EIO0000001589 (Chinese)
Modicon X80 with Unity Pro HART Analog Input/Output Modules User Guide	xxxxxxxx (English), xxxxxxxx (French), xxxxxxxx (German), xxxxxxxx (Italian), xxxxxxxx (Spanish), xxxxxxxx (Chinese)
Unity Loader User Manual	33003805 (English), 33003806 (French), 33003807 (German), 33003809 (Italian), 33003808 (Spanish), 33003810 (Chinese)
Unity Pro Operating Modes	33003101 (English), 33003102 (French),33003103 (German), 33003696 (Italian),33003104 (Spanish),33003697 (Chinese)

Title of Documentation	Reference Number
Unity Pro Program Languages and Structure Reference Manual	35006144 (English), 35006145 (French), 35006146 (German), 35013361 (Italian), 35006147 (Spanish), 35013362 (Chinese)

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

Product Related Information

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

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

Follow all local and national safety codes and standards.

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

Part I
Hardware Elements in the Modicon M580 Local Rack

Introduction

This part provides information on the Modicon M580 CPUs, power supply module and the racks on which the system modules are mounted. The physical and operational characteristics of these elements are described.

What Is in This Part?

This part contains the following chapters:

Chapter 1
M580 CPUs

Introduction

This chapter introduces you to the physical and functional characteristics of the M580 CPUs.

What Is in This Chapter?

This chapter contains the following sections:

Section 1.1
BME P58 *xxxx* CPU Functional Characteristics

Introduction

This section describes the functional characteristics of the M580 CPUs. Performance, electrical characteristics, and memory capacities of the different CPU modules are detailed.

What Is in This Section?

This section contains the following topics:

Modicon M580 CPU Characteristics

Role of the CPU in a Control System

In a modular PAC, the CPU controls and processes the application. The local rack identifies the rack that contains the CPU. In addition to the CPU, the local rack contains a power supply module and may optionally contain communication processing modules and input/out (I/O) modules.

The CPU is in charge of:

- configuring all modules and device present in the PAC configuration
- processing the application
- reading the inputs at the beginning of tasks and applying the outputs at the end of tasks
- managing the explicit and implicit communications

Modules may reside in the local rack with the CPU or they may be installed in remote drops at a distance from the local rack. The CPU has built-in capabilities to act as the RIO communications processor that manages communications between the CPU and the RIO adapters that are installed in each remote drop.

Devices can be connected to the PAC network as either distributed device clouds or distributed device sub-rings.

For detailed information about the various architectures that the M580 network supports, refer to the *Modicon M580 System Planning Guide*. For a detailed description of the RIO adapters and the options they provide for installing a remote drop, refer to the *M580 Remote I/O Modules Installation and Configuration Guide*.

Functional Considerations

The M580 CPU solves control logic for the I/O modules and distributed devices in the system. You can choose a CPU based on several operating characteristics:

- memory size
- processing power: the number of I/O points or channels that it can manage *(see page 21)*
- the speed at which the CPU can execute the control logic *(see page 22)*
- communication capabilities: the types of Ethernet ports on the CPU *(see page 40)*
- the number of local I/O modules and RIO drops that it can support *(see page 21)*
- ability to function in harsh environments: (3 CPU models are hardened to operate over extended temperature ranges and in dirty or corrosive environments *(see page 49)*

CPU References

There are seven M580 CPU references, three of which can be ordered as standard or industrially hardened modules. Industrially hardened modules have the letter H appended to the reference *(see page 49)* ID.

- BME P58 1020 and BME P58 1020 H
- BME P58 2020 and BME P58 2020 H
- BME P58 2040 and BME P58 2040 H
- BME P58 3020

- BME P58 3040
- BME P58 4020
- BME P58 4040

BME P58 *xxxx* CPU Performance Characteristics

I/O and Distributed Equipment Support

All CPUs can manage distributed equipment on a device network. This capability is called the DIO scanner service.

To manage RIO (remote) drops in the system, use the RIO scanner service. Only these CPUs provide the RIO scanner service:

- BME P58 2040 or BME P58 2040 H
- BME P58 3040
- BME P58 4040

The embedded scanner is configured via CPU IP configuration *(see page 155)*.

I/O support characteristics of the M580 CPU references:

	BME P58 References						
	1020(H)	2020(H)	2040(H)	3020	3040	4020	4040
Maximum number of discrete I/O channels	1024	2048	2048	3072	3072	4096	4096
Maximum number of analog I/O channels	256	512	512	768	768	1024	1024
Maximum number of RIO modules	–	–	64	–	128	–	128
Maximum number of distributed equipment components	64	128	64	128	64	128	64
Maximum number of Ethernet network modules apart from the CPU (1.)	2	2	2	3	3	4	4
Maximum number of local racks (main local rack + extension)	4	4	4	8	8	8	8
Maximum number of remote drops (with up to 2 racks per drop)	–	–	8	–	16	–	16
Ethernet ports:							
• Service	1	1	1	1	1	1	1
• RIO or distributed equipment	–	–	2	–	2	–	2
• Distributed equipment	2	2	–	2	–	2	–
- Not available							
1. In the maximum number of network modules, the given value does not include the CPU.							
NOTE: Those characteristics represent the maximum values that a specific CPU can manage in the system.							

Functions Performance

	BME P58 References						
	1020(H)	2020(H)	2040(H)	3020	3040	4020	4040
Simultaneous EF processed per cycle (max.)	16	32	32	48	48	80	80

Application Code Execution Performance

	BME P58 References						
	1020(H)	2020(H)	2040(H)	3020	3040	4020	4040
Boolean application execution (Kinst/ms[1.])	10	10	10	20	20	40	40
Typical execution (Kinst/ms[1.]) (65% boolean instructions + 35% fixed arithmetics)	7.5	7.5	7.5	15	15	30	30
1. Kinst/ms: 1024 instructions per millisecond							

Modicon M580 Operating States of the CPU

State Definitions

Hereafter are listed the different operating states of the CPU:

AUTOTEST: The CPU is executing its internal self-tests.

NOTE: If extender racks are connected to the local rack and line terminators are not plugged into the unused connectors on the rack extender module, the CPU remains in **AUTOTEST** after the self-test have completed.

NOCONF: The application program is not valid.

STOP: The CPU has a valid application, but it is stopped. It has set itself to predefined STOP state parameters and can be restarted when you are ready.

IDLE: The CPU has a valid application and is able to solve logic, but the application is not requiring CPU processing (the CPU has never been in RUN state). This state is not visible.

HALT: The CPU has an application, but it has stopped operating because it encountered an unexpected blocking condition. The condition that puts the CPU in a HALT state can result in a recoverable *(see page 140)* or nonrecoverable error *(see page 138)*.

RUN: The CPU is executing the application program.

WAIT: The CPU is in a transitory state while it backs up its data when a power down condition is detected.

The CPU starts again only when power is restored and the supply reserve is replenished. As it is a transitory state, it may not be viewed.

The CPU performs a warm restart *(see page 271)* to exit the WAIT state.

ERROR: The CPU is stopped because a hardware or system error is detected.

When the system is ready to be restarted, the CPU performs a cold start *(see page 269)* to exit the ERROR state.

OS DOWNLOAD: A CPU firmware download is in progress.

Monitoring the State of the CPU

The LEDs on the CPU front panel provide indications of the state of the CPU *(see page 35)*.

Electrical Characteristics of BME P58 *xxxx* CPU

Overview

The power supply module provides current to the modules installed on the rack, including the CPU. CPU current consumption contributes to the total rack consumption.

CPU Power Consumption

Typical CPU consumption with a 24 Vdc power supply:

CPU Reference	Typical Consumption
BME P58 10•0	270 mA
BME P58 20•0	270 mA
BME P58 30•0	295 mA
BME P58 40•0	295 mA

Mean Time Between Failures

For all the CPU references, the MTBF, measured at 30 °C continuous, is 600,000 hours.

BME P58 *xxxx* CPU Memory Sizes

Introduction

The following pages present the main characteristics of the BME P58 •••• CPU memory.

BME P58 *xxxx* CPUs Memory Sizes

Program and data memory capacity:

Memory Size	BME P58 References						
	1020 / 1020 H	2020 / 2020 H	2040 / 2040 H	3020	3040	4020	4040
Application global size (Kbytes)	4598	9048	9048	13558	13558	18678	18678

Maximum memory size per area:

Memory Size	BME P58 References						
	1020 / 1020 H	2020 / 2020 H	2040 / 2040 H	3020	3040	4020	4040
Maximum for saved data (Kbytes) (1.)	384	768	768	1024	1024	2048	2048
Maximum for unsaved data (Kbytes)	128	128	128	256	256	256	256
Maximum for program (Kbytes)	4096	8162	8162	12288	12288	16384	16384
1. 10 Kbytes are reserved for the system							

Size of Located Data Memory

Maximum and default size of located data according to the CPU (in Kbytes):

Object Types	Address	BME P58 References						
		1020 / 1020 H	2020 / 2020 H	2040 / 2040 H	3020	3040	4020	4040
Internal bits	`%Mi` max.	32634	32634	32634	32634	32634	32634	32634
	`%Mi` default	512	512	512	512	512	512	512
Input/Output bits	`%Ir.m.c` `%Qr.m.c`	(1.)	(1.)	(1.)	(1.)	(1.)	(1.)	(1.)
System bits	`%Si`	128	128	128	128	128	128	128
1. Memory size depends on the equipment configuration declared (I/O modules).								

Object Types	Address	BME P58 References						
		1020 / 1020 H	2020 / 2020 H	2040 / 2040 H	3020	3040	4020	4040
Internal words	`%MWi` max.	32464	32464	32464	65232	65232	65232	65232
	`%MWi` default	1024	1024	1024	2048	2048	2048	2048
Constant words	`%KWi` max.	32760	32760	32760	32760	32760	32760	32760
	`%KWi` default	256	256	256	256	256	256	256
System words	`%SWi`	168	168	168	168	168	168	168
1. Memory size depends on the equipment configuration declared (I/O modules).								

Size of Non-Located Data Memory

Non-located data types are as follows:
- Elementary Data Types (EDT)
- Derived Data Types (DDT)
- DFB and EFB function block data.

The size limit of non-located data is the global maximum memory size for data *(see page 25)* minus the size consumed by located data.

Real-Time Clock

Introduction

Your CPU has a real-time clock that:
- provides the current date and time
- lets you see the date and time of the last application shut-down

Clock Accuracy

The resolution of the real-time clock is 1 ms. Its accuracy is affected by the operating temperature of the application:

Operating Temperature	Maximum Daily Drift (Sec/Day)	Maximum Yearly Drift (Min/Year)
25 °C (77 °F) stabilized	+/- 0.9	+/- 6.8
0...60 °C (32...140 °F)	+/- 3.5	+/-22.6

Clock Back-Up

The accuracy of the real-time clock is maintained for 4 weeks when the CPU power is turned off if the temperature is below 45 °C (113 °F). If the temperature is higher, the back-up time is shorter. The real-time clock back-up does not need any maintenance.

If the back-up power is too low, system bit `%S51` is set to 1. This value indicates a loss of time when the power supply was OFF.

Current Date and Time

The CPU updates the current date and time in the system words `%SW49`–`%SW53` and `%SW70`. This data is in BCD.

System Word	Most Significant Byte	Least Significant Byte
`%SW49`	00	Days of the week in the range of values 1 - 7 (1 for Monday and 7 for Sunday)
`%SW50`	Seconds (0 - 59)	00
`%SW51`	Hours (0 - 23)	Minutes (0 to 59)
`%SW52`	Month (1 - 12)	Day in the month (1 to 31)
`%SW53`	Century (0 - 99)	Year (0 to 99)
`%SW70`	Year	Week (1 to 52)

Accessing the Date and Time

You can access the date and time:
- on the CPU debug screen
- in the program

To read the current date and time, read system words `%SW49` through `%SW53`. This operation sets system bit `%S50` to 0.

To write the current date and time, write system words `%SW50` through `%SW53`. This operation sets system bit `%S50` to 1.

When system bit `%S59` is set to 1, you can increment or decrement the current date and time values with system word `%SW59`.

The function performed by each bit in word `%SW59` is:

Bit	Function
0	Increments the day of the week
1	Increments the seconds
2	Increments the minutes
3	Increments the hours
4	Increments the days
5	Increments the months
6	Increments the years
7	Increments the centuries
8	Decrements the day of the week
9	Decrements the seconds
10	Decrements the minutes
11	Decrements the hours
12	Decrements the days
13	Decrements the months
14	Decrements the years
15	Decrements the centuries

NOTE: The function is performed when system bit `%S59` is set to 1.

Determining the Date and Time of the Last Application Shutdown

The date and time of the last application shutdown are in system words %SW54 through %SW58. They are displayed in BCD.

System Word	Most Significant Byte	Least Significant Byte
%SW54	Seconds (0 to 59)	00
%SW55	Hours (0 to 23)	Minutes (0 to 59)
%SW56	Month (1 to 12)	Day in the month (1 to 31)
%SW57	Century (0 to 99)	Year (0 to 99)
%SW58	Day of the week (1 to 7)	Reason for the last application shutdown

The reason for the last application shutdown can be displayed by reading the least significant byte of system word %SW58, which can have the following values (in BCD):

Word%SW58 Value	Meaning
1	Application switched to STOP mode.
2	Application stopped by watchdog.
4	Power loss.
5	Stop on hardware error.
6	Stop on software error (HALT instruction, SFC errors, application CRC checksum error, undefined system function call, and so on). Details on the software detected fault type are stored in %SW125.

Addressing Field Buses

Introduction

Various field buses can be addressed by either configuring the appropriate protocol or using dedicated modules and devices.

AS-i

AS-Interface bus is addressed with a Modicon X80 BMX EIA 0100 module plugged in a rack.

CANopen

CANopen is addressed with an Advantys STB island configured from Unity Pro.

The Advantys STB island is connected to the Ethernet distributed equipment network with one of the following devices:
- STB NIC 2212
- STB NIP 2212
- STB NIP 2311

The CANopen devices are linked to the following device on the Advantys STB island:
- STB XBE 2100

HART

HART communication protocol is addressed using Modicon M580 HART modules plugged in a rack. Example of modules:
- BME AHI 0812 HART analog input module
- BME AHO 0412 HART analog input module

Modbus TCP

Modbus TCP devices are connected to the CPU Ethernet distributed equipment network.

Modbus Plus

Modbus Plus is supported using a gateway module like TCSEGDB23F24FA or TCSEGDB23F24FK.

PROFIBUS-DP

A PROFIBUS remote master is connected to the Ethernet distributed equipment network. The process variables are exchanged via the DIO scanner.

PROFIBUS gateway modules: TCSEGPA23F14F or TCSEGPA23F14FK

PROFIBUS-PA

A PROFIBUS remote master and a DP/PA interface are connected to the Ethernet distributed equipment network. The process variables are exchanged via the DIO scanner.

PROFIBUS gateway modules: TCSEGPA23F14F or TCSEGPA23F14FK

Section 1.2
BME P58 *xxxx* CPU Physical Characteristics

Overview

This section describes the physical elements that can be seen and accessed on the front panel of the Modicon M580 CPUs. It describes the various communication ports, the kind of diagnostic information that can be seen on the LED displays, and several options available for industrial hardening and memory back-up.

What Is in This Section?

This section contains the following topics:

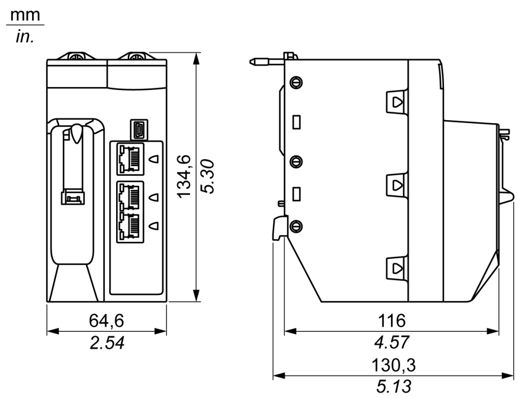
BME P58 *xxxx* CPU Position and Dimensions

CPU Position on the Local Rack

Every BME P58 •••• system requires 1 CPU module. The CPU is installed in the 2 module slots directly to the right of the power supply in the main local rack. The CPU cannot be put in any other slot locations or any other rack. If there are extended racks in the local rack configuration, assign address 00 to the rack with the CPU in it.

CPU Dimensions

BME P58 •••• CPU front and side dimensions:

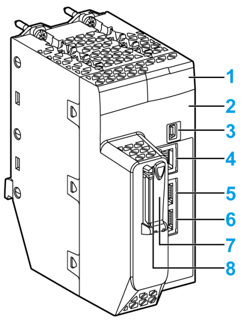


NOTE: Consider the height of the CPU when you are planning the installation of the local rack. The CPU extends below the lower edge of the rack by 29.49 mm (1.161 in.) for an Ethernet rack, and by 30.9 mm (1.217 in.) for an X Bus rack.

Front Panel View of the M580 CPUs

Introduction

All BME P58 •••• CPUs have a similar front panel. But there is one difference in the panels, depending on the CPU you are using: the function of 2 of the Ethernet RJ45 connectors (items 5 and 6).

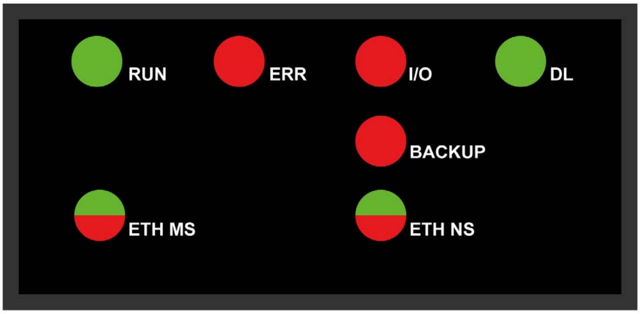


Item	Marking	Description
1	–	the LED display *(see page 35)* for CPU status and diagnostics
2	**Eth MAC Adress** **xx.xx.xx.xx.xx.xx**	the media access control (MAC) address assigned to the CPU. It is a string of six 2-digit hexadecimal numbers separated by dots.
	IP ADDRESS : . . .	space left blank for the user to write the IP address assigned to the CPU
3		the mini-B USB connector *(see page 38)* where you can attach a Unity Pro programming, or loader terminal, or HMI
4	**Service**	an RJ45 Ethernet connector *(see page 40)* for the service port
5	**Device Network**	Dual distributed equipment ports for CPUs that only support distributed devices, and dual distributed equipment or RIO ports for CPUs that support RIO scanning.
6		
7	–	Door to the SD memory card *(see page 45)* slot
8	–	Green LED that indicates memory card access. It illuminates when the CPU has access to the SD memory card and blinks when the CPU is accessing the memory card.

Diagnostic Indications from the LED Display

CPU LEDs

A 7 LED display is located on the front panel of the CPU:

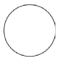
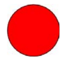
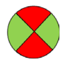


Dedicated LED Meanings

Each LED has a dedicated meaning:

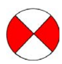
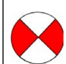
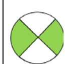
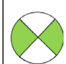
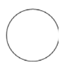
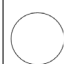
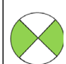
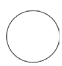
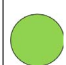
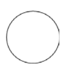
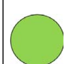
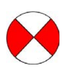
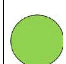
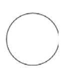
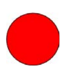
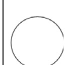
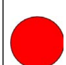
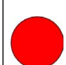
LED Indicator	Description
RUN	**ON**: The CPU is in RUN state.
ERR	**ON**: The CPU or system has detected an error.
I/O	**ON**: The CPU or system has detected an error in one or more I/O modules.
DL	Download **blinking**: during a firmware upgrade
	OFF: no firmware upgrade in progress
BACKUP	**ON**: ● Memory card or CPU flash memory is missing or inoperable. ● Memory card is not usable (bad format, unrecognized type). ● Memory card or CPU flash memory content is inconsistent with current application. ● Memory card has been removed and reinserted.
	OFF: Memory card or CPU flash memory content is valid and the application in execution memory is identical.
ETH MS	MOD STATUS: Pattern indicates the Ethernet port configuration status.
ETH NS	NET STATUS: Pattern indicates the Ethernet connection status.

The following legend shows the different LED patterns that indicate CPU states and diagnostic conditions

Symbol	Meaning	Symbol	Meaning
	off		steady red
	steady green		blinking red
	blinking green		blinking red/green

Diagnostic Indications

The LEDs provide detailed diagnostic information when you observe their pattern in combination:

Condition	CPU State	RUN	ERR	I/O	ETH MS	ETH NS
power on	autotest					
not configured (before getting a valid IP address or configuration is invalid)	NOCONF					Any pattern
configured	STOP			• **off**: no error detected • **steady red**: error detected in a module or a channel		• **off**: cable disconnected • **steady red**: duplicated IP address • **blinking green**: not connected • **steady green**: connected
	RUN					
recoverable detected error	HALT			Any pattern		Any pattern
unrecoverable detected error	–					

Condition	CPU State	RUN	ERR	I/O	ETH MS	ETH NS
power off	–					

USB Port

Overview

The USB port is a high-speed USB 2.0 connection (480 Mbps) for a Unity Pro programming or human-machine interface (HMI) panel. It can be connected to a USB port version 1.1 or later.

NOTE: Install M580 USB drivers before connecting USB cable between the CPU and the PC.

PC USB Transparency

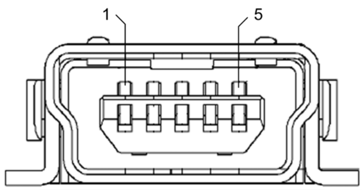
If transparency is needed between the PC connected on the USB port and M580 system device network, you must add a persistent static route in the PC routing table.

Example of command to address a device network with IP address `X.X.0.0` (for Windows):

```
route add X.X.0.0 mask 255.255.0.0 90.0.0.1 -p
```

Pin Assignments

The USB port is a mini-B USB connector with the following pin positions and pinouts:



Pin	Description
1	VBus
2	D-
3	D+
4	not connected
5	ground
shell	chassis ground

USB Cables

Use the following USB cables to connect the panel to the CPU (a type A connector on one side and the mini-B USB on the other side):

- BMX XCA USB 018: 1.8 m (5.91 ft) long
- BMX XCA USB 045: 4.5 m (14.76 ft) long

In a fixed assembly with an XBT type console connected to the CPU, connect the USB cable to a protection bar *(see page 114)*. Use the exposed part of the shield or the metal lug on the BMX XCA cable to make the connection.

Ethernet Ports

Overview

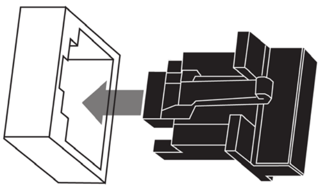
There are three RJ45 Ethernet ports on the front of the CPU, a service port, and 2 device network ports. They share some common characteristics and distinctions as described below.

Common Characteristics of the Ethernet Ports

All 3 ports have the same RJ45 connector and use the same type of Ethernet cables.

NOTE: The 3 Ethernet ports are connected to chassis ground, and the system requires an equipotential ground *(see page 110)*.

NOTE: To help prevent dust from entering the unused Ethernet ports, cover the unused ports with the stopper:

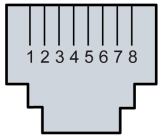


Each RJ45 connector has a pair of LED indicators:

The **ACT** LED is green, and the **LNK** LED may illuminate in either green or yellow.

LED	LED Status	Description
ACT	OFF	No activity on the Ethernet connection.
	ON / blinking green	Data is being transmitted and received on the Ethernet connection.
LNK	OFF	No link established at this connection.
	ON green	A 100 Mbps link* is established at this connection.
	ON yellow	A 10 Mbps link* is established at this connection.
* The 10/100 Mbps links support both half-duplex and full-duplex data transfer and autonegotiation.		

The pin positions, pinouts, and cable connections are the same on all three RJ45 Ethernet ports:



Pin	Description
1	TD+
2	TD-
3	RD+
4	not connected
5	not connected
6	RD-
7	not connected
8	not connected
Note: The TD pins (pin 1 and 2) and the RD pins (pins 3 and 6) can be reversed, allowing the exclusive use of straight-through cables.	

The ports have an auto MDIX capability that automatically detects the direction of the transmission.

Choose from the following Ethernet cables to connect to the Ethernet ports:
- TCS ECN 3M3M 05S2: Cat 5E Ethernet straight-through shielded cable, rated for industrial use, CE- or UL-compliant
- TCS ECN 3M3M ••••: Cat 5E Ethernet straight-through shielded cable, rated for industrial use, CE- or UL-compliant
- TCS ECE 3M3M ••••: Cat 5E Ethernet straight-through shielded cable, rated for industrial use, CE-compliant
- TCS ECU 3M3M ••••: Cat 5E Ethernet straight-through shielded cable, rated for industrial use, UL-compliant

The maximum length for a copper cable is 100 m. For distances greater than 100 m, use fiber optic cable. The CPU does not have any fiber ports on it. You may use dual ring switches (DRSs) or BMX NRP •••• fiber converter modules to handle the copper-fiber conversion.

Service Port

The service port is the uppermost of the 3 Ethernet ports on the front panel of the CPU. It can be used for:

- providing an access point that other devices or systems can use to monitor or communicate with the M580 CPU
- a standalone distributed equipment port that can support a star, daisy chain, or mesh topology of distributed equipment
- port mirroring of the CPU ports for Ethernet diagnostics. The service tool that views activity on the mirrored port may be a PC or an HMI device.

Device Network Dual Ports

When a CPU does not support RIO scanning, the 2 ports below the service port marked **Device Network** are distributed equipment ports.

The following CPUs do not support RIO scanning:

- BME P58 1020 and BME P58 1020 H
- BME P58 2020 and BME P58 2020 H
- BME P58 3020
- BME P58 4020

You may use a **Device Network** port to support a star, daisy chain, or mesh topology of distributed equipment. You may use both **Device Network** ports to support a ring topology.

Refer to the *Modicon M580 System Planning Guide* for more details regarding distributed equipment architectures.

When a CPU supports RIO scanning, the 2 ports below the service port marked **Device Network** are RIO or distributed equipment ports.

The following CPUs support RIO scanning:

- BME P58 2040 and BME P58 2040 H
- BME P58 3040
- BME P58 4040

When used as RIO ports, both ports connect the CPU to the main ring in an Ethernet daisy-chain loop.

Refer to the *Modicon M580 System Planning Guide* for more details regarding RIO drop architectures.

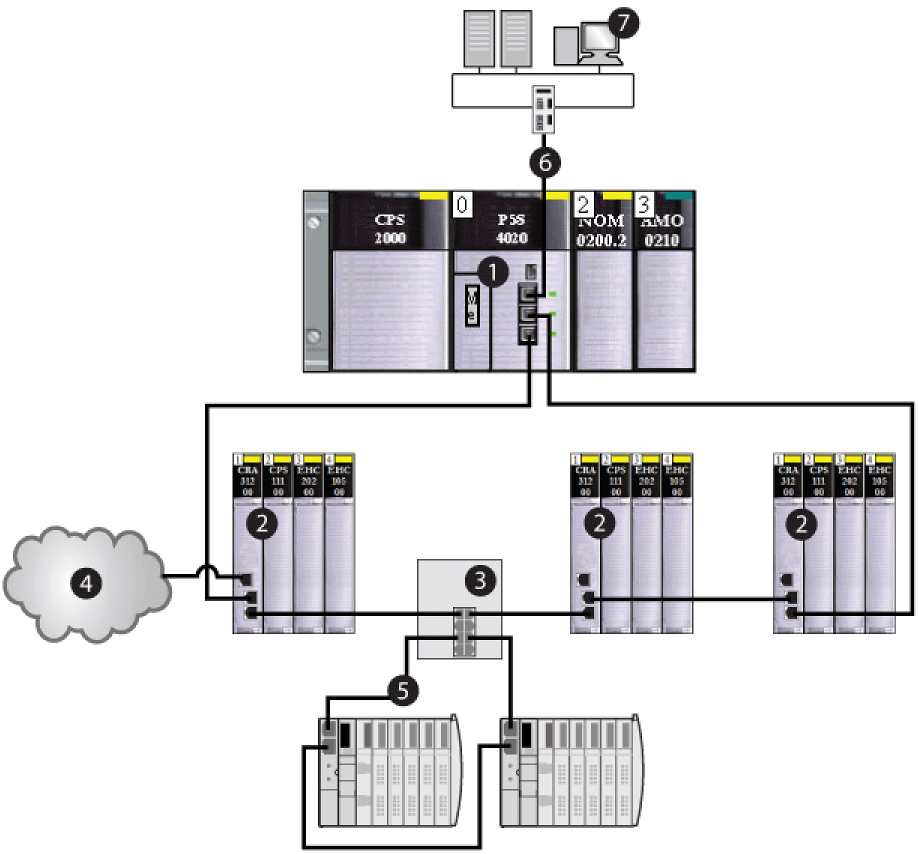
Connecting an M580 Device Network to the Control Network

Introduction

Via the service port on a CPU, connect your device network to the control network. The following figure shows a device network connected to a switch on the control network, where a SCADA system can be used to monitor and communicate with the device network.

NOTE:
Do not connect the service ports on different CPUs together through the control network.

- If transparency is needed between a device network and the control network, make the connection with a router as shown in the following figure.
- If transparency is not needed, contact the Schneider Electric PlantStruxure Competency Center for information



1 CPU managing the device network
2 RIO drop on the device network

3 DRS on the device network connecting (5) to the main ring
4 DIO cloud on the device network
5 DIO sub-ring on the device network
6 connection between the device network and the control network
7 control network

SD Memory Card for BME P58 *xxxx* CPUs

BMXRMS004GPF SD Memory Card

The SD memory card is an option that can be used for application and data storage. The SD memory card slot in the CPU housing is behind a door *(see page 34)*.

Use a BMXRMS004GPF memory card in your M580 CPU. It is a 4 GB, class A card rated for industrial use. Other memory cards, such as those used in the M340 CPUs, are not compatible with the M580 CPUs.

NOTE:
If you insert an incompatible SD memory card in the CPU:
- The CPU remains in NO_CONF state.
- The CPU **BACKUP** LED turns ON.
- The memory card access LED remains OFF.

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

Memory Card Characteristics

Global memory size	4 GB
Application backup size	64 MB
Data storage size	3.93 GB
Write/erase cycles (typical)	100,000
Operating temperature range	–40...+85 °C (–40...+185 °F)
File retention time	10 years
Memory zone for FTP access	data storage directory only

Formatting the Memory Card

The formatting procedure is described in *Formatting the Memory Card* topic in the *Unity Pro System Block Library* manual.

Memory Card Access LED

Introduction

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

Dedicated LED Meanings

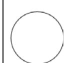
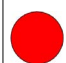
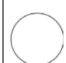
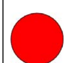
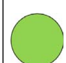
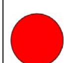
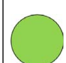
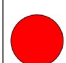
By itself, the **memory card access** LEDs have the following meanings:

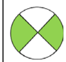
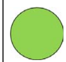
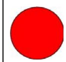
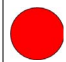
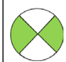
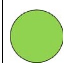
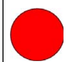
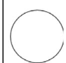
LED Status	Description
ON	The memory card is recognized, but the CPU is not accessing it.
blinking	The CPU is accessing the memory card.
OFF	The memory card can be removed from the CPU slot or the CPU does not recognize the memory card.

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

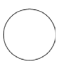
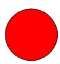
Combined LED Meanings

The LED also illuminates together with the **BACKUP** LED *(see page 35)*. Their combined patterns indicate the following diagnostic information:

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

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

The following legend shows the different LED patterns:

Symbol	Meaning	Symbol	Meaning
	off		steady red
	steady green		blinking green

Firmware Upgrade

Introduction

You can upgrade the CPU firmware by downloading a new firmware version with Unity Loader.

The firmware download can be performed by connecting to either of the following:
- CPU mini-B USB connector *(see page 38)*
- CPU **Service** port *(see page 42)*
- Ethernet network

Refer to the Unity Loader manual for a description of the download procedure (see *Unity Loader, a SoCollaborative software User Manual*).

Enabling CPU Firmware Upgrade

To enable the firmware upgrade, check the CPU security settings *(see page 156)*.

Firmware File

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

Upgrade Procedure

Follow these steps to upgrade the CPU and BME XBP ••00 rack firmware:

Step	Action
1	Install Unity Loader software provided with Unity Pro.
2	Connect the PC that is running Unity Loader to the CPU.
3	Launch Unity Loader.
4	Click **Firmware** tab.
5	In the **PC** list box, select the *.ldx* file that contains the firmware file.
6	When connected with Ethernet, check that the MAC address indicated in the **PLC** box corresponds to the MAC address marked on the CPU.
7	Check that transfer sign is green to allow transfer from PC to CPU.
8	Click **Transfer**.
9	Click **Close**.

(Hardened) Equipment

Introduction

Hardened equipment is the ruggedized version of standard equipment that can operate in extended temperature ranges and in dirty or corrosive environments. There are hardened versions of several of the CPUs, backplanes, and power supplies, as well as other components, in the M580 system.

Extended Temperature Considerations

The standard temperature range for M580 equipment is 0...60 °C (32...140 °F). Hardened equipment can operate at extended temperature range: –25...70 °C (–13...158 °F).

When used in the standard temperature range, hardened equipment has the same performance characteristics as the standard equipment. However, at the higher and lower ends of the extended temperature range (lower than 0 °C (32 °F) or higher than 60 °C (140 °F)), the hardened power supplies can have reduced power ratings *(see page 83)* that affect power calculations.

If hardened equipment is operated above or below the extended temperature limits (lower than –25 °C (–13 °F) or higher than 70 °C (158 °F)), the equipment can operate abnormally.

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION
Do not operate M580 equipment outside of its specified temperature range.
Failure to follow these instructions can result in death, serious injury, or equipment damage.

Operating in Harsh Environments

Hardened equipment has a conformal coating applied to its electronic boards. When associated with appropriate installation and maintenance, this treatment allows it to be more robust in harsh chemical environments.

Conformal coating increases the isolation capability of the circuit boards and their resistance to:
- condensation
- dusty atmospheres (conducting foreign particles)
- chemical corrosion, in sulphurous atmospheres (for example, in oil refineries or purification plants) or in atmospheres that contain halogens such as chlorine.

Hardened M580 CPU, Power Supply, and Backplane Equipment

The following hardened equipment is available:

Component	Reference
CPUs	BME P58 1020 H
	BME P58 2020 H
	BME P58 2040 H
backplanes	BME XBP 0400 H
	BME XBP 0800 H
	BME XBP 1200 H
backplane extension	BMX XBE 1000 H
power supplies	BMX CPS 3020 H
	BMX CPS 3500 H

For a list of additional M580 hardened equipment, refer to the *Modicon M580 System Planning Guide*.

Chapter 2
M580 Racks

Introduction

This chapter describes local racks and rack extender modules.

What Is in This Chapter?

This chapter contains the following sections:

Section 2.1
BME XBP *xxxx* Rack Description

Introduction

This section describes the main local racks and the extended local racks that can be used in M580 systems.

What Is in This Section?

This section contains the following topics:

Local and Remote Racks

Introduction

A module is a system component that is installed in a rack and that communicates across a bus built into the backplane of that rack. The M580 PAC is a modular system that includes a CPU, power supplies, and I/O and communication modules. The PAC also has the ability to manage distributed equipment that resides off the racks, but this equipment is optional.

Local Rack

A BME P58 •••• CPU is a module that resides in the local rack. The local rack is located at the head of the M580 PAC network. Every PAC system is managed by 1 and only 1 local rack. Also present in the local rack (and in all racks) is a power supply module *(see page 80)*.

Other modules, such as communication adapters and local X80 I/O modules, may also be present in the local rack. The presence of these other modules is optional. The presence of a CPU and a power supply is necessary in the local rack for the system to function.

This user guide focuses primarily on the local rack, where the CPU resides.

Remote Racks

If you are using an M580 CPU that provides the RIO scanner service, you may have up to 16 remote drops of X80 I/O modules *(see page 21)*. Each remote drop contains a main remote rack. In that main remote rack reside a power supply module, a BM• CRA 312 00 RIO adapter module, and the X80 I/O modules you have chosen for that drop.

For detailed information on the BM• CRA 312 00 adapters and the installation of a remote drop, refer to the *Modicon M580 Remote I/O Installation and Configuration Guide*.

Choosing an Ethernet or an X Bus Rack

One key role of a rack is to provide a communication bus for the modules in the local rack or remote drop. The Modicon M580 PAC uses 2 types of backplanes, Ethernet and X Bus. The X Bus connection is present on all M580 racks. A subset of the Modicon M580 racks contains an additional Ethernet backplane.

Ethernet is used across the backplane for:
- eX80 I/O modules, which require an Ethernet bus on the rack in order to exchange data (for example, X80 HART modules)
- third-party (PME) modules that require Ethernet
- Ethernet communication modules interlink to the CPU

For any of these cases, use an Ethernet rack. In other cases, an X-bus rack is allowed. If you use an X Bus rack for any of the cases above, the Ethernet capabilities of the modules will not work and they will not perform as expected.

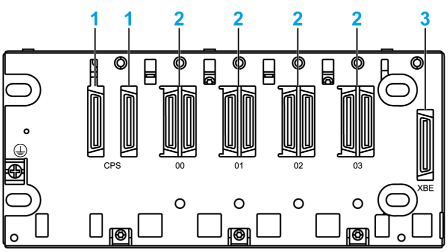
Ethernet Racks

The M580 Ethernet racks have all the features of the X Bus racks with the addition of an Ethernet communication bus across the backplane.

Ethernet (BME XBP) Rack Reference	Number of Module Slots
0400/0400 H	4
0800/0800 H	8
1200/1200 H	12 (1)
1 8 slots with X Bus and Ethernet connectors + 4 slots (slots number **02**, **08**,**10**, **11**) with X Bus connector only	

All 3 Ethernet racks are available as standard or industrially hardened modules *(see page 49)*. A hardened module has the letter **H** appended to the reference.

Here is a BME XBP 0400 (4-slot rack). The module slots in this rack contain 2 bus connectors per slot, one X Bus connector and one Ethernet bus connector:



1 power supply connectors
2 Ethernet and X Bus connectors
3 extender module connector

Any of these Ethernet racks can be used as a local or remote rack. Ethernet racks cannot be used as extended racks *(see page 59)*. Only the X Bus can be extended within the local rack or in a remote drop.

X Bus Racks

NOTE: The X Bus racks have the same commercial references as the racks that support the M340 PAC system. When these racks are used in the M580 system, you must use version PV: 02 or later. Earlier versions will work with M340 CPUs but not with M580 CPUs.

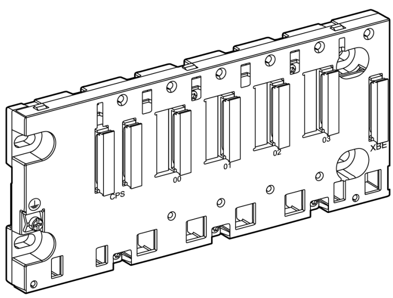
Each rack includes 1 slot with 2 connectors on the left side reserved for the power supply module. The slots that follow can be used for modules. The connector on the right can only be used to extend the rack. Racks are available with 4, 6, 8, and 12 module slots:

X Bus (BMX XBP) Rack Reference	Version	Number of Module Slots
0400/0400 H	PV:02 or later	4
0600/0600 H		6
0800/0800 H		8
1200/1200 H		12

The BMX XBP •••• (PV:02 or later) racks are available as standard or industrially hardened modules *(see page 49)*. A hardened module has the letter **H** appended to the reference.

Any of these X Bus racks can be used as a local or remote rack. They may be used as the main rack or as an extended rack.

Here is a BMX XBP 0400 (4-slot) rack. The 2 leftmost connectors are for the power supply, and the 4 module slots that follow have only one bus connector per slot. That connector is for X Bus. No Ethernet bus connectors are present.

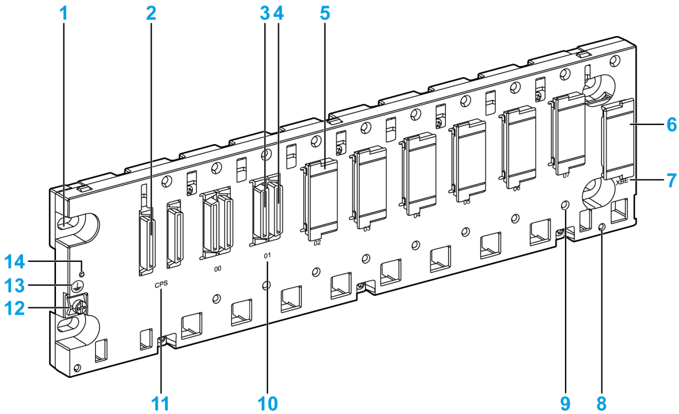


X80 Rack Characteristics

Front View

A BME XBP 0800 rack has eight X80 module slots, and each slot has both an Ethernet bus connector and an X Bus connector (items 3 and 4).

Example of BME XBP 0800 rack:



1 panel mounting hole
2 power supply module connectors
3 Ethernet connector
4 X Bus connector
5 protective cap (connectors protection against moist and dust)
6 40-pin female connector for a rack extender module
7 XBE marking for a rack extender module
8 shielding bar screw hole
9 keying hole for Ethernet module
10 marking for module location number
11 CPS marking for the power supply
12 protective ground screw
13 protective ground marking
14 rack status LED (**OK**)

Power Supply Slot

The leftmost slot, where the power supply connects (item 2), is labeled **CPS**. The power supply slot contains 2 connectors. On all racks, regardless of whether they are in a local rack or remote drop, a power supply module is needed. This slot is reserved for the power supply, and no other module types can be installed here.

Module Slots

The module slots, which are to the right of the **CPS** slot, are labeled numerically starting at **00**. For the 8-slot rack shown above, the module slots are labeled **00** through **07**.

In the main local rack, slot **00** (the first module slot after the power supply) is where the CPU is installed. In the main rack of a remote drop, slot **00** is where the (e)X80 EIO adapter module is installed. The remaining slots can be used for X80 I/O or communication modules. The number of module slots, and the presence or absence of an Ethernet connector at each module slot, depends on the rack reference you select *(see page 53)*.

Ethernet Connectors

An Ethernet communication bus is embedded in the backplane of the BME XBP *xxxx* racks.

Ethernet Rack Status LED

The green rack status LED marked **OK** is present on Ethernet racks but not on X Bus racks. The LED indicates if the rack is working properly.

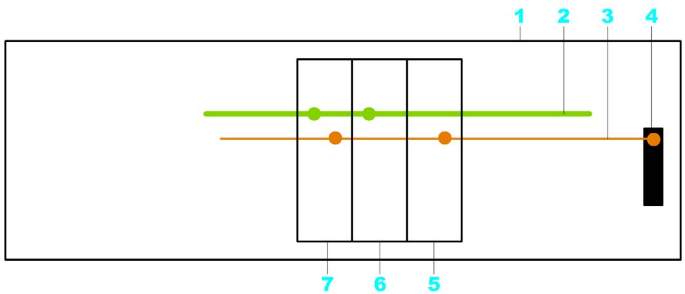
When this LED is ON, the following conditions internal to the rack have been fulfilled:
- The power rail voltages are in the rated range.
- The CPU watchdog is working properly.
- The Ethernet switch diagnostic is working properly.

When the LED is OFF, the backplane is not operational.

X Bus Connectors

All M580 racks have an X Bus connector at every module slot. Many X80 I/O modules need only X Bus to support communication across the backplane.

The following illustration shows the bus connection to the extender connector on the right side of a BME XBP ••00 rack:

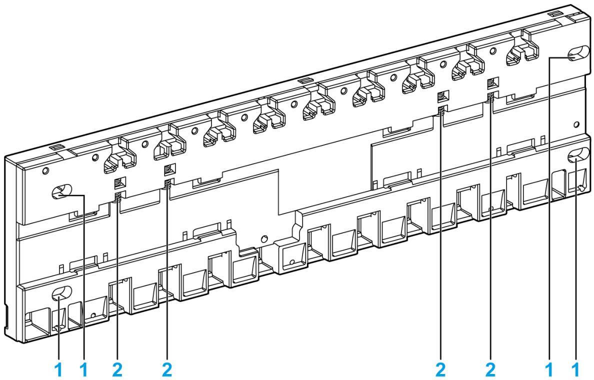


1 rack
2 Ethernet communication bus on the backplane
3 X Bus communication bus on the backplane
4 X Bus extender connector
5 Modicon X80 module

6 Ethernet only module
7 module with Ethernet and X Bus connectors

Rear View

The rear panel of an 8-slot rack, showing the mounting slots:



1 panel-mounting hole
2 spring for DIN-rail mounting

Most M580 racks may be mounted on:

- the wall of an enclosure
- a 35 mm (1.38 in) DIN rail
- Telequick mounting grids

The 12-slot (BME XBP 1200 (H) rack does not have springs like the ones shown previously (item 2). These racks cannot be mounted on a DIN rail.

Extender Racks

Overview

You may extend the number of racks in the local configuration in order to:
- increase the number of modules
- extend the area covered by the rack so that I/O modules can be installed closer to the different machines they are controlling
- include Premium I/O modules in the local rack

Only the X Bus is extended; install all eX80 modules, which require Ethernet on the backplane of the rack, in the main rack. They do not operate in extended racks.

NOTE: Depending on the type of RIO adapter you are using, you may also add an extended rack to a remote drop. Premium I/O modules are not permitted in a remote drop.
The focus of this discussion is on local racks. For more information on extended racks in remote drops, refer to the *Modicon M580 Remote I/O Modules Installation and Configuration Guide*.

Maximum Number of Racks in the Local Extension

The number of extended racks allowed in the local rack depends on the CPU you are using:
- The BME P58 1020, BME P58 2020, and BME P58 2040 CPUs support a main local rack and up to 3 extension racks. If you are using 4-, 6-, or 8-slot Premium extension racks, you may install 2 physical racks at each assigned rack address, allowing up to 6 Premium extension racks.
- The BME P58 3020, BME P58 3040, BME P58 4020, and BME P58 4040 CPUs support a main local rack with up to 7 extension racks. If you are using 4-, 6-, or 8-slot Premium extension racks, you may install 2 physical racks at each assigned rack address, allowing up to 14 Premium extension racks.

NOTE: When you use a 12-slot Premium extension rack, you may install only 1 rack at a rack address.

NOTE: When combining X80 and Premium extension racks, the X80 extension racks are chained after the main local rack and the Premium extension racks are chained last.

Assigning Rack Addresses

Assign each rack in an extension an address that is unique with respect to all other racks in the extension.
- To assign a rack address to an X80 rack, use the microswitches on the BMX XBE 1000 rack extender module *(see page 62)*, which is installed in each X80 extender rack.
- To assign a rack address to a Premium extension rack, use the microswitches on the left side of the Premium rack *(see page 69)*. Premium extension racks are connected together directly by cable and do not use a rack extender module.

The main local rack, where the CPU resides, is always rack address 00. The other racks in the extension can be assigned rack addresses in the range 01 through 07.

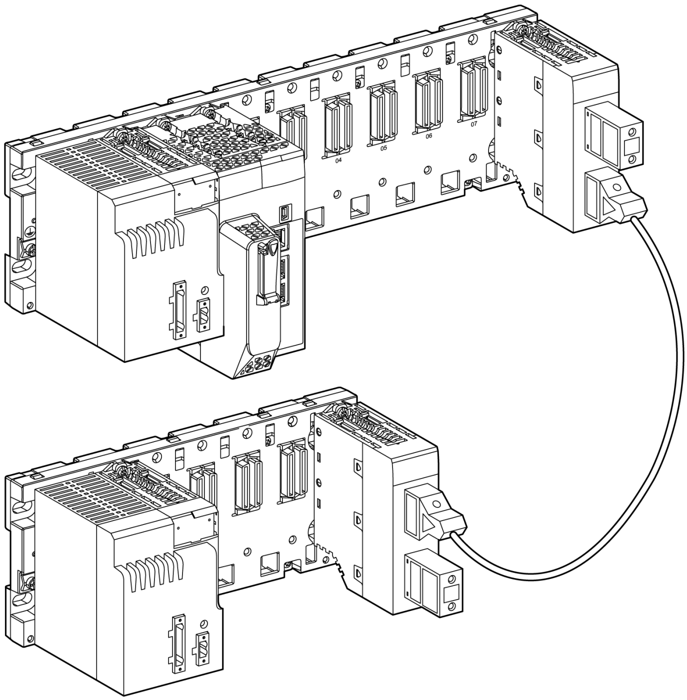
NOTE: With some Premium extension racks, you are able to install 2 physical racks at 1 rack address. To distinguish between the 2 physical racks at the same rack address, set microswitch 4 on the 2 racks to different positions, one ON and the other OFF.

Distance Between Extended Rack and the Main Rack

The maximum distance that an X80 extension rack can be from the main rack is 30 m. The maximum distance that a Premium extension rack can be from the main rack is 100 m.

Example of Topology

The followinf is an example of a main local rack with 1 extended local rack:



NOTE:
- Each rack has a power supply and a BMX XBE 1000 extender module.
- An extender cable (in this case a BMX XBC •••K cable) connects the 2 extender modules.
- The unused ports on the 2 extender modules are terminated, with a TSX line terminator on the main rack and TLY line terminator on the extender rack.

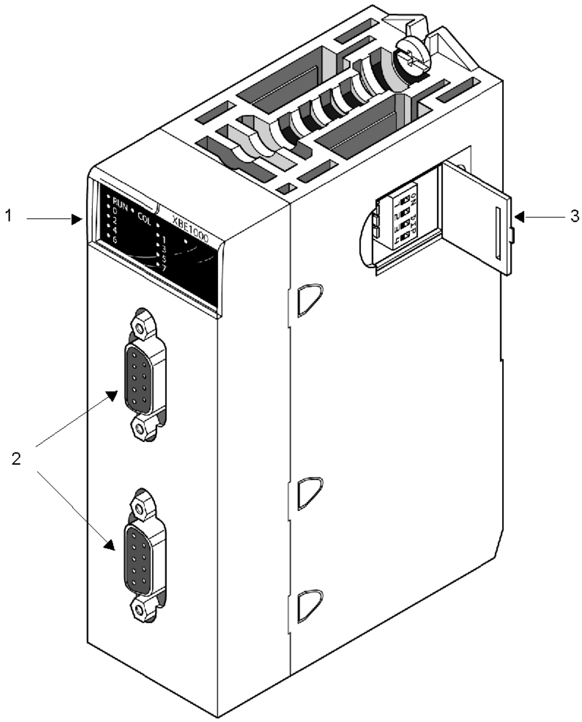
Module Consumption

- Consumption on 3.3 Vdc power supply: 22 mA
- Dissipated power on the 3.3 Vdc rack power supply: 73 mW
- Consumption on 24 Vdc rack power supply: 160 mA
- Dissipated power on the 24 Vdc rack power supply: 3.84 W

X80 Rack Extender Module

Physical Description

A BMX XBE 1000 rack extender module contains an LED diagnostic panel, a pair of connectors for the X Bus extender cables, and a set of switches for addressing the X80 extender racks.



1 rack extender status LEDs
2 female 9-pin SUB-D connectors for bus cables
3 rack address switches

Rack Address Switches

Assign an address to each X80 rack that is unique with respect to all other racks in the extension. Use the 4 microswitches on the side of the rack extender module to set each rack address.

In a local rack, as many as 7 extender racks of X80 may be added *(see page 59)*.

Switch	Rack Address							
	0	1	2	3	4	5	6	7
1	OFF	OFF	OFF	OFF	ON	ON	ON	ON

Switch	Rack Address							
	0	1	2	3	4	5	6	7
2	OFF	OFF	ON	ON	OFF	OFF	ON	ON
3	OFF	ON	OFF	ON	OFF	ON	OFF	ON
4	Not used							

By default, the rack extender module is shipped with the switches set to address **0** (all switches **OFF**). Address **0** is reserved for only the main local rack (which contains the CPU). The remaining X80 racks in an extension may be assigned addresses **1** through **7** in any order or sequence. The key requirement when you set the X80 rack addresses is that each address is unique with respect to all other rack addresses in the extension.

NOTE:
You create a condition called a *collision* if you assign:
- the same rack address to more than one X80 rack in an extension
- address **0** to any rack other than the main local rack

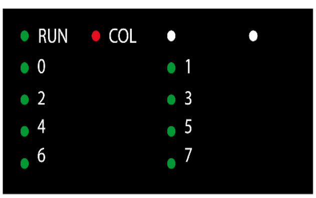
When a collision happens, one of the racks with the duplicate rack address will not operate.

To recover from a collision:

Step	Action
1	Turn OFF the power supplies in the racks that have the address mismatch.
2	Using the address switches on the rack extender module, set unique and correct rack addresses.
3	Reapply power to the racks.

Rack Extender LEDs

The LEDs on the rack extender module provide status information about the rack in which it resides:

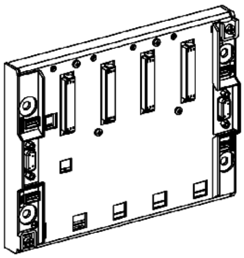
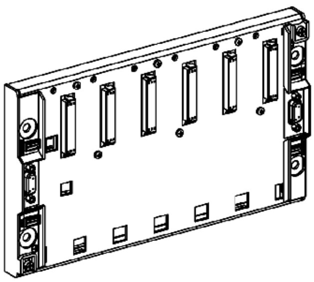
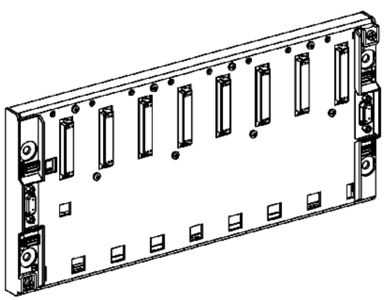


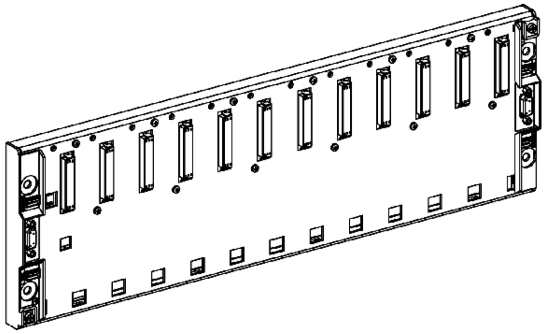
LED	Pattern	Indication
RUN (green)	ON	Module functioning normally.
	OFF	• The power supply is no longer present. • An error has been detected in the extender module.
COL (red)	ON	Rack address collision detected. • Two or more racks have been assigned the same rack address. • A rack that does not contain the CPU has been assigned address **0**.
	OFF	Each rack in the extension has a unique address.
0 to **7** (green):	ON or OFF	Rack address. Each extender module has only one address LED ON. Each extender module in the rack should have a different address LED ON.

Premium TSX RKY Extendable Racks

Premium Extendable Rack

If you install Premium equipment, use one of these 4 Premium extendable racks:

Designation	Illustration
TSX RKY 4EX	4-slot rack 
TSX RKY 6EX	6-slot rack 
TSX RKY 8EX	8-slot rack 

Designation	Illustration
TSX RKY 12EX	12-slot rack 

NOTE: Only Premium TSX RKY ••EX(C) racks are allowed. The TSX RKY ••E(C) racks are not compatible.

Premium Extendable Rack Characteristics

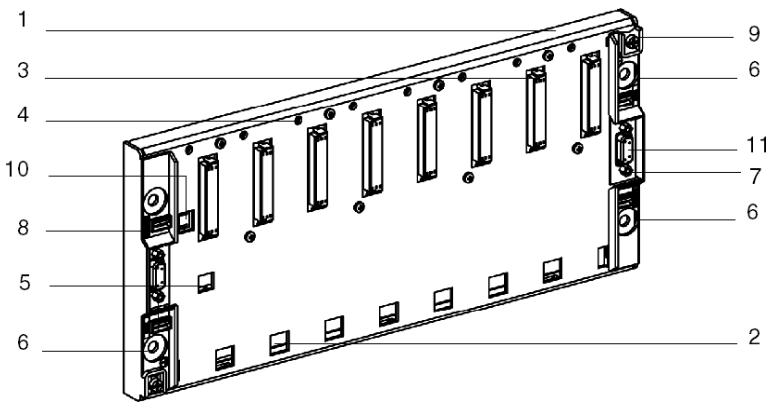
Overview

There are 2 types of Premium racks: standard and extendable racks. In the M580 system, only extendable racks are used.

Two elements distinguish an extendable rack: the microswitches on the left side of the rack (item 10 in the following figure), and the SUB-D connectors on the right side of the rack (item 11 in the following figure).

Front View

The following is an example of a TSX RKY 8EX extendable rack, which has 1 slot reserved for a power supply and 7 module slots.



1 metal frame to support the X bus backplane, support the modules, and provide rack rigidity
2 anchor-point holes for module pins
3 female 48-pin 1/2 DIN connectors for installing a module on the rack
4 holes for the module-mounting screws
5 guide hole for mounting the power supply
6 M6 screw holes for mounting the rack
7 slot for the rack address label
8 slot for the network address label
9 ground terminals for the rack
10 microswitch for setting the rack address
11 female 9-pin SUB-D connectors for extending X Bus to another rack

Slot Assignments

When racks are delivered, the module connectors for each slot have protective covers. Remove the covers before installing the modules.

The leftmost slot is reserved for the power supply. The slot is marked PS. Power supply modules have a projecting part on the back so that they cannot be mounted in any other position. The remaining slots are for all other Premium modules, and they are labeled from left to right starting with 00. In the 8-slot example above, the remaining slots are labeled 00 through 06.

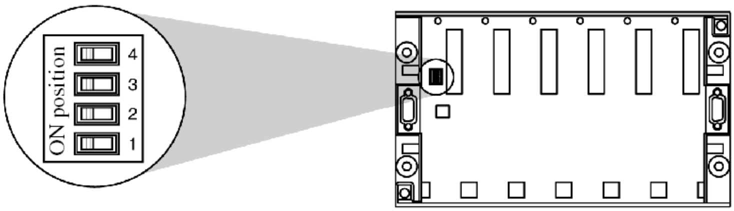
Addressing Premium Extendable Racks

Introduction

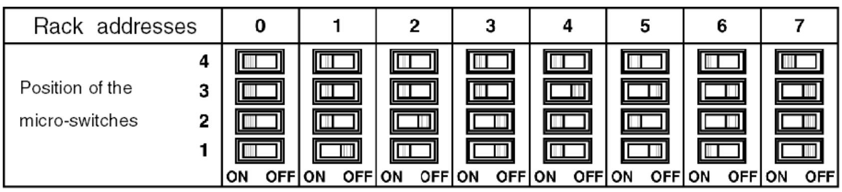
Depending on the type of M580 CPU you use, you may have a total of either 4 or 8 racks in a local rack extension.

Rack Address Microswitches

Assign each rack in the extension an address that is unique with respect to all other racks in the extension. Set the address for a Premium rack with the 4 microswitches on the left side of the rack.



Use microswitches **1** to **3** to assign the address of the rack. Use microswitch **4** to distinguish 2 racks at the same address.



NOTE: On delivery, microswitches 1, 2 and 3 are in the ON position, indicating rack address **00**. Address **00** is reserved for the main local rack, which is always an X80 rack. In all cases, you will need to change the address of a Premium extendable rack from the default setting before inserting modules in the rack.

NOTE: Set the rack address switches before mounting the power supply module.

⚠ CAUTION
RACK ADDRESS CONFLICT
Assign a unique address to each rack in the range **00** through **07**.
Reset power after setting the rack addresses.
Failure to follow these instructions can result in injury or equipment damage.

Assigning Addresses to Different Racks

Address 0: Reserved for the main local rack, which is always an X80 rack.

Addresses 1 to 7: They can be assigned in any order to the extendable racks in the station.

NOTE: The rack address coding is done before applying the power supply.

If 2 or more racks are at address 0, the rack supporting the CPU does not indicate a duplicate address.

After you have readdressed your racks to remove duplication, cycle power on the affected racks.

Rack Extender Cables and Terminators

Extender Cable BMX XBC *xxx*K and TSX CBY *xxx*K

Extender cables are available in various lengths. Different types of cable are used to extend X80 I/O racks and Premium I/O racks.

NOTE: You can use Premium extender racks only in a local rack, not in a remote drop.

Cable Reference		Length
Modicon X80	BMX XBC 008K	0.8 m (2.62 ft)
	BMX XBC 015K	1.5 m (4.92 ft)
	BMX XBC 030K	3 m (9.84 ft)
	BMX XBC 050K	5 m (16.4 ft)
	BMX XBC 120K	12 m (39.37 ft)
Premium	TSX CBY 010K	1 m (3.28 ft)
	TSX CBY 030K	3 m (9.84 ft)
	TSX CBY 050K	5 m (16.4 ft)
	TSX CBY 120K	12 m (39.37 ft)
	TSX CBY 180K	18 m (59.05 ft)
	TSX CBY 280K	28 m (91.86 ft)
	TSX CBY 380K	38 m (124.67 ft)
	TSX CBY 500K	50 m (164.04 ft)
	TSX CBY 720K	72 m (236.22 ft)
	TSX CBY 1000K	100 m (328.08 ft)
NOTE: If you install TSX CBY •••K cables, only use PV 03 or later.		

DANGER

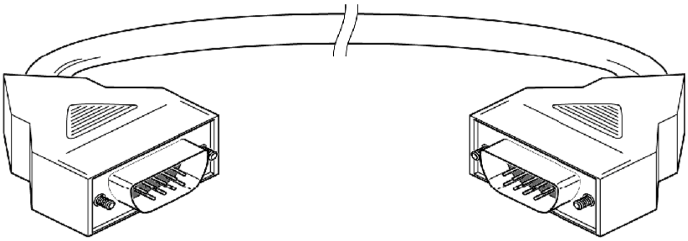
HAZARD OF ELECTRIC SHOCK

Remove power from all elements of the station (the local rack or remote drop) before inserting or extracting a BMX XBC •••K or a TSX CBY •••K cable.

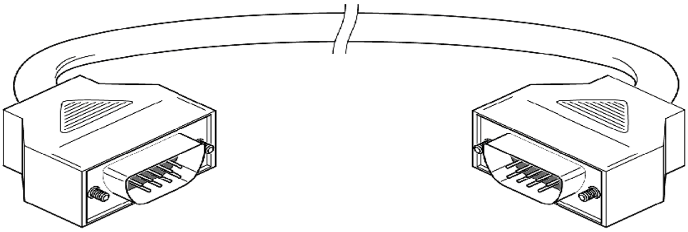
Failure to follow these instructions will result in death or serious injury.

Each cable has a male 9-pin SUB D connector that plugs on the 9-pin SUB D female connector of the rack extender modules.

Here is a BMX XBC •••K cable for an X80 I/O extender rack. It can be distinguished by its angled 45° connector.



Here is a TSX CBY •••K cable for a Premium extender rack:

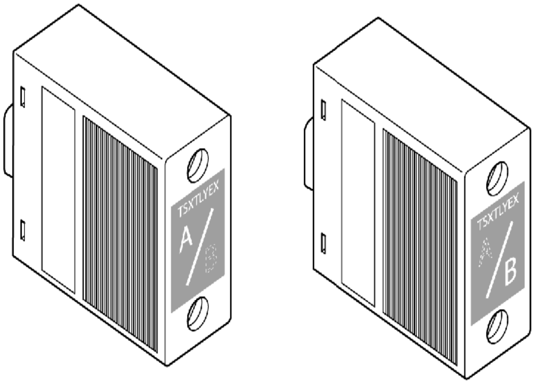


Line Terminators TSX TLY EX

Plug a line terminator at each end of the X Bus extension rack *(see page 118)*.

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION
Remove power from all elements of the station (the local rack or remote drop) before inserting or extracting a line terminator.
Failure to follow these instructions can result in death, serious injury, or equipment damage.

The following figure shows line terminators containing the adaptation components with a 9-pin SUB-D connector. They are plugged on the 9-pin SUB D connector of the extension module at each end of the extended X Bus.



TSX TLY EX line terminators are provided in pairs marked A/ and /B. An extended X Bus needs a terminator A/ at one end and a terminator /B at the other end.

Rack Firmware Upgrade

Introduction

You can upgrade the BME XBP ••00 rack firmware by downloading a new firmware version with Unity Loader through the CPU or a BME CRA 312 •0 (e)X80 adapter module.

Download the firmware by connecting to either of the following:

- the CPU mini-B USB connector *(see page 38)*
- the CPU **Service** port *(see page 42)*
- the Ethernet network

Refer to the CPU firmware upgrade *(see page 48)* procedure for a description of the download procedure.

Firmware

The firmware file is included in an **.ldx* file.

Troubleshooting

If the rack power supply is turned off during the upgrade procedure, the backplane firmware remains on the version embedded before the upgrade procedure.

Section 2.2
BME XBP *xxxx* Rack Characteristics

Introduction

This section presents the BME XBP ••00 rack performances, electrical characteristics, and dimensions.

What Is in This Section?

This section contains the following topics:

Electrical Characteristics

Introduction

The rack delivers 24 Vdc and 3.3 Vdc to supply the backplane and connected modules.

Backplane Power Consumption

Power consumption of the rack backplanes:

Rack Type	Backplane Average Current Consumption	
	3.3 Vdc Supply Power	24 Vdc Supply Power
BME XBP 0400 (H)	49 mA (162 mW)	118 mA (2.8 W)
BME XBP 0800 (H)	64 mA (211 mW)	164 mA (3.9 W)
BME XBP 1200 (H)	86 mA (283 mW)	164 mA (3.9 W)

Mean Time Between Failures

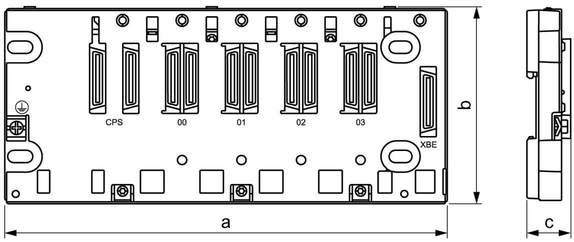
The rack MTBF is a component of the global system MTBF:

Rack Type	MTBF (Hours at 30 °C Continuous)
BME XBP 0400 (H)	2000000
BME XBP 0800 (H)	1700000
BME XBP 1200 (H)	1500000

Rack Dimensions

Rack Dimensions

The following illustration displays the overall dimensions of the BME XBP ••00 racks:

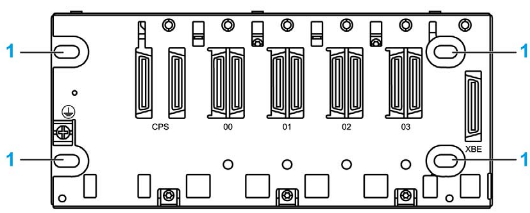


Dimensions of each BME XBP ••00 rack:

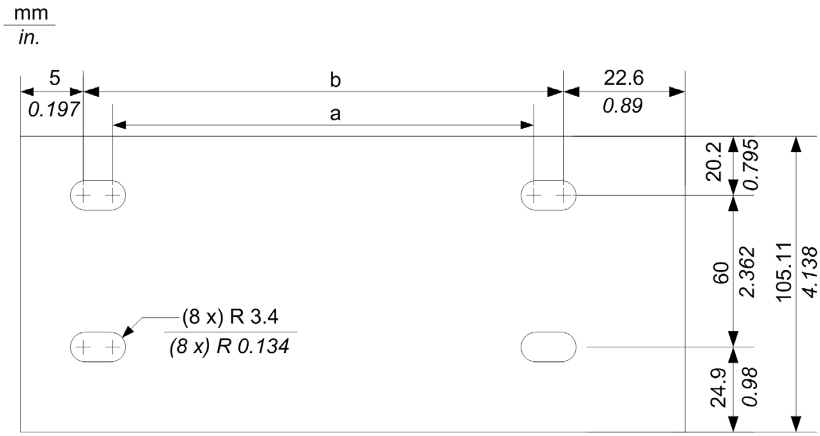
Rack Type	a		b	c
	Empty Rack	Rack With Extender Module Mounted		
BME XBP 0400 (H)	242.4 mm (9.543 in.)	243.58 mm (9.59 in.)	105.11 mm (4.138 in.)	19 mm (0.748 in.)
BME XBP 0800 (H)	372.8 mm (14.677 in.)	373.98 mm (14.724 in.)	**NOTE:** Overall height is 134.6 mm (5.299 in.) with a CPU mounted.	
BME XBP 1200 (H)	503.2 mm (19.811 in.)	504.38 mm (19.857 in.)		

Panel Fastening Holes Dimension and Location

Fastening holes are located at the 4 corners of a BME XBP ••00 rack.



1 Fastening holes



NOTE: You can use M4, M5, M6, or UNC #6 screws in the fastening holes.

Rack Type	a	b
BME XBP 0400 (H)	202.1 mm (7.957 in.)	214.8 mm (8.457 in.)
BME XBP 0800 (H)	332.5 mm (13.09 in.)	345.2 mm (13.59 in.)
BME XBP 1200 (H)	462.9 mm (18.224 in.)	475.6 mm (18.724 in.)

Chapter 3
M580-Compatible Power Supply Modules

Introduction

This chapter describes power supplies used to power the BME XBP ••00 racks.

What Is in This Chapter?

This chapter contains the following topics:

Power Supply Modules

Introduction

Main and extended local racks, as well as remote racks that contain X80 I/O modules, require one of the following power supply modules:

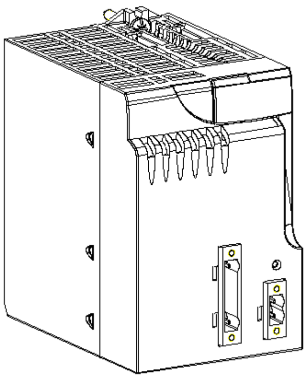
- BMX CPS 2000
- BMX CPS 2010
- BMX CPS 3020 and BMX CPS 3020 H
- BMX CPS 3500 and BMX CPS 3500 H
- BMX CPS 3540T

NOTE: The BMX CPS 3020 H, BMX CPS 3500 H, and BMX CPS 3540T are industrially hardened power supplies that can work at extended temperature ranges and in harsh environments *(see page 49)*.

The power supply you choose for each rack depends on the current requirements (alternating or direct) and the power consumption of the modules in the rack.

Illustration

The following illustration shows a BMX CPS •••• power supply module:



LED Display

Introduction

Power supply modules have a display panel with a green **OK** LED.

The BMX CPS 2000 and BMX CPS 3500 power supplies and the BMX CPS 3540T direct current power supply have an additional green **24 V** LED.

Indications

The power supply LEDs indicate the following diagnostic information:

LED	Status Indication
OK	• ON in normal operating mode • OFF when the rack power supply output voltage is below the threshold or when the RESET button is pressed
24 V	• ON in normal operating mode • OFF if the 24 Vdc sensor voltage supplied by the power supply is no longer present

Reset Button

Introduction

The power supply module has a **Reset** button on its front panel which, when pressed, triggers an initialization sequence of the modules on the rack that it supplies.

Pressing the Reset Button

When the **Reset** button is pressed, the following events occur:

- The INIT_BAC_N signal is enabled and forces all the modules on the rack to reset to 0.
- The RESET_BUTTON signal is enabled and forces the processor to reset to 0, which in turn leads to the following events:
  - The ALARM relay is forced to open state.
  - The power supply **OK** LED is switched off.

Pressing/releasing the **Reset** button triggers a cold start. The connectors around the **Reset** button are energized.

DANGER

HAZARD OF ELECTRIC SHOCK

- Do not touch the **Reset** button directly.
- Use an insulated tool to press the **Reset** button.

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

Usable Power

Introduction

When the power necessary for a rack has been calculated, the information in this section is used to select the appropriate power supply module to be installed on the rack.

Usable Power

The following table shows the power supply module usable power in the temperature range 0...60 °C (32...140 °F).

Power	BMX CPS 2000	BMX CPS 2010	BMX CPS 3020	BMX CPS 3500	BMX CPS 3540 T
total usable power (all outputs included)	20 W	17 W	32 W	36 W	36 W
usable power at the 3V3_BAC output	8.3 W (2.5 A)	8.3 W (2.5 A)	15 W (4.5 A)	15 W (4.5 A)	15 W (4.5 A)
usable power at the 24V_BAC output	16.5 W (0.7 A)	16.5 W (0.7 A)	31.2 W (1.3 A)	31.2 W (1.3 A)	31.2 W (1.3 A)
usable power at the 3V3_BAC and 24V_BAC outputs	16.5 W	16.5 W	31.2 W	31.2 W	31.2 W
usable power at the 24V_SENSORS output	10.8 W (0.45 A)	-	-	21.6 W (0.9 A)	21.6 W (0.9 A)

The power supply modules operate in an extended temperature range of -25...0 °C (-13...32 °F) and 60...70 °C (140...158 °F). The following table shows how power is derated when operation is in the extended ranges.

Power	BMX CPS 3020 H	BMX CPS 3500 H	BMX CPS 3540 T
total usable power (all outputs included)	24 W	27 W	27 W
usable power at the 3V3_BAC output	11.25 W (3.375 A)	11.25 W (3.375 A)	11.25 W (3.375 A)
usable power at the 24V_BAC output	23.4 W (0.975 A)	23.4 W (0.975 A)	23.4 W (0.975 A)
usable power at the 3V3_BAC and 24V_BAC outputs	23.4 W	23.4 W	23.4 W
usable power at the 24V_SENSORS output	-	16.2 W (0.5 A)	16.2 W (0.5 A)

NOTE: The 24V_SENSORS output is the 24 Vdc sensor power supply output and is only available on the **BMX CPS 2000/3500/3500 H/3540 T** modules.

Excessive load can cause the power supply to trip off

⚠ WARNING

UNEXPECTED EQUIPMENT OPERATION - POWER DEMAND

Do not exceed the **BMX CPS 3500 H** and **BMX CPS 3540 T** 24V_SENSORS output power rating.

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

Power Limits

Excessive load can cause the power supply to trip off.

⚠ WARNING

UNEXPECTED EQUIPMENT OPERATION - POWER DEMAND

Do not exceed the total useful power rating of the module. Use the rules below to determine the maximum power supplied to outputs.

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

When establishing the power used by the **BMX CPS 2000/3500/3500 H/3540 T** modules, follow these rules:

- Do not let the sum of the power absorbed on the 3V3_BAC, 24V_BAC, and 24V_SENSORS outputs exceed the maximum usable power of the module.
- Do not let the sum of the power absorbed on the 3V3_BAC and 24V_BAC outputs exceed the sum of their usable power.

When establishing the power used by the **BMX CPS 2010/3020/3020 H** modules:

- Do not let the sum of the power absorbed on the 3V3_BAC and 24V_BAC outputs exceed the maximum usable power of the module.

Module Power Consumption

Introduction

The power necessary for a rack depends on the type of modules installed. Calculate the global power consumption to define the power supply module to be installed on the rack.

NOTE: Unity Pro software can display the power consumption budget for a given configuration. To reach this functionality, refer to the *Consumption Management* section of *Unity Pro, Operating Modes* user guide.

The following tables give the average power consumption per module. The average value is calculated regarding the maximum and typical consumptions.

CPU Power Calculation Tables

The following tables explain how to define the global power consumption on a rack. Refer to the module current consumption table *(see page 86)*, and rack and extender module consumption table *(see page 88)* to define the total current consumed for each voltage source of the power supply.

Method to establish a power calculation for modules without **24V_Sensor** power available:

Power	Calculation	Result
power necessary on the 3.3 V rack output (P 3.3 V rack)	(current absorbed on the 3V3_BAC output by all modules (mA)) x 10^{-3} x 3.3	=...............W
power necessary on the 24 V rack output (P 24 V rack)	(current absorbed on the 24V_BAC output by all modules (mA)) x 10^{-3} x 24	=...............W
total power necessary	(P 3.3 V rack) + (P 24 V rack)	=...............W

Method to establish a power calculation for modules with **24V_Sensor** power available:

Power	Calculation	Result
power necessary on the 3.3 V rack output (P 3.3 V rack)	(current absorbed on the 3V3_BAC output by all modules (mA)) x 10^{-3} x 3.3	=...............W
power necessary on the 24 V rack output (P 24 V rack)	(current absorbed on the 24V_BAC output by all modules (mA)) x 10^{-3} x 24	=...............W
power necessary on the 24 V sensor output (P 24 V sensors)	(current absorbed on the 24V_Sensors output by all modules (mA)) x 10^{-3} x 24	=...............W
total power necessary	(P 3.3 V rack) + (P 24 V rack) + (P 24 V sensors)	=...............W

Module Current Consumption

Average current consumption for each module:

Module Type	Module		Average Current Consumption (mA)		
	Reference	Description	3,3V_BAC Output	24VR_BAC Output	24V_SENSORS Output
CPU	BME P58 10•0		–	270	–
	BME P58 20•0		–	270	–
	BME P58 30•0		–	295	–
	BME P58 40•0		–	295	–
analog	BMX AMI 0410	4 isolated high-speed analog inputs	150	45	–
	BMX AMI 0800	8 non-isolated high-speed analog inputs	150	41	–
	BMX AMI 0810	8 isolated high-speed analog inputs	150	54	–
	BMX AMM 0600	4 channel analog inputs	240	–	120
	BMX AMO 0210	2 isolated analog outputs	150	110	–
	BMX AMO 0410	4 isolated high-speed analog outputs	150	140	–
	BMX AMO 0802	8 non-isolated high-speed analog outputs	150	135	–
	BMX ART 0414	4 isolated analog inputs	150	40	–
	BMX ART 0814	8 isolated analog inputs	220	50	–
communication	BMX NOE 0100	Ethernet 1 port 10/100 RJ45	–	90	–
	BMX NOE 0110	Ethernet 1 port 10/100 RJ45	–	90	–
counting	BMX EHC 0200	2 channel high speed counter	200	40	80
	BMX EHC 0800	8 channel high speed counter	200	–	80

Module Type	Module		Average Current Consumption (mA)		
	Reference	Description	3,3V_BAC Output	24VR_BAC Output	24V_SENSORS Output
discrete inputs	BMX DAI 0805	8 discrete 200...240 Vac inputs	103	13	–
	BMX DAI 1602	16 discrete 24Vac/24Vdc inputs	90	–	60
	BMX DAI 1603	16 discrete 48 Vac inputs	90	–	60
	BMX DAI 1604	16 discrete 100...120 Vac inputs	90	–	–
	BMX DDI 1602	16 discrete 24 Vdc inputs	90	–	60
	BMX DDI 1603	16 discrete 48 Vdc inputs	75	–	135
	BMX DDI 1604T	16 discrete 125 Vdc inputs	75	–	135
	BMX DDI 3202 K	32 discrete 24 Vdc inputs	140	–	110
	BMX DDI 6402 K	64 discrete 24 Vdc inputs	200	–	110
discrete outputs	BMX DAO 1605	16 discrete outputs	100	95	–
	BMX DDO 1602	16 discrete 0.5 A outputs	100	–	–
	BMX DDO 1612	16 discrete outputs	100	–	–
	BMX DDO 3202 K	32 discrete 0.1 A outputs	150	–	–
	BMX DDO 6402 K	64 discrete 0.1 A outputs	240	–	–
	BMX DRA 0804T	8 discrete isolated outputs	100	110	–
	BMX DRA 0805	8 discrete isolated outputs	100	55	–
	BMX DRA 1605	16 discrete outputs	100	95	–

Module Type	Module		Average Current Consumption (mA)		
	Reference	Description	3,3V_BAC Output	24VR_BAC Output	24V_SENSORS Output
discrete inputs/outputs	BMX DDM 16022	8 discrete 24 Vdc inputs and 8 discrete outputs	100	–	30
	BMX DDM 16025	8 discrete 24 Vdc inputs and 8 discrete outputs	100	50	30
	BMX DDM 3202 K	16 discrete 24 Vdc inputs and 16 discrete outputs	150	–	55
motion	BMX MSP 0200	2 independent Pulse Train Output channels	200	150	–

Rack and Extender Module Consumption

Average current consumption for each rack

Family	Rack Reference	Average Current Consumption (mA)	
		3,3V_BAC Output	24V_BAC Output
BMX XBP •••• (PV:02 or later) rack	BMX XBP 0400 (H)	340	–
	BMX XBP 0600 (H)	510	–
	BMX XBP 0800 (H)	670	–
	BMX XBP 1200	50	–
	BMX XBP 1200 (H)	250	–
BME XBP ••00 rack	BME XBP 0400 (H)	49	118
	BME XBP 0800 (H)	64	164
	BME XBP 1200 (H)	86	164
rack extender module	BMX XBE 1000	22	160

Chapter 4
Standards, Certifications, and Conformity Tests

Introduction

This chapter describes the operational standards for modules in an M580 PAC system. Agency certifications, environmental conditions, and mechanical characteristics of the modules are detailed.

What Is in This Chapter?

This chapter contains the following topics:

Standards and Certifications

Introduction

M580 PACs have been designed to comply with the relevant standards and rules for electrical equipment in an industrial automation environment.

NOTE: The M580 PAC standard and certifications are consistent with Modicon X80 and M340 module values.

Industrial Standards

Requirements specific to the PAC functional characteristics, immunity, robustness, and safety:

- IEC/EN 61131-2 completed by IEC 61010-2-201
- CSA 22.2 No.142 completed by CSA-E 61131-2
- UL 508

Merchant Navy Certification

The products are designed to comply with major merchant navy agencies requirements (IACS).

More details on merchant navy certifications are available on Schneider Electric website: www.schneider-electric.com.

European Directives for EC Marking

- low voltage: 2006/95/EC
- electromagnetic compatibility: 2004/108/EC

Installation in Classified Ex Area

- For USA and Canada: Hazardous locations class I, division 2, groups A, B, C, and D according to CSA 22.2 No.213, or ISA12.12.01, or FM3611
- For other countries: EC ATEX (directive 94/9/EC), or IECEx in defined atmosphere zone 2 (gas) and/or zone 22 (dust) according to IEC/EN 60079-0, IEC/EN 60079-15, and IEC/EN 60079-31

More details on certifications and Ex installation guides are available on Schneider Electric website: www.schneider-electric.com.

Specific Countries

- For Australia and New Zealand: ACMA requirements for RCM marking (formerly C-Tick)
- For Russia and eastern countries: GOST and EAC

Environmental Friendly Design

- Hazardous substances
  This product is compliant with:

- WEEE, Directive 2002/96/EC
- RoHS, Directive 2011/65/EU
- RoHS China, Standard SJ/T 11363-2006
- REACh regulation EC 1907/2006

NOTE: Documentation about sustainable development is available on Schneider Electric website (Product Environmental Profile and End of Life Instructions, RoHS and REACh certificates).

- End of life (WEEE)
  This product contains electronic boards. It must be disposed of in specific treatment channels.

Service Conditions and Recommendations Relating to Environment

Operating Temperature/Hygrometry/Altitude

Condition		Standard M580 Components	Hardened M580 Components
temperature	operation	0...+60 °C (+32...+140 °F)	–25...+70 °C (–13...+158 °F)
	storage	–40...+85 °C (–40...+185 °F)	–40...+85 °C (–40...+185 °F)
relative humidity (without condensation)	cyclical humidity	5...95% up to +55 °C (+131 °F)	5...95% up to +55 °C (+131 °F)
	continuous humidity	5...93% up to +55 °C (+131 °F)	5...93% up to +60 °C (+140 °F)
altitude	operation	• 0...2000 m (0...6562 ft): full specification for temperature and isolation • 2000...4000 m (6562...13123 ft): ○ temperature derating: +1 °C/400 m (+1.8 °F/1312 ft) ○ isolation loss: 150 Vdc/1000 m (150 Vdc/3280 ft)	

Supply Voltage

Operating conditions relative to the supply voltage:

Power Supply		BMX CPS References				
		2010	3020 (H)	3500 (H)	2000	3540 T
Voltage	Rated	24 Vdc	24...48 Vdc	100...240 Vac	100...240 Vac	125 Vdc
	Limit	18...31.2 Vdc	18...62.4 Vdc	85...264 Vac	85...264 Vac	100...150 Vdc
Frequency	Rated	–	–	50...60 Hz	50...60 Hz	–
	Limit	–	–	47...63 Hz	47...63 Hz	–
Micro-power outages	Duration	≤ 10 ms (1.)	≤ 10 ms (1.)	≤ 1/2 period	≤ 1/2 period	≤ 50 ms at 125 Vdc
	Repetition	≥ 1 s	≥ 1 s	≥ 1 s	≥ 1 s	≥ 1 s
Harmonic rate		–	–	10 %	10 %	–
Residual ripple included (0 to peak)		5 %	5 %	–	–	5 %
1. Limited to 1 ms at maximum load with minimum supply (18 Vdc).						

Conformity Tests

Installation Wiring and Maintenance

Devices must be installed, wired, and maintained in compliance with the instructions provided in the Grounding and Electromagnetic Compatibility of PLC Systems, Basic Principles and Measures, User Manual *(see page 12)* and Control Panel Technical Guide, How to protect a machine from malfunctions due to electromagnetic disturbance *(see page 12)*.

Equipment and Personnel Safety (EC)

Name of Test	Standards	Level
Dielectric strength and insulation resistance	IEC/EN 61131-2 IEC 61010-2-201 UL CSA	Dielectric: 2 **Un** + 1000 V; t = 1 min Insulation: ● **Un** ≤ 50 V: 10 MΩ ● 50 V ≤ **Un** ≤ 250 V: 100 MΩ
Continuity of earth	IEC/EN 61131-2 IEC 61010-2-201 UL CSA	30 A, R ≤ 0.1 Ω, t = 2 min
Leakage current	UL CSA	≤ 3.5 mA after disconnecting
Protection offered by enclosure	IEC/EN 61131-2 IEC 61010-2-201	IP 20 and protection against standardized pins
Impact withstand	IEC/EN 61131-2 IEC 61010-2-201 UL CSA	sphere of 500 g, fall from 1.3 m (energy 6.8 J minimum)
Stored energy injury risk	IEC/EN 61131-2 IEC 61010-2-201	● Non-permanent connection: 37% **Un** after 1 s ● Permanent connection: 37% **Un** after 10 s
Overload	IEC/EN 61131-2 IEC 61010-2-201 UL CSA	50 cycles, **Un**, 1.5 **In** t = 1 s ON + 9 s OFF
Endurance	IEC/EN 61131-2 IEC 61010-2-201 UL CSA	**In**, **Un** 12 cycles: t = 100 ms ON + 100 ms OFF 988 cycles: t = 1 s ON + 1 s OFF 5000 cycles: t = 1 s ON + 9 s OFF
Temperature rise	IEC/EN 61131-2 UL CSA IECEx	ambient temperature: +60 °C (for ruggedized range *(see page 92)*: +70 °C)
Un Nominal voltage **In** Nominal current		

NOTE: (EC): tests required by European directives EC and based on IEC/EN 61131-2 standards.

Immunity to L.F. Interference (EC)

Name of Test	Standards	Level
Voltage and frequency variations	IEC/EN 61131-2 IEC/EN 61000-6-2 IEC 61000-4-11	0.85 **Un**, 1.10 **Un** 0.94 **Fn**, 1.04 **Fn** 4 steps t = 30 min
	IACS E10 IEC 61000-4-11	0.80 **Un**, 1.20 **Un** 0.90 **Fn**, 1.10 **Fn** t = 1.5 s/5 s
Direct voltage variations	IEC/EN 61131-2 IEC 61000-4-29 IACS E10 (PLC not connected to charging battery)	0.85 **Un** + ripple: 5% peak 1.2 **Un** + ripple: 5% peak 2 steps t = 30 min
Third harmonic	IEC/EN 61131-2	H3 (10% **Un**) 0° / 180° 2 steps t = 5 min
Immunity to conducted low frequency (only IACS)	IACS E10	For ac: H2...H15 (10% **Un**), H15...H100 (10...1% **Un**), H100...H200 (1% **Un**) For dc: H2...H200 (10% **Un**)
Voltage interruptions	IEC/EN 61131-2 IEC/EN 61000-6-2 IEC 61000-4-11 IEC 61000-4-29 IACS E10	Power supply immunity: 1 ms for dc **PS1** / 10 ms for ac or dc **PS2** Check operating mode for longer interruptions For IACS: 30 s for ac or dc
	IEC/EN 61131-2 IEC/EN 61000-6-2 IEC 61000-4-11	For ac **PS2**: ● 20% **Un**, t0: 1/2 period ● 40% **Un**, cycle 10/12 ● 70% **Un**, cycle 25/30 ● 0% **Un**, cycle 250/300
Voltage shut-down and start-up	IEC/EN 61131-2	**Un**...0...**Un**; t = **Un** / 60 s **Umin**...0...**Umin**; t = **Umin** / 5 s **Umin**...0.9 **Udl**...**Umin**; t = **Umin** / 60 s
Umin minimum voltage **Udl** detection level when powered **Un** nominal voltage **Fn** nominal frequency **PS1** applies to PLC supplied by battery **PS2** applies to PLC energized from ac or dc supplies		

Name of Test	Standards	Level
Magnetic field	IEC/EN 61131-2 IEC/TS 61000-6-5 IEC 61000-4-8 (for MV power stations: IEC 61850-3)	Power frequency: 50/60 Hz 100 A/m continuous 1000 A/m, t = 3 s 3 axes
	IEC 61000-4-10 (for MV power stations: IEC 61850-3)	Oscillatory: 100 kHz–1 MHz, 100 A/m t=9 s 3 axes
Conducted common mode disturbances range 0…150 kHz	IEC 61000-4-16 (for MV power stations: IEC 61850-3)	For remote systems: ● 50/60 Hz and dc, 300 V, t = 1 s ● 50/60 Hz and dc, 30 V, t = 1 min ● 5 Hz...150 kHz, sweep 3...30 V
Umin minimum voltage **Udl** detection level when powered **Un** nominal voltage **Fn** nominal frequency **PS1** applies to PLC supplied by battery **PS2** applies to PLC energized from ac or dc supplies		

NOTE: (EC): tests required by European directives EC and based on IEC/EN 61131-2 standards.

Immunity to H.F. Interference (EC)

Name of Test	Standards	Level
Electrostatic discharges	IEC/EN 61131-2 IEC/EN 61000-6-2 IEC 61000-4-2 IACS E10	6 kV contact 8 kV air 6 kV indirect contact
Radiated radio frequency electromagnetic field	IEC/EN 61131-2 IEC/EN 61000-6-2 IEC 61000-4-3 IACS E10	15 V/m, 80 MHz...3 GHz Sinus amplitude modulated 80%,1 kHz + internal clock frequencies
Electrical fast transient burst	IEC/EN 61131-2 IEC/EN 61000-6-2 IEC 61000-4-4 IACS E10	For ac and dc main supplies: 2 kV in common mode / 2 kV in wire mode For ac and dc auxiliary supplies, ac unshielded I/Os: 2 kV in common mode For analog, dc unshielded I/Os, communication, and all shielded lines: 1 kV in common mode

Name of Test	Standards	Level
Surge	IEC/EN 61131-2 IEC/EN 61000-6-2 IEC 61000-4-5 IACS E10	For ac and dc main and auxiliary supplies, ac unshielded I/Os: 2 kV in common mode / 1 kV in differential mode For analog, dc unshielded I/Os: 0.5 kV in common mode / 0.5 kV in differential mode For communication and all shielded lines: 1 kV in common mode
Conducted disturbances induced by radiated electromagnetic fields	IEC/EN 61131-2 IEC/EN 61000-6-2 IEC 61000-4-6 IACS E10	10 V, 0.15...80 MHz Sinus amplitude modulated 80%, 1 kHz + spot frequencies
Damped oscillatory wave	IEC/EN 61131-2 IEC/EN 61000-4-18 IACS E10	For ac and dc main supplies and ac auxiliary supplies, ac unshielded I/Os: 2.5 kV in common mode / 1 kV in differential mode For dc auxiliary supplies, analog, dc unshielded I/Os: 1 kV in common mode / 0.5 kV in differential mode For communication and all shielded lines: 0.5 kV in common mode

NOTE: These tests are performed without a cabinet, with devices fixed on a metal grid and wired as per the recommendations in the Grounding and Electromagnetic Compatibility of PLC Systems, Basic Principles and Measures, User Manual *(see page 12)*.

NOTE: (EC): tests required by European directives EC and based on IEC/EN 61131-2 standards.

Electromagnetic Emissions (EC)

Name of Test	Standards	Level
Conducted emission	IEC/EN 61131-2 FCC part 15 IEC/EN 61000-6-4 CISPR 11&22, Class A, Group 1 IACS E10	150...500 kHz: quasi-peak 79 dB (µV/m); average 66 dB (µV/m) 500 kHz...30 MHz: quasi-peak 73 dB (µV/m); average 60 dB (µV/m) ac and dc power (general power distribution zone): • 10...150 kHz: quasi-peak 120...69 dB (µV/m) • 150 kHz...0.5 MHz: quasi-peak 79 dB (µV/m) • 0.5...30 MHz: quasi-peak 73 dB (µV/m) ac and dc power (bridge and deck zone for evaluation): • 10…150 kHz: quasi-peak 96…50 dB (µV/m) • 150 kHz…0.35 MHz: quasi-peak 60...50 dB (µV/m) • 0.35…30 MHz: quasi-peak 50 dB (µV/m)

Name of Test	Standards	Level
Radiated emission	IEC/EN 61131-2 FCC part 15 IEC/EN 61000-6-2 CISPR 11&22, Class A, Group 1 IACS E10	30...230 MHz: quasi-peak 40 dB (µV/m) (at 10 m); 50 dB (µV/m) (at 3 m) 230 MHz...1 GHz: quasi-peak 47 dB (µV/m) (at 10 m); 57 dB (µV/m) (at 3 m) For general power distribution zone: ● 0.15...30 Mhz: quasi-peak 80...50 dB (µV/m) (at 3 m) ● 30...100 MHz: quasi-peak 60...54 dB (µV/m) (at 3 m) ● 100 MHz...2 GHz: quasi-peak 54 dB (µV/m) (at 3 m) ● 156...165 MHz: quasi-peak 24 dB (µV/m) (at 3 m)

NOTE: (EC): tests required by European directives EC and based on IEC/EN 61131-2 standards.

Immunity to Climatic Variations (Power On)

Name of Test	Standards	Level
Dry heat	IEC 60068-2-2 (Bb & Bd)	+60 °C, t = 16 h (for ruggedized range *(see page 92)*: +70 °C, t = 16 h)
	IACS E10	+60 °C, t = 16 h and +70 °C, t = 2 h (for ruggedized range: +70 °C, t = 16 h)
Cold	IEC 60068-2-1 (Ab & Ad) IACS E10	0 °C...–25 °C, t = 16 h + power on at 0 °C (for ruggedized range: power on at –25 °C)
Damp heat, steady state (continuous humidity)	IEC 60068-2-78 (Cab) IACS E10	+55 °C, 93% relative humidity, t = 96 h (for ruggedized range: +60 °C)
Damp heat, cyclic (cyclical humidity)	IEC 60068-2-30 (Db) IACS E10	+55...+25 °C, 93...95% relative humidity, 2 cycles t = 12 h + 12 h
Change of temperature	IEC 60068-2-14 (Na & Nb)	0...+60 °C, 5 cycles t = 6 h + 6 h (for ruggedized range: –25...+70 °C)

Withstands to Climatic Variations (Power Off)

Name of Test	Standards	Level
Dry heat	IEC/EN 61131-2 IEC 60068-2-2 (Bb & Bd) IEC/EN 60945	+85 °C, t = 96 h
Cold	IEC/EN 61131-2 IEC 60068-2-1 (Ab & Ad) IACS E10	–40 °C, t = 96 h

Name of Test	Standards	Level
Damp heat, cyclic (cyclical humidity)	IEC/EN 61131-2 IEC 60068-2-30 (Db)	+55...+25 °C, 93...95 % relative humidity, 2 cycles t = 12 h + 12 h
Change of temperature (thermal shocks)	IEC/EN 61131-2 IEC 60068-2-14 (Na & Nb)	–40...+85 °C, 5 cycles t = 3 h + 3 h

Immunity to Mechanical Constraints (Power On)

Name of Test	Standards	Level
Sinusoidal vibrations	IEC/EN 61131-2 IEC 60068-2-6 (Fc)	• Basic IEC/EN 61131-2: 5...150 Hz, +/– 3.5 mm amplitude (5...8.4 Hz), 1 g (8.4...150 Hz) • Specific profile: 5...150 Hz, +/– 10.4 mm amplitude (5...8.4 Hz), 3 g (8.4...150 Hz) • For basic and specific, endurance: 10 sweep cycles for each axis
	IACS E10	3...100 Hz, 1 mm amplitude (3...13.2 Hz), 0.7 g (13.2...100 Hz) Endurance at each resonance frequency: 90 min for each axis, amplification coefficient < 10
	IEC 60068-2-6	Sismic analysis: 3...35 Hz, 22.5 mm amplitude (3...8.1 Hz), 6 g (8.1...35 Hz)
Shocks	IEC/EN 61131-2 IEC 60068-2-27 (Ea)	30 g, 11 ms; 3 shocks/direction/axis **NOTE:** In case of using fast actuators (response time ≤ 15 ms) driven by relay outputs: 15 g, 11 ms; 3 shocks/direction/axis.
		25 g, 6 ms; 100 bumps/direction/axis (bumps) **NOTE:** In case of using fast actuators (response time ≤ 15 ms) driven by relay outputs: 15 g, 6 ms; 100 bumps/direction/axis.
Free fall during operation	IEC/EN 61131-2 IEC 60068-2-32 (Ed Method 1)	1 m, 2 falls

Withstand to Mechanical Constraints (Power Off)

Name of Test	Standards	Level
Random free fall with packaging	IEC/EN 61131-2 IEC 60068-2-32 (Method 1)	1 m, 5 falls

Name of Test	Standards	Level
Flat free fall	IEC/EN 61131-2 IEC 60068-2-32 (Ed Method 1)	10 cm, 2 falls
Controlled free fall	IEC/EN 61131-2 IEC 60068-2-31 (Ec)	30° or 10 cm, 2 falls
Plugging / Unplugging	IEC/EN 61131-2	For modules and connectors: ● operations: 50 for permanent connections, 500 for non-permanent connections

Specific Environment

Name of Test	Standards	Level
Corrosion areas - gas, salt, dust	ISA S71.4	Mixed flowing gases: class G3, 25 °C, 75 % relative humidity, t = 14 days
	IEC 60721-3-3	Mixed flowing gases: class 3C3, 25 °C, 75 % relative humidity, t = 14 days
	IEC 60068-2-52	Salt spray: test Kb, severity 2
	IEC 60721-3-3	Sand/Dust: class 3S3

Protective Enclosure

The M580 PACs are enclosed equipment designed to an IP20 level of ingress protection. For installation in industrial manufacturing workshops or in heat and humidity processing environments, install the M580 PAC in an IP54 enclosure.

NOTE: For IP20 compliance, use a BMX XEM 010 protective cover on any empty slots in a rack.

A system may be installed outside an enclosure if it is operating in a restricted-access room not exceeding pollution level 2 (for example, a control room with no machines or dust-producing activities).

Part II
Installing a Local Rack

Introduction

Installation and assembly of the M580 system is a methodical process described in the following topics.

What Is in This Part?

This part contains the following chapters:

Chapter 5
Installation and Assembly of M580 Racks and Extender Module

Overview

This chapter explains how to install M580 racks and extender module.

What Is in This Chapter?

This chapter contains the following topics:

Planning the Installation of the Local Rack

Introduction

The size and number of racks and the kinds of modules installed on the racks are significant considerations when you are planning an installation. That installation may be either inside or outside an enclosure. The height, width, and depth of the installed system head as well as the spacing between the local and the extender racks need to be well understood.

⚠ WARNING
UNEXPECTED EQUIPMENT OPERATION
Install the racks lengthways and horizontally to facilitate ventilation.
Failure to follow these instructions can result in death, serious injury, or equipment damage.

Modules such as the power supply, CPU, and I/O are cooled by natural convection. Install them on a horizontally installed rack as illustrated in this manual to maintain the necessary thermal cooling. Other rack mounting positions may cause overheating and unexpected equipment operation.

Clearance Around the Racks

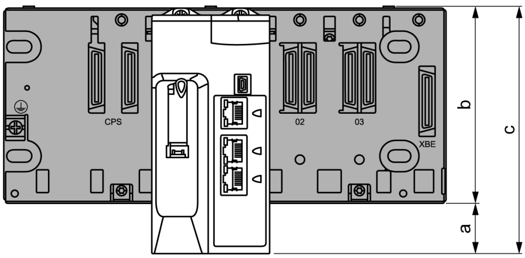
Leave a minimum space of 12 mm (0.472 in.) on the right side of each rack for cooling.

When your plan calls for extender racks, leave a minimum space of 35 mm (1.378 in.) in front of the modules. The BMX XBE 1000 rack extender module requires this clearance for the local bus connector and terminator.

Spacing Requirements for an M580 CPU in a Local Main Rack

⚠ WARNING
OVERHEATING AND UNEXPECTED EQUIPMENT OPERATION
Maintain proper thermal clearances when installing the racks.
Failure to follow these instructions can result in death, serious injury, or equipment damage.

In the main local rack, allow additional clearance at the bottom of the rack for the M580 CPU. The following illustration shows the mounting dimensions when an X Bus rack is used and when an Ethernet rack is used. The overall height dimension of the main local rack in both cases is 134.6 mm (5.299 in.).



a Additional space below the rack to accommodate the height of the CPU. For an X Bus rack, the value is 30.9 mm (1.217 in.); for an Ethernet rack, the value is 29.49 mm (1.161 in.).

b The height of the rack. For an X bus rack, the height is 103.7 mm (4.083 in.); for an Ethernet rack, the height is 105.11 mm (4.138 in.).

c The height of the main local rack, 134.6 mm (5.299 in.).

Thermal Considerations Inside an Enclosure

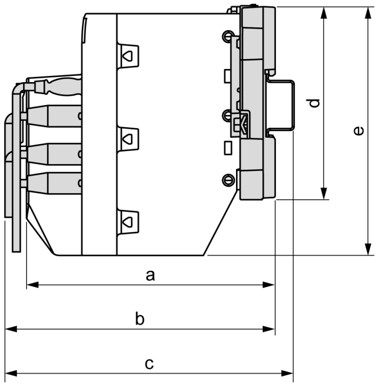
If the racks are installed in an enclosure, you need to facilitate air circulation. Use an enclosure that allows the following minimum clearances:

- 80 mm (3.15 in.) above the top of the modules on the rack
- 60 mm (2.36 in.) below the bottom of the modules on the rack
- 60 mm (2.36 in.) between modules and wiring ducts

The minimum depth of the enclosure is:

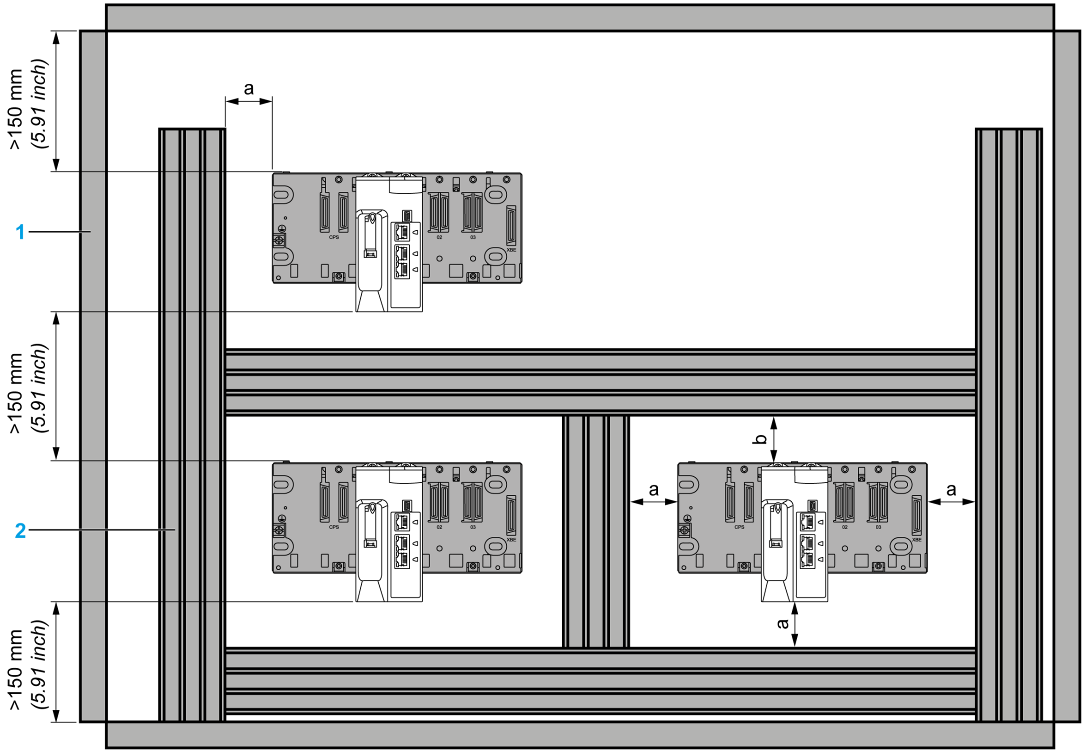
- 150 mm (5.91 in.) if the rack is fastened to a plate
- 160 mm (6.30 in.) if the rack is mounted on a 15 mm (0.59 in.) DIN rail
- If BMX XBE 1000 rack extender modules are connected, the use of BMX XBC •••K cables with connectors angled at 45° is recommended.

Here is a side view of a rack on a DIN rail with modules and cables mounted in an enclosure:



a enclosure depth: 135 mm (5.315 in.)
b wiring + module depth: > 146 mm (5.748 in.)
c wiring + module + DIN rail depth: > 156 mm (6.142 in.)
d rack height: for an X Bus rack 103.7 mm (4.083 in.); for an Ethernet rack, 105.11 mm (4.138 in.)
e module height: 134.6 mm (5.299 in.)

The following illustration shows the rules of installation in a cabinet:



1 installation or casing
2 wiring duct or tray
a side and bottom clearance: > 60 mm (2.36 in.)
b top clearance: > 80 mm (3.15 in.)

Mounting the Racks

Introduction

Ethernet and X Bus racks may be mounted on:

- DIN rails
- walls
- Telequick mounting grids

NOTE: Mount the racks on a properly grounded metallic surface to allow the PAC to operate correctly in the presence of electromagnetic interference.

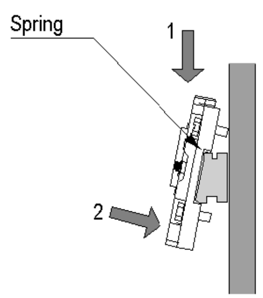
Mounting on a DIN Rail

Most racks can be mounted on DIN rails that are 35 mm (1.38 in.) wide and 15 mm (0.59 in.) deep.

NOTE: Racks longer than 400 mm (15.75 in.) and support more than 8 module slots are not compatible with DIN rail mounting. Do not mount a BME XBP 1200 (H), or BMX XBP 1200 (PV:02 or later) (H) rack on a DIN rail.

NOTE: When mounted on a DIN rail, the system is more susceptible to mechanical stress *(see page 98)*.

Mounting a rack on a DIN rail:

Step	Action	Illustration
1	Position the rack on the top of the DIN rail and press down the top of the rack to compress the springs in contact with the DIN rail.	
2	Tilt the bottom of the rack backwards to flatten it against the DIN rail.	
3	Release the rack to lock it.	

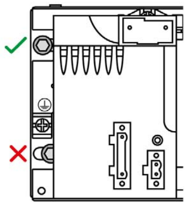
To remove a rack from a DIN rail:

Step	Action
1	Press down the top of the rack to compress the springs in contact with the DIN rail.
2	Tilt the bottom of the rack forward to disengage it from the DIN rail.
3	Release the freed rack.

Mounting on a Wall

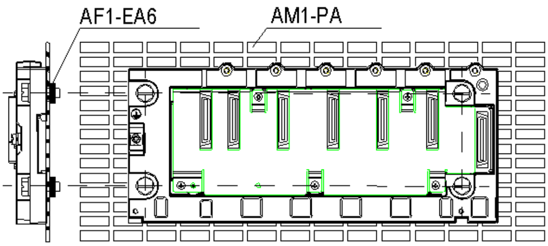
You can mount a rack on a wall inside or out of an enclosure with M4, M5, M6, or UNC #6 screws inserted in the fastening holes *(see page 77)*.

Place the 2 left side screws (near the power supply) as close as possible to the left edge of the rack. This enables you to access the screws after the power supply is mounted.



Mounting on Telequick Grid AM1-PA and AM3-PA Mounting Grids

You can mount a rack on a Telequick AM1-PA or AM3-PA mounting grid using M4, M5, M6, or UNC #6 screws.



Grounding the Rack and Power Supply Module

Grounding the Rack

To ground the racks, connect a ground cable between the protective earth ground of the installation and the screw located on the left-hand side of the rack, close to the power supply module. This screw is used to connect two 1.5...2.5 mm^2 cables.

Ground every rack in the PAC system.

Grounding the Power Supply Module

Ground each power supply module in the system.

DANGER

HAZARD OF ELECTRIC SHOCK

Ground the power supplies by connecting the protective earth ground terminal on each power supply module to the protective earth ground of the installation. Connect them in either of the following ways:

- Connect the protective earth ground terminal of the power supply to the protective earth ground of the installation with a separate cable, independent of the rack ground cable.
- Connect the protective earth ground terminal of the power supply to the ground screw of the rack (where the rack itself is grounded).

Do not connect anything else to the power supply ground.

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

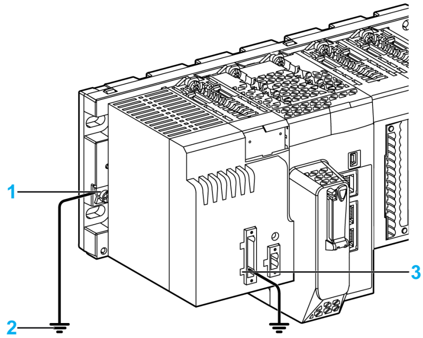
DANGER

HAZARD OF ELECTRIC SHOCK

- Use only cables with ring or spade lugs and check that there is a good ground connection.
- Make sure that grounding hardware is tightened properly.

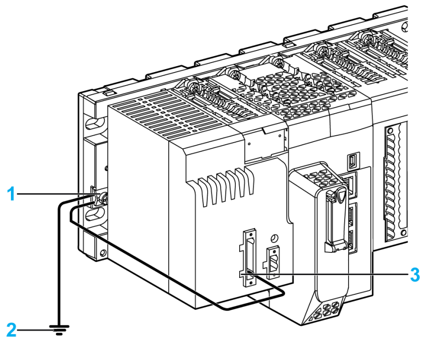
Failure to follow these instructions will result in death or serious injury.

The following illustration shows how the rack and the power supply module are grounded using 2 independent ground cables:



1 rack protective earth ground screw
2 protective earth ground
3 power supply module terminal block (PE)

The following illustration shows how the rack and the power supply module are grounded using the PE terminals connected to each other:



1 rack protective earth ground screw
2 protective earth ground
3 power supply module terminal block (PE)

Previous wiring illustration is possible only if the cable extremities (which are screwed to the grounding bus of the rack) have ring or spade lugs able to maintain permanent fastening even if the screw is slack.

Grounding Installed Modules

Grounding CPUs and Power Supplies

DANGER

HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH

Check that ground connection contacts are available and not bent out of shape. If they are bent or not available, do not use the module and contact your Schneider Electric representative.

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

WARNING

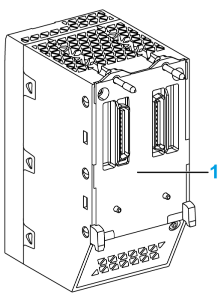
UNINTENDED EQUIPMENT OPERATION

Tighten the clamping screws of the modules. A bad module connection can lead to an unexpected behavior of the system.

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

Grounding the Modules

All modules are equipped with ground connection contacts at the rear for grounding purposes (following example shows a CPU module):



1 ground connection contact

These contacts connect the grounding bus of the modules to the grounding bus of the rack.

BMX XEM 010 Protective Cover for Unused Module Slots

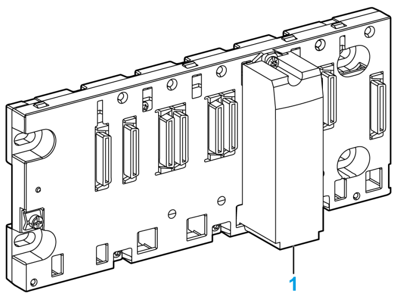
Introduction

If a rack has unused module slots, install a BMX XEM 010 cover to keep dust and other objects out of the slots and to comply with IP20 ingress protection requirements. *(see page 99)*

BMX XEM 010 covers are sold in sets of 5.

Illustration

Install and attach a BMX XEM 010 cover to the rack like a normal module. Here a cover is placed in an unused module slot in a BME XBP 0400 rack:



1 BMX XEM 010 cover

BMX XSP *xxxx* Protection Bar

Introduction

Connect the cable shielding directly to the ground and not to the module shielding to help protect the system from electromagnetic perturbations.

Use a protection bar in the 3 following cases:
- counting module with 10-pin, 16-pin, and 20-pin terminal blocks
- analog module with 20-pin terminal block and 40-pin connector
- processor connected to an XBT console via the USB port

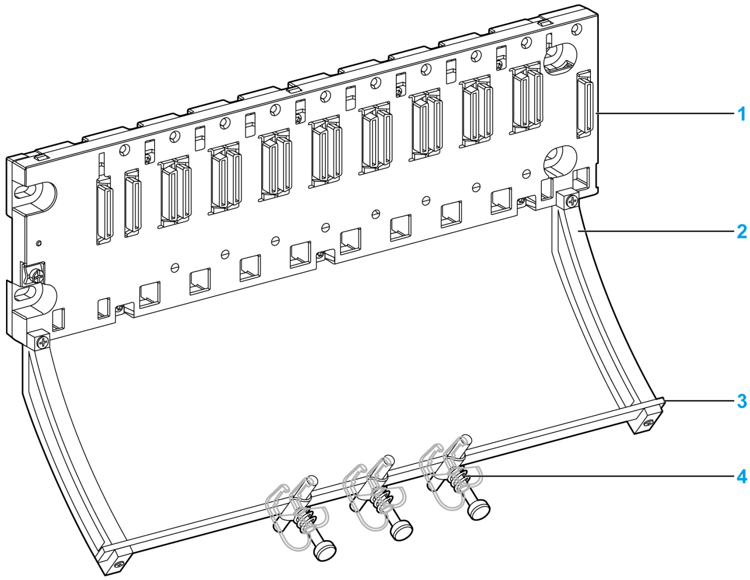
Fasten the protection bar at each end of the rack to provide a connection between the cable and the grounding screw.

Protection Bar Kits References

The protection bar kit references are as follows:
- BMX XSP 0400 bar is fastened to a:
  - BMX XBP 0400 (PV:02 or later) (H) rack
  - BME XBP 0400 (H) rack
- BMX XSP 0600 bar is fastened to a:
  - BMX XBP 0600 (PV:02 or later) (H) rack
- BMX XSP 0800 bar is fastened to a:
  - BMX XBP 0800 (PV:02 or later) (H) rack
  - BME XBP 0800 (H) rack
- BMX XSP 1200 bar is fastened to a:
  - BMX XBP 1200 (PV:02 or later) (H) rack
  - BME XBP 1200 (H) rack

Each kit includes the following components:
- 1 metallic bar
- 2 bases
- 1 set of spring locking clamp rings to fasten the cables to the protection bar.

Example of protection bar fastened to a Modicon M580 rack:



1 rack
2 base
3 metallic bar
4 clamp ring

Clamp rings are sold in sets of 10 and are available under the following references:
- STB XSP 3010: small rings to fasten USB connection cables
- STB XSP 3020: large rings to fasten analog and counting modules connection cables

NOTE: A protection bar does not modify the volume required when installing and uninstalling modules.

Connecting a Console to a CPU

2 connection cables are available to connect a human-machine interface to the CPU USB port:
- BMX XCA USB 018: 1.8 m cable
- BMX XCA USB 045: 4.5 m cable

Each cable ends with 2 different connectors:
- Type A USB: console connector.
  A metallic ground connection is provided close to the connector to be screwed to a grounded object
- Type mini-B USB: CPU connector.
  A metallic ground connection is provided close to the connector to be screwed to a grounded object.

A bare section of cable is provided close to the connector to be fastened to the protection bar with a clamp ring.

Modicon X80 Rack Extender Module Installation

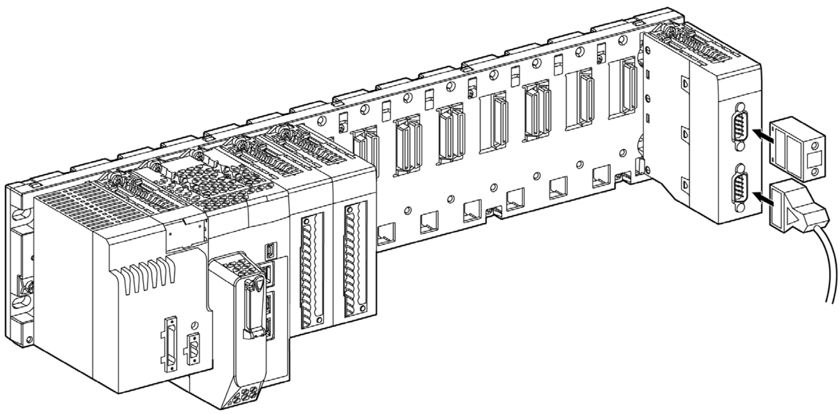
Introduction

When your installation has more than one rack in the local rack or at a remote drop, install a BMX XBE 1000 rack extender module on the main rack and the extended racks. Rack extender modules are connected together by X bus extension cables.

Extender Module Placement in an X80 Rack

This module goes in each rack in the slot marked **XBE** on the right side of the rack.

The following illustration shows a main local rack set up to support extended racks. On the left side of the rack are the power supply, the CPU, and some X80 I/O modules. On the right side of the rack is a BMX XBE 1000 extender module:



Extension Cables

The BMX XBE 1000 rack extender modules on each rack are connected with BMX XBC •••K or TSX CBY •••K extension cables *(see page 71)*. A BMX XBC •••K cable is used to connect to an X80 I/O extension. A TSX CBY •••K cable is used to connect to a Premium I/O extension.

NOTE: Premium I/O extensions are permitted in the local rack only. You cannot use Premium I/O in a remote drop.

Line Terminators in X80 Racks

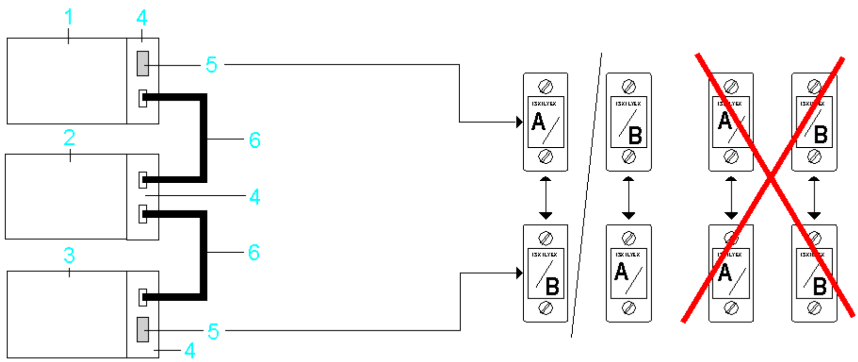
DANGER

HAZARD OF ELECTRIC SHOCK

Remove power from all elements of the station (the local rack or remote drop) before inserting or extracting a line terminator.

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

Terminate the unconnected 9-pin SUB-D connectors on any BMX XBE 1000 modules. One connector in the main rack and one connector in the last rack in the extension are unused. Insert a TSX TLY EX line terminator in each of the unused connectors *(see page 72)*:



1 X80 main rack
2 first X80 extension rack
3 last X80 extension rack
4 BMX XBE 1000 modules in each rack
5 TSX TLY EX line terminator in the main rack and the last rack
6 BMX XBC •••K or TSX CBY •••K extension cables between each rack

Line terminators are labeled A/ or /B. An extended rack needs to use one line terminator labeled A/ and one labeled /B. If you terminate the unused connector in the main rack with anA/ terminator, then you need to terminate the unused connector in the last rack with a /B terminator.

Line Terminators in X80 Rack with Premium Extendable Racks

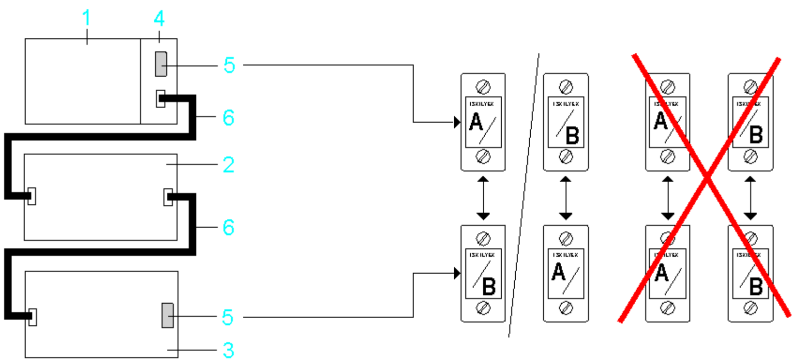
DANGER

HAZARD OF ELECTRIC SHOCK

Remove power from all elements of the station (the local rack or remote drop) before inserting or extracting a line terminator.

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

Unconnected 9-pin SUB-D connectors on any BMX XBE 1000 modules or Premium extendable rack need to be terminated. One connector in the main rack and one connector in the last rack in the extension are unused. Insert a TSX TLY EX line terminator in each of the unused connectors *(see page 72)*:



1 X80 main rack
2 First Premium extension rack
3 Last Premium extension rack
4 BMX XBE 1000 module
5 TSX TLY EX line terminator in the main rack and the last rack
6 BMX XBC •••K or TSX CBY •••K extension cables between each rack

Line terminators are labeled A/ or /B. An extended rack needs to use one line terminator labeled A/ and one labeled /B. If you terminate the unused connector in the main rack with anA/ terminator, then you need to terminate the unused connector in the last rack with a /B terminator.

Extender Module Installation in an X80 Rack

The BMX XBE 1000 rack extender module is installed similarly to the other modules in the rack with these special considerations:

- The **XBE** slot is not a standard module slot. It is reserved for a BMX XBE 1000 rack extender module. No other module type can be installed in the **XBE** slot.
- The BMX XBE 1000 rack extender module cannot be installed in any slot other than the **XBE** slot.
- If a BMX XBE 1000 rack extender module is not present in the main rack of the extension, none of the extender racks will be operational.
- If a BMX XBE 1000 rack extender module is not present in an extended rack, that rack will not be operational.
- Each rack with a BMX XBE 1000 rack extender module in it needs to be assigned an address from 00 to 08. The address assigned to each rack in an extension needs to be unique with respect to all other racks in the extension. Rack addresses are set manually using the 4 microswitches on the side of the BMX XBE 1000 rack extender module *(see page 62)*.
- The main rack in the extension needs to be given address 00, which is the factory default setting for the switches.

DANGER

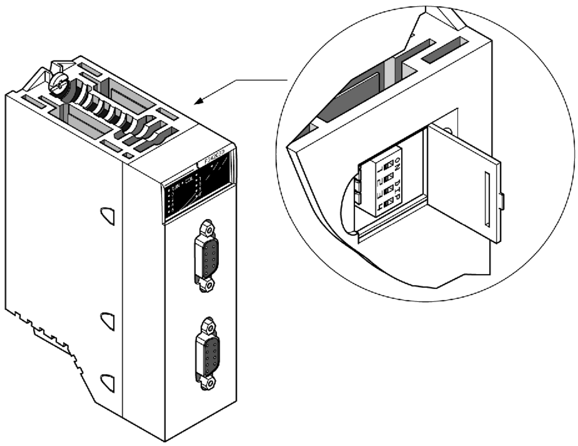
HAZARD OF ELECTRIC SHOCK

Remove all power sources before installing the rack extender module.

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

Follow these steps to install a rack extender module in a rack:

Step	Action
1	Remove all power sources to the rack.

Step	Action
2	Using the microswitches on the side of the rack extender module, set a unique address for that rack from 00 to 08: 
3	Insert the rack extender module in the slot labeled **XBE**.
4	Connect each rack in the extension to the rack immediately before it and immediately after it using the appropriate extension cable.
5	Terminate the unused connector on the extender module in the main rack and the unused connector on the last rack in the extension. Use a line terminator labeled **A/** on one end of the extension and a line terminator labeled **/B** on the other end of the extension.

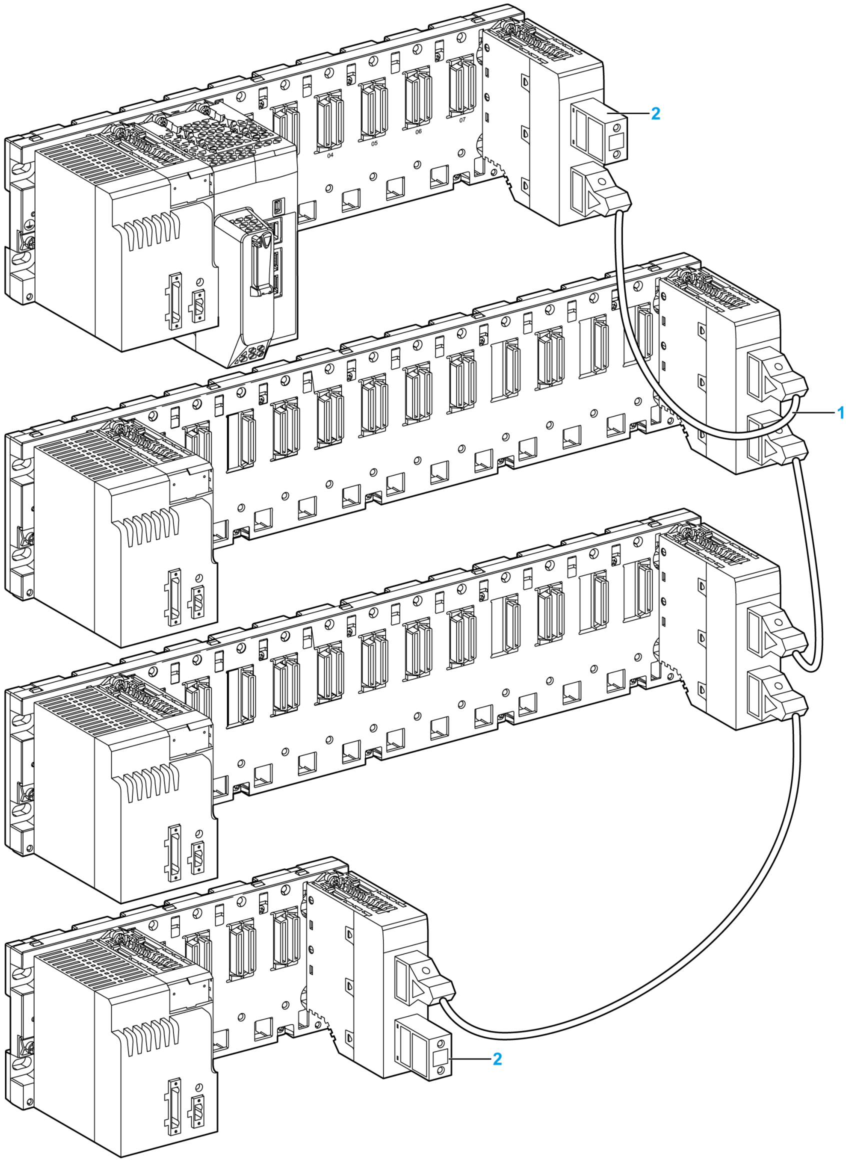
Extender Module Grounding

The BMX XBE 1000 rack extender module has ground connection contacts *(see page 112)*.

Building a Modicon M580 System Using BME XBP ••00 Racks

Thanks to the BMX XBE 1000 extender modules and cables, a specific quantity of racks *(see page 59)* can be added to a local or remote drop main rack.

Example of Modicon X80 main rack with extension racks and extender modules and cables:



1 The same station can contain racks of different sizes that are interconnected by extension cables.
2 The extender modules located at the extremities of the interconnected cables are terminated.

Chapter 6
Installation of the Power Supply, CPU, and Modules in a M580 Rack

Overview

This chapter explains how to install the modules in a M580 rack.

What Is in This Chapter?

This chapter contains the following topics:

Definition of Protection Devices at the Start of the Line

Introduction

It is recommended that you install a protection device at the start of the line on the power supply network, including the following elements:

- circuit breaker
- fuse

The following information allows definition of the minimum caliber circuit breaker and fuse for a given power supply module.

Choice of Line Circuit Breaker

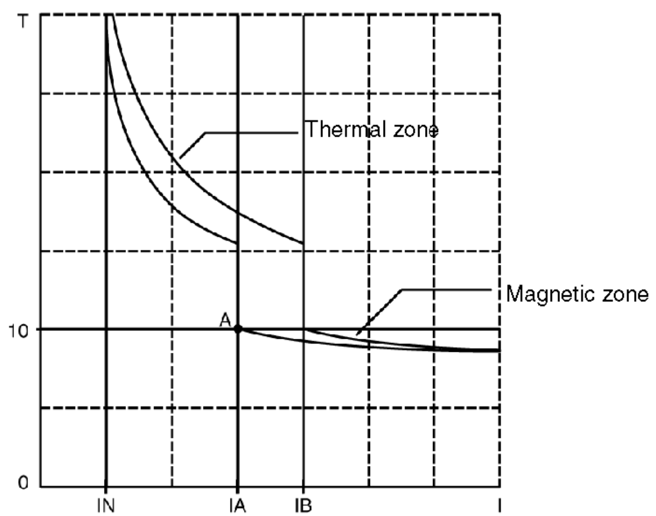
When you choose the caliber of the line circuit breaker, consider:

- nominal input current (Imrs)
- signaling current (I)
- current characteristic (It)

The choice of minimum circuit breaker caliber is made according to the following rules:

- IN circuit breaker caliber greater than the power supply nominal input current (Irms)
- maximum circuit breaker caliber greater than the power supply signaling current (I)
- current characteristic (It) at point A of the curve greater than the power supply characteristic (It)

The following graph shows an example of characteristics provided by a circuit breaker manufacturer:



Choice of Line Fuse

When you choose the caliber of the line fuse, consider:

- current characteristic (I^2t)

The choice of minimum fuse caliber is made according to the following rules:

- IN fuse caliber greater than 3 times the power supply nominal input current Irms
- fuse current characteristic I^2t greater than 3 times the power supply characteristic I^2t

The following table shows the characteristics of each power supply module:

Power Supply Module		BMX CPS 2000	BMX CPS 3500	BMX CPS 3540 T	BMX CPS 2010	BMX CPS 3020
nominal input current Irms	at 24 Vdc	-	-	-	1 A	1.65 A
	at 48 Vdc	-	-	-	-	0.83 A
	at 115 Vac	0.61 A	1.04 A	-	-	-
	at 125 Vdc	-	-	0.36 A	-	-
	at 230 Vac	0.31 A	0.52 A	-	-	-
signaling current I (1)	at 24 Vdc	-	-	-	30 A	30 A
	at 48 Vdc	-	-	-	-	60 A
	at 115 Vac	30 A	30 A	-	-	-
	at 125 Vdc	-	-	30 A	-	-
	at 230 Vac	60 A	60 A	-	-	-
current characteristic It	at 24 Vdc	-	-	-	0.15 As	0.2 As
	at 48 Vdc	-	-	-	-	0.3 As
	at 115 Vac	0.03 As	0.05 As	-	-	-
	at 125 Vdc	-	-	0.05 As	-	-
	at 230 Vac	0.06 As	0.07 As	-	-	-
current characteristic I^2t	at 24 Vdc	-	-	-	0.6 A^2s	1 A^2s
	at 48 Vdc	-	-	-	-	3 A^2s
	at 115 Vac	0.5 A^2s	1 A^2s	-	-	-
	at 125 Vdc	-	-	2 A^2s	-	-
	at 230 Vac	2 A^2s	3 A^2s	-	-	-
1 values at initial power-up and at 25 °C (77 °F)						

Power Supply, CPU, and Module Guidelines

Introduction

A valid local rack contains at least a power supply and a CPU. A valid remote rack contains at least an adapter module, a power supply, and an X80 module.

Module Guidelines

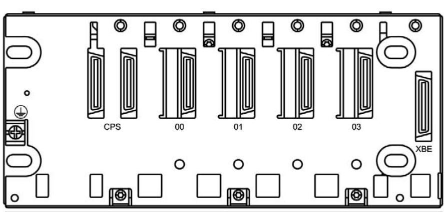
Rack Position	Rack Type	Slots Marking				
		CPS (X80) PS (Premium)	00	01	02	...n [(1)]
local	main rack	power supply	CPU		module	module
	X80 extension rack	power supply	module	module	module	module
	Premium extension rack	power supply	module	module	module	module
remote drop	main rack	power supply	(e)X80 EIO adapter module	module	module	module
	extension rack	power supply	module	module	module	module
1 slots from number **03** to last numbered slot of the rack						

NOTE: When your installation has more than one rack in the local rack or at a remote drop, the BMX XBE 1000 rack exender module goes in the slot marked **XBE** of the X80 racks.

Check that the CPU is installed in the 2 slots marked **00** and **01** on the local rack before powering up the system. If the CPU is not installed in these 2 slots, the CPU will start in **NO_CONF** state and use the configured IP address (not the default IP address).

Rack Markings

Example of BMX XBP •••• (PV:02 or later) rack with slot markings:



Installing the CPU Module

Introduction

A BME P58 •••• CPU can be installed in the following racks:
- BMX XBP •••• (PV:02 or later) X Bus rack
- BME XBP ••00 Ethernet rack

Installation Precautions

BME P58 •••• CPU is powered by the rack bus so the rack power supply must be turned off before installing the CPU.

DANGER

HAZARD OF ELECTRIC SHOCK

Remove all power sources before installing the CPU.

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

Remove the protective cover from the rack slot connectors before plugging the module in the rack.

WARNING

UNEXPECTED EQUIPMENT OPERATION

Check that the CPU does not contain an unsupported SD memory card before powering up the CPU.

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

NOTE: Check that the memory card slot door is closed after a memory card is inserted in the CPU.

NOTE: Refer to `%SW97` to check the status of the SD card.

Grounding Considerations

DANGER

ELECTRICAL SHOCK HAZARD

- Switch off the power supply to the PAC at both ends of the connection before inserting or removing an Ethernet cable.
- Use suitable insulation equipment when inserting or removing all or part of this equipment.

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

Do not apply power to an Modicon X80 rack until connections are made at both ends of the Ethernet cable. For example, connect the cable to both the CPU and another device (adapter module) or DRS before you turn on the power.

Refer to your system hardware reference manual for details about the DRSs.

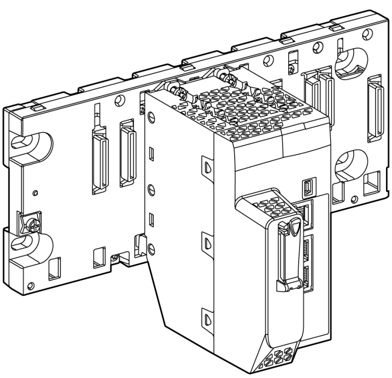
Use fiber-optic cable to establish a communications link when it is not possible to equalize the potential between the 2 grounds.

NOTE: Refer to the ground protection information provided in the Grounding and Electromagnetic Compatibility of PLC Systems, Basic Principles and Measures, User Manual *(see page 12)* and Control Panel Technical Guide, How to protect a machine from malfunctions due to electromagnetic disturbance *(see page 12)*.

Situation

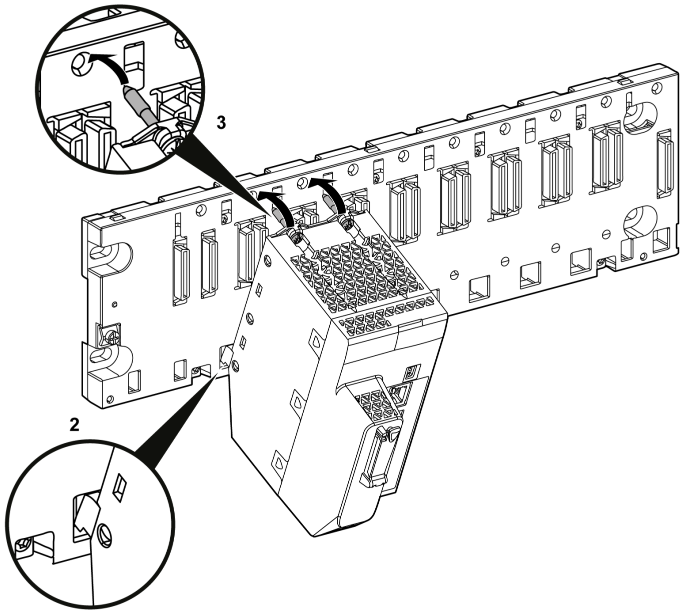
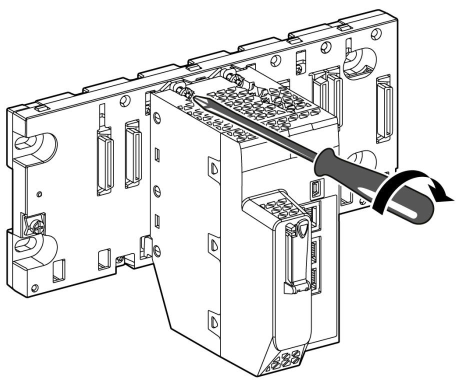
The CPU module is installed in the 2 slots of the rack marked **00** and **01**. If the CPU is not installed in these 2 slots, the CPU will start in **NO_CONF** state and use the configured IP address (not the default IP address)

Example of BME P58 •••• CPU installed in a BME XBP 0400 rack:



Installing the CPU in the Rack

Procedure to follow to install a CPU module in a rack:

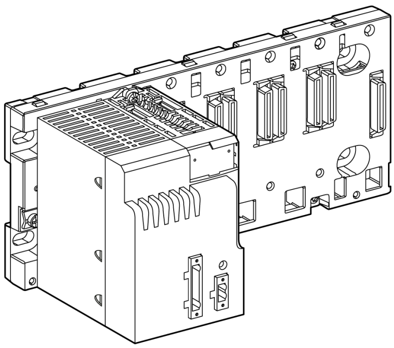
Step	Action	Illustration
1	Verify that: ● the power supply is turned off ● if an SD memory card is used, check that it is supported by the CPU ● the connectors protective covers are removed ● the CPU is placed on the slots marked **00** and **01**	
2	Position the locating pins situated at the rear of the module (on the bottom part) in the corresponding slots in the rack.	
3	Swivel the module towards the top of the rack so that the module sits flush with the back of the rack. The module is now set in position.	
4	Tighten the 2 screws on top of the CPU to maintain the module in place on the rack. Tightening torque: 1.5 N.m (1.106 lbf ft) max.	

Installing a Power Supply Module

Introduction

Install the power supply module in the first 2 slots of each rack marked **CPS**.

Example of power supply module installed in a BME XBP 0400 rack:



NOTE: The power supply module design only allows it to be placed in the dedicated slots.

Installing the Power Supply Module in a Rack

To install a BMX CPS •••• power supply module in a rack, follow the procedure for installing a BME P58 •••• CPU *(see page 127)*.

Grounding the Power Supply Module

The power supply is equipped with ground connection contacts *(see page 112)*.

Installing an SD Memory Card in a CPU

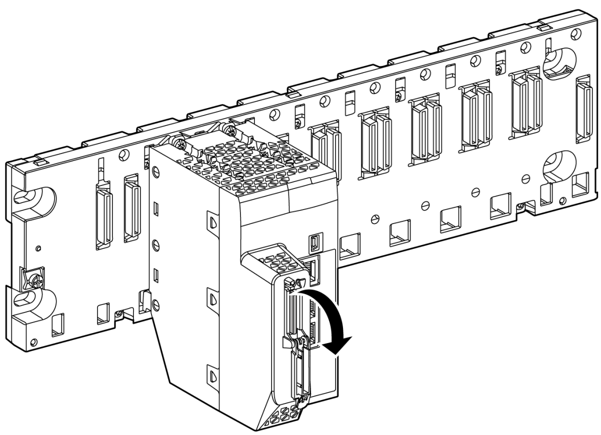
Memory Card Maintenance

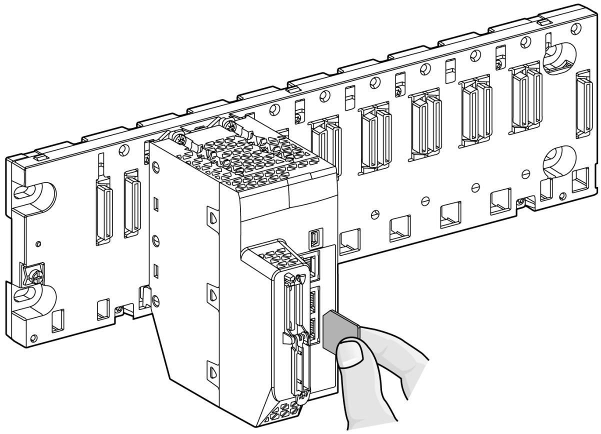
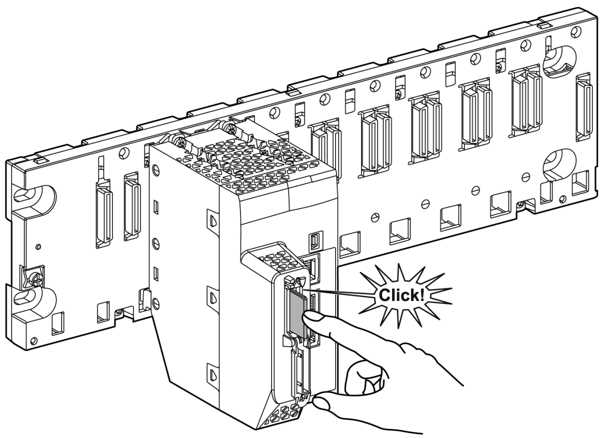
To keep the memory card in normal working order:

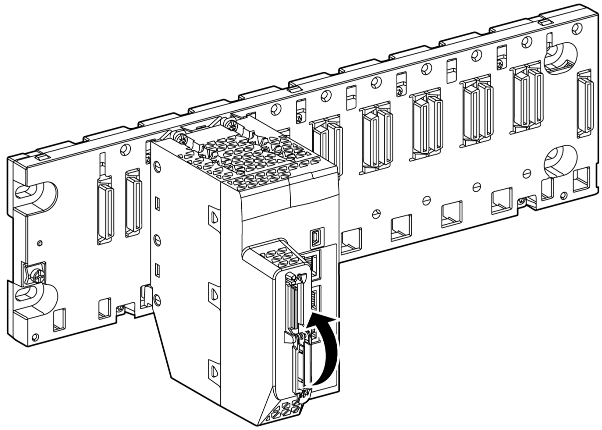
- Avoid removing the memory card from its slot when the CPU accesses the card (memory card access green LED ON or blinking).
- Avoid touching the memory card connectors.
- Keep the memory card away from electrostatic and electromagnetic sources as well as heat, sunlight, water, and moisture.
- Avoid impact on the memory card.
- Before sending a memory card by post, check the postal service security policy. In some countries, the postal service exposes mail to high levels of radiation as a security measure. These high levels of radiation may erase the contents of the memory card and render it unusable.
- If a card is extracted without generating a rising edge of the bit %S65 and without checking that the memory card access green LED is OFF, the data (files, application, and so on) may be lost or become unreliable.

Memory Card Insertion Procedure

Procedure for inserting a memory card into a BME P58 •••• CPU:

Step	Description	Illustration
1	Open the SD memory card protective door by pulling the top of the door towards you.	

Step	Description	Illustration
2	Insert the card in its slot.	
3	Push the memory card until you hear a click. **Result:** The card should now be clipped into its slot. **Note:** Insertion of the memory card does not force an application restore.	

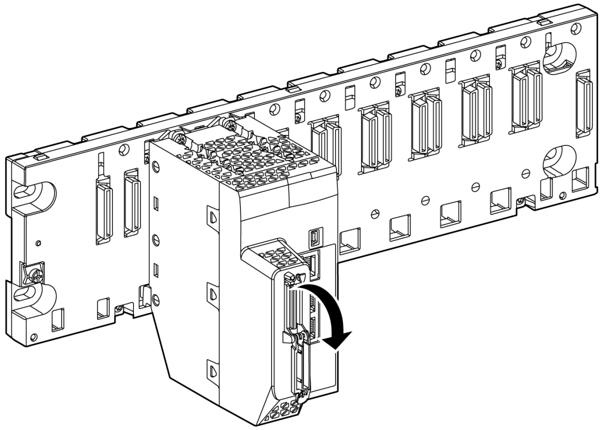
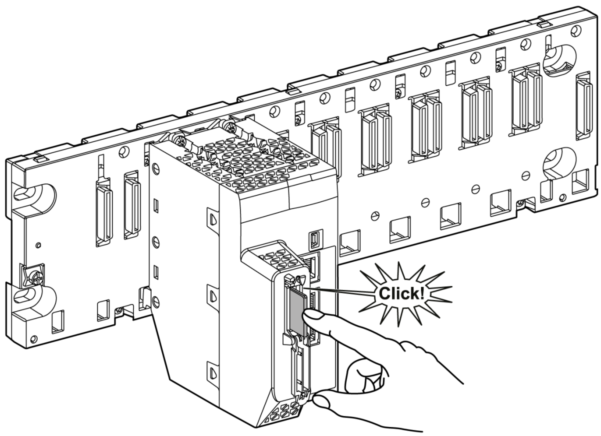
Step	Description	Illustration
4	Close the memory card protective door.	

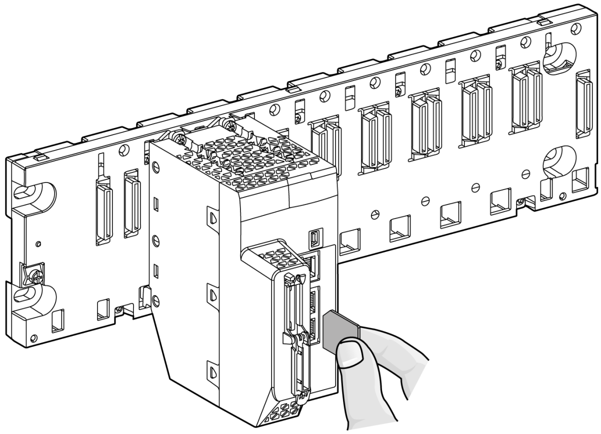
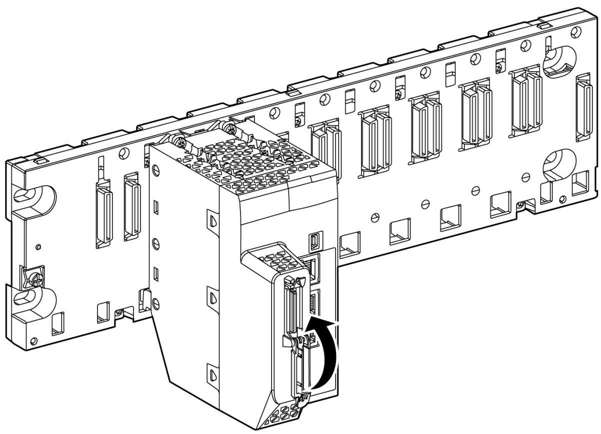
Memory Card Removal Procedure

NOTE: Before removing a memory card, a rising edge on bit %S65 needs to be generated. If a card is extracted without generating a rising edge of the bit `%S65` and without checking that the memory card access green LED is OFF, the data may be lost.

Procedure for removing a memory card from a BME P58 •••• CPU:

Step	Description	Illustration
1	Generate a rising edge on bit `%S65`.	–
2	Check that the memory card access green LED is OFF.	–

Step	Description	Illustration
3	Open the SD memory card protective door by pulling the top of the cover towards you.	
4	Push the memory card until you hear a click, then release the pressure on the card. **Result:** The card should unclip from its slot.	

Step	Description	Illustration
5	Remove the card from its slot. **Note:** The memory card access green LED is ON when the memory card is removed from the CPU.	
6	Close the memory card protective door.	

Chapter 7
M580 Diagnostics

Introduction

This chapter provides information on diagnostics that can be performed via hardware indications (based on LED status) and system bits or words when necessary. The entire M580 system diagnostics is explained in the *Modicon M580 System Planning Guide*.

The CPU manages different types of detected error:

- detected errors that can be recovered and do not change the PAC behavior unless specific options are used
- detected errors that cannot be recovered and lead the CPU to the halt state
- CPU or system detected errors that lead the CPU to an error state

What Is in This Chapter?

This chapter contains the following topics:

Blocking Errors

Introduction

Blocking detected errors caused during the execution of the application program do not cause system errors but they stop the CPU. The CPU goes into the HALT state *(see page 23)*.

Diagnostics

Visual indications of a blocking error are the **ERR** LED on the CPU front panel *(see page 35)*.

A description of the error is provided in system word `%SW125`.

The address of the instruction that was executing when the blocking error occurred is provided by system words `%SW126` through `%SW127`.

`%SW125` system word values and corresponding detected error description:

%SW125 Value (hex)	**Blocking Detected Error Description**
0•••	execution of an unknown function
0002	SD card signature feature (used with `SIG_CHECK` and `SIG_WRITE` functions)
2258	execution of the HALT instruction
2259	execution flow different than the reference flow
23••	execution of a CALL function towards an undefined subroutine
81F4	SFC node incorrect
82F4	SFC code inaccessible
83F4	SFC work space inaccessible
84F4	too much initial SFC steps
85F4	too much active SFC steps
86F4	SFC sequence code incorrect
87F4	SFC code description incorrect
88F4	SFC reference table incorrect
89F4	SFC internal index calculation detected error
8AF4	SFC step status not available
8BF4	SFC memory too small after a change due to a download
8CF4	transition/action section inaccessible
8DF4	SFC work space too small
8EF4	version of the SFC code older than the interpreter

%SW125 Value (hex)	Blocking Detected Error Description
8FF4	version of the SFC code more recent than the interpreter
90F4	poor description of an SFC object: NULL pointer
91F4	action identifier not authorized
92F4	poor definition of the time for an action identifier
93F4	macro step cannot be found in the list of active steps for deactivation
94F4	overflow in the action table
95F4	overflow in the step activation/deactivation table
9690	error detected in the application CRC check (checksum)
DE87	calculation detected error on numbers with decimal points
DEB0	watchdog overrun
DEF0	division by 0
DEF1	character string transfer detected error
DEF2	capacity exceeded
DEF3	index overrun
DEF7	SFC execution detected error
DEFE	SFC steps undefined

Restarting the Application

After a blocking error has occurred, the halted CPU needs to be initialized. The CPU can also be initialized by setting the %S0 bit to 1.

When initialized, the application behaves as follows:
- the data resume their initial value
- tasks are stopped at end of cycle
- the input image is refreshed
- outputs are controlled in fallback position

The RUN command then allows the application to be restarted.

Non-blocking Errors

Introduction

A non-blocking error is caused by an input/output error on the rack backplane bus (X Bus or Ethernet) or through execution of an instruction, which can be processed by the user program and does not modify the CPU status.

Errors Linked to I/O Diagnostics

A non-blocking error linked to the I/O is diagnosed with the following indications:

- CPU **I/O** LED pattern: steady ON
- module **I/O** LED pattern: steady ON
- system bits (type of error):
  - `%S10` set to 0: I/O error detected on one of the modules on the rack (channel power supply detected error, or broken channel, or module not compliant with the configuration, or inoperative module, or module power supply detected error)
  - `%S16` set to 0: I/O error detected in the task in progress
  - `%S40–%S47` set to 0: I/O error detected on rack address 0 to 7
- system bits and words combined with the channel having an error detected (I/O channel number and type of detected error) or I/O module Device DDT information (for modules configured in Device DDT addressing mode):
  - bit `%Ir.m.c.ERR` set to 1: channel error detected (implicit exchanges)
  - word `%MWr.m.c.2`: the word value indicates the type of error detected on the specified channel and depends on the I/O module (implicit exchanges)

Errors Linked to Execution of the Program Diagnostics

A non-blocking error linked to execution of the program is diagnosed with the following system bits and words:

- system bits (type of error detected):
  - `%S15` set to 1: character string manipulation error detected
  - `%S18` set to 1: capacity overrun, error detected on a floating point, or division by 0
  - `%S20` set to 1: index overrun
- system word (nature of the error detected):
  - `%SW125` *(see page 138)* (always updated)

NOTE: The CPU can be forced to the HALT state *(see page 23)* on program execution recoverable error.

There are 2 ways to force a CPU to stop when non-blocking errors linked to the execution of the program are detected:

- Use the diagnostic program function accessible through Unity Pro programming software.
- set the system bit `%S78` (HALTIFERROR) to 1.

CPU or System Errors

Introduction

CPU or system errors are related either to the CPU (equipment or software) or to the rack internal bus wiring. The system can no longer operate correctly when these errors occur.

A CPU or system error causes the CPU to stop in ERROR mode and requires a cold restart. Before applying a cold restart, set the CPU to STOP mode to keep the PAC from returning to ERROR mode.

Diagnostics

A CPU or system error is diagnosed with the following indications:
- CPU I/O LED pattern: steady on
- system word `%SW124` value defines the detected error source:
  - 80 hex: system watchdog error or rack internal bus wiring error
  - 81 hex: rack internal bus wiring error
  - 90 hex: interruption not foreseen, or system task pile overrun

CPU Application Compatibility

Application Compatibility

Ability to download and execute applications built on a different BME P58 CPU reference:

Download and Execute on the CPUs: An Application Built on the Following CPUs:	1020	2020	2040	3020	3040	4020	4040
1020	X	X	–	X	–	X	–
2020	–	X	–	X	–	X	–
2040	–	–	X	–	X	–	X
3020	–	–	–	X	–	X	–
3040	–	–	–	–	X	–	X
4020	–	–	–	–	–	X	–
4040	–	–	–	–	–	–	X
X yes – no							

An application built on a BME P58 3020 CPU can only be downloaded or executed on a BME P58 3020 or a BME P58 4020 CPU.

Part III
Configuring the CPU in Unity Pro

Introduction

This part describes how to configure a M580 system with Unity Pro.

What Is in This Part?

This part contains the following chapters:

Chapter 8
M580 CPU Configuration

Introduction

The chapter describes the configuration of the M580 CPU.

What Is in This Chapter?

This chapter contains the following sections:

Section 8.1
Unity Pro Projects

Overview

Use this section to add an M580 CPU to your Unity Pro application.

NOTE: For detailed information about using Unity Pro, refer to the online help and documentation DVD that come with Unity Pro.

What Is in This Section?

This section contains the following topics:

Creating a Project in Unity Pro

Introduction

You may have already created a project in Unity Pro and installed a power supply and an M580 CPU. If not, use the following steps to create a new Unity Pro project with these components:

- M580 CPU *(see page 17)*
- a power supply *(see page 79)*

Creating and Saving a Unity Pro Project

Create a Unity Pro project:

Step	Action
1	Open Unity Pro.
2	Open the **New Project** window (**File** → **New...**).
3	In the **PLC** window, expand the **Modicon M580** node and select a CPU. In the **Rack** window, expand the **Modicon M580 local drop** node and select a rack.
4	Click **OK**. Unity Pro displays the **Project Browser**.
5	Open the **Save As** dialog (**File** → **Save**).
6	Enter a **File name** to name your Unity Pro project and click **Save**. Unity Pro saves your project to the specified path location. (See the note below.)

Changing the Default Storage Location (Optional)

You can change the default location that Unity Pro uses to store project files before you click **Save** (above):

Step	Action
1	Open the **Options Management** window (**Tools** → **Options**).
2	In the left pane, navigate to **Options** → **General** → **Paths**.
3	In the right pane, type in a new path location for the **Project path**. You can also edit these items: ● **Import/Export file path** ● **XVM path** ● **Project settings templates path**
4	Click **OK** to close the window and save your changes.

Power Supply Selection

A default power supply is automatically added to the rack in a new Unity Pro project. To use a different power supply, follow these steps:

Step	Action
1	In the **Project Browser**, double-click **PLC Bus** to display a graphical representation of the hardware rack: ● The selected M580 CPU is in the second position. ● A default power supply appears in the first position. ● Unity Pro automatically opens the **Hardware Catalog** that corresponds to the **PLC bus** tab.
2	Select (left-click) the power supply in the **PLC bus**.
3	Press the **Delete** key to remove the power supply.
4	Double-click the first slot of the **PLC bus** to open the **New Device** list.
5	Double-click the preferred power supply to make it appear in the **PLC bus**.
6	Save your project (**File** → **Save**).

Configuring the Size and Location of Inputs and Outputs

Introduction

Use the following steps to configure the size and starting positions of inputs and outputs. Your own project configuration may differ.

Setting Global Addresses and Operating Mode Parameters

Edit the communication module inputs and outputs:

Step	Action
1	Double-click the left mouse button on the image of the M580 CPU in the **PLC Bus** to view its properties.
2	Select the **Configuration** tab.
3	You can check the **Operating mode** boxes to enable these parameters in your application: ● **Run/Stop input** (default: **Not Selected**) ● **Memory protect** (default: **Not Selected**) ● **Automatic start in Run** (default: **Not Selected**) ● **Initialize %MWi on cold start** (default: **Selected**) ● **Cold Start Only** (default: **Not Selected**)
4	Select the size of the global addresses: ● **%M** (maximum value: 32,634) ● **%MW** (maximum value: 65,232) ● **%KW** (maximum value: 32,760) ● **%S** (maximum value: 128) ● **%SW** (maximum value: 168)
5	Select the **Online modification in RUN or STOP** check box (in the **Configuration Online Modification** field) to use the change configuration on the fly (CCOTF) feature.
6	Select **Edit → Validate** (or click the ☑ toolbar button) to save the configuration.

NOTE: After you validate module settings for the first time, you cannot edit the module name. If you subsequently decide to change the module name, delete the existing module from the configuration, then add and rename a replacement module.

Completing the Ethernet Network Configuration

After you configure these settings, configure the CPU settings beginning with its Channel Properties *(see page 188)*. Then configure the Ethernet network devices.

Project Management

Downloading the Application to the CPU

Download the Unity Pro application to the CPU through one of its ports or through a connection to an Ethernet communication module:

Method	Connection
USB port	If the CPU and the PC that is running Unity Pro both have USB ports, you can download the application to the CPU directly through the USB ports *(see page 38)* (version 1.1 or later).
Ethernet port	If the CPU and the PC that is running Unity Pro both have Ethernet ports, you can download the application to the CPU directly through the Ethernet ports. (The PC and the CPU must be on the same network.)
communication module	You can download the application to the CPU by connecting Unity Pro to the IP address of the communication module.

NOTE: More details are provided in the *Downloading CPU Applications* topic in the *Modicon M580 System Planning Guide*.

Converting Legacy Applications To M580

For details on this conversion process, contact your Schneider Electric customer support.

Project Backup and Restore

Automatic and manual accesses are carried out between the CPU application RAM *(see page 262)* and the CPU flash memory (and the memory card if inserted) to:

- restore a project in the CPU from the flash memory (and the memory card if inserted):
  - automatically after a power cycle
  - automatically on a warm restart
  - automatically on a cold start
  - manually with Unity Pro command: **PLC** → **Project Backup** → **Backup Restore**.

  NOTE: If a memory card is inserted with a different application than the application in the CPU, the application is transferred from the memory card to the CPU application RAM when the restore function is carried out.

- save the CPU project in the flash memory (and the memory card if inserted):
  - automatically after an online modification is performed in the application RAM
  - automatically after a download
  - automatically on detection of `%S66` system bit rising edge
  - manually with Unity Pro command: **PLC** → **Project Backup** → **Backup Save**.

- compare the CPU project and the flash memory project:
  - manually with Unity Pro command: **PLC** → **Project Backup** → **Backup Compare**

NOTE:

When a valid memory card is inserted *(see page 45)* with a valid application, the application backup and restoral operations are performed as follows:

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

Section 8.2
Configuring the CPU with Unity Pro

Introduction

Use the instructions in this section to configure the M580 CPU in Unity Pro.

NOTE: Some configuration features for the M580 CPU are accessed through the Unity Pro **DTM Browser**. Those configuration instructions appear elsewhere in this document *(see page 186)*.

What Is in This Section?

This section contains the following topics:

Unity Pro Configuration Tabs

Accessing the Unity Pro Configuration Tabs

To access the CPU configuration parameters for RIO and distributed equipment, choose one of the following methods. First, in the **Project Browser**, double-click **Project → Configuration → PLC bus**, then:

- In the **PLC bus** dialog box, double-click the drawing with 3 Ethernet ports in the middle of the CPU.
- Double-click **PLC bus → BME XBP xxxx → BME P58 xxxx → EIO**.

NOTE: You can accept the automatically assigned IP address or change the IP address in the Unity Pro **IP Config** tab.

NOTE: Maximum Ethernet security is enforced by default. Before configuring Ethernet parameters, set up the required security level *(see page 156)* otherwise some services (firmware update, RIO, web access service) will not be available.

Unity Pro Configurable Parameters for RIO and Distributed Equipment

This table indicates the available Unity Pro configuration tabs for the M580 CPUs:

Unity Pro Tab	CPU with Embedded RIO Scanner	CPU Without Embedded RIO Scanner (DIO Scanner Only)
IPConfig	X	X
Security	X	X
RSTP	X	X
SNMP	X	X
NTP	X	X
Switch	–	X
QoS	–	X
Service Port	X	X
Advanced Settings	–	X
X yes – no		

About Unity Pro Configuration

Accessing Configuration Settings

Follow these steps to access the configuration settings for the M580 CPU in Unity Pro:

Step	Action
1	Open Unity Pro.
2	Open a Unity Pro project that includes a M580 CPU in the configuration.
3	Open the **Project Browser** (**Tools** → **Project Browser**).
4	Double-click **PLC bus** in the **Project Browser**.
5	In the virtual rack, double-click the Ethernet ports of the M580 CPU to see these configuration tabs: ● **IpConfig** ● **Security** ● **RSTP** ● **SNMP** ● **NTP** ● **Switch** ● **QoS** ● **Service Port** ● **Advanced Settings** These configuration tabs are described in detail in the following pages.

IPConfig Tab

IPConfig Parameters

IP address configuration field on the **IP Config** tab:

Parameter	Default Value	Description
Main IP address	192.168.10.1	the IP address of the CPU
IP address A	192.168.11.1	the IP address of the RIO/DIO scanner **NOTE:** If you change **IP address A**, the system may recalculate all IP addresses (including those of the drops) to keep all devices in the same subnetwork.
IP address B	–	used for Hot Standby
Subnetwork mask	255.255.0.0	This bit mask identifies or determines the IP address bits that correspond to the network address and the subnetwork portion of the address. (The value can be changed to any valid value in the subnetwork.)
Gateway address	192.168.10.1 (when not used)	This is the IP address of the default gateway to which messages for other networks are transmitted.

CRA IP address configuration field on the **IP Config** tab:

Parameter	Description
Drop N°	drop number
Device Name	device name (for the (e)X80 adapter module)
IP Address	When an RIO drop is added, the adapter module is automatically assigned an IP address. (You can change this IP address in the **IP Address** column, but we recommend that you accept the automatically assigned IP address.)

Security Tab

Introduction

Security tab:

Field	Parameter	Value	Comment
FTP	–	**Disabled** (default)	Disable firmware upgrade, SD memory card data remote access, data storage remote access, and device configuration management using FDR service. **NOTE:** Data storage is still operational.
		Enabled	–
TFTP	–	**Disabled** (default)	Disable the ability to read RIO drop configuration and device configuration management using FDR service for a few devices.
		Enabled	–
HTTP	–	**Disabled** (default)	Disable the web access service.
		Enabled	–
Achilles level 2	–	**Enabled** (default)	Enable this feature to increase Ethernet frame filtering to improve the level of security and robustness.
		Disabled	Disable this feature to increase increases system performance by reducing the Ethernet frame filtering capability.
Access Control	–	**Enabled** (default)	Deny Ethernet access to the Modbus and Eip server by unauthorized network devices.
		Disabled	–
Enforce Security and **Unlock Security**	–	–	(See the following paragraph for more details.)
Authorized addresses (1.)	**IP Address**	0.0.0.0 ... 255.255.255.255	See the information for security fields (below).
	Subnet	**Yes/No**	
	Subnet mask	0.0.0.0 ... 255.255.255.255	
1. Field can be modified when **Access Control** field is set to **Enabled**.			

Schneider Electric recommends disabling services that are not being used.

For cyber security details, refer to the *Cyber Security* chapter in the *Modicon M580 System Planning Guide*.

NOTE: Set the **Security** tab parameters before you download the application to the CPU. The default settings (maximum security level) reduce the communication capacities and port access.

Enforce Security and Unlock Security Fields

Click **Enforce Security** to set previous fields to the maximum security level. This is the **Security** tab default setting.

- **FTP**, **TFTP**, and **HTTP** are set to **Disabled**
- **Achilles level 2** and **Access Control** are set to **Enabled**

Click **Unlock Security** to set previous fields to the minimum security level:

- **FTP**, **TFTP**, and **HTTP** are set to **Enabled**
- **Achilles level 2** and **Access Control** are set to **Disabled**

NOTE: Each field can be set individually once the global setting is applied.

Defining the List of Authorized Addresses

The list of authorized addresses applies to the only devices that can communicate with the M580 CPU via the port 502 server or the EtherNet/IP server. The list also applies to CPU firmware downloads.

When access control is enabled, add the IP addresses of the authorized addresses. Devices can only communicate with authorized addresses. To define the list of authorized addresses, you can enter one of the following:

- an IP address in the **IP Address** table column with **NO** selected in the **Subnet** column
- a subnet address in the **IP Address** table column with **YES** selected in the **Subnet** column and a subnet mask entered in the **Subnet Mask** column

NOTE: The subnet in the **IP Address** column can be the subnet itself or any IP address of the subnet. If you enter a subnet without a subnet mask, a detected error displays stating that the screen cannot be validated.

You can enter up to 128 authorized IP addresses.

RSTP Tab

RSTP Parameters for CPU With RIO Scanner

Use RSTP to design a network with redundant wiring so that RIO and distributed equipment communication automatically finds an alternate path if a communication disruption occurs (for example, a cable breaks or a device becomes inoperable). This method does not require you to manually enable or disable communication paths.

Changing these parameters can affect sub-ring diagnostics, I/O determinism, and network recovery times.

RSTP tab:

Field	Parameter	Value	Comment
RSTP Operational State	Bridge Priority	Root(0)	default
		Backup Root(4096)	reserved
		Participant(32768)	–

RSTP Parameters for CPU Without RIO Scanner (DIO Scanner Only)

The Ethernet ports on the front of the M580 CPU support the *rapid spanning tree protocol* (RSTP). RSTP is an OSI layer 2 protocol defined by IEEE 802.1D 2004. RSTP performs these services:

- It creates a loop-free logical network path for Ethernet devices that are part of a topology that includes redundant physical paths.
- It automatically restores network communication by activating redundant links in the event the network experiences a loss of service.

RSTP software, operating simultaneously in all network switches, obtains information from each switch, which enables the software to create a hierarchical logical network topology. RSTP is a flexible protocol that can be implemented on many physical topologies, including ring, mesh, or a combination of ring and mesh.

NOTE: RSTP can be implemented only when all network switches are configured to support RSTP.

RSTP tab:

Field	Parameter	Value	Comment
RSTP Operational State	Bridge Priority	Root(0)	–
		Backup Root(4096)	–
		Participant(32768)	default

Field	Parameter	Value	Comment
Bridge parameters	Force version	2	you cannot edit this value.
	Forward delay (ms)	21000	
	Maximum Age Time (ms)	40000	
	Transmit Hold Count	40	
	Hello Time (ms)	2000	
Port 3 Parameters	–	–	you cannot edit these field parameters
Port 4 Parameters	–	–	You cannot edit these field parameters

SNMP Tab

Introduction

Use the **SNMP** tab in Unity Pro to configure SNMP parameters for these modules:
- embedded RIO/DIO scanner
- (e)X80 adapter module on the RIO drop

An SNMP v1 agent is a software component of the SNMP service that runs on these modules to allow access to diagnostic and management information for the modules. You can use SNMP browsers, network management software, and other tools to access this data. In addition, the SNMP agent can be configured with the IP addresses of 1 or 2 devices (typically PCs that run network management software) to be the targets of event-driven trap messages. Such messages inform the management device of events like cold starts and the inability of the software to authenticate a device.

Use the **SNMP** tab to configure the SNMP agents for communication modules in the local rack and RIO drops. The SNMP agent can connect to and communicate with 1 or 2 SNMP managers as part of an SNMP service. The SNMP service includes:
- authentication checking by the Ethernet communication module, of any SNMP manager that sends SNMP requests
- management of events or traps

SNMP Parameters

These parameters are found on the Unity Pro **SNMP** tab:

Field	Parameter	Value	Description
IP address managers	**IP address manager1**	0.0.0.0 ... 255.255.255.255	The address of the first SNMP manager to which the SNMP agent sends notices of traps.
	IP address manager 2		The address of the second SNMP manager to which the SNMP agent sends notices of traps.
Agent	**Location (SysLocation)**	31 characters (maximum)	device location
	Contact (SysContact)		description of the person to contact for device maintenance
	Enable SNMP Manager	check box selected or deselected	check box deselected (default): You can edit the **Location** and **Contact** parameters. check box selected: You cannot edit the **Location** and **Contact** parameters.
Community names	**Set**	15 characters (maximum)	password that the SNMP agent requires to read commands from an SNMP manager (default = **public**)
	Get		
	Trap		

Field	Parameter	Value	Description
Security	Enable “Authentication failure” trap	check box selected or deselected	check box deselected (default): not enabled check box selected: The SNMP agent sends a trap notification to the SNMP manager if an unauthorized manager sends a **Get** or **Set** command to the agent.

Online Behavior

You can perform tests online to verify that the IP addresses of the managers are not:
- multicast
- loopback
- broadcast

NTP Tab

Introduction

When the PAC is configured as an NTP client, the network time service (SNTP) synchronizes the clock in the M580 CPU to that of the time server. The synchronized value is used to update the clock in the CPU. Typical time service configurations utilize redundant servers and diverse network paths to achieve high accuracy and reliability.

When the PAC is configured as an NTP server, it can synchronize client clocks (such as a BM• CRA 312 00 adapter). The CPU's internal clock is then used as reference clock for NTP services. When only BM• CRA 312 00 adapters are configured as NTP clients, the accuracy of this server allows time discrimination of 20 ms.

NOTE: Refer to the *Modicon M580 Remote I/O Installation and Configuration Guide* for detailed information about timestamping performance.

These are some features of the time synchronization service:
- periodic time correction obtained from the reference-standard time server
- automatic switchover to a backup (secondary) time server if an error is detected with the normal time server system
- controller projects use a function block to read the accurate clock, allowing project events or variables to be time stamped

NTP Parameters for CPU With RIO Scanner

NTP tab:

Field	Parameter	Value	Comment
NTP	–	**Disabled**	default: no access to the NTP configuration and **NTP Server Configuration** tab
		NTP Client	The RIO/DIO scanner consumes NTP data and the **NTP Server Configuration** tab needs to be configured.
		NTP Server	The RIO/DIO scanner acts as an NTP server.
NTP Server Configuration	**Primary NTP Server IP address**	192.168.11.1	the IP address of the NTP server, from which the RIO/DIO scanner first requests a time setting
	Secondary NTP Server IP address	0.0.0.0	the IP address of the backup NTP server, from which the RIO/DIO scanner requests a time setting after not receiving a response from the primary NTP server
	Polling Period	20	The time (in seconds) between updates from the NTP server. To obtain optimal accuracy (and if the network allows it), reduce the polling rate to small values.

NTP Parameters for CPU Without RIO Scanner (DIO Scanner Only)

NTP tab:

Field	Parameter	Value	Comment
NTP	–	**Disabled**	default: no access to the NTP configuration and **NTP Server Configuration** tab
		NTP Client	The RIO/DIO scanner consumes NTP data and the **NTP Server Configuration** tab needs to be configured.
		NTP Server	The RIO/DIO scanner acts as an NTP server.
NTP Server Configuration	–	–	you cannot edit these field parameters

NTP Client Mode

To establish the accurate Ethernet system network time, the system performs the following at power up:

- requires the CPU to boot
- uses the CPU to obtain the time from the SNTP server
- requires a predefined interval until time is accurate; your configuration determines how long before time is accurate
- may require several updates to achieve peak accuracy

Once an accurate time is received, the service sets the status in the associated time service register.

The time service clock value starts at 0 until fully updated from the CPU.

Model	Starting Date
Modicon M580 with Unity Pro	January 1st 1980 00:00:00.00

Stop or run PAC:

- Stop and run have no effect on the accuracy of the clock.
- Stop and run have no effect on the update of the clock.
- A transition from one mode to the other has no effect on the accuracy of the Ethernet system network time.

Download application:

- The status clock value associated with the time service register in the M580 CPU is reinitialized after an application is downloaded or after an SNTP server swap. The time is accurate after 2 polling periods.

Switch Tab

Description

The **Switch** tab is only available for CPUs without RIO scanner (DIO scanner only).

Switch tab:

Field	Parameter	Value	Comment
ETH1	–	–	you cannot edit these field parameters
ETH2	Enabled	Yes	default
		No	–
	Baud Rate	Auto 10/100 Mbits/sec	–
		100 Mbits/sec Half duplex	–
		100 Mbits/sec Full duplex	–
		10 Mbits/sec Half duplex	–
		10 Mbits/sec Full duplex	–
ETH3	Enabled	Yes	default
		No	–
	Baud Rate	Auto 10/100 Mbits/sec	–
		100 Mbits/sec Half duplex	–
		100 Mbits/sec Full duplex	–
		10 Mbits/sec Half duplex	–
		10 Mbits/sec Full duplex	–
Backplane	–	–	you cannot edit these field parameters

NOTE: ETH1 port is a dedicated service port and the Ethernet backplane network is dedicated to the communication between modules on the rack. The switch parameters for those 2 ports cannot be configured in the **Switch** tab.

QoS Tab

Description

The M580 CPU can be configured to perform Ethernet packet tagging. The CPU supports the OSI layer 3 quality of service (QoS) standard defined in RFC-2475. When you enable QoS, the M580 CPU adds a *differentiated services code point* (DSCP) tag to each Ethernet packet that it transmits to indicate the priority of that packet.

QoS Tab

The **QoS** tab is only available for CPUs without RIO scanner (DIO scanner only).

Field	Parameter	Value	Comment
802.1Q Tagging	–	Enabled	default
		Disabled	–
PTP	DSCP PTP Event Priority	59	–
	DSCP PTP General Priority	47	–
Ethernet IP Traffic	DSCP Value For I/O Data Schedule Priority Messages	47	–
	DSCP Value For Explicit Message	27	–
	DSCP Value For I/O Data Urgent Priority Messages	55	–
	DSCP Value For I/O Data High Priority Messages	43	–
	DSCP Value For I/O Data Low Priority Messages	31	–
Modbus TCP Traffic	DSCP Value For I/O Messages	43	–
	DSCP Value For Explicit Message	27	–
Network Time Protocol Traffic	DSCP Value For Network Time Protocol Messages	59	–

QoS tagging lets you prioritize the Ethernet packet streams based on the type of traffic in that stream.

To implement QoS settings in your Ethernet network:
- Use network switches that support QoS.
- Consistently apply DSCP values to network devices and switches that support DSCP.
- Confirm that switches apply a consistent set of rules for sorting DSCP tags, when transmitting and receiving Ethernet packets.

Service Port Tab

Service Port Parameters

These parameters are on the Unity Pro **Service Port** tab:

Field	Parameter	Value	Comment
Service Port	–	**Enabled**	Enable port and edit port parameters.
	–	**Disabled**	Disable port (no access to parameters).
Service Port Mode	–	**Access**	default This mode supports Ethernet communications.
	–	**Mirroring**	In port mirroring mode, data traffic from one or more of the other ports is copied to this port. A connected tool can monitor and analyze port traffic. **NOTE:** In this mode, the **Service** port acts like a read-only port. That is, you cannot access devices (ping, connection to Unity Pro, and so on) through the **Service** port.
Access Port Configuration	**Service Port Number**	**ETH1**	You cannot edit the value in the **Service Port Number** field.
Port Mirroring Configuration	**Source Port(s)**	**Internal Port**	all Ethernet traffic for the module
		ETH2	RIO/DIO scanner: Ethernet traffic through the first device network port
			adapter modules: Ethernet traffic through the first device network port
		ETH3	RIO/DIO scanner: Ethernet traffic through the second device network port
			adapter modules: Ethernet traffic through the second device network port
		Backplane Port	RIO/DIO scanner: Ethernet traffic through the Ethernet port on the backplane

On Line Behavior

The **Service Port** parameters are stored in the application, but you can reconfigure (change) the parameters in connected mode. Values that you reconfigure in connected mode are sent to the (e)X80 adapter module or the RIO/DIO scanner in explicit messages.

The changed values are not stored, so a mismatch can exist between the parameters that are being used and those that are in the stored application.

If the module does not respond to the explicit messages, a message appears.

Limitations

The **Service** port on the CPU and the adapter modules have the same limitations as the cloud port of the dual-ring switch (DRS). Therefore, the CPU **Service** port and the DRS's cloud port can be connected to the same equipment.

The maximum load the module can process from distributed equipment:
- 5 Mbps: per **Service** port
- 20 Mbps: total distributed equipment traffic on the main ring

For more information about considerations that apply to the use of the distributed equipment cloud port on the DRS and the **Service** port on the CPU, refer to the *DRS Predefined Configuration Files* topic in the *Modicon M580 System Planning Guide*.

Advanced Settings Tab

Description

The **Advanced Settings** tab is only available for CPUs without RIO scanner (DIO scanner only).

Advanced Settings tab:

Field	Parameter	Value	Comment
EtherNet/IP Timeout Settings	**FW_Open I/O Connection Timeout** (msec)	4960	–
	FW_Open EM Connection Timeout (msec)	3000	–
	EM Connection RPI (msec)	10000000	–
	EM Request Timeout (sec)	10	–
EtherNet/IP Scanner Behavior	**Allow RESET via explicit message**	**Disabled**	default
		Enabled	–
	Behavior when CPU state is STOP	**Idle**	default
		STOP	–

Section 8.3
The Unity Pro FDT/DTM Interface

Overview

The section describes the use of DTMs within Unity Pro.

What Is in This Section?

This section contains the following topics:

Device Type Managers

Definition

A device type manager (DTM) is a device driver provided by a field device vendor. The DTM contains device-specific information and provides a graphical user interface.

In an M580 configuration, you can configure the properties in the DTM to monitor and configure the corresponding EtherNet/IP or Modbus TCP device.

A DTM is not a standalone application. It requires a field device tool (FDT) frame application to run.

DTMs in Unity Pro

Unity Pro incorporates the FDT/DTM approach to integrating intelligent remote devices into your process control application:

- The FDT container in Unity Pro interfaces to any module with a DTM.
- When you add a device to a Unity Pro configuration, the DTM that corresponds to that device appears in the Unity Pro **DTM Browser** (**Tools → DTM Browser**). From the **DTM Browser** you can open the **Device Editor** to configure the parameters in the DTM

Device manufacturers may provide a DTM for each of its EtherNet/IP or Modbus TCP devices or other products. However, an EtherNet/IP or Modbus TCP device without a DTM can be configured in Unity Pro with one of these methods:

- Configure a generic DTM provided by Unity Pro.
- Import the device's EDS file *(see page 235)*. A generic EDS DTM is automatically installed when you set up Unity Pro. Unity Pro populates the DTM parameters based on the content of the imported EDS file.

DTM Types

There are two kinds of DTMs:

- **master (communication) DTM:** This DTM is both a device DTM and a communication DTM. The master DTM is a pre-installed component of Unity Pro.
- **generic DTM:** The Unity Pro FDT container is the integration interface for any device's communication DTM.

Ethernet Configuration Tool User Interface

Overview

Use the M580 DTM to configure and monitor the M580 CPU and network communications:

- **Device Editor**: Configure Ethernet communication modules, remote devices, and their common Ethernet connections.
- **Diagnostic window**: Monitor the real-time operation of network devices and diagnose their condition.

Connecting and Disconnecting a Device or Module DTM

A device or module DTM can be connected to or disconnected from the physical device or module:

- **connected**: Use the DTM to monitor and diagnose the real-time operations of the device or module.
- **disconnected**: Use the DTM to configure a communications module or a remote device by editing its properties.

You can connect the M580 DTM to or disconnect a DTM from the M580 CPU with the contextual pop-up menu in the **DTM Browser** (**Tools → DTM Browser**). A connected DTM is displayed in **bold** text in the **DTM Browser**. A disconnected DTM is displayed in normal text.

Place Unity Pro in the online or offline operating mode with commands in the Unity Pro **PLC** menu.

Connect the M580 DTM to or disconnect it from the M580 CPU:

Step	Action
1	In the **DTM Browser** select the M580 DTM.
2	Right-click to open a pop-up menu.
3	Scroll to one of these commands: ● **Connect** ● **Disconnect** **NOTE:** You can also use the Unity Pro menu (**Edit → Connect** or **Edit → Disconnect**).

Device Editor

The properties that are available for viewing and editing depend upon the device that is selected in the **DTM Browser**. Use the **Device Editor** to display and configure device properties:

- **connected**: When the communication module and DTM are connected, the **Device Editor** opens in read/write mode.
- **disconnected**: When the communication module and DTM are disconnected, the **Device Editor** opens in read-only mode.

Use these steps to access the **Device Editor**. (You may have to disconnect the Ethernet communication module from its DTM.)

Step	Action
1	In the **DTM Browser**, select the Ethernet communication module node.
1	Right-click and scroll to **Disconnect**.
2	Right-click on the module again and scroll to **Device menu → Configuration** to open the **Device Editor** window.

Diagnostic Window

Use the **Diagnostic Window** to monitor hardware performance:

- Colored LED icons indicate the operating status of the Ethernet communication module, remote devices, and their connections.
- View diagnostic data for the communication module, local slaves, and Ethernet connections.

The **Diagnostic Window** can be displayed only when the communication module is connected to its DTM.

Access the **Diagnostic Window** from the **DTM Browser**. If necessary, you may need to first connect the Ethernet communication module to its DTM.

Step	Action
1	In the **DTM Browser**, select the Ethernet communication module node and click the right mouse button, then select **Connect** in the pop-up menu.
2	In the **DTM Browser**, again select the Ethernet communication module node and click the right mouse button. The same pop-up menu opens.
3	Select **Device menu → Diagnostics** in the pop-up menu to open the **Diagnostic Window**.

DTM Browser

Overview

The **DTM Browser** displays a hierarchical list of DTMs that have been added to your Unity Pro project as nodes on a connectivity tree. Each DTM node corresponds to a module or device in your Ethernet network.

Node Types

View the DTM type:

Step	Action
1	Open the **DTM Browser** in Unity Pro (**Tools → DTM Browser**).
2	Right-click the device DTM in the **DTM Browser**.
3	Scroll to **Properties** to open the DTM properties.
4	Open the **Device information** tab to see the DTM **Type** in the list of **Properties**.

This table describes the DTM node types shown in the **Type** description on the **Device Information** tab:

DTM	Description
communication	Any communication DTM can be plugged directly under the root node (Host PC) and is at the first level. A communication DTM supports gateway DTMs or device DTMs as children if their protocols are compatible.
device	A device DTM manages the corresponding target device.
master communication	A master communication DTM is both a device DTM and a communication DTM. The master DTM is a pre-installed component of Unity Pro.
gateway (see note)	A gateway DTM (defined in the FDT specification) supports other gateway DTMs or device DTMs as children if their protocols are compatible. A gateway DTM requires communication capabilities from a parent DTM.

NOTE: Unlike the device DDT for typical end-point devices, the input and output structures in the gateway device DDT can be the same and can be reused across instances of different gateway DTMs when the I/O layout does not change. (This is not a fixed behavior. The gateway DTM can generate different DDTs for inputs and outputs when instructed to do so.)

Node Names in the DTM Browser

When it is inserted into the **DTM browser**, a DTM has a default name with these components: <channel: address> device name:

Element	Description
channel	This is the name of the protocol that uses the DTM to communicate with the target device. This name is read from the DTM and is set by the device vendor.
address	The bus address of the device can be either of these: ● the connection point on its parent gateway network ● the slot number in the modular device parent internal bus **Example**: the device IP address for Ethernet-based protocols
device name	The default name is determined by the vendor in the device DTM, but it can be edited by the user.

Node Status

The **DTM Browser** graphically displays the status of each DTM node in the connectivity tree:

Status	Description
Built/Not-built	A blue check mark superimposed on a device icon in the **DTM Browser** indicates that a modification has been made to the DTM since the last build of the Unity Pro application.. This means that some property of the node has changed, and the information stored in the physical device is no longer consistent with the local project.
Connected/ Disconnected	A connected DTM is denoted in **bold** text. An unconnected DTM appears in plain text. **NOTE:** ● Connecting a DTM to its physical device automatically connects all higher level parent nodes up to the root node. ● Disconnecting a DTM from its physical device automatically disconnects all its lower level child nodes. **NOTE:** Connecting or disconnecting a DTM to or from its device does not also connect or disconnect Unity Pro to or from the PAC. DTMs can be connected/disconnected while Unity Pro is either offline or online.
Installed/Not-installed	A red superimposed on a device icon in the **DTM Browser** indicates that the DTM for that device is not installed on the PC.

Handling Invalid Nodes

If the red appears (above), install the corresponding DTM on your PC and close the Unity Pro application. Reopen Unity Pro and update the catalog *(see page 237)*.

DTM Browser Menu Commands

Overview

The **DTM Browser** includes a pop-up, contextual (right-click) menu that displays commands for the currently selected node in the browser. The list of available commands consists of:

- universal commands, as determined by the selected node level:
  - host PC node (level 1)
  - communication module node (level 2)
  - remote device node (level 3)
- device-specific commands, as determined by the device DTM

Host PC Node Commands

Right-click the **Host PC** node in the **DTM browser** to access these commands:

Name	Description
Add...[1]	Open the **Add** dialog. It contains a subset of the **Hardware Catalog**, from which you can select a communication module DTM.
Check DTM devices[1]	Check the current project for invalid DTMs or DTMs that are not installed in the PC. If the results include invalid or not-installed DTMs, they are displayed in the **User errors** tab in the information window and a red **X** is superimposed over the corresponding icons in the **DTM Browser**.
DTM services...	Display the communication DTMs selection, as well as the device topology, their respective IP addresses, and connection state. In this dialog, for each device you can connect, disconnect, load from devices, or store to devices. You can also choose to stop communication or continue activity when detected errors occur.
DTM hardware catalog...	Display the **DTM catalog** tab of the **Hardware Catalog** dialog.
Expand all[2]	Display every DTM in the project.
Collapse all[2]	Display only the communication DTMs in the project.
1. This command also appears in the Unity Pro **Edit** menu. 2. This command also appears in the Unity Pro **View** menu.	

Communication Module and Remote Device Node Commands

Right-click the M580 CPU in the **DTM Browser** to perform these tasks:

Name	Description
Open[1]	Open the **Device Editor** for the selected CPU. **NOTE:** You can also double-click the left mouse button on the DTM in the **DTM Browser** to open this window.
Add... [1]	Open the **Add** dialog to display the subset of the **Hardware Catalog** that allows you to select a device or gateway DTM that needs to be added under the selected DTM. **NOTE:** Unity Pro filters the content of the **Add** dialog so that it displays only the device and gateway DTMs that are supported by the parent communication DTM.
Delete[1]	If the selected DTM allows this function, this deletes the selected DTM and its sub-node DTMs from the DTM topology.
Field bus discovery	This scans the connected physical devices to create the corresponding field bus topology.
Sort by address	The configured physical devices are scanned and listed by their respective IP addresses.
Connect[1]	This connects the DTM to its physical device on the network. This connection does not depend on the PAC online/offline status of the Unity Pro project application. **NOTE:** Connecting a gateway or device DTM implicitly connects its parent DTM.
Disconnect[1]	This disconnects the DTM from its physical device. This disconnection depends on the PAC online/offline status of the Unity Pro project application. **NOTE:** Disconnecting a gateway or device DTM implicitly disconnects its parent DTM.
Load data from device[1]	This loads data from the corresponding physical device on the network to the DTM.
Store data to device[1]	This loads data from the DTM to the corresponding physical device on the network.
Copy	This command is disabled.
Paste	This command is disabled.
Device menu	This command opens a sub-menu that contains device-specific commands, as determined by the device vendor.
Properties[1]	Opens the DTM's **Properties** window.
Print device[1]	If this optional function is supported by a DTM, this function displays the detailed device information (including configuration settings) in the PC's default internet browser, which can then be printed. **NOTE:** Device information can be printed: ● for only one device DTM at a time. ● only when the DTM is disconnected from the physical device.
1. This command also appears in the Unity Pro **Edit** menu. 2. This command also appears in the Unity Pro **View** menu.	

Name	Description
Zoom in[2]	Make this selection to display only the selected module in the connectivity tree of the **DTM Browser**.
Zoom out[2]	Make this selection to display the entire connectivity tree of the **DTM Browser**.
Expand all[2]	This displays all child DTMs below the selected DTM.
Collapse all[2]	This displays only the selected DTM.
1. This command also appears in the Unity Pro **Edit** menu. 2. This command also appears in the Unity Pro **View** menu.	

Communication Module Commands

Find the **Device menu** list:

Step	Action
1	Select the M580 DTM in the **DTM Browser**.
2	Right-click and scroll to **Device menu**.
3	Examine the list of possible commands.

This is the **Device menu** sub-menu:

Name	Description
Offline Parameter	This command is enabled or disabled according to the DTM.
Online Parameter	This command is enabled or disabled according to the DTM.
Compare	When supported, this command compares 2 devices, either online or offline.
Configuration	This opens the **Device Editor** for the selected communication module when the module and its DTM are disconnected.
Observe	This command is enabled or disabled according to the DTM.
Diagnosis	This opens the **Diagnosis Window** for the selected communication module when the module and its DTM are connected.

Name		Description
Additional functions	**Add EDS to library**	Open the **EDS File Wizard** to add a device EDS file to the Unity Pro EDS device library. Unity Pro displays the contents of EDS files as DTMs for use in the **DTM Browser** and **Device Editor**.
	Remove EDS from library	Removes a device EDS file from the Unity Pro EDS device library.
	FTP Upload	(reserved)
	FTP Download	(reserved)
	TFTP Upload	(reserved)
	TFTP Download	(reserved)
	Online Action (see note)	Open the **Online Action** window. Depending upon the protocol(s) a remote device supports, you can use the **Online Action** window to: ● Ping a remote EtherNet/IP or Modbus TCP device ● view and write Ethernet parameters in a remote EtherNet/IP device ● view and write to port configuration properties in a remote EtherNet/IP device
	EtherNet/IP Explicit Message (see note)	Open the **Ethernet/IP Explicit Message** window to send explicit messages to EtherNet/IP remote devices.
	Modbus Explicit Message (see note)	Open the **Modbus TCP Explicit Message** window to send explicit messages to Modbus TCP remote devices.
	About	Get information about the selected hardware (model, version, manufacturer, etc.).
NOTE: These commands are available only when the `DTM` is in the connected state.		

Fieldbus Discovery Service

Introduction

Use the field bus discovery service (in the table below) with your Unity Pro application to detect and add control network devices that are situated on a network. This discovery service is available only when the M580 DTM is physically connected to its PAC.

Only first-level devices below the communication DTM are detected.

Performing Field Bus Discovery

The results of the scanning process is compared to the registered DTMs in the DTM catalog of the computer. If a match is found in the DTM catalog for a scanned device, the results are accompanied with a matching type that gives the accuracy of the match.

These are the available matching types:

- **exact match:** All identification attributes match. The correct device type was found.
- **generic match:** At least the **Vendor** and device **Type ID** attributes match. The support level of DTM is `Generic Support`.
- **uncertain match:** At least the **Vendor** and device **Type ID** attributes match. The support level of the DTM is **not** `Generic Support`.

Use the field bus discovery service:

Step	Action
1	In the **DTM Browser**, right-click the M580 CPU commuication DTM.
2	Scroll to **Field bus discovery** to open the **Field bus discovery** window.
3	Select a channel and a protocol: ● if the DTM has more than one channel ● if the channel supports more than one protocol
4	Click **OK** to detect devices on the selected channel. **NOTE:** The field bus discovery service limits its search to only the range of IP addresses that is pre-configured for the selected channel in the **Channel Properties** page *(see page 188)*.
5	If at least one matched device has been found, the **Field bus discovery** dialog displays a list of **Scanned Devices**.
6	Use the controls of the **Field bus discovery** dialog to select the devices to add to your Unity Pro application.
7	The device properties dialog opens, displaying the default name for the first discovered device to be added. In the **General** page of the device properties dialog, type the **Alias name** for the device to be added, then click **OK**. **Result**: The dialog closes, then re-opens if there is another device to be added to the application. **NOTE:** Repeat this step for each additional discovered device.

Step	Action
9	After all devices have been added to the application, configure each device for operation as part of the application: ● Disconnect the Ethernet communication module from its DTM. In the **DTM Browser**, select the Ethernet communication module, then select **Edit → Disconnect**. ● Configure the new device properties in the DTMs for both the Ethernet communication module and the newly added remote device.

Field Bus Discovery Dialog

If at least one matched device has been found, the **Field bus discovery** dialog box is displayed listing the scanned and matched devices. Select the matched devices to be created in the Unity Pro project (which then shows up in the **Selected Devices** list.

This dialog presents these lists:

List	Content
Scanned Devices	This list contains all devices (matched and unmatched) that were found during the scan.
Matched Devices	This list contains the matched DTMs found in the workstation DTM catalog for the device that you selected in the **Scanned Devices** list. Each time a scanned device is selected in the **Scanned Devices** list, the contents of the **Matched Devices** list is updated to display the matched device DTMs found for the selected scanned device. The matching process can yield one or more matched devices for a given scanned device. In this case, only one DTM was discovered for the selected scanned device.
Selected Devices	This list displays the device DTMs that have been selected in the **Matched Devices** list, which will be added to the Unity Pro project.

The lists use these colored icons:

Color	Indication
green	The device has been selected.
yellow	The device has been matched.
red	The device has **not** been matched.
black	Information about the address of the scanned device: ● In the **Scanned Devices** list, the device has an address identical to one of the DTMs in the Unity Pro project. ● In the **Matched Devices** list, the device will be assigned an address identical to one of the DTMs in the Unity Pro project.
NOTE: An icon can consist of 2 colors. A search can discover a device that has a matching DTM and an IP address that is identical to a device that has already been added to the Unity Pro application. In that case, the icon next to the discovered device would be half yellow and half black before it is selected and half green and half black after it is selected.	

This dialog has these buttons:

Button	Use this button to...
Add All 	Automatically add the respective (according to the matching types listed above) device DTM for each found device in the **Matched Devices** list to the **Selected Devices** list.
Add One 	Add the matched device DTM selected in the **Matched Devices** list.
Remove 	Remove one or more devices from the **Selected Devices** list.
OK	Insert the device DTMs in the **Selected Devices** list into the Unity Pro project. If there are one or more devices in the **Selected Devices** list that have he same address in the Unity Pro project, a message box opens asking if you want to continue. If you click **OK**, all devices in the Unity Pro project that have identical addresses as the selected devices are **deleted** and **replaced** by the DTMs selected in the **Selected Devices** list.
Cancel	Cancel the field bus discovery scan and do nothing. All information in the 3 lists is discarded.

Configuring DTM Properties

Introduction

You can edit and view parameters in the **Device List** that is associated with the M580 DTM.

Open the Device List

View the **Device List**:

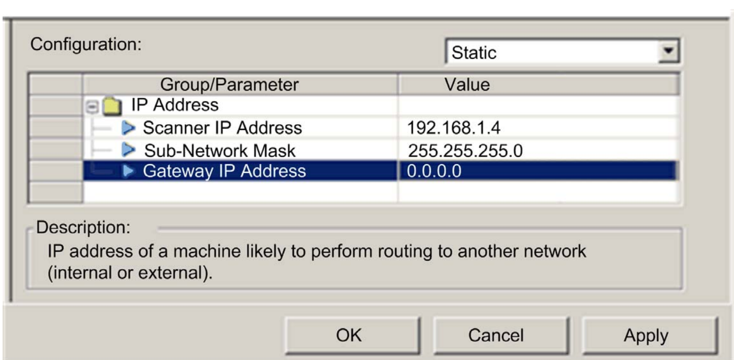
Step	Action
1	Open the **DTM Browser** in Unity Pro (**Tools** → **DTM Browser**).
2	Double-click the M580 DTM in the **DTM Browser**.
3	In the configuration tree associated with the M580 DTM, click **Device List**.

The **Device Editor** displays these icons next to the device properties:

Icon	Access	Description
	read-only	This property value cannot be edited on this page.
	read-write	This property value can be edited on this page.
	—	Expand (+) the folder icon to view associated properties.

Displaying Property Definitions

When you select a property in the list, a description for that property often appears in the **Description** field:



Configuring Properties

Configure the **Device Editor** properties:

Step	Action	
2	While you edit a parameter, Unity Pro displays an icon next to the field you are editing and in the navigation tree. These icons refer to value of the parameter that is being edited:	
3		The entered value is not valid. The **Apply** button does not work until a valid value is entered.
		This parameter has changed. The **Apply** button does not work until the value is corrected.
4	Click one of these buttons: ● **Apply**: Save your changes and keep the page open. ● **OK**: Save your changes and close the page. ● **Cancel**: Cancel changes. **NOTE:** Your changes do not take effect until they are successfully downloaded from your PC to the CPU and from the CPU to the communication modules and network devices.	

Uploading and Downloading DTM-Based Applications

Introduction

Use the Unity Pro instructions below to perform these tasks:

- Download an application file from your PC to the PAC.
- Upload an application file from the PAC to your PC. To successfully complete an upload, the application file needs to include specific upload-related information as part of the application.

Downloading DTM-Based Applications

Unity Pro applications that include DTM files require more memory than traditional Unity Pro applications. These products employ DTMs for network configuration:

- 140 NOC 771 01 Ethernet communication module for Quantum
- TSX ETC 101 Ethernet communication module for Premium
- BMX NOC 0401 Ethernet communication module for M340
- 140 NOC 78• 00 Ethernet communication module for Quantum
- BME P58 •••• CPU for M580

In some cases, the configurations created for these modules (and the data associated with them) require more memory than is available in the CPU.

If the amount of memory required by an application exceeds the amount of memory that is available in the CPU, Unity Pro informs you of this condition during the build process and disallows the application download to the PAC.

When this situation occurs, exclude the additional upload-related information from the application to complete the build and enable the application download. To do this, make this configuration change in Unity Pro:

Step	Action
1	In the main menu, select **Tools → Project Settings...** to open the **Project Settings** window.
2	In the left pane of the **Project Settings** window, select **General → PLC embedded data**.
3	In the right pane, de-select **Upload information**.
4	Click **OK** to save your changes and close the **Project Settings** window.

After the **Upload information** setting is disabled, you can build the application and download it to the PAC (PLC).

NOTE: An application in which the **Upload information** setting has been disabled cannot later be uploaded from the PLC to the PC.

Uploading DTM-Based Applications

DTM-based applications that were successfully downloaded to Unity Pro while the project's **Upload information** setting was enabled can later be uploaded from the PLC to the PC if the target PC has these files installed on it:

- a Unity Pro version that is equal to or higher than the version used to create the application
- the master DTMs for the modules in the configuration
  NOTE: The Ethernet Configuration Tool installation CD contains the Master DTMs for all the Ethernet communication modules, referenced above.
- the device DTMs for all DTM-based devices attached to the network (the DTMs are of the same or higher revision as each device DTM used in the configuration)
- the device EDS files for any EtherNet/IP device used in the configuration (the EDS files are of the same or higher revision as each device EDS file used in the configuration)

After all the above components have been installed on the target PC, you can upload a DTM-based Unity Pro application from a PLC.

NOTE: Install the above DTM components on the target PC *before* attempting the upload. Enable the EDS to get complete application data from the DTMs. Otherwise, the DTMs display default data.

Section 8.4
Configuring the M580 CPU with DTMs in Unity Pro

Introduction

Some configuration features for the M580 CPU are accessed through its corresponding M580 DTM in the Unity Pro **DTM Browser**.

Use the instructions in this section to configure the M580 CPU through the DTM.

What Is in This Section?

This section contains the following topics:

About DTM Configuration in Unity Pro

Introduction

The configuration of the M580 CPU through standard Unity Pro features is described elsewhere in this guide *(see page 152)*.

Some configuration that is specific to a particular device (like the M580 CPU) is done through a corresponding device type manager (DTM) in Unity Pro. This section describes that configuration.

Accessing Configuration Settings

Follow these steps to access the configuration settings in the DTM for the M580 CPU in Unity Pro:

Step	Action
1	Open Unity Pro.
2	Open a Unity Pro project that includes a M580 CPU in the configuration.
3	Open the **DTM Browser** (**Tools → DTM Browser**).
4	Double-click the DTM that corresponds to the M580 CPU in the **DTM Browser** to open the device editor of the DTM.
5	These headings appear in the configuration tree of the M580 DTM: ● **Channel Properties** ● **Services** ● **EtherNet/IP Local Slaves** ● **Device List** ● **Logging**

Channel Properties Page

Description

Perform these tasks on the **Channel Properties** page:

- Select the IP interface of the PC to perform these actions:
  - Connect a module or device DTM to physical devices.
  - Send explicit messages to Modbus TCP and EtherNet/IP devices.
- View your PC's IP address settings.

To display this page, select the **Channel Properties** node in the navigation tree located on the left side of the **Device Editor**.

NOTE: Refer to the topic Configuring Properties in the Device Editor *(see page 182)* for property editing instructions.

Accessing Channel Properties

Follow these steps to access the channel properties for the M580 CPU in Unity Pro:

Step	Action
1	Open Unity Pro.
2	Open a Unity Pro project that includes a M580 CPU in the configuration.
3	Open the **DTM Browser** (**Tools** → **DTM Browser**).
4	Double-click the DTM that corresponds to the M580 CPU in the **DTM Browser** to open the device editor of the DTM.
5	Select the **Channel Properties** heading in the configuration tree.

Properties

These tables describe the editible properties on the **Channel Properties** page.

Source Address properties:

Name	Description
Source IP Address	A list of IP addresses assigned to network interface cards installed on your PC. (Select the interface address that matches the IP address of your M580 CPU.) The source IP address is automatically detected based on the configured main IP address of the CPU. **NOTE:** If the configured main IP address of the CPU *(see page 155)* is not in subnet of any of the IP configured on the interface cards of the PC, then the first interface card IP is suggested by default.
Sub-Network Mask	The (read-only) subnet mask is associated with the selected source IP address (above).

EtherNet/IP Network Detection area:

Name	Description
Begin detection range address	The starting IP address of the address range for automatic field bus discovery of EtherNet/IP devices.
End detection range address	The ending IP address of the address range for automatic field bus discovery of EtherNet/IP devices.

Modbus Network Detection properties:

Name	Description
Begin detection range address	The starting IP address of the address range for automatic field bus discovery of Modbus TCP devices.
End detection range address	The ending IP address of the address range for automatic field bus discovery of Modbus TCP devices.

TCP/IP Monitoring

Expand (+) the **Channel Properties** heading in the configuration tree and select the **TCP/IP** item at level 1.

The read-only information on this page monitors the IP parameters that were configured in Unity Pro *(see page 155)*.

Configuring DHCP and FDR Address Servers

DHCP and FDR Address Servers

The M580 CPU includes both a dynamic host communication protocol (DHCP) and a fast device replacement (FDR) server. The DHCP server provides IP address settings to networked Ethernet devices. The FDR server provides operating parameter settings to replacement Ethernet devices that are equipped with FDR client functionality.

Accessing the Address Server

Follow these steps to access the channel properties for the M580 CPU in Unity Pro:

Step	Action
1	Open Unity Pro.
2	Open a Unity Pro project that includes a M580 CPU in the configuration.
3	Open the **DTM Browser** (**Tools → DTM Browser**).
4	Double-click the DTM that corresponds to the M580 CPU in the **DTM Browser** to open the device editor of the DTM.
5	Expand (+) the **Services** heading in the configuration tree.
6	Select the **Address Server** item in the configuration tree at level 1 to see the address server configuration.

Configuration

Configure the address server to perform these tasks:
- Enable and disable the CPU FDR service.
- View an automatically generated list of all devices included in the CPU configuration, displaying for each device:
  - IP addressing parameters
  - whether the device IP addressing parameters are provided by the CPU embedded DHCP server

Manually add remote devices that are not part of the CPU configuration to the CPU DHCP client list.

NOTE: Remote devices added in this way are equipped with DHCP client software and are configured to subscribe to the CPU IP addressing service.

Enabling the FDR Service

To enable the CPU FDR service, set the **FDR Server** field to **Enabled**. To disable the service, toggle the same field to **Disabled**.

Any networked Ethernet device equipped with FDR client functionality can subscribe to the CPU FDR service. The CPU can store up to 1 MB of FDR client operating parameter files. When this file storage capacity is reached, the CPU cannot store any additional client FDR files.

The CPU can store FDR client files for up to 128 devices, depending on the size of each stored file. For example, if the size of each FDR client file is small – not more than 8 Kb – the CPU could store up to the maximum of 128 parameter files.

Viewing the Auto-Generated DHCP Client List

The list of **Automatically Added Devices** includes a row for each remote device that is:
- part of the CPU configuration
- configured to subscribe to the CPU DHCP addressing service

NOTE: You cannot add devices to this list in this page. Instead, use the configuration pages for the remote device to subscribe to this service.

This table describes the available properties:

Property	Description
Device No	The number assigned to the device in the Unity Pro configuration.
IP Address	The client device IP address.
DHCP	TRUE indicates that the device subscribes to the DHCP service.
Identifier Type	Indicates the mechanism used by the server to recognize the client (MAC address or DHCP device name).
Identifier	The actual MAC address or DHCP device name.
Netmask	The client device subnet mask.
Gateway	The IP address a DHCP client device will use to access other devices that are not located on the local subnet. A value of 0.0.0.0 constrains the DHCP client device by allowing it to communicate only with devices on the local subnet.

Manually Adding Remote Modules to the DHCP Service

Remote modules that are part of the CPU configuration – and which have subscribed to the CPU IP addressing service – automatically appear in the **Automatically Added Devices** list.

Other remote modules that are not part of the CPU configuration can be manually added to the CPU DHCP IP addressing service.

Manually add networked Ethernet modules that are not part of the CPU configuration to the CPU IP addressing service:

Step	Description
1	In the **Address Server** page, click the **Add** button in the **Manually Added Devices** field to instruct Unity Pro to add an empty row to the list.

<table>
<tr><th>Step</th><th colspan="2">Description</th></tr>
<tr><td rowspan="6">2</td><td colspan="2">In the new row, configure the following parameters for the client device:</td></tr>
<tr><td>IP Address</td><td>Type in the IP address of the client device.</td></tr>
<tr><td>Identifier Type</td><td>Select the type of value the client device will use to identify itself to the FDR server:
● MAC address
● device Name</td></tr>
<tr><td>Identifier</td><td>Depending upon the identifier type, type in the client device setting for the MAC address or name.</td></tr>
<tr><td>Netmask</td><td>Type in the client device subnet mask.</td></tr>
<tr><td>Gateway</td><td>Type in the gateway address that remote devices can use to communicate with devices located on other networks. Use 0.0.0.0 if remote devices will not communicate with devices located on other networks.</td></tr>
<tr><td>3</td><td colspan="2">Refer to the topic Configuring Properties in the Device Editor *(see page 182)* for instructions on how to apply edited properties to networked devices.</td></tr>
</table>

Logging

Description

Unity Pro maintains a log of events for:
- the Unity Pro embedded FDT container
- each Ethernet communication module DTM, and
- each EtherNet/IP remote device DTM

Events relating to the Unity Pro FDT container are displayed in the **FDT log event** page of the **Output Window**.

Events relating to a communication module or remote EtherNet/IP device are displayed:
- in configuration mode: in the **Device Editor**, by selecting the **Logging** node in the left pane
- in diagnostic mode: in the **Diagnostics** window, by selecting the **Logging** node in the left pane

Logging Attributes

The **Logging** window displays the result of an operation or function performed by Unity Pro. Each log entry includes the following attributes:

<table>
<tr><th>Attribute</th><th colspan="2">Description</th></tr>
<tr><td>Date/Time</td><td colspan="2">The time the event occurred, displayed in the format: yyyy-mm--dd hh:mm:ss</td></tr>
<tr><td rowspan="4">Log Level</td><td colspan="2">The level of event importance. Values include:</td></tr>
<tr><td>Information</td><td>A successfully completed operation.</td></tr>
<tr><td>Warning</td><td>An operation that Unity Pro completed, but which may lead to a subsequent error.</td></tr>
<tr><td>Error</td><td>An operation that Unity Pro was unable to complete.</td></tr>
<tr><td>Message</td><td colspan="2">A brief description of the core meaning of the event.</td></tr>
<tr><td>Detail Message</td><td colspan="2">A more detailed description of the event, which may include parameter names, location paths, etc.</td></tr>
</table>

Section 8.5
Configuring the M580 CPU as an EtherNet/IP Adapter

Introduction

This section describes the configuration of an M580 CPU as an EtherNet/IP adapter using a functionality called *local slave*. The CPU supports up to three instances of local slaves.

In its role as a EtherNet/IP adapter, the M580 CPU initiates no messages. Instead, it responds to:
- implicit messaging requests (from a scanner device in the network)
- explicit messaging requests (directed to the communication module's assembly object from other devices on the network)
  NOTE: If no local slave instance is enabled, the M580 CPU can respond to explicit messaging requests directed at its CIP objects other than the assembly object.

What Is in This Section?

This section contains the following topics:

Introducing the Local Slave

Introduction

You can configure the M580 CPU to perform the functionality of an EtherNet/IP adapter using *local slave* functionality.

Configure local slaves in the M580 master communication DTM in the Unity Pro **DTM Browser**. The CPU supports up to 3 instances of local slaves (**Local Slave 1**, **Local Slave 2**, and **Local Slave 3**).

Messaging

In its role as an EtherNet/IP adapter, the CPU does not initiate messages. Instead, it responds to these requests:

- **implicit:** Implicit messaging requests come from a scanner device in the network.
- **explicit:** Explicit messaging requests are directed to the CPU's assembly object from other devices on the network.

NOTE: If no local slave instance is enabled in the M580 CPU DTM, the CPU can respond to explicit messaging requests directed at its CIP objects other than the assembly object.

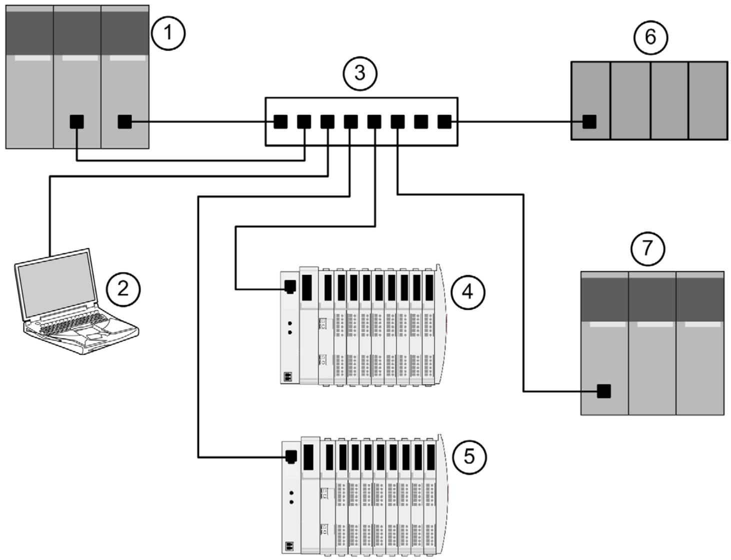
Local Slave Networking Example

The local slave functionality allows other scanners on the network to read from and write to the M580 CPU using messaging. Each local slave instance can accept one exclusive owner connection and one listen-only connection. Through a local slave, a remote scanner can access the M580 CPU's CIP Assembly object. The local slave function is especially beneficial for peer-to-peer data exchanges at a repetitive rate.

NOTE:

- The M580 CPU can provide 3 local slave adapter instances, while simultaneously performing as a scanner.
- The M580 CPU supports a total of 128 EtherNet/IP connections for the distributed device scanner and the local slaves. When you configure 1 local slave, 127 available EtherNet/IP scanner connections remain.
- The local slave is exclusively an EtherNet/IP functionality.

This is a sample local slave configuration:



This sample configuration includes these scanners and adapters:

- A primary PAC (1) with one local slave instance enabled. The PAC performs the following functions:
  - scans I/O data from remote devices (4 and 5)
  - scans input data from its own local slave instance
- A third party scanner (6)—which lacks adapter capability, and therefore cannot itself be scanned by the primary PAC (1)—performs the following functions:
  - collects data from other sources (not part of this network)
  - writes data to inputs of the primary PAC's local slave
  - scans the primary PAC's local slave's output data via an exclusive owner connection
- A secondary scanner (7), which also scans the primary PAC's local slave—for the very same output data scanned by the third party scanner—via a listen only connection

NOTE:

- Because the third party scanner (6) and the secondary scanner (8) both receive the same data produced by the local slave, the requested packet interval (RPI) settings of the third party scanner's exclusive owner connection are the same as those of the secondary scanner's listen only connection.
- By enabling a local slave on the primary PAC (1):
  - PAC (1) allows the third party PAC (6) to write to it at a repetitive rate, even if PAC (6) is not capable of acting as an adapter.
  - the secondary PAC (7) is able to scan the primary PAC (1) at a repetitive rate, rather than through application intensive explicit messaging.

The topics in this section show you how to use Unity Pro software installed in the PC (2, above) to configure a local slave, and to create input and output items in support of the peer-to-peer data transfers between and among scanners.

Configuring a Local Slave

Description

The M580 CPU presents 3 identical **Local Slave** configuration pages. Use each page to configure a separate local slave instance.

To display this page, select a **Local Slave** node in the navigation tree on the left side of the **Device Editor**.

The following steps describe a sample configuration for **Local Slave 1**.

Access Configuration

Access the local slave properties:

Step	Action
1	Double-click the M580 master communication DTM in the Unity Pro **DTM Browser**.
2	Expand **EtherNet/IP Local Slaves**.
3	Select one of the three available local slaves.

Configuration Overview

In the following configuration example you will create a single local slave instance:

Stage	Description
1	Enable and name the local slave.
2	Specify the size of the local slave input and output assemblies.

Configuration Example: Local Slave 1

In the sample network configuration *(see page 195)*, the application in the third-party PAC produces data, which is available in the PAC's M580 CPU as inputs. In this example, the third-party device produces the following information:

- production totals for manufacturing line A
- production totals for manufacturing line B
- the number of production interruption events for line A
- the number of production interruption events for line B

It is required that some information is passed to a third-party device (for example, confirmation that data from the third-party device has been received by the PAC). That information is accessible in the third-party device as input data. In this example, the third-party device is programmed to scan Local Slave 1 for this confirmation.

Makes selections for the sample configuration:

Step	Action
1	At **Active Configuration**, select **Enabled**.
2	Leave the **Comment** field blank.
3	At **Connection Bit**, accept the auto-generated value.

When configuring inputs and outputs in both the local slave and the third-party PAC, associate the inputs and outputs in this manner:

Associate these local slave items:	With these third-party PAC items:
outputs (T -> O): assembly instance 101	inputs: assembly instance 101
inputs (O -> T): assembly instance 102	outputs: assembly instance 102

Enabling and Naming the Local Slave

These are the settings in the **Properties** area:

- **Number:** This unique number (or identifier) is assigned to the device. By default, Unity Pro assigns these numbers:
  - **033:** local slave 1
  - **034:** local slave 2
  - **035:** local slave 3
- **Active Configuration:**
  - **Enabled:** Enable the local slave.
  - **Disabled:** Disable the local slave and save the current local slave settings.
- **Comment:** This optional free text comment field can contain 80 characters maximum.
- **Connection Bit:** This predefined integer (385... 387) indicates the offset of the connection's bits. (For example, the LS1 connection bit equals 385).
  - health bit (in the CPU's input area)
  - control bit (in the CPU's output area)

Configuring the Size of Local Slave Input and Output Assemblies

Use the **Assembly** section of the **Local Slave** page to configure the size of the local slave inputs and outputs. The assembly numbers are non-editable and are assigned by Unity Pro in this manner:

Assembly number	Local slave number	Used for connection
101	1	T->O [1]
1. In this table: ● O indicates the originator (scanner) device ● T indicates the target (adapter) device		

Assembly number	Local slave number	Used for connection
102	1	O->T Exclusive Owner
103	1	Configuration
199	1	O->T Listen Only
111	2	T->O
112	2	O->T Exclusive Owner
113	2	Configuration
200	2	O->T Listen Only
121	3	T->O
122	3	O->T Exclusive Owner
123	3	Configuration
201	3	O->T Listen Only
1. In this table: ● O indicates the originator (scanner) device ● T indicates the target (adapter) device		

The **Local Slave** assembly settings include these:

Setting	Description
Outputs (T->O)	A read-only value (see preceding table).
Outputs (T->O) Size	An integer (1 ... 509) that represents the maximum size (in bytes) that are reserved for local slave outputs.
Inputs (O->T)	A read-only value (see table, above).
Inputs (O->T) Size	An integer (0 ... 505) that represents the maximum size (in bytes) that is reserved for local slave inputs.
Configuration	A read-only value (see table, above).
Configuration Size	A read-only value set to **0**.

NOTE: When using explicit messaging to read the Ethernet communication module's assembly object, allocate sufficient room for the response. The size of the response equals the sum of: the assembly size + Reply service (1 byte) + General Status (1 byte).

Makes selections for the sample configuration:

Step	Action
1	At **Outputs (T->O)**, accept the read-only value (101).
2	At **Outputs (T->O) Size**, enter 2 (for the 2 output bytes).
3	At **Inputs (O->T)**, accept the read-only value (102).
4	At **Inputs (O->T) Size**, enter 8 (for the 8 input bytes).

Step	Action
5	At **Configuration**, accept the read-only value (103)
6	At **Configuration Size**, accept the read-only value (0)

Local Slave Structure and Variables Names

Each input and output that Unity Pro creates for your application has both a non-editable **Structure Name** (used by Unity Pro to internally identify input and output items) and an editable **Variable Name**.

Use the **I/O Structure Name** section of the **Local Slave** page to perform these tasks:
- View and edit local slave input and output variable names.
- View non-editable local slave structure names.

This table describes the input and output property settings:

Setting	Description
Structure Name	The read-only name for input or output structures. By default, it contains these components: • the prefix T_ • the alias device name (example: BMEP58_ECPU) • the local slave number (LS1, LS2, LS3) • the suffix (_IN or _OUT) An example default structure name for an input for local slave 2 would be T_BMEP58_ECPU_LS2_IN.
Variable Name	The editable base name for input or output variables. By default, it contains these components: • the alias device name (example: BMEP58_ECPU) • the local slave number (LS1, LS2, LS3) • the suffix (_IN or _OUT) An example default variab le name for an out for local slave 3 would be T_BMEP58_ECPU_LS3_OUT.

NOTE: Restore the default variable names at any time by clicking on the **Default Name** button.

For each configured local slave of the M580 CPU, a single (non-editable) device DDT with the type name T_BMEP58_ECPU_LS* is automatically created. The _LS* suffix denotes the configured slave. For example, T_BMEP58_ECPU_LS1 is the type name of the device DDT that is created when local slave 1 is configured. T_BMEP58_ECPU_LS2 and T_BMEP58_ECPU_LS3 correspond to local slaves 2 and 3 respectively. If no local slave is configured, the LS variable and its type is not generated.

Each local slave has internal DDTs for input and output areas:

Area	DDT Name	Description
inputs	T_BMEP58_ECPU_LS*_IN	This DDT input corresponds to the configured local slave.

Area	DDT Name	Description
outputs	T_BMEP58_ECPU_LS*_OUT	This DDT output corresponds to the configured local slave.

A specific local slave is again identified by the suffix _LS*. When local slave 1 is configured, for example, the variable LS1 has 2 internal input and output variables of type T_BMEP58_ECPU_LS1_IN and T_BMEP58_ECPU_LS1_OUT respectively.

NOTE: The I/O layout for the configured local slave always corresponds to the maximum size of the input and output of the local slave. Even though the DDT displays the maximum number of fields, the number of available variables depends on the value configured in the M580 CPU DTM. The scan scans only the configured size in the DTM. The configuration of input and output sizes *(see page 199)* is done in the master DTM.

Local Slave Inputs and Outputs

Introduction

The M580 CPU serves as an adapter when the **Active Configuration** field is set to **Enabled** in the configuration window for one (or more) of the module's local slave nodes.

When a local slave instance of an Ethernet communication module is enabled, the designated memory location allocated to that instance is exposed to, and can be accessed by, other devices.

The I/O data exchange between the remote device and the local slave is configured as part of the remote device's configuration settings.

Configuring the I/O Items

You can configure input and output items in groups of 1 or more single bits, 8-bit bytes, 16-bit words, 32-bit dwords, or 32-bit IEEE floating values. The number of items you create depends upon the data type and size of each item.

The process for creating and defining I/O items for the local slave is the same as for any adapter class device and depends upon the type of items you wish to create.

In support of the ongoing configuration example, these items are required:
- 4 input word items
- 1 output word item

NOTE: The items created, below, are designed to hold data received from, or sent to, the third-party scanner. In addition to these items, it is necessary to include logic in the application programs in which the Ethernet communication module and the third-party scanner, respectively, are included. Writing this code is beyond the scope of this example.

Accessing Input and Output Word Items

Follow these steps to view and configure the input and output word items for the M580 CPU in Unity Pro:

Step	Action
1	Open the Unity Pro **DTM Browser** (**Tools → DTM Browser**).
2	Double-click the DTM that corresponds to the M580 CPU in the **DTM Browser** to open the device editor of the DTM.
3	Expand (+) the **Device List** in the device editor.
4	Expand a local slave instance in the **Device List**.

NOTE: The Unity Pro instructions for creating input and output items are of a general nature. You may wish to refer to these topics to configure different kinds of nodes before you continue with this general description:

- Configure an EtherNet/IP node *(see page 216)*.
- Configure a Modbus TCP node *(see page 219)*.
- Configure the M580 CPU as a network adapter *(see page 223)*.

Creating Input and Output Word Items

Create input and output word items for local slave 01:

Step	Action
1	Expand the local slave instance in the **Device List**.
2	Expand the **Exclusive Owner**.
3	Select the **Input** tab to create input word items or select the **Output** tab to create output word items.
4	Enter text in the **Default Item Name Root** input box.
5	In the table, select the first two rows (0 and 1).
6	Click the **Define Items** button to open the **Item Name Definition** dialog.
7	Select a data type from the **New Item(s) Data Type** pull-down menu.
8	Click **OK** to display the new item in the table. (The **Default Item Name Root** assigned above appears in the **Item Name** column.)
9	Click **Apply** to save the new items and leave the tab open.

Repeat these steps for each new word item you want to create and save your changes (**File → Save**).

Using Local Slave Inputs and Outputs

The inputs and outputs created above are used as follows:

- A third-party device updates the values of the variables you created above.
- The M580 CPU updates value of the DataReceived variable in the third-party device at the configured RPI.

Section 8.6
DTM Device Lists

Introduction

This section describes the connection of an M580 CPU to other network nodes through the Unity Pro **DTM Browser**.

What Is in This Section?

This section contains the following topics:

Device List Configuration and Connection Summary

Introduction

View the read-only properties of the M580 CPU in the Unity Pro **Device List**:

Step	Action
1	Double-click the M580 DTM in the **DTM Browser** to see the device configuration.
2	Select **Device List**.

NOTE: Refer to these topics to configure and connect different types of devices:
- Configuring an EtherNet/IP Node *(see page 216)*
- Configuring a communication adapter *(see page 194)*

Configuration Data

The **Device List** displays read-only properties in the **Configuration Summary**. These settings are configured elsewhere in Unity Pro (as noted in the following tables).

Input data image (**Inputs**):

Name	Description	Value set by...
Input Current Size	The cumulative number of inputs (Input size) actually used in the application	**General** page in the device editor for a selected remote device and connection

Output data image (**Outputs**):

Name	Description	Value set by...
Output Current Size	The cumulative number of outputs (Output size) actually used in the application	**General** page in the device editor for a selected remote device and connection

Maximum and actual numbers for devices, connections, and packets (**Configuration Size**):

Name	Description	Value set by...
Maximum Number of DIO Devices	The maximum number of distributed devices that can be added to the configuration.	predefined
Current Number of DIO Devices	The number of distributed devices in the current configuration.	network design in the Unity Pro device editor
Maximum Number of DIO Connections	The maximum number of connections to distributed devices that can be managed by the module.	predefined
Current Number of DIO Connections	The number of connections to distributed devices in the current configuration.	network design in the Unity Pro device editor

Name	Description	Value set by...
Maximum Number of Packets	The maximum number of packets per second the module is able to manage.	predefined
Current Number of Input Packets	Total number of input packets (traffic) per second, based on the current number of modules and its configured input data.	network design in the Unity Pro device editor
Current Number of Output Packets	Total number of output packets (traffic) per second, based on the current number of modules and its configured output data.	network design in the Unity Pro device editor
Current Number of Total Packets	Total number of packets (traffic in both directions) per second, based on the current number of modules and its configured I/O data.	network design in the Unity Pro device editor

Recommended PLC times (**PLC Scan Time**):

Name	Description	Value set by...
Minimum PLC Scan Time	The estimated cycle time to process inputs and outputs, equal to the sum of estimates for communication over both the backplane and the network.	predefined
Module Exchange Time	The estimated additional time contributed by the EtherNet/IP module to perform the I/O management. This value is included in the "minimum PLC scan time" value.	predefined

Request / Connection Data

The **Device List** page displays columns of request and connection data in the **Request / Connection Summary** area:

Column	Description
Connection Bit	The offset for both the connection’s health bit and control bit.
Task	The task that is associated with this connection.
Input Object	The ID of the input object associated with the connection. (See the note after the table.)
Output Object	The ID of the output object associated with the connection. (See the note after the table.)
Device	The device **Number** as set in the **Properties** configuration page for the local slave or remote device.
Device Name	A unique name associated with the device that owns the connection

Column	Description
Type	The target device type: • EtherNet/IP • Local Slave • Modbus TCP
Address	The target device IP address for remote devices. (It does not apply to local slaves.)
Rate (msec)	The RPI (for EtherNet/IP) or the repetitive rate (for Modbus TCP), in ms.
Input Packets per Sec	The number of input (T->O) packets per second exchanged over this connection.
Output Packets per Sec	The number of output (O->T) packets per second exchanged over this connection.
Packets per Sec	The total number of packets per second exchanged over this connection in both Input and output directions.
Bandwidth usage	The total bandwidth used by this connection (total bytes traffic)
Size In	The number of input words configured for this remote device.
Size Out	The number of output words configured for this remote device.

NOTE: The numeric identifiers in the **Input Object** and **Output Object** columns represent the objects associated with a single device connection (scan line). For example, if an EtherNet/IP connection has an input object of 260 and an output object of 261, the corresponding control bits for this connection are in the DEVICE_CNX_CTRL_256_271 field in the M580 CPU device DDT. Object 260 is the fifth bit and object 261 is in the sixth bit in this field. There can be multiple connections for a device. Set the corresponding bits to conrtol the input and output objects for these connections.

Configuring Device Properties

Overview

Use the **Properties** configuration page to view and configure settings for a local slave or remote device:
- Assign a numeric address to the device.
- Include or exclude device inputs and outputs in the Unity Pro project.
- Specify variable and structure names for device s and outputs.
- Determine how I/O items will be managed.

To display this page, select the device name, which is found under the **Device List** node in the left pane of the **Device Editor**, then click the **Properties** tab.

NOTE: Refer to the topic Configuring Properties in the Device Editor *(see page 182)* for instructions on how to edit properties.

Access the Page

View the **Properties** and **Assembly** information on the M580 DTM device configuration page:

Step	Action
1	Go to the M580 DTM device configuration page.
2	Expand (or double-click on) **EtherNet/IP Local Slaves**.
3	Select Local Slave 1 to view the **Properties** and **Assembly** settings.

Device Properties

Properties settings:

Setting	Description
Number	The relative position of the device in the list (36...163). By default, this number is assigned sequentially to devices in the project, beginning with the number **033** that is assigned to the first local slave.
Active Configuration	• **Enable:** adds this device to the Unity Pro project configuration • **Disable:** removes this device from the Unity Pro project configuration **NOTE:** • Changing this setting also changes the addresses of items in project memory. Selecting **Enable** adds the device's inputs and outputs to project memory; selecting **Disable** removes these inputs and outputs from memory. • Alternatively, if you enable the **I/O Communication Control** service, you can turn ON and OFF the connection between a communication module and a remote device by toggling the output CONTROL_BIT for that connection. This leaves unchanged the size of the project's input and output data images.

IO Structure Name settings:

Setting	Description
Structure Name	The read-only name for input and output structures. By default, it is the concatenation of: • The prefix T_ • the string DEVICE_ • the device number
Variable Name	The editable base name for input and output variables. By default, it is the concatenation of: • the string DEVICE_ • the device number
Default Name	Click this button to restore the default variable and structure names.

The table contains the **Items** settings:

Setting	Description
Import mode	• **Automatic:** I/O items are taken from the device DTM and updated if the items list in the device DTM changes. Items cannot be edited in the **Device Editor**. • **Manual:** I/O items are added when the device DTM is first added to Unity Pro. Thereafter, all I/O item edits are made manually in the **Device Editor**. Changes to the device DTM do not impact the I/O items list.
Reimport Items	Imports the I/O items list from the device DTM, overwriting any manual I/O item edits. Enabled only when **Import mode** is set to **Manual**.

Device DDT Names for the M580 CPU

Introduction

This topic describes the Unity Pro **Device DDT** tab that is associated with the M580 CPU.

Access the M580 CPU Variables

View the variables for the M580 CPU:

Step	Action
1	In Unity Pro, open the **Project Browser** (**Tools** → **Project Browser**).
2	In the **Project Browser**, double-click on **Variables & FB instances** to open the **Variables** tab.
3	In the **Variables** tab, expand the fields associated with the M580 CPU (BMEP58_CPU) by clicking the plus (+) sign.

The variables associated with the CPU are listed in the **Name** column. The corresponding variable descriptions are in the **Comment** column.

Access the Device DDT Variables

View the variables associated with EtherNet/IP or Modbus devices:

Step	Action
1	Open the **Variables** tab (as shown in the above table).
2	In the **Variables** tab, expand the fields associated with an EtherNet/IP or Modbus device.

These inputs and outputs associated with EtherNet/IP or Modbus devices are described:

Name	Description
Freshness	This is a global bit: ● **1:** All input objects below (**Freshness_1**, **Freshness_2**, etc.) for the associated device are true (**1**) and provide up-to-date data. ● **0:** One or more inputs (below) is not connected and does not provide up-to-date data.
Freshness_1	These bits represent individual input objects for the device: ● **1:** The input object in this row is true (**1**) and provides up-to-date data. ● **0:** The input object is not connected (**0**) and does not provide up-to-date data.
Freshness_2	
Freshness_3	
...	
(available)	The rows after the **Freshness** data are organized in groups of **Inputs** and **Outputs** that have user-defined names. The number of input and output rows depends on the maximum number of connections that a particular device supports.

Configuring Modbus TCP Request Settings

Overview

Use the **Request Setting** page to configure scanner connection information for a remote Modbus TCP device.

To display this page, select a remote Modbus TCP device in the **Device List** node in the left pane of the **Device Editor**, then click the **Request Settings** tab.

NOTE: Refer to the topic Configuring Properties in the Device Editor *(see page 182)* for instructions on how to edit properties.

Configuring the Request Settings Page

The **Request Settings** page includes these settings:

Setting	Description
Connection Bit	The offset for both this connection health bit and control bit.
Unit ID	The number of the device (or module) that is the target of the connection. A value of: ● The number 255 (default) accesses the Ethernet communication module itself. ● All other numbers (0...254) identify the device number of the target device behind a Modbus TCP to Modbus gateway. **NOTE:** When accessing data in the Ethernet communication module itself, use 255. When accessing data in the application running in the PAC, use a value from 0 to 254. (A value of 1 is recommended.)
Health Timeout (ms)	The maximum allowed period, in milliseconds, between device responses, from 0 to 120000 ms, in intervals of 5 ms. When this setting is exceeded, the health timeout bit is set to 1. (Default = 1500 ms.)
Repetitive Rate (ms)	The rate at which data will be scanned (0 ... 60000 ms) in intervals of 5 ms. (Default = 60 ms.)
RD Address	The address (0 ... 65535) in the remote device of the first word from which the communication module reads data.
RD Length	The number of words in the remote device (0 ... 125) that the communication module will read.
Last Value	The behavior of inputs in the application in the event communication is lost: ● Hold Value (the default) ● Set To Zero
WR Address	Address (0 ... 65535) in the remote device of the first word to which the communication module writes data.
WR Length	The number of words in the remote device (0 ... 120) that the communication module will write.

Configuring Communication Module Connection Settings

Overview

Use the **Connection Settings** page to view and edit (when available) connection properties from the perspective of the communication module.

To open this page, in the left pane of the **Device Editor** expand the navigation tree and, under the **Device List** node, select **<remote device>** → **<connection>**, where:

- **<remote device>** represents the name of the selected remote device appearing in the **Device List**, and
- **<connection>** represents the name of the selected connection, which depends upon the types of connections supported by the remote device and the particular connection type selected in the connection configuration settings for the remote device DTM.

NOTE: The name of the connection displayed in the **Device List** depends upon the types of connections supported by the remote device and the particular connection type selected in the connection configuration settings for the remote device DTM.

Refer to the topic Configuring Properties in the Device Editor *(see page 182)* for instructions on how to edit properties

Communication Module Connection Properties

The following connection settings can be viewed or configured in the **Request / Connection Summary** table in the M580 DTM:

Setting	Description
Connection Bit	Offset of the connection health bit in the status byte array of the scanner device DDT.
Request Packet Interval (RPI)	The refresh period, 2...65535 ms, for this connection. **NOTE:** This parameter can be set in the device DTM.
Input Fallback Mode	Use this setting to configure the behavior of inputs in the application when communications are lost: ● **Hold Value** ● **Set to Zero**

EtherNet/IP Connection Information

Overview

Use this read-only page to view connection properties for the remote device. An EtherNet/IP connection provides a communication link between two or more devices. Properties for a single connection are configured in the DTMs for each of the connected devices (typically a communication module and a remote device). The read-only properties viewable in this page can be configured in the general page of the connection node configuration for the DTM of the remote device.

To open this page, in the left pane of the **Device Editor** expand the navigation tree and, under the **Device List** node, select **<remote device>** → **<connection>**, where:

- **<remote device>** represents the name of the selected remote device appearing in the **Device List**, and
- **<connection>** represents the name of the selected connection, which depends upon the types of connections supported by the remote device and the particular connection type selected in the connection configuration settings for the remote device DTM.

Remote Device Connection Properties

A connection to a remote Schneider Electric device can present the following properties:

Setting	Description
Input size	The number of bytes reserved for input data, from 0 to 505.
Input mode	The transmission type: ● Multicast ● Point to Point
Input type	Ethernet packet type—fixed or variable length—to be transmitted. **NOTE:** The Ethernet communication module supports only **Fixed** length packets.
Input priority	The transmission priority. The value depends upon the device DTM. Values can include: ● Low ● High ● Scheduled
Input trigger	The transmission trigger. Values can include: ● Cyclic ● Change of state or application
Output size	The number of bytes reserved for output data, from 0 to 509.
Output mode	The transmission type: ● Multicast ● Point to Point
Output type	Ethernet packet type – fixed or variable length –to be transmitted. **NOTE:** The Ethernet communication module supports only **Fixed** length packets.

Setting	Description
Output priority	The transmission priority. The value depends upon the device DTM. Values can include: • Low • High • Scheduled

Configuring a EtherNet/IP Node

Introduction

Follow these directions to configure an EtherNet/IP node in the **Device List** through the Unity Pro **DTM Browser**.

Accessing the EtherNet/IP Node

Follow these steps to view the available DTMs:

Step	Action
1	Open a Unity Pro project that includes the M580 CPU.
2	Open the Unity Pro **DTM Browser** (**Tools** → **DTM Browser**).
3	Double-click the DTM that corresponds to the M580 CPU.
4	In the DTM configuration tree, expand (+) the **Device List** to view the network nodes that correspond to the available DTMs.
5	Select an EtherNet/IP node from the **Device List** to see the **Properties** and **Address Settings** tabs for that node type.

Property Settings

On the **Properties** configuration tab you can view and configure settings for an EtherNet/IP device:

- Assign a numeric address to the device.
- Include or exclude device inputs and outputs in the Unity Pro project.
- Specify variable and structure names for device inputs and outputs.
- Determine how I/O items will be managed.

NOTE: Refer to the topic Configuring Properties in the Device Editor *(see page 182)* for instructions on how to edit properties.

These settings are available on the **Properties** tab:

Field	Setting	Description
Properties	**Number**	This value (0 ... 127) represents the relative position of the device in the list. By default, this number is assigned sequentially to devices in the project. (The number 000 is assigned to the first local slave.)
	Active Configuration	**Enable:** Add the selected device to the Unity Pro project configuration. **Disable:** Remove the selected device from the Unity Pro project configuration. **NOTE:** Changing this setting also changes the addresses of items in project memory. Selecting Enable adds the device's inputs and outputs to project memory; selecting Disable removes these inputs and outputs from memory. Alternatively, if you enable the I/O Communication Control service, you can turn the connection between a communication module and a remote device on and off by toggling the output CONTROL_BIT for that connection. This leaves unchanged the size of the project's input and output data images.
IO Structure Name	**Structure Name**	This read-only field contains the name for input and output structures.
	Variable Name	This editible field contains the base name for input and output variables.
Items Management	**Import Mode**	**Automatic:** I/O items are taken from the device DTM and updated if the items list in the device DTM changes. Items cannot be edited in the device editor. **Manual:** I/O items are added when the device DTM is first added to Unity Pro. Thereafter, all I/O item edits are made manually in the device editor. Changes to the device DTM have no impact on impact the I/O items list.

Address Settings

Use the **Address Settings** tab to view and edit the IP address settings for an EtherNet/IP device.

NOTE: Refer to the topic Configuring Properties in the Device Editor *(see page 182)* for instructions on how to edit properties.

These settings are available on the **Address Setting** tab:

Field	Setting	Description
Change Address	**IP Address**	This field contains the IP address of the M580 CPU.

Field	Setting	Description
Address Server	**DHCP for this device**	Scroll to enable or disable DHCP.
	Identified by	Identify the M580 CPU by its **MAC Address** or **Device Name**.
	Identifier	Unity Pro adds the device name in this field.
	Subnet Mask	Unity Pro applies the same subnet mask that is used for the M580 CPU.
	Gateway (see note)	This field contains the gateway address.

NOTE: The default gateway address of the devices is the IP address of the CPU. You can change this IP address when DHCP is enabled.

Connection Settings and Information

You can monitor the node connections through the DTM device editor:

Step	Action
1	Expand (+) the EtherNet/IP device to see its sub-nodes.
2	Select a sub-node to view these tabs: **Connection Settings** and **Connection Information** tabs.

This information is read-only. Refer to the information in this section for configuring connection settings *(see page 213)*.

Configuring a Modbus TCP Node

Introduction

Follow these directions to configure a Modbus TCP node in the **Device List** through the Unity Pro **DTM Browser**.

Accessing the EtherNet/IP Node

Follow these steps to view the available DTMs:

Step	Action
1	Open a Unity Pro project that includes the M580 CPU.
2	Open the Unity Pro **DTM Browser** (**Tools** → **DTM Browser**).
3	Double-click the DTM that corresponds to the M580 CPU.
4	In the DTM configuration tree, expand (+) the **Device List** to view the network nodes that correspond to the available DTMs.
5	Select a Modbus TCP node from the **Device List** to see the configuration tabs for that node type.

Property Settings

On the **Properties** configuration tab you can view and configure settings for a Modbus TCP device:

- Assign a numeric address to the device.
- Include or exclude device inputs and outputs in the Unity Pro project.
- Specify variable and structure names for device inputs and outputs.
- Determine how I/O items will be managed.

NOTE: Refer to the topic Configuring Properties in the Device Editor *(see page 182)* for instructions on how to edit properties.

These settings are available on the **Properties** tab:

Field	Setting	Description
Properties	**Number**	This value (0 ... 127) represents the relative position of the device in the list. By default, this number is assigned sequentially to devices in the project. (The number 000 is assigned to the first local slave.)
	Active Configuration	**Enable:** Add the selected device to the Unity Pro project configuration. **Disable:** Remove the selected device from the Unity Pro project configuration. **NOTE:** Changing this setting also changes the addresses of items in project memory. Selecting Enable adds the device's inputs and outputs to project memory; selecting Disable removes these inputs and outputs from memory. Alternatively, if you enable the I/O Communication Control service, you can turn the connection between a communication module and a remote device on and off by toggling the output CONTROL_BIT for that connection. This leaves unchanged the size of the project's input and output data images.
IO Structure Name	**Structure Name**	This read-only field contains the name for input and output structures.
	Variable Name	This editible field contains the base name for input and output variables.
Items Management	**Import Mode**	**Automatic:** I/O items are taken from the device DTM and updated if the items list in the device DTM changes. Items cannot be edited in the device editor. **Manual:** I/O items are added when the device DTM is first added to Unity Pro. Thereafter, all I/O item edits are made manually in the device editor. Changes to the device DTM have no impact on impact the I/O items list.

Address Settings

Use the **Address Settings** tab to view and edit the IP address settings for a Modbus TCP device.

NOTE: Refer to the topic Configuring Properties in the Device Editor *(see page 182)* for instructions on how to edit properties.

These settings are available on the **Address Settings** tab:

Field	Setting	Description
Change Address	**IP Address**	This field contains the IP address of the M580 CPU.

Field	Setting	Description
Address Server	**DHCP for this device**	Scroll to enable or disable DHCP.
	Identified by	Identify the M580 CPU by its **MAC Address** or **Device Name**.
	Identifier	Unity Pro adds the device name in this field.
	Subnet Mask	Unity Pro applies the same subnet mask that is used for the M580 CPU.
	Gateway	This field contains the gateway address.

Request Settings

Use the **Request Setting** page to configure scanner connection information for a remote Modbus TCP device.

To display this page, select a remote Modbus TCP node in the **Device List** in the left pane of the device editor and select the **Request Settings** tab.

NOTE: Refer to the topic Configuring Properties in the Device Editor *(see page 182)* for instructions on how to edit properties.

The **Request Settings** tab contains these settings:

Column	Description
Connection Bit	The offset for both this connection health bit and control bit.
Unit ID	The number of the device (or module) that is the target of the connection. A value of: ● Number 255 (default) accesses the Ethernet communication module itself. ● Other numbers (0 ... 254) identify the device number of the target device, behind a Modbus TCP to Modbus gateway. **NOTE:** When accessing data in the Ethernet communication module itself, use 255. When accessing data in the application running in the PAC, use a value from 0 to 254. (A value of 1 is recommended.)
Health Time Out (ms)	The maximum allowed period (0 ... 120000 ms,) between device responses in intervals of 5 ms. When this setting is exceeded, the health timeout bit is set to 1. (Default = 1500 ms.)
Repetive Rate (ms)	The rate at which data will be scanned (0 ... 60000 ms) in intervals of 5 ms. (Default = 60 ms.)
RD Address	Address (0 ... 65535) in the remote device of the first word from which the communication module reads data.
RD Length	The number of words in the remote device (0 ... 125) that the communication module will read.
Last Value	The behavior of inputs in the application in the event communication is lost: ● Hold Value (the default) ● Set To Zero

Column	Description
WR Address	Address (0 ... 65535) in the remote device of the first word to which the communication module writes data.
WR Length	The number of words in the remote device (0 ... 120) that the communication module will write.

Input and Output Items

Expand (+) the Modbus TCP node to see the Modbus TCP sub-nodes (input and output items).

Select an input item to see the **Input** and **Input (bit)** tabs.

The **Input** and **Output** tabs contain this information:

Field	Setting	Description
(table)	**Offset/Device**	These columns represent the byte address.
	Offset/Connection	
	Item Name	item name
Default Item Name Root		This field contains an editible name.
Define Item(s)	(button)	Click this button to open the **Item Name Definition** dialog box, where you can scroll to the data type for your selected item.

The **Input (bit)** and **Output (bit)** tabs contain this information:

Column	Description
Offset/Device	These columns represent the byte address.
Offset/Connection	
Position in Byte	This value indicates the bit position (within the byte) of each discrete output item.
Item Name	item name
Default Item Name Root	This field contains an editible name.
Define Item(s)	Click this button to open the **Item Name Definition** dialog box, where you can scroll to the data type for your selected item.

Configuring a Network Adapter

Introduction

Follow these directions to configure an adapter module or CPU with adapter functionality (for example, the M580 CPU) in the **Device List** through the Unity Pro **DTM Browser**.

Accessing the Network Adapter

Follow these steps to view the available DTMs:

Step	Action
1	Open a Unity Pro project that includes the M580 CPU.
2	Open the Unity Pro **DTM Browser** (**Tools** → **DTM Browser**).
3	Double-click the DTM that corresponds to the M580 CPU.
4	In the DTM configuration tree, expand (+) the **Device List** to view the network nodes that correspond to the available DTMs.
5	Select a device that can act as an adapter from the **Device List** to see the configuration tabs for that node type.

Property Settings

On the **Properties** configuration tab you can view and configure settings for the adapter device:

- Assign a numeric address to the device.
- Include or exclude device inputs and outputs in the Unity Pro project.
- Specify variable and structure names for device inputs and outputs.
- Determine how I/O items will be managed.

NOTE: Refer to the topic Configuring Properties in the Device Editor *(see page 182)* for instructions on how to edit properties.

These settings are available on the **Properties** tab:

Field	Setting	Description
Properties	**Number**	This value (0 ... 127) represents the relative position of the device in the list. By default, this number is assigned sequentially to devices in the project. (The number 000 is assigned to the first local slave.)
	Active Configuration	**Enable:** Add the selected device to the Unity Pro project configuration. **Disable:** Remove the selected device from the Unity Pro project configuration. **NOTE:** Changing this setting also changes the addresses of items in project memory. Selecting Enable adds the device's inputs and outputs to project memory; selecting Disable removes these inputs and outputs from memory. Alternatively, if you enable the I/O Communication Control service, you can turn the connection between a communication module and a remote device on and off by toggling the output CONTROL_BIT for that connection. This leaves unchanged the size of the project's input and output data images.
IO Structure Name	**Structure Name**	This read-only field contains the name for input and output structures.
	Variable Name	This editible field contains the base name for input and output variables.
Items Management	**Import Mode**	**Automatic:** I/O items are taken from the device DTM and updated if the items list in the device DTM changes. Items cannot be edited in the device editor. **Manual:** I/O items are added when the device DTM is first added to Unity Pro. Thereafter, all I/O item edits are made manually in the device editor. Changes to the device DTM have no impact on impact the I/O items list.

Address Settings

Use the Address Settings tab to view and edit the IP address settings for a remote device.

NOTE: Refer to the topic Configuring Properties in the Device Editor *(see page 182)* for instructions on how to edit properties.

These settings are available on the **Address Setting** tab:

Field	Setting	Description
Change Address	**IP Address**	This field contains the IP address of the M580 CPU.

Field	Setting	Description
Address Server	**DHCP for this device**	Scroll to enable or disable DHCP.
	Identified by	Identify the M580 CPU by its **MAC Address** or **Device Name**.
	Identifier	Unity Pro adds the device name in this field.
	Subnet Mask	Unity Pro applies the same subnet mask that is used for the M580 CPU.
	Gateway	This field contains the gateway address.

Connection Settings and Information

You can monitor the node connections through the DTM device editor:

Step	Action
1	Expand (+) the adapter device to see its sub-nodes.
2	Select a sub-node to view these tabs: **Connection Settings** and **Connection Information** tabs.

This information is read-only. Refer to the information in this section for configuring connection settings *(see page 213)*.

Section 8.7
DTM Online Action

Online Action

Introduction

You can view and configure the settings in the **Online Action** menu when the M580 CPU is connected through the Unity Pro **DTM Browser**.

Accessing Online Action

Follow these directions to access the **Online Action** settings for the M580 CPU:

Step	Action
1	Open the **DTM Browser** in Unity Pro (**Tools** → **DTM Browser**).
2	Select the M580 DTM in the **DTM Browser**.
3	Connect the DTM to the Unity Pro application (**Edit** → **Connect**).
4	Right-click the M580 DTM.
5	Scroll to the **Online Action** menu (**Device menu** → **Additional functions** → **Online Action**).
6	3 tabs appear: ● **Ethernet/IP Objects** ● **Port Configuration** ● **Ping**

EtherNet/IP Objects

Displays object parameters value when available.

Click **Refresh** to update the displayed values.

Port Configuration

Configure and read the service port mode:

Field	Description
Service Port Mode	● **Access** (default) ● **Mirroring** **NOTE:** This mode can also be set in the CPU configuration tabs *(see page 166)*.
Access Port Configuration	Displays the access port configuration information (refer to CPU configuration tabs *(see page 166)*).

Field	Description
Port Mirroring Configuration	Displays the port mirroring configuration (refer to CPU configuration tabs *(see page 166)*).

Ping

Field	Parameter	Description
Address	**IP Address**	Type the IP address to ping.
Ping	**Ping**	Click to ping the address set.
	Ping Result	Displays the ping result.
	Repeat (100ms)	Select this parameter to repeat ping if no reply is received.
	Stop on Error	Select this parameter to stop repeating ping if an error is detected when **Repeat (100ms)** is selected.
	Clear	Click to clear the **Ping Result** display.

Section 8.8
Explicit Messaging

Introduction

The connection of the M580 CPU to a Unity Pro project (through the Unity Pro **DTM Browser**) allows for the configuration of EtherNet/IP and Modbus TCP explicit messages.

What Is in This Section?

This section contains the following topics:

Sending Explicit Messages to EtherNet/IP Devices

Introduction

Use the **EtherNet/IP Explicit Message** window to send an explicit message from Unity Pro to the M580 CPU.

An explicit message can be connected or unconnected:

- **connected:** A connected explicit message contains both path information and a connection identifier to the target device.
- **unconnected:** An unconnected message requires path (addressing) information that identifies the destination device (and, optionally, device attributes).

You can use explicit messaging to perform many different services. Not every EtherNet/IP device supports every service.

Accessing the Page

Before you can perform explicit messaging, connect the DTM for the M580 CPU to the CPU itself:

Step	Action
1	Open the **DTM Browser** in Unity Pro (**Tools** → **DTM Browser**).
2	Select the M580 DTM in the **DTM Browser**.
3	Right-click the M580 DTM.
4	Scroll to the EtherNet/IP explicit messaging page (**Device menu** → **Additional functions** → **EtherNet/IP Explicit Message**).

Configuring Settings

Configure the explicit message using these settings on the **EtherNet/IP Explicit Messaging** page:

Field	Setting
Address	**IP Address:** The IP address of the target device that is used to identify the target of the explicit message.
	Class: The **Class** integer (1 ... 65535) is the identifier of the target device that is used in the construction of the message path.
	Instance: The **Instance** integer (0 ... 65535) is the class instance of the target device that is used in the construction of the message path.
	Attribute: Check this box to enable the **Attribute** integer (0 ... 65535), which is the specific device property that is the target of the explicit message that is used in the construction of the message path.

Field	Setting
Service	**Number:** The **Number** is the integer (1 ... 127) associated with the service to be performed by the explicit message. **NOTE:** If you select **Custom Service** as the named service, type in a service number. This field is read-only for all other services.
	Name: Select the service that the explicit message is intended to perform.
	Enter Path(hex): Check this box to enable the message path field, where you can manually enter the entire path to the target device.
Data(hex)	**Data(hex):** This value represents the data to be sent to the target device for services that send data.
Messaging	**Connected:** Select this radial button to make the connection.
	Unconnected: Select this radial button to end the connection.
Response(hex)	The **Response** area contains the data sent to the configuration tool by the target device in hexadecimal format.
Status	The **Status** area displays messages that indicate whether or not the explicit message has succeeded.
Button	**Send to Device**: When your explicit message is configured, click **Send to Device**.

Click the **Close** button to save the changes and close the window.

Sending Explicit Messages to Modbus Devices

Introduction

Use the Modbus explicit messaging window to send an explicit message from Unity Pro to the M580 CPU.

You can use explicit messaging to perform many different services. Not every Modbus TCP device supports every service.

Accessing the Page

Before you can perform explicit messaging, connect the DTM for the M580 CPU to the CPU itself:

Step	Action
1	Open the **DTM Browser** in Unity Pro (**Tools** → **DTM Browser**).
2	Select the M580 DTM in the **DTM Browser**.
3	Right-click the M580 DTM.
4	Scroll to the EtherNet/IP explicit messaging page (**Device menu** → **Additional functions** → **Modbus Explicit Message**).

Configuging Settings

Configure the explicit message using these settings on the **Modbus Explicit Messaging** page:

Field	Setting
Address	**IP Address:** The IP address of the target device that is used to identify the target of the explicit message.
	Start Address: This setting is a component of the addressing path.
	Quantity: This setting is a component of the addressing path.
	Read Device Id Code: This read-only code represents the service that the explicit message is intended to perform.
	Object Id: This read-only identifier specifies the object that the explicit message is intended to access.
	Unit Id: This integer represents the device or module that is the target of the connection: ● **255:** (default): Use this value to access the M580 CPU itself. ● **0 ... 254:** Use these values to identify the device number of the target device behind a Modbus TCP to Modbus gateway.
Service	**Number:** This integer (0 ... 255) represents the service to be performed by the explicit message.
	Name: Select the integer (0 ... 255) that represents the service that the explicit message is intended to perform.
Data	**Data(hex):** This value represents the data to be sent to the target device for services that send data.

Field	Setting
Response	The **Response** area displays any data sent to the configuration tool by the target device in hexadecimal format.
Status	The **Status** area displays messages indicating whether or not the explicit message has succeeded.
Button	**Send to Device:** After your explicit message is configured, click **Send to Device**.

Click the **Close** button to save the changes and close the window.

Section 8.9
Hardware Catalog

Overview

This section describes the Unity Pro **Hardware Catalog**.

What Is in This Section?

This section contains the following topics:

Adding a DTM to the Unity Pro Hardware Catalog

Introduction

The Unity Pro **Hardware Catalog** is a list of modules and devices that can be added to Unity Pro projects. EtherNet/IP and Modbus TCP devices are located in the hardware catalog's **DTM Catalog** page. Each device in the catalog is represented by a DTM that defines the parameters of the module or device.

More details on project and DTM are provided in *Project Management with DTMs* topic in the *Unity Pro Operating Modes* manual.

EDS and DTM Files

Not all devices in the market today offer device-specific DTMs. Some devices are instead defined by a device-specific electronic data sheet (EDS) file. Unity Pro displays each EDS file in the form of a DTM. In this way, you can use Unity Pro to configure these Ethernet/IP devices — defined by an EDS file — in the same way you would configure a DTM-defined device.

Other devices lack both a DTM and an EDS file. You can configure these devices by using a generic DTM that is included in the **DTM Catalog** page.

A Manufacturer Defined Process

Before a DTM can be used by the Unity Pro **Hardware Catalog**, install the DTM on the host PC that is running Unity Pro.

Refer to the device documentation provided by the device manufacturer to install the device DTM on your PC.

NOTE: After a device DTM is successfully installed on your PC, update the Unity Pro **Hardware Catalog** *(see page 237)* so the new DTM is visible in the catalog and available to be added to a Unity Pro project.

Add an EDS File to the Unity Pro Hardware Catalog

Overview

Use a Unity Pro wizard to add one or more EDS files to the Unity Pro **Hardware Catalog**.

The wizard simplifies the process of adding EDS files to the catalog and provides a redundancy check (in case you attempt to add duplicate EDS files to the catalog).

NOTE: The Unity Pro **Hardware Catalog** displays a partial collection of DTMs and EDS files registered with the ODVA. This library includes DTMs and EDS files for products not manufactured or sold by Schneider Electric. The non-Schneider Electric EDS files are identified by vendor in the catalog. Please contact the identified device manufacturer for inquiries regarding the corresponding non-Schneider Electric EDS files.

Adding EDS Files

Add one or more EDS files to the library:

Step	Action
1	If the **DTM Browser** is not already open, in the Unity Pro main menu select **Tools** → **DTM Browser**.
2	In the **DTM Browser**, select a master communication module.
3	Right-click on the master communication module and scroll to **Device menu** → **Add EDS to library**. **Result**: The introductory page of the wizard opens.
4	Click **Next** to go to the next page.
5	In the **Select the Location of the EDS File(s)** section, select one of these: ● **Add File(s)**: Add one or more EDS files that you will select in a moment. ● **Add all the EDS from the Directory**: Add all files from a folder that you will select. ❍ Select **Look in Subfolders** to also add EDS files in subfolders beneath the folder you will select.
6	Click the **Browse** button to open the **Open** dialog.
7	Navigate to and select one of these: ● one or more EDS files ● a folder containing EDS files
8	After you make your selections, click **Open**. **Result**: The dialog closes, and your selection appears in the **Directory or File Name** field.
9	Click **Next**. **Result**: The wizard compares the selected EDS files against existing files in the library.
10	(Conditional) If one or more selected EDS files is a duplicate, a **File Already Exists** message opens. **Close** the message.

Step	Action
11	The next page of the wizard opens. It indicates the status of each device you attempted to add: ● A green check mark ✓ indicates the EDS file can be added. ● A blue informational icon indicates a redundant file. ● A red exclamation point ! indicates an invalid EDS file. (Optional) Select a file in the list, then click **View Selected File** to open it.
12	Click **Next** to add the non-duplicate files. **Result**: Page 4 of the wizard opens, indicating the action is complete.
13	Click **Finish** to close the wizard.

The next step is to update the Unity Pro **Hardware Catalog**, so that the newly added device is available for inclusion in a Unity Pro project.

Updating the Unity Pro Hardware Catalog

Updating Hardware Catalog

After you use the manufacturer's instructions to install a module or device DTM on your PC, update the Unity Pro **Hardware Catalog**. The update makes the new Ethernet module or device available to your Unity Pro application.

Update the **Hardware Catalog**:

Step	Action
1	Open the Unity Pro **Hardware Catalog** (**Tools** → **Hardware Catalog**).
2	Select the **DTM catalog** tab to display the list of DTMs.
3	Select the DTM that corresponds to the new module or device.
4	Click the **Update** buttonto refresh the window.
5	Wait for the progress bar to finish updating the **Hardware Catalog**.

A message in the **User Errors** tab (at the bottom of the Unity Pro screen) confirms that the DTM catalog update is finished.

Remove an EDS File from the Unity Pro Hardware Catalog

Overview

You can remove a module or device from the list of available devices in the Unity Pro **Hardware Catalog** by removing its **EDS** file. When you remove an EDS file from the library, the device or module is no longer displayed by Unity Pro in the **DTM Catalog** page of the **Hardware Catalog** window.

However, removing an EDS file from the library does not delete the file. Instead, the EDS file remains in its stored location and can again be added to the catalog at a future time.

Removing an EDS File from the Catalog

Remove an EDS file from the catalog:

Step	Action
1	If the **DTM Browser** is not already open, select **Tools** → **DTM Browser** in the Unity Pro main menu.
2	In the **DTM Browser**, select an M580 CPU.
3	Right-click and scroll to **Device menu** → **Additional functions** → **Remove EDS from library**.
4	Use the selection lists in the heading of this window to specify how EDS files are displayed: ● **Display:** To filter the list of displayed EDS files, make a selection: ❍ **All EDS** (no filtering) ❍ **Only Devices** ❍ **Only Chassis** ❍ **Only Modules** ● **Sort by:** To sort the list of displayed EDS files, make a selection: ❍ **File Name** ❍ **Manufacturer** ❍ **Category** ❍ **Device Name** ● **Display Name:** To see a description of each device, make a selection: ❍ **Catalot Name** ❍ **Product Name**
5	In the **Device Library** tree control, navigate to and select the EDS file you want to remove.
6	(Optional) Click the **View Selected File** button to display the read-only contents of the selected EDS file.
7	Click the **Delete Selected File** button to open a message box.
8	Click **Yes** to remove the selected EDS file from the list.
9	When you have finished removing EDS files, click **Close**.
10	The next step is to update the **Hardware Catalog**.

Section 8.10
M580 CPU Embedded Web Pages

Introduction

The M580 CPU includes a Hypertext Transfer Protocol (HTTP) server. The server transmits web pages for the purpose of monitoring, diagnosing, and controlling remote access to the communication module. The server provides easy access to the CPU from standard internet browsers.

What Is in This Section?

This section contains the following topics:

Introducing the Embedded Web Pages

Introduction

Use the embedded web server pages to:

- display real-time diagnostic data for both the M580 CPU and other networked devices
- read the values of and write values to Unity Pro application variables
- manage and control access to the embedded web pages by assigning separate passwords for:
  - viewing the diagnostic web pages
  - using the Data Editor to write values to Unity Pro application variables

Requirements

The embedded web server in the M580 CPUs displays data in standard HTML web pages. Access the embedded web pages on a PC, iPad, or Android tablet with these browsers:

- Internet Explorer (v8 or later)
- Google Chrome (v11 or later)
- Mozilla Firefox (v4 or later)
- Safari (v5.1.7 or later)

M580 CPU Diagnostic Web Pages

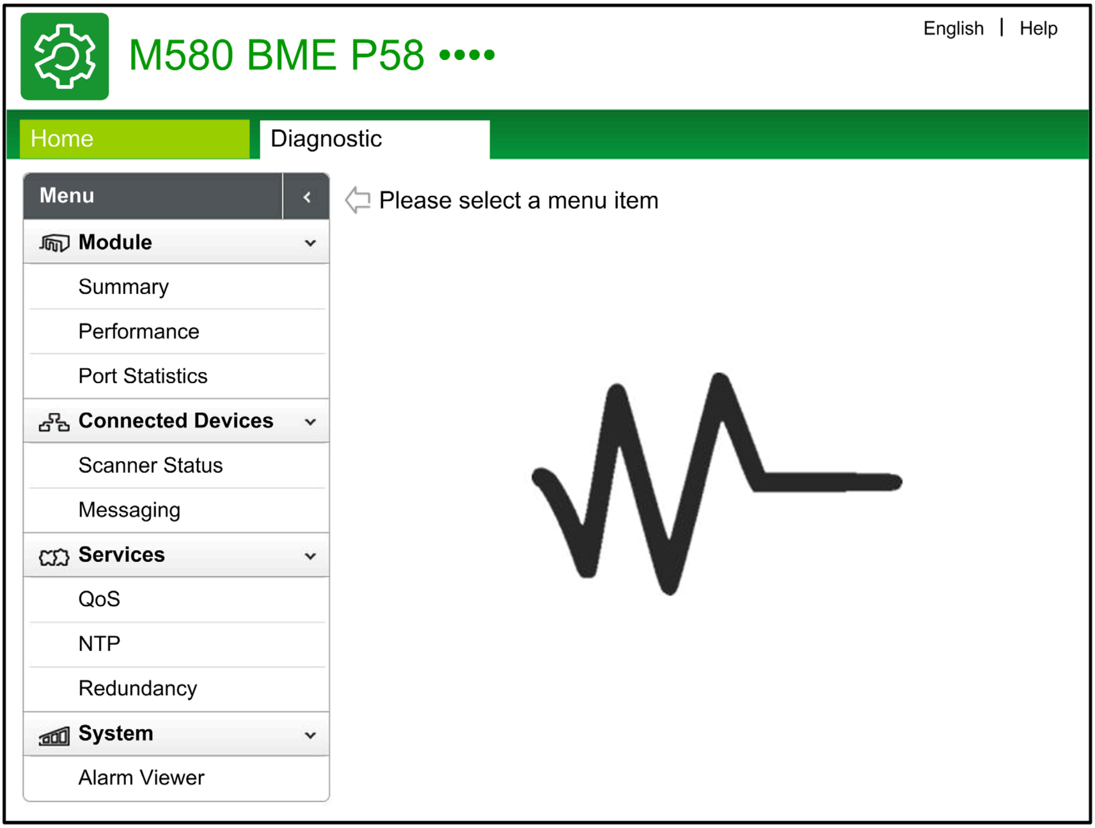
Accessing the Web Site

Access the **Diagnostic** tab:

Step	Action
1	Open an Internet browser.
2	In the address bar, enter the IP address of the M580 CPU *(see page 155)*.
3	Press **Enter** and wait for the **Home** page to open.

Navigating the Web Pages

Click the **Diagnostic** tab to navigate through the diagnostic web pages:



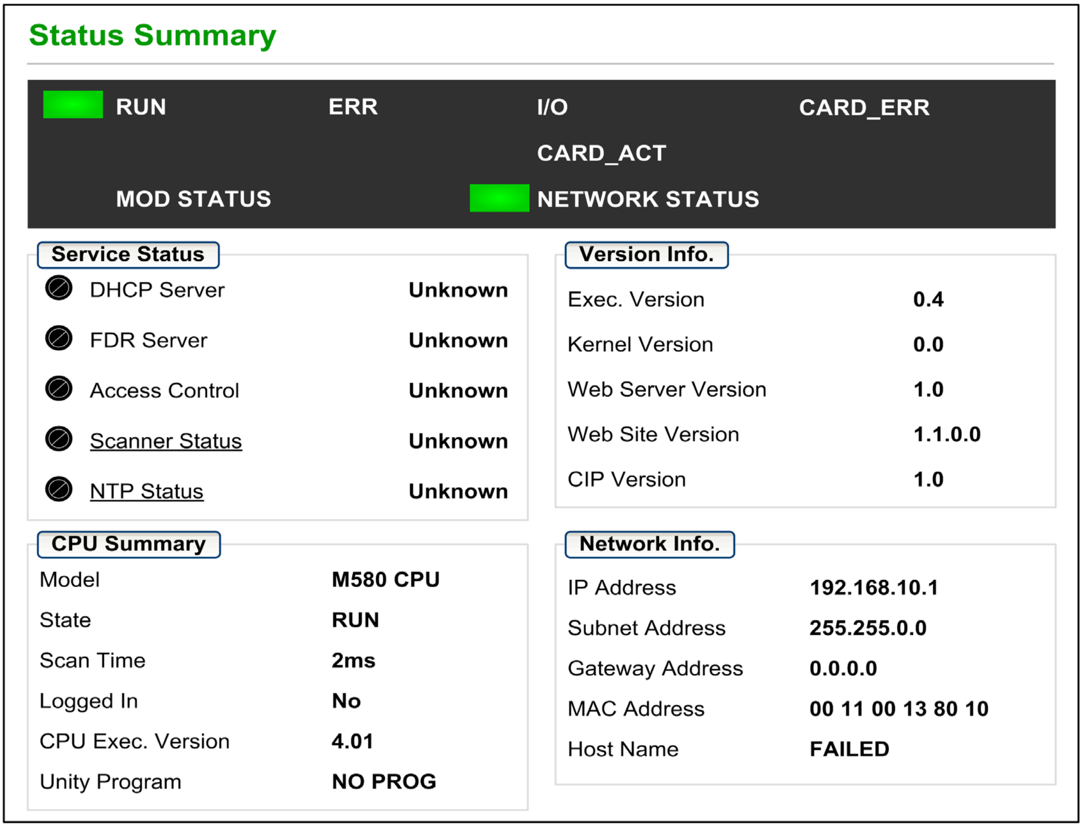
Access these pages by expanding the **Menu** on the **Diagnostic** tab:

- **Status Summary** *(see page 243)*
- **Performance** *(see page 245)*
- **Port Statistics** *(see page 247)*
- **I/O Scanner** *(see page 248)*
- **Messaging** *(see page 250)*
- **QoS** *(see page 251)*
- **Network Time Service** *(see page 252)*
- **Redundancy** *(see page 254)*

Status Summary

Open the Page

Access the **Status Summary** page on the **Diagnostics** tab (**Menu** → **Module** → **Summary**):



NOTE: This page is updated every 5 seconds.

Diagnostic Information

The objects on this page provide status information:

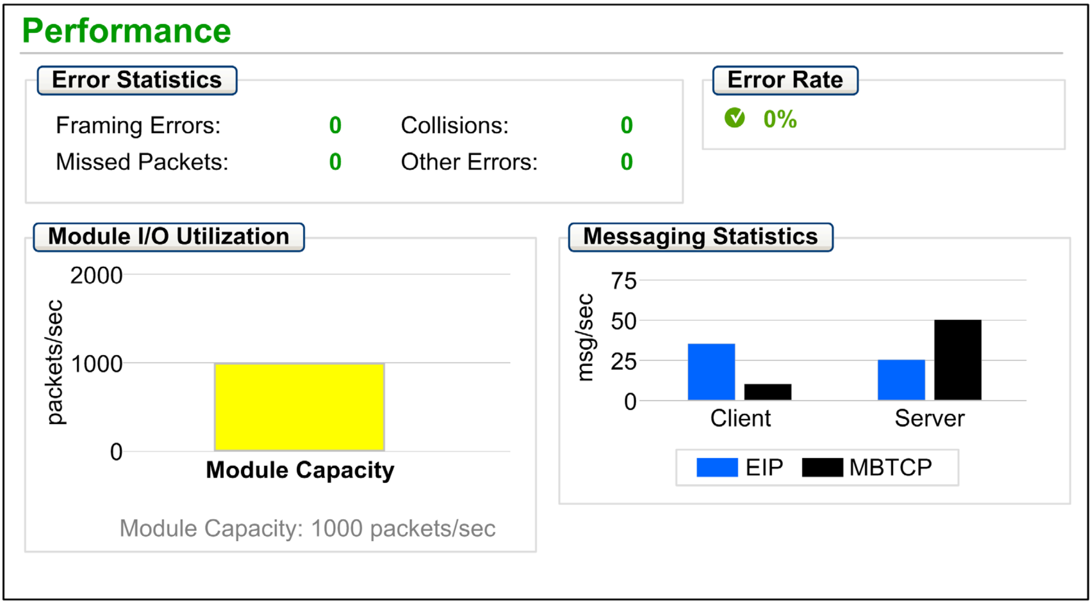
Parameters	Description
LEDs	The black field contains LED indicators (**RUN**, **ERR**, etc.). **NOTE:** The diagnostics information associated with the LED activity is described elsewhere *(see page 35)*.

Parameters	Description	
Service Status	green	The available service is operational and running.
	red	An error is detected in an available service.
	black	The available service is not present or not configured.
Version Info.	This field describes the software versions that are running on the CPU.	
CPU Summary	This field describes the CPU hardware and the applications that are running on the CPU.	
Network Info.	This field contains network and hardware address information and connectivity that corresponds to the CPU.	

Performance

Open the Page

Access the **Performance** page from the **Diagnostics** tab (**Menu** → **Module** → **Performance**):



NOTE: This page is updated every 5 seconds.

Diagnostic Information

This table describes the performance statistics:

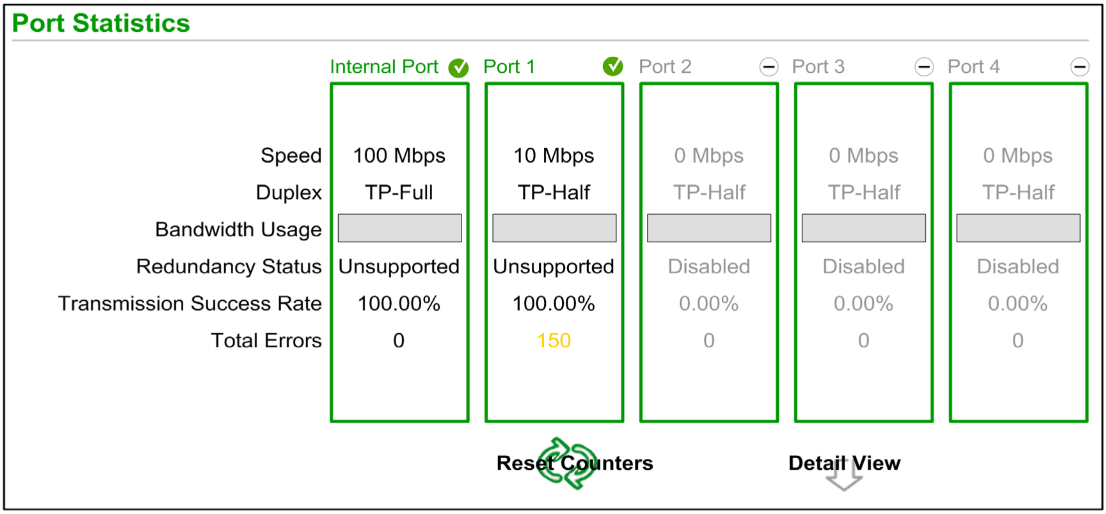
Field	Description
Error Statistics	This area contains the detected errors in the diagnostics data for the CPU. (Reset these counters to 0 with the **Reset Counters** button.)
Error Rate	This percentage represents the total number of packets divided by the number of packets that are not associated with detected errors.
Total Bandwidth Utilization	This value indicates the percentage of the available bandwidth that the CPU is using.
Module I/O Utilization	This graph shows the total number of packets (per second) the CPU can handle at once. (See the note below.)
Processor Utilization	This value represents the limit for processor use (as a percentage of the total capacity of the CPU).

Field	Description
Messaging Statistics	This graph shows the number of Modbus/TCP or EtherNet/IP *(see page 289)* messages per second for the client or server. (See the note below.)
System Bandwidth Monitor	These graphs show the percentage of bandwidth consumed by the Modbus messaging and I/O scanning services. (See the note below.)
NOTE: Move the mouse over the dynamic graphs to see the current numeric values.	

Port Statistics

Open the Page

Access the **Port Statistics** page from the **Diagnostics** tab (**Menu** → **Module** → **Port Statistics**):



NOTE: This page is updated every 5 seconds.

Diagnostic Information

This page shows the statistics for each port on the CPU. This information is associated with the configuration of the Ethernet ports *(see page 40)* and the configuration of the service/extended port *(see page 166)*.

The names of active ports are green. The names of inactive ports are gray.

Reset or expand the available information with these buttons:
- **Reset Counters**: Reset all dynamic counters to 0.
- **Detail View**: Expand the list of port statistics.

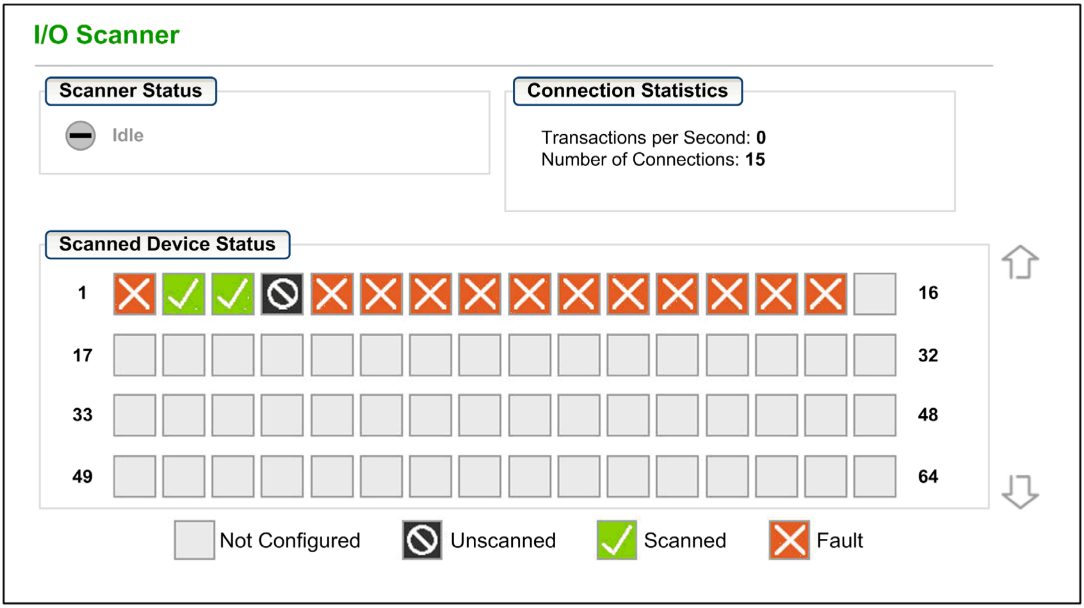
Detail View

Click **Detail View** to expand the list of parameters:

I/O Scanner

Open the Page

Access the **I/O Scanner** page from the **Diagnostics** tab (**Menu** → **Connected Devices** → **Scanner Status**):



NOTE: This page is updated every 5 seconds.

Diagnostic Information

This table describes the scanner status and connection statistics:

Scanner Status	**Enabled**	The I/O scanner is enabled.
	Disabled	The I/O scanner is disabled.
	Idle	The I/O scanner is enabled but not running.
	Unknown	The I/O scanner returns unexpected values from the device.
Connection Statistics	**Transactions per Second**	
	Number of Connections	

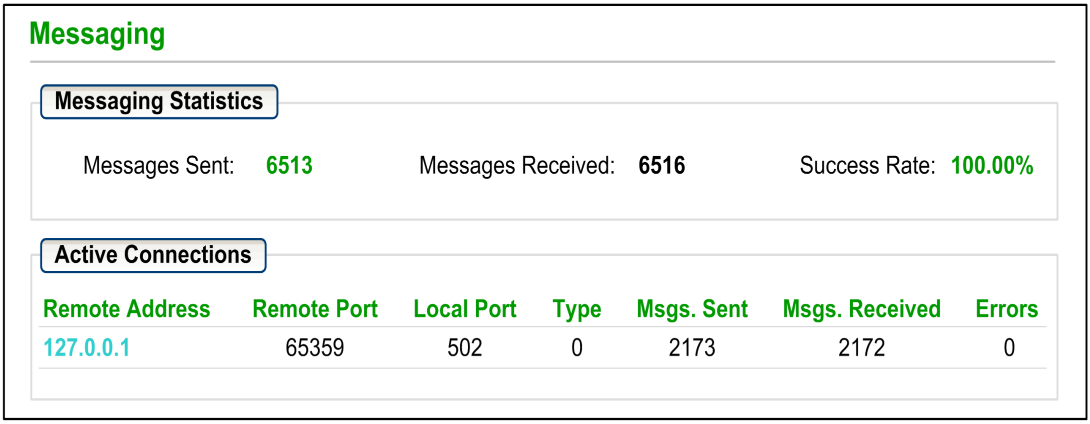
In the **Scanned Device Status** display, the colors that appear in each block indicate these states for specific remote devices:

Color	Status
gray	There is an unconfigured device.
black	The scanning of the specific device has been intentionally disabled.
green	A device is being scanned successfully.
red	A device that is being scanned is returning detected errors.

Messaging

Open the Page

Access the **Messaging** page from the **Diagnostics** tab (**Menu** → **Connected Devices** → **Messaging**):



NOTE: This page is updated every 5 seconds.

Diagnostic Information

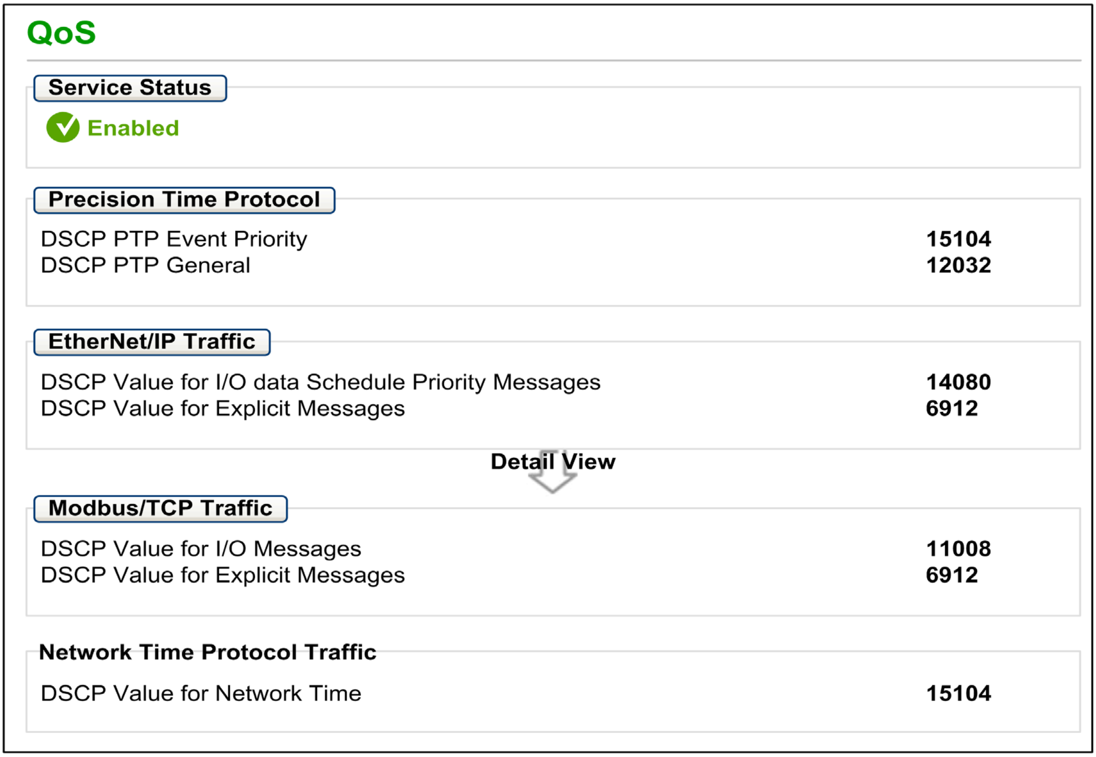
This page shows current information for open TCP connections on port 502:

- **Messaging Statistics**: This field contains the total number of sent and received messages on port 502. These values are not reset when the port 502 connection is closed. Therefore, the values indicate the number of messages that have been sent or received since the module was started.
- **Active Connections**: This field shows the connections that are active when the **Messaging** page is refreshed.

QoS

Open the Page

Access the **QoS** (quality of service) page from the **Diagnostics** tab (**Menu** → **Services** → **QoS**):



NOTE: This page is updated every 5 seconds.

Diagnostic Information

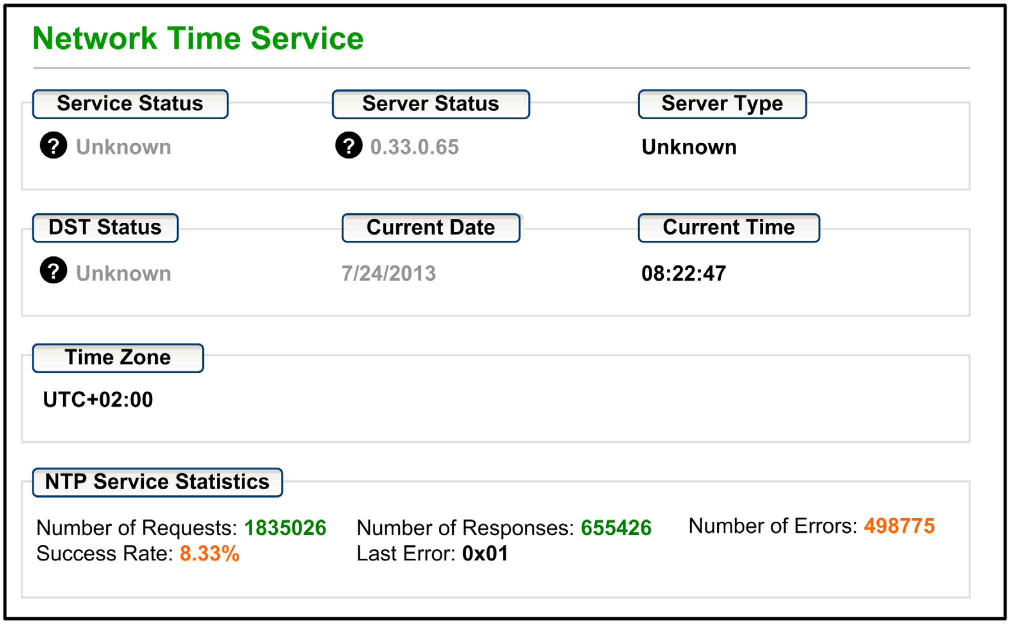
This page displays information about the QoS service. Configure this service in Unity Pro *(see page 165)*.

When you enable QoS, the module adds a differentiated services code point (DSCP) tag to each Ethernet packet it transmits, thereby indicating the priority of that packet.

Network Time Service

Open the Page

Access the **Network Time Service** page from the **Diagnostics** tab (**Menu** → **Services** → **NTP**):



NOTE: This page is updated every 5 seconds.

Diagnostic Information

This page displays information about the NTP service. Configure this service in Unity Pro *(see page 162)*.

The Network Time Service synchronizes computer clocks over the Internet for the purposes of event recording (sequence events), event synchronization (trigger simultaneous events), or alarm and I/O synchronization (time stamp alarms):

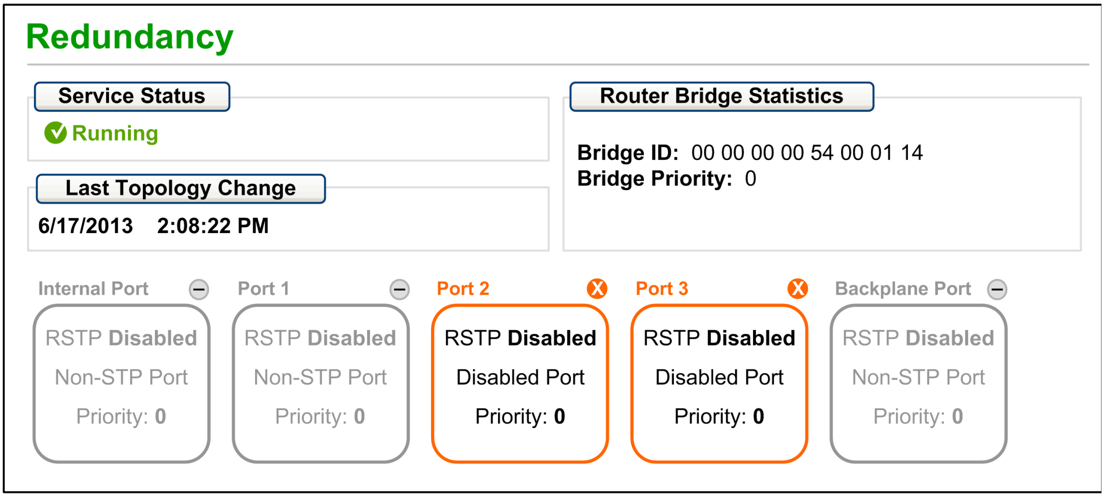
Field	Description	
Service Status	Running	The NTP service is correctly configured and running.
	Disabled	The NTP service is disabled.
	Unknown	The NTP service status is unknown.

Field	Description	
Server Status	green	The server is connected and running.
	red	A bad server connection is detected.
	gray	The server status is unknown.
Server Type	**Primary**	A primary server polls a master time server for the current time.
	Secondary	A secondary server requests the current time only from a primary server.
DST Status	**Running**	DST (daylight saving time) is configured and running.
	Disabled	DST (daylight saving time) is disabled.
	Unknown	The DST status is unknown.
Current Date	This is the current date in the selected time zone.	
Current Time	This is the current time in the selected time zone.	
Time Zone	This field shows the time zone in terms of plus or minus Universal Time, Coordinated (UTC).	
NTP Service Statistics	These fields show the current values for service statistics.	
	Number of Requests	This field shows the total number of requests sent to the NTP server.
	Success Rate	This field shows the percentage of successful requests out of the total number of requests.
	Number of Responses	This field shows the total number of responses received from the NTP server.
	Last Error	This field contains the error code of the last error that was detected during the transmission of an email message to the network.
	Number of Errors	This field contains the total number of email messages that could not be sent to the network or that have been sent but not acknowledged by the server.

Redundancy

Open the Page

Access the **Redundancy** page on the **Diagnostic** tab (**Menu** → **Services** → **Redundancy**):



NOTE: This page is updated every 5 seconds.

Diagnostic Information

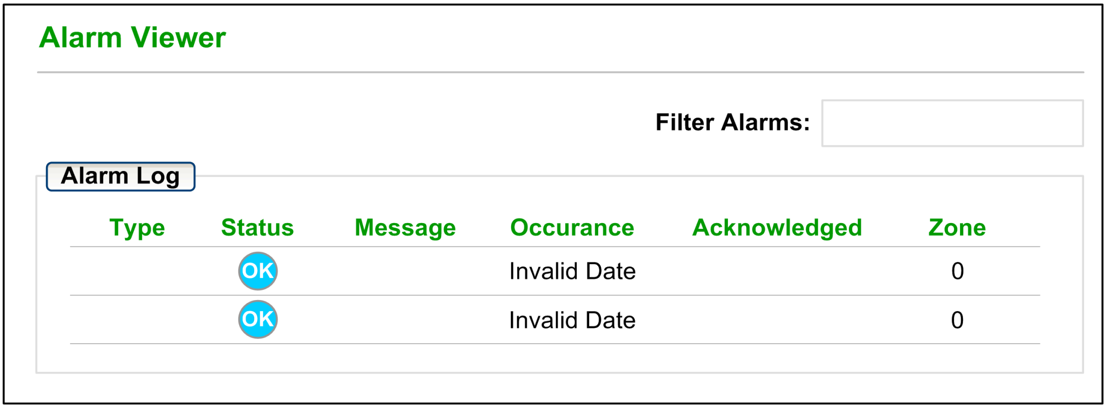
This page displays values from the RSTP configuration in Unity Pro *(see page 158)*:

Field	Description	
Service Status	This is the status (**Enabled** or **Disabled**) of the RSTP bridge on the corresponding CPU.	
Last Topology Change	These values represent the date and time that the last topology change was received for the corresponding **Bridge ID**.	
Redundancy Status	green	The designated Ethernet port is learning or formatting information.
	yellow	The designated Ethernet port is discarding information.
	gray	RSTP is disabled for the designated Ethernet port.
Router Bridge Statistics	**Bridge ID**	This unique bridge identifier is the concatenation of the bridge RSTP priority and the MAC address.
	Bridge Priority	In Unity Pro, configure the RSTP operating state *(see page 158)* of the **Bridge ID**.

Alarm Viewer

Open the Page

Access the **Alarm Viewer** page from the **Diagnostics** tab (**Menu** → **System** → **Alarm Viewer**):



NOTE: This page is updated every 5 seconds.

Diagnostic Information

The **Alarm Viewer** page reports detected application errors. You can read, filter, and sort information about alarm objects on this page. Adjust the type of information displayed by the **Alarm Viewer** in the **Filter Alarms** box.

This table describes the components of the page:

Column	Value	
Type	This column describes the alarm type.	
Status	STOP	You need to acknowledge the alarm.
	ACK	An alarm has been acknowledged.
	OK	An alarm does not require acknowledgment.
Message	This column contains the text of the alarm message.	
Occurance	This column contains the date and time that the alarm occurred.	
Acknowledged	This column reports the acknowledged status of the alarm.	
Zone	This column contains the area or geographical zone from which the alarm comes (0: common area).	

Chapter 9
M580 CPU Programming and Operating Modes

Overview

This chapter provides information on M580 CPU I/O exchanges, tasks, memory structure, and operating modes.

What Is in This Chapter?

This chapter contains the following sections:

Section 9.1
I/O and Task Management

Overview

This section presents information on M580 I/O addressing and management, tasks allowed, and I/O scanning capabilities.

What Is in This Section?

This section contains the following topics:

I/O Exchanges

I/O Vision

Each module uses a structure that represents inputs, outputs, control, and diagnostic data. The structures can be represented using:

- topological addressing / IODDT
- Device DDT

I/O Module Location	I/O Family	Topological Addressing / IODDT	Device DDT
local rack	(e)X80	X	X
	Premium	X	–
RIO	(e)X80	–	X
distributed equipment	Schneider Electric or third party	–	X
X Supported. When both visions are supported, select one of the exchange types when adding the equipment. – Not supported.			

Adding an I/O Module in Unity Pro

When you insert an I/O module on a rack in Unity Pro, the type of addressing appears in the bottom of the **New Device** dialog box. Choose between the following:

- **I/O data type: Topological** (default)
- **I/O data type: Device DDT**

NOTE: If you want to change the type of addressing you selected when you added an I/O module to your application, delete the module from your application and then insert the module again selecting the appropriate addressing type.

Exchanges Types

I/O modules in an M580 system can be controlled, read, or written with 2 types of exchanges:

- implicit exchanges
  Implicit exchanges are performed automatically on each cycle of the task (MAST, FAST, AUX0, AUX1) associated with the I/O modules. They are used to read and write inputs or outputs to the modules.
- explicit exchanges
  Explicit exchanges are performed on application request. They are used for detailed diagnostics and to set/read command and adjust parameters. They use specific function blocks.
  An acknowledgment or reply is sent once the requested action is performed. This reply may be received a few cycles after the request was sent.
  NOTE: Explicit exchanges are performed in a MAST task.

Explicit Exchanges

Function block usage depending on the module location and I/O vision selected for the module:

I/O Module Location	I/O Vision	Function Block
Local rack	Topological addressing / IODDT	READ_PARAM
		READ_STS
		READ_TOPO_ADDR
		RESTORE_PARAM
		SAVE_PARAM
		WRITE_CMD
		WRITE_PARAM
		READ_VAR
		WRITE_VAR
		DATA_EXCH
	Device DDT	READ_PARAM_MX
		READ_STS_MX **NOTE:** MOD_FAULT parameter is not automatically updated; a READ_STS_MX must be performed.
		RESTORE_PARAM_MX
		SAVE_PARAM_MX
		WRITE_CMD_MX
		WRITE_PARAM_MX
RIO and local rack	Device DDT	READ_STS_MX
		WRITE_CMD_MX

The function blocks mentioned in previous table are detailed in the *Explicit Exchange* part of *Unity Pro, I/O Management, Block Library manual*, and in the *Extended* part of *Unity Pro, Communication, Block Library manual*.

CPU Tasks

Introduction

An M580 CPU can execute single-task and multi-task applications. Unlike a single-task application which only executes master tasks, a multi-task application defines the priorities of each task.

There are 4 tasks available (see *Application Program Structure* chapter in *Unity Pro Program Languages and Structure Reference Manual*) and 2 types of event tasks:

- MAST
- FAST
- AUX0
- AUX1
- I/O event in a local rack only
- Timer event in a local rack only

Task Characteristics

The time model, task period, and maximum number of tasks per CPU are defined as follows:

Task	Time Model	Task Period (ms)		BME P58 References			
		Range	Default Value	10•0 (H)	20•0 (H)	30•0	40•0
MAST(1.)	cyclic(2.) or periodic	1...255	20	1	1	1	1
FAST	periodic	1...255	5	1	1	1	1
AUX0	periodic	10...2550 by 10	100	1	1	1	1
AUX1	periodic	10...2550 by 10	200	1	1	1	1
1. MAST task is mandatory. 2. When set to cyclic mode, the minimum cycle time is 8 ms if there is a RIO network and 1 ms if there is no RIO network in the system.							

Section 9.2
BME P58 *xxxx* CPU Memory Structure

Memory Structure

CPU Memory

3 types of memories are available in a BME P58 •••• CPU:

- non-persistent application RAM: used to run the application program and store temporary data
- flash memory: used to back up the application program and a copy of %MW values
- optional SD memory card to store application and data in parallel to the CPU flash memory

NOTE: The optional SD memory card is usefull to backup the application in case of hardware issue. The SD memory card with backup data can be inserted in a new CPU to run the application.

Application Download to the CPU Memory

CPU memory involved during an application download from a programming terminal:

- Application is transferred into the non-persistent application RAM.
- If a memory card is inserted, working and not write protected, then an internal backup is performed in the memory card.
- The application backup is performed in the the flash memory.

NOTE: A write protected memory card inserted disables the application download.

Application Upload from the CPU Memory

The application upload reads and uploads the non-persistent application RAM content.

Application Online Modification Backup

An application program modification is performed in the CPU non-persistent memory with an automatic backup performed as follows:

- If a memory card is inserted, working and not write protected, then the backup is performed in the memory card.
- The application backup is performed in the flash memory.

NOTE: The online modification is disabled when a write protected memory card is inserted.

Application Memory Self Modification

The user code may modify the application content (for example to save I/O parameters or replace variables initial value by the current value).

In such a case, only the non-persistent application RAM content is modified.

To back up the application in the memory card and to the flash memory, use the system bit `%S66`.

Section 9.3
BME P58 *xxxx* CPU Operating Modes

Overview

This section provides information on the CPU operating modes.

What Is in This Section?

This section contains the following topics:

Managing Run/Stop Input

Managing Remote Run/Stop Access

When configuring the M580 CPU, you can help prevent remote commands/requests from accessing the CPU **Run/Stop** modes. Select the respective **Run/Stop input** and **Run/Stop only by input** check boxes according to the following table parameters to determine the type of remote access for your system.

Run/Stop Input	Run/Stop Only By Input	Description
–	–	Allows remote access to run/stop input by request only.
X	–	Allows remote access to run/stop input.
X	X	Denies remote access to run/stop input.
X: check box selected –: check box deselected		

Power Cut and Restore

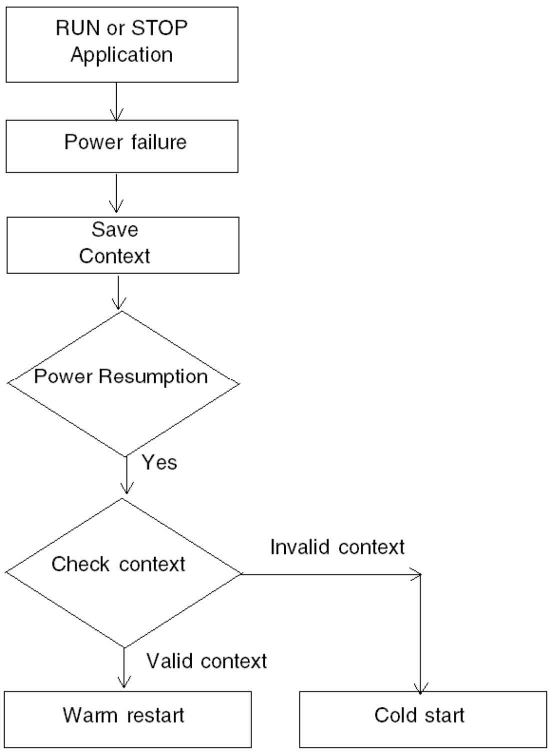
Introduction

If the duration of the outage is lower than the power supply filtering time, it has no effect on the program which continues to run normally.

If the duration of the outage is higher than the power supply filtering time, the program is interrupted and power restoration processing is activated. The CPU then restarts in warm restart or cold start as described in the following diagram.

Illustration

Power cycle phases:



Power Supply Filtering Times

The BMX CPS 2000, BMX CPS 3500, and BMX CPS 3540T power supplies, which provide Vac power, have a filtering time of 10 ms.

The BMX CPS 2010 and BMX CPS 3020 power supplies, which provide Vdc power, have a filtering time of 1 ms.

Power Outage Processing Phases

When power to the system is lost, it recovers in 3 phases:

Phase	Description
1	On power outage, the system saves the application context, the values of application variables, and the state of the system on internal flash memory.
2	The system sets all the outputs into fallback state (state defined in configuration).
3	On power restoral, some actions and checks are done to verify if warm restart is available: ● restore internal flash memory application context ● verify application and context validity If all checks are correct a warm restart *(see page 271)* is performed, otherwise a cold start *(see page 268)* is carried out.

Cold Start

CPU Cold Start Causes

Cold start causes and resulting CPU state:

Cause	Resulting CPU State
End of the application download.	STOP
Application restored from flash memory is different than the one in the non-persistent application RAM. Use case: • application restored from a memory card if a compatible memory card is in the card slot • Application restored from the CPU flash memory	STOP(1.)
Application restored from persistent memory with Unity Pro command **PLC → Project backup → ....** is different than the one in the non-persistent application RAM: • application restored from a memory card if a compatible memory card is in the card slot • Application restored from the CPU flash memory	STOP(1.)
Power supply **RESET** button pressed.	STOP(1.)
Power supply **RESET** button pressed less than 500 ms after a power down.	STOP(1.)
Power supply **RESET** button pressed after a CPU detected error, except in the case of a watchdog detected error (halt state).	STOP(2.)
Init requested with one of the 3 following means: • `%S0` system bit set to 0 • INIT request • **Cold Start** command in Unity Pro	The CPU does not change its state. It only initializes the application. It is a simulation of cold start.
Restoral after power down with a loss of context.	STOP(1.)
1. CPU state is set to RUN if **Automatic start in Run** option is selected. 2. **Automatic start in Run** option does not set the CPU to RUN state.	

Loading or transferring an application to the CPU involves initialization of unlocated variables.

You need to assign a topological address to the data if the process requires keeping the current values of the data when transferring the application.

To save the located variables, avoid the initialization of the `%MWi` by unchecking **Initialize %MWi on cold start** parameter in the CPU configuration screen.

NOTE: Pressing the **RESET** button on the power supply resets `%MWi` and initial values are loaded.

NOTE: Do not press the **RESET** button on the power supply if you do not want `%MWi` to be reset and loaded with initial values.

Executing a Cold Start

Use these steps to perform a cold start:

<table>
<tr><th>Phase</th><th>Description</th></tr>
<tr><td>1</td><td>The startup is performed in RUN or in STOP state depending on one of the 2 following conditions:
The status of the Automatic start in Run parameter defined in the CPU configuration. If the parameter is selected, the start will be performed in RUN.
The state of the I/O defined in the Run/Stop input parameter in the CPU configuration.
Program execution is resumed at the start of the cycle.</td></tr>
<tr><td>2</td><td>The system carries out the following:
Disable FAST, AUX, and event tasks.
MAST task is executed until the end of data initialization.
Initialize data (bits, I/O image, words, and so on) with the initial values defined in the data editor (value set to 0 if no other initial value has been defined). For <code>%MW</code> words, the values can be retrieved on a cold start when these conditions are met:
The Initialize %MWi on cold start parameter is not checked in the CPU configuration screen,
The internal flash memory has a valid backup (see %SW96).
NOTE: If the number of <code>%MW</code> words exceeds the backup size during the save operation the remaining words are set to 0.
Initialize elementary function blocks (initial data).
Initialize data declared in the DFBs: either to 0 or to the initial value declared in the DFB type.
Initialize system bits and words.
Position charts to initial steps.
Cancel any forcing action.
Initialize message and event queues.
Send configuration parameters to all I/O and application-specific modules.</td></tr>
<tr><td>3</td><td>To start a cycle, the system performs these tasks:
Relaunch the MAST task with the <code>%S0</code> (cold start) and <code>%S13</code> (first cycle in RUN) system bits set to 1. <code>%SW10</code> (first cycle after cold start) system word is set to 0.
Reset the <code>%S0</code> and <code>%S13</code> system bits to 0 and set each bit of <code>%SW10</code> system word to 1 at the end of this first cycle of the MAST task.
Activate the FAST and AUX tasks and event processing at the end of the first cycle of the MAST task.</td></tr>
</table>

Processing a Cold Start by Program

Test `%SW10.0` system bit to detect a cold start and adapt the program consequently.

NOTE: It is possible to test the `%S0` system bit on the first execution cycle if the **Automatic start in RUN** parameter is selected. If it is not selected, the CPU starts in STOP state and the bit `%S0` switches to 1 on the first cycle after start (not visible for the program).

Output Changes

As soon as a power outage is detected the outputs are set in the fallback position configured (programmed fallback value or current value).

On power down, the outputs are not driven and remain at 0.

After power restoral, the outputs remain at 0 until they are updated by the task.

Warm Restart

Introduction

A warm restart occurs after a power cycle.

Executing a Warm Restart

Phase	Description
1	Program execution does not resume from the element where the power outage occurred. The remaining program is discarded during the warm restart. Each task restarts from the beginning.
2	The system carries out the following: ● Restore the application variables value, ● Set `%S1` system bit to 1. ● Initialize message and event queues, ● Send configuration parameters to all I/O and application-specific modules, ● If the application was reserved, the CPU removes the reservation. ● Reset communication. ● If needed, the CPU configures the I/O modules with the current adjustment parameters. ● Disable FAST, AUX, and event tasks.
3	The system performs a restart cycle during which it: ● Restarts the MAST task from beginning of cycle, ● Sets `%S1` system bit to 0 when the MAST task is completed. ● Enable FAST, AUX, and event tasks at the end of the first MAST task cycle. ● CPU state set to the value before power down. If the CPU was in HALT state, it is set to STOP state.

Processing a Warm Restart by Program

On warm restart, if the application needs to be processed in a particular way, the program needs to test that `%S1` system bit is set to 1 at the start of the MAST task program.

SFC Warm Restart Specific Features

The warm start on Modicon M580 CPU is not considered as a real warm start by the CPU. SFC interpreter does not depend on tasks.

SFC publishes a `ws_data` memory area to the OS that contains SFC section-specific data to be saved on power down.

At the beginning of chart processing the active steps are saved to ws_data and processing is marked to be in a critical section. At the end of chart processing the critical section is unmarked.

If a power down hits into the critical section, it could be detected if this state is active at the beginning (as the scan is aborted and MAST task is restarted from the beginning). In this case, the workspace may be inconsistent and is restored from the saved data.

Additional information from `SFCSTEP_STATE` variable in located data area is used to reconstruct the state machine.

When a power down occurs, the following is performed:
- During first scan, `%S1` = 1, MAST task is executed but FAST and event tasks are not executed.

On power restoral, the following is performed:
- clear chart, deregister diagnostics, keep set actions
- set steps from saved area
- set step times from `SFCSTEP_STATE`
- suppress execution of the P / P1 actions
- restores elapsed time for timed actions

NOTE: SFC interpreter is independent, if the transition is valid, the SFC chart evolves while `%S1` = 1.

Output Changes

As soon as a power outage is detected the outputs are set in the fallback position configured: either programmed fallback value or current value.

After power restoral, the outputs remain at 0 until they are updated by the task.

Appendices

Appendix A
Derived Data Types

Device DDT Names for the M580 CPU

Introduction

This topic describes the Unity Pro **Device DDT** tab for an M580 CPU in a local rack. A derived data type (DDT) is a set of elements with the same type (ARRAY) or with different types (structure).

The default device DDT name is `BMEP58_ECPU`, of `T_BMEP58_ECPU` type.

Access the Device DDT Tab

In Unity Pro:

Step	Action	Comment
1	Open the **Data Editor** in the Unity Pro **Project Browser**.	Path: **Tools → Data Editor**
2	Select the **Device DDT** checkbox.	

Parameters

Use the Unity Pro **Device DDT** tab to configure parameters for the CPU RIO head on the local rack:

Parameter		Description
Implicit device DDT	Name	the default name of the device DDT
	Type	module type (uneditable)
Goto details		link to the DDT data editor screen

Standalone Configuration

This table describes the fields in the `BMEP58_ECPU` implicit device DDT type that is used with the CPU RIO head module in standalone configurations.

`ETH_STATUS` (WORD):

Name	Type	Bit	Description
`PORT1_LINK`	BOOL	0	0 = Port 1 link is down.
			1 = Port 1 link is up.
`PORT2_LINK`	BOOL	1	0 = Port 2 link is down.
			1 = Port 2 link is up.

Name	Type	Bit	Description
PORT3_LINK	BOOL	2	0 = Port 3 link is down.
			1 = Port 3 link is up.
ETH_BKP_PORT_LINK	BOOL	3	0 = Ethernet backplane link is down.
			1 = Ethernet backplane link is up.
HSBY_LINK	BOOL	4	(reserved)
REDUNDANCY_STATUS	BOOL	5	0 = Redundant path is not available.
			1 = Redundant path is available.
SCANNER_OK	BOOL	6	0 = Scanner is not present.
			1 = Scanner is present.
GLOBAL_STATUS	BOOL	7	0 = At least 1 service is not operating normally.
			1 = All services are operating normally.
(reserved)	BYTE	8–15	(reserved)

NOTE: You can monitor breaks in the RIO main ring by diagnosing the `REDUNDANCY_STATUS` bits in the CPU module device DDT. The system detects and reports in this bit a main ring cable break that persists for at least 5 seconds.
Within the `REDUNDANCY_STATUS` bit:
- 0: The cable is broken or the device is stopped.
- 1: The loop is present and healthy.

`SERVICE_STATUS` (WORD):

Name	Type	Bit	Description
RSTP_SERVICE	BOOL	0	0 = RSTP service is not operating normally.
			1 = RSTP service is operating normally or disabled.
(reserved)	BOOL	1	(reserved)
PORT502_SERVICE	BOOL	2	0 = Ethernet Port 502 service is not operating normally.
			1 = Ethernet Port 502 service is operating normally or disabled.
SNMP_SERVICE	BOOL	3	0 = SNMP service is not operating normally.
			1 = SNMP service is operating normally or disabled.
MAIN_IP_ADDRESS_STATUS	BOOL	4	0 = IP address is a duplicate or unassigned.
			1 = Assigned IP address is not a duplicate.

Name	Type	Bit	Description
ETH_BKP_FAILURE	BOOL	5	0 = Ethernet backplane hardware has a detected issue. 1 = Ethernet backplane hardware is operating properly.
ETH_BKP_ERROR	BOOL	6	0 = Ethernet backplane error detected. 1 = Ethernet backplane is operating properly.
EIP_SCANNER	BOOL	7	0 = Service not operating normally. 1 = Service operating normally.
MODBUS_SCANNER	BOOL	8	0 = Service not operating normally. 1 = Service operating normally.
NTP_SERVER	BOOL	9	0 = SNTP server not operating normally. 1 = SNTP server operating normally.
SNTP¨_CLIENT	BOOL	10	0 = Service not operating normally. 1 = Service operating normally.
WEB_SERVER	BOOL	11	0 = Service not operating normally. 1 = Service operating normally.
FIRMWARE_UPGRADE	BOOL	12	0 = Service not operating normally. 1 = Service operating normally.
FTP	BOOL	13	0 = Service not operating normally. 1 = Service operating normally.
FDR_SERVER	BOOL	14	0 = Service not operating normally. 1 = Service operating normally.
EIP_ADAPTER	BOOL	15	0 = EIP adapter (server) service not operating normally. 1 = EIP adapter (server) service operating normally.

SERVICE_STATUS2 (WORD):

Name	Type	Bit	Description
A_B_IP_ADDRESS_STATUS	BOOL	0	0 = Duplicate IP or no IP address assigned. 1 = IP addresses correctly assigned.
LLDP_SERVICE	BOOL	1	0 = LLDP service is not operating normally. 1 = LLDP service is operating normally or disabled.
(reserved)	–	2–15	(reserved)

`ETH_PORT_1_2_STATUS` (BYTE):

Name	Bit	Description
Ethernet ports function and RST role coded on 2 bits	0–1	Ethernet port 1 function
	2–3	Ethernet port 1 RSTP role
	4–5	Ethernet port 2 function
	6–7	Ethernet port 2 RSTP role

Port function and RSTP role features description:

Feature	Value	Description
port function	0	disabled
	1	access port
	2	port mirror
	3	device network port
RSTP role	0	alternate
	1	backup
	2	designated
	3	root

`ETH_PORT_3_BKP_STATUS` (BYTE):

Name	Bit	Description
Ethernet ports function and RST role coded on 2 bits	0–1	Ethernet port 3 function
	2–3	Ethernet port 3 RSTP role
	4–5	backplane Ethernet function (2 bits value): ● 0: backplane without Ethernet network ● 3: backplane with Ethernet network
	6–7	(reserved)

Port function and RSTP role features description:

Feature	Value	Description
port function	0	disabled
	1	access port
	2	port mirror
	3	device network port

Feature	Value	Description
RSTP role	0	alternate
	1	backup
	2	designated
	3	root

`IN_PACKETS` (UINT):

Type	Bit	Description
UINT	0–7	number of packets received on the interface (internal ports)

`IN_ERRORS` (UINT):

Type	Bit	Description
UINT	0–7	number of inbound packets that contain detected errors

`OUT_PACKETS` (UINT):

Type	Bit	Description
UINT	0–7	number of packets sent on the interface (internal ports)

`OUT_ERRORS` (UINT):

Type	Bit	Description
UINT	0–7	number of outbound packets that contain detected errors

`CONF_SIG` (UDINT):

Type	Bit	Description
UDINT	0–15	Signatures of all files on local module FDR server

CRA_CNX_HEALTH (ARRAY [1...16] OF BYTE):

Name	Type	Rank	Description
CRA_CNX_HEALTH[1]	BYTE	0	CRA module number 1 connection health status.
CRA_CNX_HEALTH[2]	BYTE	1	CRA module number 2 connection health status.
CRA_CNX_HEALTH[3]	BYTE	2	CRA module number 3 connection health status.
...			
CRA_CNX_HEALTH[16]	BYTE	15	CRA module number 16 connection health status.
NOTE: Each byte provides details of input and output per task for a dedicated CRA module: ● Bit 0: FAST task input ● Bit 1: MAST task input ● Bit 2: AUX0 task input ● Bit 3: AUX1 task input ● Bit 4: FAST task output ● Bit 5: MAST task output ● Bit 6: AUX0 task output ● Bit 7: AUX1 task output			

DEVICE_CNX_HEALTH (ARRAY [1..7] OF WORD):

Name	Type	Rank	Description
DEVICE_CNX_HEALTH[1]	WORD	0	DIO connection status (DIO 1 to 16, 1 bit per DIO)
DEVICE_CNX_HEALTH[2]	WORD	1	DIO connection status (DIO 17 to 32, 1 bit per DIO)
DEVICE_CNX_HEALTH[3]	WORD	2	DIO connection status (DIO 33 to 48, 1 bit per DIO)
DEVICE_CNX_HEALTH[4]	WORD	3	DIO connection status (DIO 49 to 64, 1 bit per DIO)
DEVICE_CNX_HEALTH[5]	WORD	4	DIO connection status (DIO 65 to 80, 1 bit per DIO)
DEVICE_CNX_HEALTH[6]	WORD	5	DIO connection status (DIO 81 to 96, 1 bit per DIO)
DEVICE_CNX_HEALTH[7]	WORD	6	DIO connection status (DIO 97 to 112, 1 bit per DIO)
DEVICE_CNX_HEALTH[8]	WORD	7	DIO connection status (DIO 113 to 128, 1 bit per DIO)
NOTE: DIO number = DIO device number from the mapping list - 32. For example, DIO device number 37 in the mapping list corresponds to DIO number 5 (37 - 32) in the above table.			

CRA_CNX_CTRL (ARRAY [1..16] OF BYTE):

Name	Type	Rank	Description
CRA_CNX_CTRL[1]	BYTE	0	CRA module number 1 connection CTRL status.
CRA_CNX_CTRL[2]	BYTE	1	CRA module number 2 connection CTRL status.
CRA_CNX_CTRL[3]	BYTE	2	CRA module number 3 connection CTRL status.
CRA_CNX_CTRL[4]	BYTE	3	CRA module number 4 connection CTRL status.
CRA_CNX_CTRL[5]	BYTE	4	CRA module number 5 connection CTRL status.
CRA_CNX_CTRL[6]	BYTE	5	CRA module number 6 connection CTRL status.
CRA_CNX_CTRL[7]	BYTE	6	CRA module number 7 connection CTRL status.
CRA_CNX_CTRL[8]	BYTE	7	CRA module number 8 connection CTRL status.
CRA_CNX_CTRL[9]	BYTE	8	CRA module number 9 connection CTRL status.
CRA_CNX_CTRL[10]	BYTE	9	CRA module number 10 connection CTRL status.
CRA_CNX_CTRL[11]	BYTE	10	CRA module number 11 connection CTRL status.
CRA_CNX_CTRL[12]	BYTE	11	CRA module number 12 connection CTRL status.
CRA_CNX_CTRL[13]	BYTE	12	CRA module number 13 connection CTRL status.
CRA_CNX_CTRL[14]	BYTE	13	CRA module number 14 connection CTRL status.
CRA_CNX_CTRL[15]	BYTE	14	CRA module number 15 connection CTRL status.
CRA_CNX_CTRL[16]	BYTE	15	CRA module number 16 connection CTRL status.
NOTE: Each byte provides details of input and output per task for a dedicated CRA module: ● Bit 0: FAST task input ● Bit 1: MAST task input ● Bit 2: AUX0 task input ● Bit 3: AUX1 task input ● Bit 4: FAST task output ● Bit 5: MAST task output ● Bit 6: AUX0 task output ● Bit 7: AUX1 task output			

DIO_CTRL (T_DIO_CTRL):

Name	Type	Rank	Description
DEVICE_CNX_CTRL_256_271	WORD	0	device connection CTRL bits 256 to 271
DEVICE_CNX_CTRL_272_287	WORD	1	device connection CTRL bits 272 to 287
DEVICE_CNX_CTRL_288_303	WORD	2	device connection CTRL bits 288 to 303
NOTE: Device connection refers to objects containing inside connections, each bit referring to a specific task.			

Name	Type	Rank	Description
DEVICE_CNX_CTRL_304_319	WORD	3	device connection CTRL bits 304 to 319
DEVICE_CNX_CTRL_320_335	WORD	4	device connection CTRL bits 320 to 335
DEVICE_CNX_CTRL_336_351	WORD	5	device connection CTRL bits 336 to 351
DEVICE_CNX_CTRL_352_367	WORD	6	device connection CTRL bits 352 to 367
DEVICE_CNX_CTRL_368_383	WORD	7	device connection CTRL bits 368 to 383
DEVICE_CNX_CTRL_384_399	WORD	8	device connection CTRL bits 384 to 399
DEVICE_CNX_CTRL_400_415	WORD	9	device connection CTRL bits 400 to 415
DEVICE_CNX_CTRL_416_431	WORD	10	device connection CTRL bits 416 to 431
DEVICE_CNX_CTRL_432_447	WORD	11	device connection CTRL bits 432 to 447
DEVICE_CNX_CTRL_448_463	WORD	12	device connection CTRL bits 448 to 463
DEVICE_CNX_CTRL_464_479	WORD	13	device connection CTRL bits 464 to 479
DEVICE_CNX_CTRL_480_495	WORD	14	device connection CTRL bits 480 to 495
DEVICE_CNX_CTRL_496_511	WORD	15	device connection CTRL bits 496 to 511
NOTE: Device connection refers to objects containing inside connections, each bit referring to a specific task.			

Glossary

!

%I

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

%IW

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

%M

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

%MW

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

%Q

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

%QW

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

%SW

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

A

adapter

An adapter is the target of real-time I/O data connection requests from scanners. It cannot send or receive real-time I/O data unless it is configured to do so by a scanner, and it does not store or originate the data communications parameters necessary to establish the connection. An adapter accepts explicit message requests (connected and unconnected) from other devices.

advanced mode

In Unity Pro, advanced mode is a selection that displays expert-level configuration properties that help define Ethernet connections. Because these properties should be edited only by people with a good understanding of EtherNet/IP communication protocols, they can be hidden or displayed, depending upon the qualifications of the specific user.

architecture

Architecture describes a framework for the specification of a network that is constructed of these components:

- physical components and their functional organization and configuration
- operational principles and procedures
- data formats used in its operation

ARRAY

An `ARRAY` is a table containing elements of a single type. This is the syntax: `ARRAY [<limits>] OF <Type>`

Example: `ARRAY [1..2] OF BOOL` is a one-dimensional table with two elements of type `BOOL`.

`ARRAY [1..10, 1..20] OF INT` is a two-dimensional table with 10x20 elements of type `INT`.

ART

(*application response time*) The time a PAC application takes to react to a given input. ART is measured from the time a physical signal in the PAC turns on and triggers a write command until the remote output turns on to signify that the data has been received.

AUX

An (AUX) task is an optional, periodic processor task that is run through its programming software. The AUX task is used to execute a part of the application requiring a low priority. This task is executed only if the MAST and FAST tasks have nothing to execute. The AUX task has 2 sections:
- IN: Inputs are copied to the IN section before execution of the AUX task.
- OUT: Outputs are copied to the OUT section after execution of the AUX task.

B

BCD

(*binary-coded decimal*) Binary encoding of decimal numbers.

BOOL

(*boolean type*) This is the basic data type in computing. A `BOOL` variable can have either of these values: 0 (`FALSE`) or 1 (`TRUE`).

A bit extracted from a word is of type `BOOL`, for example: `%MW10.4`.

BOOTP

(*bootstrap protocol*) A UDP network protocol that can be used by a network client to automatically obtain an IP address from a server. The client identifies itself to the server using its MAC address. The server, which maintains a pre-configured table of client device MAC addresses and associated IP addresses, sends the client its defined IP address. The BOOTP service utilizes UDP ports 67 and 68.

broadcast

A message sent to all devices in the subnet.

C

CCOTF

(*change configuration on the fly*) A feature of Unity Pro that allows a PAC hardware change in the system configuration while the system is operating and not impacting other active operations.

CIP™

(*common industrial protocol*) A comprehensive suite of messages and services for the collection of manufacturing automation applications (control, safety, synchronization, motion, configuration and information). CIP allows users to integrate these manufacturing applications with enterprise-level Ethernet networks and the internet. CIP is the core protocol of EtherNet/IP.

class 1 connection

A CIP transport class 1 connection used for I/O data transmission via implicit messaging between EtherNet/IP devices.

class 3 connection

A CIP transport class 3 connection used for explicit messaging between EtherNet/IP devices.

connected messaging

In EtherNet/IP, connected messaging uses a CIP connection for communication. A connected message is a logical relationship between 2 or more application objects on different nodes. The connection establishes a virtual circuit in advance for a particular purpose, such as frequent explicit messages or real-time I/O data transfers.

connection

A virtual circuit between 2 or more network devices, created prior to the transmission of data. After a connection is established, a series of data is transmitted over the same communication path, without the need to include routing information, including source and destination address, with each piece of data.

connection originator

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

connectionless

Describes communication between 2 network devices, whereby data is sent without prior arrangement between the 2 devices. Each piece of transmitted data also includes routing information, including source and destination address.

control network

An Ethernet-based network containing PACs, SCADA systems, an NTP server, PCs, AMS, switches, etc. Two kinds of topologies are supported:

- flat: All modules and devices in this network belong to same subnet.
- 2 levels: The network is split into an operation network and an inter-controller network. These 2 networks can be physically independent, but are generally linked by a routing device.

CPU

(*central processing unit*) The CPU, also known as the processor or controller, is the brain of an industrial manufacturing process. It automates a process as opposed to relay control systems. PACs are computers suited to survive the harsh conditions of the industrial environment.

D

DDT

(*derived data type*) A derived data type is a set of elements with the same type (`ARRAY`) or with different types (structure).

determinism

For a defined application and architecture, you can predict that the delay between an event (change of value of an input) and the corresponding change of a controller output is a finite time *t*, smaller than the deadline required by your process.

Device DDT (DDDT)

A Device DDT is a DDT predefined by the manufacturer and not modifiable by user. It contains the I/O language elements of an I/O module.

device network

An Ethernet-based network within an RIO network that contains both RIO and distributed equipment. Devices connected on this network follow specific rules to allow RIO determinism.

DFB

(*derived function block*) DFB types are function blocks that can be defined by the user in ST, IL, LD or FBD language.

Using these DFB types in an application makes it possible to:

- simplify the design and entry of the program
- make the program easier to read
- make it easier to debug
- reduce the amount of code generated

DHCP

(*dynamic host configuration protocol*) An extension of the BOOTP communications protocol that provides for the automatic assignment of IP addressing settings, including IP address, subnet mask, gateway IP address, and DNS server names. DHCP does not require the maintenance of a table identifying each network device. The client identifies itself to the DHCP server using either its MAC address, or a uniquely assigned device identifier. The DHCP service utilizes UDP ports 67 and 68.

DIO

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

DIO cloud

A group of distributed equipment that is not required to support RSTP. DIO clouds require only a single (non-ring) copper wire connection. They can be connected to some of the copper ports on DRSs, or they can be connected directly to the CPU in the *local rack*. DIO clouds cannot be connected to *sub-rings*.

DIO network

A network containing distributed equipment, in which I/O scanning is performed by a CPU with DIO scanner service on the local rack. DIO network traffic is delivered after RIO traffic, which takes priority in an RIO network.

distributed equipment

Any Ethernet device (Schneider Electric device, PC, servers, or third-party devices) that supports exchange with a PAC or other Ethernet communication service.

DNS

(*domain name server/service*) A service that translates an alpha-numeric domain name into an IP address, the unique identifier of a device on the network.

domain name

An alpha-numeric string that identifies a device on the internet, and which appears as the primary component of a web site's uniform resource locator (URL). For example, the domain name *schneider-electric.com* is the primary component of the URL *www.schneider-electric.com*.

Each domain name is assigned as part of the domain name system, and is associated with an IP address.

Also called a host name.

DRS

(*dual-ring switch*) A ConneXium extended managed switch that has been configured to operate on an Ethernet network. Predefined configuration files are provided by Schneider Electric to downloaded to a DRS to support the special features of the main ring / sub-ring architecture.

DSCP

(*Differentiated Service Code Points*) This 6-bit field is in the header of an IP packet to classify and prioritize traffic.

DST

(*daylight saving time*) DST is also called *summer time* and is a practice consisting of adjusting forward the clock near the start of spring and adjusting it backward near the start of autumn.

DT

(*date and time*) The `DT` type, encoded in BCD in a 64-bit format, contains this information:

- the year encoded in a 16-bit field
- the month encoded in an 8-bit field
- the day encoded in an 8-bit field
- the time encoded in an 8-bit field
- the minutes encoded in an 8-bit field
- the seconds encoded in an 8-bit field

NOTE: The 8 least significant bits are not used.

The `DT` type is entered in this format:

DT#<Year>**-**<Month>**-**<Day>**-**<Hour>**:**<Minutes>**:**<Seconds>

This table shows the upper/lower limits of each field:

Field	Limits	Comment
Year	[1990,2099]	Year
Month	[01,12]	The leading 0 is displayed; it can be omitted during data entry.
Day	[01,31]	For months 01/03/05/07/08/10/12
	[01,30]	For months 04/06/09/11
	[01,29]	For month 02 (leap years)
	[01,28]	For month 02 (non-leap years)
Hour	[00,23]	The leading 0 is displayed; it can be omitted during data entry.
Minute	[00,59]	The leading 0 is displayed; it can be omitted during data entry.
Second	[00,59]	The leading 0 is displayed; it can be omitted during data entry.

DTM

(*device type manager*) A DTM is a device driver running on the host PC. It provides a unified structure for accessing device parameters, configuring and operating the devices, and troubleshooting devices. DTMs can range from a simple graphical user interface (GUI) for setting device parameters to a highly sophisticated application capable of performing complex real-time calculations for diagnosis and maintenance purposes. In the context of a DTM, a device can be a communications module or a remote device on the network.

See FDT.

E

EDS

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

EF

(*elementary function*) This is a block used in a program which performs a predefined logical function.

A function does not have any information on the internal state. Several calls to the same function using the same input parameters will return the same output values. You will find information on the graphic form of the function call in the [*functional block (instance)*]. Unlike a call to a function block, function calls include only an output which is not named and whose name is identical to that of the function. In FBD, each call is indicated by a unique [number] via the graphic block. This number is managed automatically and cannot be modified.

Position and configure these functions in your program in order to execute your application.

You can also develop other functions using the SDKC development kit.

EFB

(*elementary function block*) This is a block used in a program which performs a predefined logical function.

EFBs have states and internal parameters. Even if the inputs are identical, the output values may differ. For example, a counter has an output indicating that the preselection value has been reached. This output is set to 1 when the current value is equal to the preselection value.

EN

`EN` stands for **EN**able; it is an optional block input. When the `EN` input is enabled, an `ENO` output is set automatically.

If `EN` = 0, the block is not enabled; its internal program is not executed, and `ENO` is set to 0.

If `EN` = 1, the block's internal program is run and `ENO` is set to 1. If a runtime error is detected, `ENO` is set to 0.

If the `EN` input is not connected, it is set automatically to 1.

ENO

`ENO` stands for **E**rror **NO**tification; this is the output associated with the optional input `EN`.

If `ENO` is set to 0 (either because `EN` = 0 or if a runtime error is detected):

- The status of the function block outputs remains the same as it was during the previous scanning cycle that executed correctly.
- The output(s) of the function, as well as the procedures, are set to 0.

Ethernet

A 10 Mb/s, 100 Mb/s, or 1 Gb/s, CSMA/CD, frame-based LAN that can run over copper twisted pair or fiber optic cable, or wireless. The IEEE standard 802.3 defines the rules for configuring a wired Ethernet network; the IEEE standard 802.11 defines the rules for configuring a wireless Ethernet network. Common forms include 10BASE-T, 100BASE-TX, and 1000BASE-T, which can utilize category 5e copper twisted pair cables and RJ45 modular connectors.

EtherNet/IP™

A network communication protocol for industrial automation applications that combines the standard internet transmission protocols of TCP/IP and UDP with the application layer common industrial protocol (CIP) to support both high speed data exchange and industrial control. EtherNet/IP employs electronic data sheets (EDS) to classify each network device and its functionality.

explicit messaging

TCP/IP-based messaging for Modbus TCP and EtherNet/IP. It is used for point-to-point, client/server messages that include both data, typically unscheduled information between a client and a server, and routing information. In EtherNet/IP, explicit messaging is considered class 3 type messaging, and can be connection-based or connectionless.

explicit messaging client

(*explicit messaging client class*) The device class defined by the ODVA for EtherNet/IP nodes that only support explicit messaging as a client. HMI and SCADA systems are common examples of this device class.

F

FAST

An event-triggered (FAST) task is an optional, periodic processor task that identifies high priority, multiple scan requests, which is run through its programming software. A FAST task can schedule selected I/O modules to have their logic solved more than once per scan. The FAST task has 2 sections:

- IN: Inputs are copied to the IN section before execution of the FAST task.
- OUT: Outputs are copied to the OUT section after execution of the FAST task.

FBD

(*function block diagram*) A graphical programming language that works like a flowchart. By adding simple logical blocks (`AND`, `OR`, etc.), each function or function block in the program is represented in this graphical format. For each block, the inputs are on the left and the outputs on the right. Block outputs can be linked to inputs of other blocks in order to create complex expressions.

FDR

(*fast device replacement*) A service that uses configuration software to replace an inoperable product.

FDT

(*field device tool*) The technology that harmonizes communication between field devices and the system host.

FTP

(*file transfer protocol*) A protocol that copies a file from one host to another over a TCP/IP-based network, such as the internet. FTP uses a client-server architecture as well as separate control and data connections between the client and server.

full duplex

The ability of 2 networked devices to independently and simultaneously communicate with each other in both directions.

function block diagram

See FBD.

G

gateway

A device that interconnects 2 different networks sometimes with different network protocols. When used to connect networks based on different protocols, a gateway converts a datagram from one protocol stack into the other. When used to connect 2 IP-based networks, a gateway (also called a router) has 2 separate IP addresses, one on each network.

H

harsh environment

Resistance to hydrocarbons, industrial oils, detergents and solder chips. Relative humidity up to 100%, saline atmosphere, significant temperature variations, operating temperature between -10°C and + 70°C, or in mobile installations.

HART

(*highway addressable remote transducer*) A bi-directional communication protocol for sending and receiving digital information across analog wires between a control or monitoring system and smart devices.

HART is the global standard for providing data access between host systems and intelligent field instruments. A host can be any software application from a technician's hand-held device or laptop to a plant's process control, asset management, or other system using any control platform.

high-capacity daisy chain loop

Often referred to as HCDL, a high-capacity daisy chain loop uses dual-ring switches (DRSs) to connect device sub-rings (containing RIO drops or distributed equipment) and/or DIO clouds to the Ethernet RIO network.

HMI

(*human machine interface*) System that allows interaction between a human and a machine.

HTTP

(*hypertext transfer protocol*) A networking protocol for distributed and collaborative information systems. HTTP is the basis of data communication for the web.

I

I/O scanner

An Ethernet service that continuously polls I/O modules to collect data, status, event, and diagnostics information. This process monitors inputs and controls outputs. This service supports both RIO and DIO logic scanning.

IEC 61131-3

International standard: programmable logic controllers

Part 3: programming languages

IGMP

(internet group management protocol) This internet standard for multicasting allows a host to subscribe to a particular multicast group.

IL

(*instruction list*) This language is a series of basic instructions. It is very close to assembly language used to program processors. Each instruction is made up of an instruction code and an operand.

implicit messaging

UDP/IP-based class 1 connected messaging for EtherNet/IP. Implicit messaging maintains an open connection for the scheduled transfer of control data between a producer and consumer. Because an open connection is maintained, each message contains primarily data, without the overhead of object information, plus a connection identifier.

INT

(*INTeger*) (encoded in 16 bits) The upper/lower limits are as follows: -(2 to the power of 15) to (2 to the power of 15) - 1.

Example: `-32768`, `32767`, `2#1111110001001001`, `16#9FA4`.

inter-controller network

An Ethernet-based network that is part of the control network, and provides data exchange between controllers and engineering tools (programming, asset management system (AMS)).

IODDT

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 PAC module. Each application expert module possesses its own IODDTs.

IP address

The 32-bit identifier, consisting of both a network address and a host address assigned to a device connected to a TCP/IP network.

isolated DIO network

An Ethernet-based network containing distributed equipment that does not participate in an RIO network.

L

LD

(*ladder diagram*) A programming language that represents instructions to be executed as graphical diagrams very similar to electrical diagrams (contacts, coils, etc.).

literal value of an integer

A literal value of an integer is used to enter integer values in the decimal system. Values may be preceded by the "+" and "-" signs. Underscore signs (_) separating numbers are not significant.

Example:

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

local rack

An M580 rack containing the CPU and a power supply. A local rack consists of 1 or 2 racks: the main rack and the extended rack, which belongs to the same family as the main rack. The extended rack is optional.

local slave

The functionality offered by Schneider Electric EtherNet/IP communication modules that allows a scanner to take the role of an adapter. The local slave enables the module to publish data via implicit messaging connections. Local slave is typically used in peer-to-peer exchanges between PACs.

M

M580 Ethernet I/O device

An Ethernet device that provides automatic network recovery and deterministic RIO performance. The time it takes to resolve an RIO logic scan can be calculated, and the system can recover quickly from a communication disruption. M580 Ethernet I/O devices include:

- local rack (including a CPU with Ethernet I/O scanner service)
- RIO drop (including an Ethernet RIO adapter module)
- DRS switch with a predefined configuraton

main ring

The main ring of an Ethernet RIO network. The ring contains RIO modules and a local rack (containing a CPU with RIO scanner service) and a power supply module.

MAST

A master (MAST) task is a deterministic processor task that is run through its programming software. The MAST task schedules the RIO module logic to be solved in every I/O scan. The MAST task has 2 sections:

- IN: Inputs are copied to the IN section before execution of the MAST task.
- OUT: Outputs are copied to the OUT section after execution of the MAST task.

MB/TCP

(*Modbus over TCP protocol*) This is a Modbus variant used for communications over TCP/IP networks.

MIB

(*management information base*) A virtual database used for managing the objects in a communications network. See SNMP.

Modbus

Modbus is an application layer messaging protocol. Modbus provides client and server communications between devices connected on different types of buses or networks. Modbus offers many services specified by function codes.

multicast

A special form of broadcast where copies of the packet are delivered to only a specified subset of network destinations. Implicit messaging typically uses multicast format for communications in an EtherNet/IP network.

N

network

There are 2 meanings:

- In a ladder diagram:
  A network is a set of interconnected graphic elements. The scope of a network is local, concerning the organizational unit (section) of the program containing the network.
- With expert communication modules:
  A network is a set of stations that intercommunicate. The term *network* is also used to define a group interconnected graphic elements. This group then makes up part of a program that may comprise a group of networks.

NIM

(*network interface module*) A NIM resides in the first position on an STB island (leftmost on the physical setup). The NIM provides the interface between the I/O modules and the fieldbus master. It is the only module on the island that is fieldbus-dependent — a different NIM is available for each fieldbus.

NTP

(*network time protocol*) Protocol for synchronizing computer system clocks. The protocol uses a jitter buffer to resist the effects of variable latency.

O

O->T

(*originator to target*) See originator and target.

operation network

An Ethernet-based network containing operator tools (SCADA, client PC, printers, batch tools, EMS, etc.). Controllers are connected directly or through routing of the inter-controller network. This network is part of the control network.

originator

In EtherNet/IP, a device is considered the originator when it initiates a CIP connection for implicit or explicit messaging communications or when it initiates a message request for un-connected explicit messaging.

P

PAC

programmable automation controller. The PAC is the brain of an industrial manufacturing process. It automates a process as opposed to relay control systems. PACs are computers suited to survive the harsh conditions of the industrial environment.

port 502

Port 502 of the TCP/IP stack is the well-known port that is reserved for Modbus communications.

port mirroring

In this mode, data traffic that is related to the source port on a network switch is copied to another destination port. This allows a connected management tool to monitor and analyze the traffic.

Q

QoS

(*quality of service*) The practice of assigning different priorities to traffic types for the purpose of regulating data flow on the network. In an industrial network, QoS is used to provide a predictable level of network performance.

R

rack optimized connection

Data from multiple I/O modules are consolidated in a single data packet to be presented to the scanner in an implicit message in an EtherNet/IP network.

RIO drop

One of the 3 types of RIO modules in an Ethernet RIO network. A RIO drop is an M580 rack of I/O modules that are connected to an Ethernet RIO network and managed by an Ethernet remote adapter module. A drop can be a single rack or a main rack with an extended rack.

RIO network

An Ethernet-based network that contains 3 types of RIO devices: a local rack, an RIO drop, and a ConneXium extended dual-ring switch (DRS). Distributed equipment may also participate in an RIO network via connection to DRSs.

RPI

(requested packet interval) The time period between cyclic data transmissions requested by the scanner. EtherNet/IP devices publish data at the rate specified by the RPI assigned to them by the scanner, and they receive message requests from the scanner at each RPI.

RSTP

(*rapid spanning tree protocol*) Allows a network design to include spare (redundant) links to provide automatic backup paths if an active link stops working, without the need for loops or manual enabling/disabling of backup links.

S

scanner

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

scanner class device

A scanner class device is defined by the ODVA as an EtherNet/IP node capable of originating exchanges of I/O with other nodes in the network.

service port

A dedicated Ethernet port on the M580 RIO modules. The port may support the following major functions (depending on the module type):

- port mirroring: for diagnostic use
- access: for connecting HMI/Unity Pro/ConneXview to the PAC
- extended: to extend the device network to another subnet
- disabled: disables the port, no traffic is forwarded in this mode

SFC

(*sequential function chart*) Used to graphically represent in a structured manner the operation of a sequential PAC. This graphical description of the PAC's sequential behavior and of the various resulting situations is created using simple graphic symbols.

simple daisy chain loop

Often referred to as SDCL, a simple daisy chain loop contains RIO modules only (no distributed equipment). This topology consists of a local rack (containing a CPU with RIO scanner service), and one or more RIO drops (each drop containing an RIO adapter module).

SMTP

(*simple mail transfer protocol*) An email notification service that allows controller-based projects to report alarms or events. The controller monitors the system and can automatically create an email message alert with data, alarms, and/or events. Mail recipients can be either local or remote.

SNMP

(*simple network management protocol*) Protocol used in network management systems to monitor network-attached devices. The protocol is part of the internet protocol suite (IP) as defined by the internet engineering task force (IETF), which consists of network management guidelines, including an application layer protocol, a database schema, and a set of data objects.

SNTP

(*simple network time protocol*) See NTP.

SOE

(*sequence of events*) The process of determining the order of events in an industrial system and correlating those events to a real-time clock.

ST

(*structured text*) The structured literal language is a developed language similar to computer programming languages. It can be used to organize a series of instructions.

sub-ring

An Ethernet-based network with a loop attached to the main ring, via a dual-ring switch (DRS) on the main ring. This network contains RIO or distributed equipment.

subnet mask

The 32-bit value used to hide (or mask) the network portion of the IP address and thereby reveal the host address of a device on a network using the IP protocol.

switch

A multi-port device used to segment the network and limit the likelihood of collisions. Packets are filtered or forwarded based upon their source and destination addresses. Switches are capable of full-duplex operation and provide full network bandwidth to each port. A switch can have different input/output speeds (for example, 10, 100 or 1000Mbps). Switches are considered OSI layer 2 (data link layer) devices.

T

T->O

(*target to originator*) See target and originator.

target

In EtherNet/IP, a device is considered the target when it is the recipient of a connection request for implicit or explicit messaging communications, or when it is the recipient of a message request for un-connected explicit messaging.

TCP

(*transmission control protocol*) A key protocol of the internet protocol suite that supports connection-oriented communications, by establishing the connection necessary to transmit an ordered sequence of data over the same communication path.

TCP/IP

Also known as *internet protocol suite*, TCP/IP is a collection of protocols used to conduct transactions on a network. The suite takes its name from 2 commonly used protocols: transmission control protocol and internet protocol. TCP/IP is a connection-oriented protocol that is used by Modbus TCP and EtherNet/IP for explicit messaging.

TFTP

(*trivial file transfer protocol*) A simplified version of *file transfer protocol* (FTP), TFTP uses a client-server architecture to make connections between 2 devices. From a TFTP client, individual files can be uploaded to or downloaded from the server, using the user datagram protocol (UDP) for transporting data.

TIME_OF_DAY

See `TOD`.

TOD

(*time of day*) The `TOD` type, encoded in BCD in a 32-bit format, contains this information:
- the hour encoded in an 8-bit field
- the minutes encoded in an 8-bit field
- the seconds encoded in an 8-bit field

NOTE: The 8 least significant bits are not used.

The TOD type is entered in this format: xxxxxxxx: **`TOD#`**`<Hour>`:`<Minutes>`:`<Seconds>`

This table shows the upper/lower limits of each field:

Field	Limits	Comment
Hour	[00,23]	The leading 0 is displayed; it can be omitted during data entry.
Minute	[00,59]	The leading 0 is displayed; it can be omitted during data entry.
Second	[00,59]	The leading 0 is displayed; it can be omitted during data entry.

Example: `TOD#23:59:45`.

TR

(*transparent ready*) Web-enabled power distribution equipment, including medium- and low-voltage switch gear, switchboards, panel boards, motor control centers, and unit substations. Transparent Ready equipment allows you to access metering and equipment status from any PC on the network, using a standard web browser.

trap

A trap is an event directed by an SNMP agent that indicates one of these events:
- A change has occurred in the status of an agent.
- An unauthorized SNMP manager device has attempted to get data from (or change data on) an SNMP agent.

U

UDP

(*user datagram protocol*) A transport layer protocol that supports connectionless communications. Applications running on networked nodes can use UDP to send datagrams to one another. Unlike TCP, UDP does not include preliminary communication to establish data paths or provide data ordering and checking. However, by avoiding the overhead required to provide these features, UDP is faster than TCP. UDP may be the preferred protocol for time-sensitive applications, where dropped datagrams are preferable to delayed datagrams. UDP is the primary transport for implicit messaging in EtherNet/IP.

UTC

(*coordinated universal time*) Primary time standard used to regulate clocks and time worldwide (close to former GMT time standard).

V

variable

Memory entity of type `BOOL`, `WORD`, `DWORD`, etc., whose contents can be modified by the program currently running.

VLAN

(*virtual local area network*) A local area network (LAN) that extends beyond a single LAN to a group of LAN segments. A VLAN is a logical entity that is created and configured uniquely using applicable software.

Index

H

I

L

M

N

O

P

T

U

W