User Manual
GFK-2222AU
Jul 2022

PACSystems™ RX3i and RSTi-EP CPU

REFERENCE MANUAL

EMERSON

Warnings and Caution Notes as Used in this Publication

⚠ WARNING

Warning notices are used in this publication to emphasize that hazardous voltages, currents, temperatures, or other conditions that could cause personal injury exist in this equipment or may be associated with its use.

In situations where inattention could cause either personal injury or damage to equipment, a Warning notice is used.

⚠ CAUTION

Caution notices are used where equipment might be damaged if care is not taken.

Note: *Notes merely call attention to information that is especially significant to understanding and operating the equipment.*

These instructions do not purport to cover all details or variations in equipment, nor to provide for every possible contingency to be met during installation, operation, and maintenance. The information is supplied for informational purposes only, and Emerson makes no warranty as to the accuracy of the information included herein. Changes, modifications, and/or improvements to equipment and specifications are made periodically and these changes may or may not be reflected herein. It is understood that Emerson may make changes, modifications, or improvements to the equipment referenced herein or to the document itself at any time. This document is intended for trained personnel familiar with the Emerson products referenced herein.

Emerson may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not provide any license whatsoever to any of these patents.

Emerson provides the following document and the information included therein as-is and without warranty of any kind, expressed or implied, including but not limited to any implied statutory warranty of merchantability or fitness for particular purpose.

Table of Figures viii

Section 1 Introduction 1

1.1 Revisions in this Manual 2
1.2 PACSystems Control System Overview 7
1.2.1 Programming and Configuration 7
1.2.2 Process Systems 7
1.2.3 PACSystems CPU Models 8
1.3 RX3i Overview 9
1.4 RSTi-EP Overview 12
1.5 Migrating Series 90 Applications to PACSystems 14
1.6 Documentation 14
1.6.1 PACSystems Manuals 14
1.6.2 PACEdge Manuals 14
1.6.3 RX3i Manuals 15
1.6.4 RSTi-EP Manuals 15
1.6.5 Series 90 Manuals 15
1.6.6 Distributed I/O Systems Manuals 15

Section 2 CPU Features & Specifications 16

2.1 Common CPU Features 17
2.1.1 Features are shared by all PACSystems CPU models 17
2.1.2 Features Shared by Certain PACSystems CPU Models 18
2.1.3 Firmware Storage in Flash Memory 19
2.1.4 Operation, Protection, and Module Status 19
2.1.5 Ethernet Global Data 19
2.1.6 Embedded PROFINET Controller 20
2.1.7 OPC UA 21
2.1.8 Removable Data Storage Devices (RDSDs) 22
2.1.9 Uploading a Project from the CPU to the RDSD 23
2.1.10 Downloading a Project from the RDSD to an RX3i CPU 25
2.1.11 CPU Over-Temperature Monitoring and Behavior 30
2.2 RX3i CPU Features and Specifications 31
2.2.1 CPE400 and CPL410 37
2.2.2 CPE330 53

2.2.3 CPE302, CPE305[51] and CPE310 63
2.2.4 CPU315 and CPU320/CRU320 76
2.3 RSTi-EP CPU Features and Specifications 80
2.3.1 CPE100/CPE115 85
2.3.2 EPXCPE205, EPXCPE210, EPXCPE215, EPXCPE220, and EPXCPE240 93

Section 3 CPU Configuration 109

3.1 Configuring the CPU 110
3.2 Configuration Parameters 111
3.2.1 Settings Parameters 111
3.2.2 Modbus TCP Address Map 117
3.2.3 SNTP 118
3.2.4 Time 119
3.2.5 Scan Parameters 120
3.2.6 Memory Parameters 123
3.2.7 Fault Parameters 126
3.2.8 Redundancy Parameters (Redundancy CPUs Only) 128
3.2.9 Transfer List 128
3.2.10 COM1 and COM2 Parameters 129
3.2.11 Scan Sets Parameters 134
3.2.12 Power Consumption Parameters 134
3.2.13 Access Control 134
3.2.14 OPC UA Parameters 136
3.3 Storing (Downloading) Hardware Configuration 136
3.4 Configuring the Embedded Ethernet Interface 137
3.4.1 Establishing Initial Ethernet Communications 138
3.4.2 Setting a Temporary IP Address 139

Section 4 CPU Operation 141

4.1 CPU Sweep 143
4.1.1 Parts of the CPU Sweep 144
4.1.2 CPU Sweep Modes 148
4.2 Program Scheduling Modes 151
4.3 Window Modes 151
4.4 Data Coherency in Communications Windows 152
4.5 Run/Stop Operations 152
4.5.1 CPU STOP Modes 153

4.5.2 STOP-to-RUN Mode Transition 156
4.5.3 RUN Mode Store 156
4.5.4 RUN Mode Store to User Flash 164
4.6 Flash Memory Operation 165
4.6.1 RUN/STOP Switch Operation 166
4.7 Logic/Configuration Source and CPU Operating Mode at Power-Up 166
4.7.1 CPU Mode when Memory Not Preserved/Power-up Source is Flash 168
4.7.2 CPU Mode when Memory Preserved 169
4.8 Clocks and Timers 170
4.8.1 Elapsed Time Clock 170
4.8.2 Time-of-Day Clock 170
4.8.3 Watchdog Timer 171
4.9 System Security 173
4.9.1 Passwords and Privilege Levels - Legacy Mode 174
4.9.2 OEM Protection – Legacy Mode 179
4.9.3 Enhanced Security for Passwords and OEM Protection 180
4.9.4 Legacy/Enhanced Security Comparison 182
4.10 PACSystems I/O System 183
4.10.1 I/O Configuration 184
4.10.2 I/O System Diagnostic Data Collection 187
4.10.3 Power-Up and Power-Down Sequences 188

Section 5 Communications 195

5.1 Ethernet Communications 196
5.1.1 Embedded Ethernet Interfaces 196
5.1.2 Ethernet Interface Modules 201
5.2 Serial Communications 201
5.2.1 Serial Port Communications Capabilities 201
5.2.2 Configurable STOP Mode Protocols 202
5.2.3 Serial Port PIN Assngments 203
5.2.4 Serial Port Electrical Isolation 206
5.2.5 Serial Cable Lengths and Shielding 207
5.2.6 Maximum Cable Lengths for RSTi-EP CPE205/210/215/220/240 207
5.2.7 Serial Port Baud Rates 208
5.2.8 Programmable Coprocessor Module (PCM) 210

Section 6 Serial I/O, SNP & RTU Protocols 211

6.1 Configuring Serial Ports Using COMMREQ Function 65520 212
6.1.1 COMMREQ Function Example 212
6.1.2 Timing 213
6.1.3 Sending Another COMMREQ to the Same Port 213
6.1.4 Invalid Port Configuration Combinations 213
6.1.5 COMMREQ Command Block Parameter Values 214
6.1.6 Example COMMREQ Data Block for Configuring Message mode Protocol 215
6.1.7 Example COMMREQ Command Blocks for Serial Port Setup function 215
6.1.8 Example COMMREQ Command Block for Configuring SNP Protocol 216
6.1.9 Example COMMREQ Data Block for Configuring RTU Protocol 217
6.1.10 Example COMMREQ Data Block for Configuring Serial I/O Protocol 218
6.2 Serial I/O Protocol 219
6.2.1 Calling Serial I/O COMMREQs from the CPU Sweep 219
6.2.2 Compatibility 219
6.2.3 Status Word for Serial I/O COMMREQs 220
6.2.4 Serial I/O COMMREQ Commands 222
6.2.5 Overlapping COMMREQs 222
6.2.6 Initialize Port Function (4300) 223
6.2.7 Set Up Input Buffer Function (4301) 224
6.2.8 Flush Input Buffer Function (4302) 225
6.2.9 Read Port Status Function (4303) 225
6.2.10 Write Port Control Function (4304) 227
6.2.11 Cancel COMMREQ Function (4399) 228
6.2.12 Autodial Function (4400) 230
6.2.13 Write Bytes Function (4401) 231
6.2.14 Read Bytes Function (4402) 233
6.2.15 Read String Function (4403) 234
6.3 RTU Slave Protocol 236
6.3.1 Message Format 237
6.3.2 Cyclic Redundancy Check (CRC) 241
6.3.3 RTU Message Descriptions 245
6.3.4 RTU Scratch Pad 262
6.3.5 Communication Errors 262
6.3.6 RTU Slave/SNP Slave Operation with Programmer Attached 265

6.4 SNP Slave Protocol 266
6.4.1 Permanent Datagrams....266
6.4.2 Communication Requests (COMMREQs) for SNP 267
Appendix A: Performance Data 268
A.1: Boolean Execution Measurements (ms per 1000 Boolean executions)....268
A.2: Instruction Timing....269
A.3: RX3i & RSTi-EP Instruction Times 271
A.4: Overhead Sweep Impact Times 272
A.5: Base Sweep Times 272
A.6: Base Sweep vs. Full Sweep Phases....274
A.7: What the Sweep Impact Tables Contain....275
A.8: Programmer Sweep Impact Times....276
A.9: I/O Scan and I/O Fault Sweep Impact 277
A.10: Sweep Impact of Local I/O Modules 277
A.11: RX3i I/O Module Types 278
A.12: RX3i I/O Module Sweep Impact Times 279
A.13: Worksheet A: I/O Module Sweep Time....280
A.14: Sweep Impact of Genius I/O and GBCs 281
A.15: Sweep Impact Time of Genius I/O and GBCs 281
A.16: Worksheet B: Genius I/O Sweep Time 282
A.17: Ethernet Global Data Sweep Impact 283
A.18: Exchange Overhead....283
A.19: Data Transfer Time 284
A.20: Worksheet C: Ethernet Global Data Sweep Time....286
A.21: EGD Sweep Impact for RX3i CPE330 and CPE400/CPL410 286
A.22: EGD Sweep Impact for RX3i CPE302/CPE305/CPE310 and RSTi-EP CPE100/CPE115 Embedded Ethernet Interface 286
A.23: Example Calculation for EGD Utilization on RX3i CPE302/CPE305/CPE310 and RSTi-EP CPE100/CPE115....287
A.24: Normal Sweep – EGD on RX3i CPE302/CPE305/CPE310 and RSTi-EP CPE100/CPE115 Embedded Ethernet Interface 288
A.25: Constant Sweep - EGD on RX3i CPE302/CPE305/CPE310 and RSTi-EP CPE100/CPE115 Embedded Ethernet Interface 290
A.26: Sweep Impact of Intelligent Option Modules 290
A.27: Fixed Sweep Impact Times of RX3i Intelligent Option Modules .291

A.28: PROFINET Controller (PNC001) and PROFINET I/O Sweep Impact 292
A.29: DSM314 Sweep Impact 294
A.30: I/O Interrupt Performance and Sweep Impact 294
A.31: I/O Interrupt Block Performance and Sweep Impact Times 295
A.32: Dropped Interrupts 295
A.33: Worksheet D: Programmer, IOM, I/O Interrupt Sweep Time 296
A.34: Timed Interrupt Performance 297
Appendix B: User Memory Allocation 298
B.1: Items that Count Against User Memory 299
B.2: User Program Memory Usage 300
B.3: %L and %P Program Memory 300
B.4: Program Logic and Overhead 300

General Contact Information 301

Technical Support 301

Table of Figures

Figure 1: Configuring an Embedded PROFINET Controller 20
Figure 2: Example ipConfig.xml 28
Figure 2: CPE400 and CPL410 Front View and Features 37
Figure 3: Underside Ports & Connectors 45
Figure 4: CPE400 Micro-SD & USB Connectors 46
Figure 5:Location of RTC Battery on CPE400/CPL410 47
Figure 6: LAN3 Interconnects for Hot Standby Redundancy 50
Figure 7: Display Port Connector 52
Figure 8: Front Display Port 53
Figure 9: CPE330 Run/Stop Switch and RDSD Switches 55
Figure 10: Location and Orientation of Real-Time Clock Battery in CPE330 59
Figure 11: IC695CPE302/CPE305-Axxx Front View 63
Figure 12: IC695CPE310 Front View 63
Figure 13: CPE302/305-Bxxx 64
Figure 13: External Features of CPE302/CPE305-Axxx 66
Figure 14: External Features of CPE310 66
Figure 14: External Features of CPE302/CPE305-Bxxx 66
Figure 15: Accessing Real-Time Clock Battery (CPE302-Axxx, CPE305-Axxx, and CPE310 70
Figure 18: Sample Tool for Battery Removal 72
Figure 19: Remove Screws Before Opening Cover 72
Figure 20: Use Tweezers to Remove Battery. 73
Figure 17: IC695CPU320 Front View 76
Figure 18: CPU310 Front View 78
Figure 19: CPE100, Front, Top, and Bottom Views and Features 85
Figure 20: CPE100/CPE115 Membrane Pushbutton and Module Status LEDs 87
Figure 21: State Diagram for CPE100/CPE115 Run/Stop Operation 88
Figure 22: Typical Multi-Tier LAN Application (Star/Bus Topology) 90
Figure 23: Typical Multi-Tier LAN Application (Ring Topology) 91
Figure 25: EPXCPE205 94
Figure 26: EPXCPE210/215/220/240 95
Figure 2: 3-Position Switch RUN EN/RUN DIS/STOP 97
Figure 27: MicroSD 102
Figure 24: Embedded Ethernet Interface Configuration 137
Figure 25: Set Temporary IP Address 140
Figure 26: Major Phases of a Typical CPU Sweep 144
Figure 27: Typical Sweeps in Normal Sweep Mode 148
Figure 28: Typical Sweeps in Constant Sweep Mode 149
Figure 29: Typical Sweeps in Constant Window Mode 150
Figure 30: CPU Sweep in Stop-I/O Disabled and Stop-I/O Enabled Modes 153
Figure 36: Connect Timeout and Request Timeout 164
Figure 31: CPE330 Overlapping Local IP Subnet Example 197
Figure 32: Expected Response Path 198
Figure 33: Actual Response Path 199

Figure 34: COM1 Port CPE400/CPL410 204
Figure 35: COMMREQ Example 212
Figure 36: RTU Message Transactions 237
Figure 37: RTU Read Output Table Example 240
Figure 38: CRC Register Operation 242
Figure 39: RTU Read Output Table Message Format 246
Figure 40: RTU Read Input Table Message Format 247
Figure 41: RTU Read Registers Message Format 248
Figure 42: RTU Read Analog Inputs Message Format 249
Figure 43: RTU Force Single Output Message Format 250
Figure 44: RTU Preset Single Register Message Format 251
Figure 45: RTU Read Exception Status Message Format 252
Figure 46: RTU Loopback/Maintenance Message Format 252
Figure 47: RTU Force Multiple Outputs Message Format 254
Figure 48: RTU Preset Multiple Registers Message Format 255
Figure 49: RTU Report Device Type Message Format 256
Figure 50: RTU Read Scratch Pad Memory Message Format 261
Figure 51: Interrupt Execution Considerations 295

Section 1 Introduction

This manual contains general information about PACSystems CPU operation and product features.

Section 1 provides a **general introduction** to the PACSystems family of products, including new features, product overviews, and a list of related documentation.

CPU Features & Specifications are provided in Section 2.

Installation procedures for the different platforms are described in their respective manuals as given below:

1. PACSystems RX7i Installation Manual, GFK-2223.
2. *PACSystems RX3i System Manual,* GFK-2314.
3. RSTi-EP User Manual, GFK-2958.

CPU Programming is covered in *PACSystems RX3i and RSTi-EP CPU Programmer's Reference Manual,* GFK-2950. It provides an overview of program structure and describes the various languages which may be used, their syntax and operation, and provides examples.

CPU Configuration is described in Section 3. Configuration using the proprietary PAC Machine Edition™ (PME) programming and configuration software package determines the characteristics of CPU, System, and module operation. It also establishes the program references used by each module in the system. For details on the configuration of RX3i Ethernet Interface modules, refer to *PACSystems RX3i and RSTi-EP TCP/IP Ethernet Communications User Manual,* GFK-2224.

CPU Operation is described in Section 4.

Ethernet Communications and Serial Communications are described in Section 5.

Serial I/O, SNP & RTU Protocols are described in Section 6.

Performance Data, including Instruction Timing, is provided in Appendix A.

User Memory Allocation is described in Appendix B.

1.1 Revisions in this Manual

Note: A given feature may not be implemented on all PACSystems CPUs. To determine whether a feature is available on a given CPU model and firmware version, please refer to the *Important Product Information* (IPI) document provided for the CPU version that you are using.

Rev	Date	Description
AU	Jul-2022	▪ Updates to sections 4.10.1 and 4.10.3 to address the new Legacy Client/Server Protocol Memory Access feature and other Password and Privilege Level enhancements. ▪ Updates to support the release of CPE302/305-Bxxx. ▪ Added note about RMS to flash performance in section 3.2.13 ▪ Addition of RSTi-EP EPXCPE2XX controllers. ▪ Updates to Access Control List ▪ Updates to Flash Memory and Override Tables ▪ Correction to the maximum number of supported simplex & redundant PROFINET devices.
AT	Sep-2021	▪ Updates to CPE400/CPL400 in support of PACEdge Software updates.
AQ	May-2021	▪ Updates regarding the number of supported PROFINET devices for the CPL410.
AP	Nov-2020	▪ Revised instructions for the STOP-Halt auto-recovery feature for the latest firmware release 10.05 in the section entitled ▪ STOP-Halt Mode.
AN	Jun-2020	▪ Updated discussion of STOP-Halt mode recovery to include new auto-recovery functionality in sections 4.5.1.3 and 4.8.3.
AM	May-2020	▪ Updated CPE/CPL Features and Specifications in Section 2.2 ▪ Updated Tables 4-3 and 4-4 in Section *Privilege Level Request from PAC Machine Edition*
AL	Feb-2020	▪ Addition of DNP3 to CPE400
AK	Oct-2019	▪ Updated to reflect updated capabilities of FW 9.90 ▪ Following Emerson's acquisition of this product, changes have been made to apply appropriate branding and registration of the product with required certification agencies.

Rev	Date	Description
AJ	Jun-2019	▪ Updated Appendix A on Boolean Execution for clarity.
AH	Apr-2019	▪ CPE400/CPL410 added the ability to recover from STOP-Halt mode. ▪ CPE330 supports HSB CPU redundancy with Single RMX. This support includes CRU320 compatible mode. Dual RMX configuration is no longer a requirement.1
AG	Nov-2018	▪ CPE330/CPE400/CPL410 increased block count from 512 to 768 including _Main
AF	Aug-2018	▪ SoE and other minor changes for CPE330 ▪ European DST correction
AE	Jul-2018	▪ Addition of IC695CPL410, CPU . ▪ Addition of IC695PNS101.
AD	Apr-2018	▪ Added CPE115 module
AC	Feb-2018	▪ Updated throughout for addition of CPE302 (initial firmware version 9.40). ▪ CPE400 Serial IO feature added ▪ New Authorized Firmware Update feature noted (part of RX3i firmware version 9.40).
AB	Oct-2017	▪ Added Redundancy features for CPE400 ▪ Updated Field Agent information for CPE400. ▪ Updated Section 2.2 for new features of CPE400. ▪ Added Section 0 on CPU Over-Temperature behavior.
AA	Sep-2017	▪ Addition of support for Media Redundancy Protocol (MRP) on CPE100.
Z	May-2017	▪ Addition of RSTi-EP EPSCPE100 (new product) and updated other relevant sections. ▪ Addition of Simple Network Time Protocol (SNTP), Coordinated Universal Time (UTC), and Daylight Savings Time (DST) features forCPE305, CPE310, CPE330, and CPE400.

[1] *Requires PME 9.50 SIM 14 or later and CPE330 firmware 9.75 or later.*

Rev	Date	Description
Y	Dec-2016	▪ Added section on CPE400 and incorporated into CPU comparison table (section 2.2). This section also introduces Field Agent and documents how to set up Embedded Field Agent for the CPE400 (section ▪ Added section 2.1.6, Embedded PROFINET Controller. ▪ Update of Energy Pack to include ACC403 and compatibility matrix. ▪ Added compatibility mode information for CPE330 with CPU320 & CRU320
X	Feb-2016	▪ Corrected Ethernet Indicators CPE305 & CPE310 table.
W	Aug-2015	▪ Addition of support for Ethernet Global Data (Class 1) on CPE330
V	Jun-2015	▪ Addition of RX3i CPE330 (new product) and related Ethernet considerations. ▪ Update of Energy Pack to include ACC402 and compatibility matrix. ▪ Addition of HART® Pass-Through feature (see page 10). ▪ Addition of CPU Comparison Charts ▪ Update of Communications Section (Section 5). Removed original Sections 5-11 (Sections dealing with CPU programming) and Section 14 (Diagnostics). These are now in PACSystems RX3i CPU Programmer's Reference Manual, GFK-2950 (Sections 2-8 and Section 9 respectively).
U	Nov-2014	▪ New section for EGD Sweep Impact for RX3i CPE302/CPE305/CPE310 and RSTi-EP CPE100/CPE115 Embedded Ethernet Interface.

Rev	Date	Description
T	Oct-2014	▪ Support for OPC UA using embedded Ethernet port in CPE305/CPE310 with CPU firmware 8.20. ▪ Support for Ethernet Global Data (EGD Class 1) using embedded Ethernet port in CPE305/CPE310 with CPU firmware 8.30 Sweep impact of EGD on Embedded Ethernet interface. Direct replacement for S90-30 IC693CPU374. ▪ New communications capabilities are provided by: o IC695PNS001 – PROFINET Scanner Module o IC695GCG001 – Genius Communications Gateway via PROFINET o IC695EDS001 – Ethernet-based DNP3 Outstation
S	July-2013	▪ Support for Modbus/TCP Server, SRTP channels, and Modbus/TCP client channels on RX3i CPE305/CPE310 embedded Ethernet interface – Section 2 & Section 5 ▪ Support for Access Control List – Section 3 ▪ Modbus TCP/IP mapping for CPE305/CPE310 – Section 30 ▪ Enhanced Security Passwords and OEM Protection – Section 4 ▪ Serial I/O protocol enhancements (Data Set Ready, Ring Indicator, and Data Carrier Detect) – Section 6 ▪ Diagnostics for PROFINET alarms and PROFINET network faults, including #PNIO_ALARM, SA0030 – refer to PACSystems RX7i and RX3i CPU Programmer's Reference Manual, GFK-2950 Section 3 & Section 9. ▪ Instruction executions times measured for RX3i CPU320/CRU320 – Appendix A: ▪ Sweep impact times for new modules: IC694MDL758, IC694APU300-CA and later, IC695PNS001, IC694ALG442, IC694ALG220, IC694MDL645, and IC694MDL740–Appendix A:

Rev	Date	Description
earlier		▪ Added instructions for replacing the RX3i CPE305/CPE310 real-time clock battery: Section 2. ▪ Corrected definitions of reverse acting and direct-acting modes for PID functions: refer to PACSystems RX7i and RX3i CPU Programmer's Reference Manual, GFK-2950 Section 7. ▪ Expanded data for Boolean execution measurements – Appendix A: ▪ Re-instated instruction times for RX7i CPE030/CRE030/CPE040 release 6.0 as published in version Q of the manual (unintentionally omitted from version R) – Appendix A: ▪ Compatibility information for volatile memory backup batteries has been consolidated in the PACSystems Battery and Energy Pack Manual, GFK-2741 – throughout

1.2 PACSystems Control System Overview

The PACSystems controller environment combines performance, productivity, openness, and flexibility. The PACSystems control system integrates advanced technology with existing systems. The result is seamless migration that protects your investment in I/O and application development.

1.2.1 Programming and Configuration

PAC Machine Edition programming software provides a universal engineering development environment for all programming, configuration, and diagnostics of PACSystems. A PACSystems CPU is programmed and configured using the programming software to perform process and discrete automation for various applications. The CPU communicates with I/O and smart option modules through a rack-mounted backplane. It communicates with the programmer and/or HMI devices via the Ethernet ports or the serial ports COM1 and COM2 using Serial I/O, or Modbus RTU slave protocols.

1.2.2 Process Systems

PACSystems CPUs with firmware version 5.0 and later support PAC Process Systems (PPS). PPS is a complete, tightly integrated, seamless process control system using PACSystems, HMI/SCADA, and PAC Production Management Software to provide control, optimization, and performance management to manage and monitor batch or continuous manufacturing. It delivers the tools required to design, implement, document, and maintain an automated process. For information about purchasing PPS software, refer to the Support website.

1.2.3 PACSystems CPU Models

Family	Catalog #	Description
RSTi-EP Standalone CPUs	EPSCPE100	1 MB user memory
	EPSCPE115	1.5 MB user memory
RX3i Standalone CPUs with embedded Ethernet/PROFINET Interface	IC695CPE400	64 MB user memory with PACEdge + Connext
	IC695CPL410	64 MB user memory with PACEdge + WebHMI
RX3i CPUs with embedded Ethernet/PROFINET Interface	IC695CPE330	64 MB user memory
RX3i CPUs with embedded Ethernet Interface[2]	IC695CPE302	2 MB user memory[3]
	IC695CPE305	6 MB user memory[4]
	IC695CPE310	10 MB user memory
RX3i CPUs	IC695CPU310	10 MB user memory
	IC695CPU315	20 MB user memory
	IC695CPU320	64 MB user memory
	IC695NIU001+ versions –AAAA & later	For information, refer to the *PACSystems RX3i Ethernet Network Interface Unit User's Manual*, GFK-2439
	IC695NIU001	For information, refer to the *PACSystems RX3i Ethernet Network Interface Unit User's Manual*, GFK-2439
RX3i Redundancy CPU	IC695CRU320	64 MB user memory

[2] *The RX3i CPE302/CPE305/CPE310 embedded Ethernet interface provides a maximum of two programmer connections. It does not support the full set of Ethernet interface features described in this manual. For a summary of RX3i embedded Ethernet interface features, refer to PACSystems RX3i TCP/IP Ethernet Communications User Manual, GFK-2224K or later.*

[3] *CPE302-Axxx and -Bxxx both have 2 MB of user memory.*

[4] *CPE305-Axxx has 5 MB of user memory and the -Bxxx has 6 MB of user memory.*

1.3 RX3i Overview

The RX3i control system hardware consists of an RX3i universal backplane and up to seven Series 90-30 expansion or remote racks. The CPU can be in any slot in the universal backplane except the last slot, which is reserved for the serial bus transmitter, IC695LRE001.

The RX3i supports user-defined Function Blocks (LD logic only) and Structured Text programming.

The RX3i universal backplane uses a dual bus that provides both:

- High-speed PCI for fast throughput of new advanced I/O.
- Serial backplane for easy migration of existing Series 90-30 I/O

The RX3i universal backplane and Series 90-30 expansion/remote racks support the Series 90-30 Genius Bus Controller and Motion Control modules, and most Series 90-30/RX3i discrete and analog I/O with catalog prefixes IC693 and IC694. RX3i modules with catalog prefixes IC695, including the Ethernet and other communications modules can only be installed in the universal backplane. See the *PACSystems RX3i System Manual,* GFK-2314 for a list of supported modules.

RX3i supports hot standby (HSB) CPU redundancy, which allows a critical application or process to continue operating if a failure occurs in any single component. A CPU redundancy system consists of an active unit that actively controls the process and a backup unit that is synchronized with the active unit and can take over the process if it becomes necessary. Each unit must have a redundancy CPU (See section 2.2 for CPUs that support redundancy). For the backplane-based CPU redundancy, the redundancy communication path is provided by IC695RMX128 Redundancy Memory Xchange (RMX) modules set up as redundancy links. For the Ethernet-based CPU redundancy, the redundancy communication path is provided by Ethernet connections between the redundant CPUs. For details on the operation of PACSystems redundancy systems, refer to the *PACSystems Hot Standby CPU Redundancy User Manual,* GFK-2308.

RX3i communications features include:

- Open communications support includes Ethernet, PROFIBUS, PROFINET, Modbus TCP, Ethernet Global Data (EGD), DNP3, and serial protocols.
- On the CPE400 and CPL410, the Ethernet Port and Serial Port located on the underside and one of the USB ports are controlled by the PACEdge; all front-panel ports are controlled by the RX3i PLC, except for the one USB port mentioned above.
- The CPE302[13], CPE305, CPE310, CPE330, CPE400, and CPL410 CPUs provide an embedded Ethernet interface that is used to connect to the programmer (PAC Machine Edition).
- Effective with RX3i CPE310/CPE305 firmware version 7.30, or CPE330 firmware version 8.50, the embedded Ethernet port on the CPU provides support for Service Request Transfer Protocol (SRTP) channels and Modbus TCP. This feature is available on all firmware versions of CPE400 and CPL410.
- Effective with CPE310/CPE305 firmware version 8.20, or CPE330 firmware version 8.45, the CPE embedded Ethernet port supports OPC UA Server. This feature is available on all firmware versions of CPE400 and CPL410. Refer to *PACSystems RX3i TCP/IP Ethernet Communications User Manual,* GFK-2224 version M or higher (Section 10).
- Effective with RX3i firmware version 8.30[5], the CPE310/CPE305 CPUs also support Ethernet Global Data (EGD). Before that firmware version, EGD was only available in the RX3i via the

[5] *PAC Machine Edition Release 8.50 SIM 7 is required for EGD Class 1 on Embedded Ethernet interface of CPE305/CPE310.*

RX3i Ethernet Interface Module (IC695ETM001). With this upgrade, these CPUs are positioned as a direct replacement for S90-30 IC693CPU374.

- Effective with RX3i firmware version 8.60[6], the CPE330 supports Ethernet Global Data (EGD) Class 1. This feature is available on all firmware versions of CPE400[7], CPL410[8], CPE100, CPE115, and CPE302[13].
- The rack-based IC695ETM001 Ethernet Interface has dual RJ45 ports connected through an auto-sensing switch. This eliminates the need for rack-to-rack switches or hubs. The ETM001 supports upload, download, and online monitoring, and provides 32 SRTP channels with a maximum of 48 simultaneous SRTP server connections. It also supports Modbus TCP. For details on Ethernet Interface capabilities, refer to *PACSystems RX3i TCP/IP Ethernet Communications User Manual,* GFK-2224.
- PROFIBUS communications via the PROFIBUS Master module, IC695PBM300. For details, refer to the *PACSystems RX3i PROFIBUS Modules User's Manual,* GFK-2301.
- PROFINET communications via any supported PROFINET Controller and any supported PROFINET Scanner.
- Supported PROFINET Controllers include the embedded PROFINET Controller function offered by IC695CPL410, IC695CPE400, IC695CPE330, and the rack-mounted PROFINET Controller module IC695PNC001.
- Supported PROFINET Scanners include the RX3i PROFINET Scanner modules IC695PNS001[9], IC695PNS101, the RX3i IC695CEP001, and the VersaMax PROFINET Scanner modules IC200PNS001 and IC200PNS002.
- For details, refer to the PACSystems RX3i PROFINET IO-Controller Manual, GFK-2571F or later, and PACSystems RX3i PROFINET Scanner Manual, GFK-2737F or later.
- Effective with the release of IC695CEP001 and IC694CEE001, the RX3i may be configured to control a remote drop consisting of one or two I/O modules. The RX3i interface to the remote drop is managed by the PROFINET Controller, IC695PNC001.
- Effective with the release of IC695GCG001, the RX3i may be equipped to control a Genius Bus. The RX3i interface to the Genius Gateway is managed by the PROFINET Controller, IC695PNC001. Refer to *PACSystems RX3i Genius Communications Gateway User Manual,* GFK-2892.
- Effective with the release of IC695EDS001, the RX3i may be configured as a DNP3 Outstation. Refer to *PACSystems RX3i DNP3 Outstation Module IC695EDS001User's Manual,* GFK-2911.
- Effective with the release of IC695EIS001, the RX3i may be configured to act as an IEC 104 Server. Refer to *PACSystems RX3i IEC 104 Server Module IC695EIS001 User's Manual,* GFK-2949.
- PROFINET Scanner User Manual, GFK-2883.
- HART Pass Through allows an RX3i CPU to communicate HART asset management data between HART-capable I/O modules and PC-based asset management tools. This entails the

---

[6] *PAC Machine Edition Release 8.60 SIM 5 is required for EGD Class 1 on both LAN1 and LAN2 of CPE330. This PME version also supports Advanced Configuration Parameters for EGD on CPE330. Alternately, PME Release 8.60 (not SIM 5) supports EGD on CPE330 LAN1 only, and does not support Advanced Configuration Parameters for EGD.*

[7] *PAC Machine Edition Release 9.00 SIM 8 or later is required for native configuration support of the CPE400.*

[8] *PAC Machine Edition Release 9.50 SIM 10 or later is required for native configuration support of the CPL410.*

[9] *IC695PNS001 firmware version 2.40 added support for a number of I/O modules not previously supported, as documented in PACSystems RX3i PROFINET Scanner Important Product Information, GFK-2738L.*

usage of PC-based applications, RX3i Analog modules with HART functionality, and (optionally) supporting PROFINET products. HART Pass-Through operation is described in the *PACSystems HART Pass-Through User Manual*, GFK-2929.

- The following RX3i CPUs support HART Pass-Through: IC695CPE305, IC695CPE310, IC695CPU315, IC695CPU320, IC695CRU320, IC695CPE330[10,11] (firmware version 8.50 or later). All versions of IC695CPE302[13] support this feature.

The following RX3i analog modules support HART:

- IC695ALG626
- IC695ALG628
- IC695ALG728

If used for HART Pass-Through, the supporting RX3i PROFINET Controller and PROFINET Scanner must also contain HART-compatible firmware:

- IC695CPL410 and IC695CPE400 (using embedded PROFINET Controller)
- IC695CPE330-ABAH Firmware Release 8.90 (using embedded PROFINET Controller)
- IC695PNC001-AK firmware version 2.20
- IC695PNS001-ABAH firmware version 2.30[12]
- IC695PNS101-AAAA firmware version 3.10
- IC695CEP001-AAAD firmware version 2.30.
- IC695CMM002 and IC695CMM004 expand the serial communications capability of the RX3i system. Refer to *PACSystems RX3i Serial Communications Modules User's Manual,* GFK-2460.
- CPE310, CPU310, CPU315, CPU/CRU320, and NIU001 provide two serial ports, one RS-232 and one RS-485.
- CPE400 (firmware version 9.40), CPL410, CPE302, and CPE305 each provide one RS-232 serial port.
- CPE330 provides no serial ports.
- Effective with CPE302 firmware version 9.40[13], CPE305/CPE310/CPE400[14] firmware version 9.20, or CPE330 firmware version 9.21, the CPE embedded Ethernet interface supports Simple Network Time Protocol (SNTP) Client, Coordinated Universal Time (UTC), and Daylight Savings Time (DST). Refer to *PACSystems RX3i and RSTi-EP TCP/IP Ethernet Communications User Manual,* GFK-2224 version Q or higher. CPL410 supports this feature in all firmware versions.
- Effective with RX3i firmware version 9.40, the Authorized Firmware Update functionality is available. Users may now authorize access to firmware updates using a custom password. Details are included in the revised firmware update instructions.

---

[10] *When used to support HART Pass Through, CPE330 must do so via a PNC001 and cannot employ its embedded PROFINET feature for this purpose.*

[11] *IC695CPE330 firmware version 8.95 added support for the Remote Get HART Device Information COMMREQ.*

[12] *IC695PNS001 firmware version 2.41 added support for the Remote Get HART Device Information COMMREQ not previously supported, as documented in PACSystems RX3i PROFINET Scanner Important Product Information, GFK-2738L. The syntax and usage for this COMMREQ are described in the PACSystems RX3i System Manual, GFK-2314M or later.*

[13] *PAC Machine Edition Release 9.50 SIM 7 or later is required for CPE302 configuration.*

[14] *PAC Machine Edition Release 9.00 SIM 10, or 9.50 SIM 2, or later is required for SNTP Client, UTC, and DST support.*

- Effective with CPE330 firmware version 9.60, Sequence of Events functionality is available. Refer to *PACSystems RX3i Sequence of Events User Manual*, GFK-3050.
- Effective with CPE330 firmware version 9.75, Hot Standby CPU redundancy is supported with a single RMX per rack. Refer to the *PACSystems Hot Standby CPE Redundancy User Manual*, GFK-2308.
- CPE400 and CPL410 firmware version 9.75 provide a mechanism to recover from STOP-Halt mode using the OLED Display and without removing the Energy Pack.
- Effective with firmware version 10.05, Auto-Recovery from STOP-Halt mode functionality is available. This supersedes/deprecates the mechanism to recover from STOP-Halt mode using the OLED Display on the CPE400 and CPL410.

1.4 RSTi-EP Overview

RSTi-EP CPUs make it possible to incorporate the entire PACSystems programming suite in stand-alone applications or as auxiliary control in larger process applications that use RX3i. They allow the user to leverage the power and flexibility of PACSystems in smaller applications.

At a high level, CPE100/CPE115 supports real-time application status, remote diagnostics and:

- Dual LAN interfaces with four Ethernet ports
- Built-in RS-232, RS-485 serial port
- Support for a range of communications protocols, including PROFINET
- Up to 1 MB of non-volatile user memory.
- All in just 1.5” (38.1mm) of DIN rail space.

At a high-level EPXCPE205, EPXCPE210, EPXCPE215, EPXCPE220, and EPXCPE240 support real-time application status, remote diagnostics and:

- Dual LAN interfaces (can change EPXCPE205 mode to single LAN)
- Built-in RS232 serial port
- Support for a range of communications protocols, including PROFINET with embedded PNC
- Up to 4 MB of non-volatile user memory
- Supports USB (RDSD, RDSD set IP)Supports uSD card (ServReq 56/57 data retention/restore)
- Supports secure Web firmware updates

RSTi-EP CPUs support two independent Ethernet LANs at 1 Gbps for the CPE200 series and 10/100 Mbps for the CPE100 series. LAN1 has only one port and is dedicated to high-speed Ethernet and whereas LAN2 is comprised of two or three switched ports (labeled as 1 and 2 on the CPE200 series and 2, 3 & 4 on the CPE 100 series) configurable as either a second embedded Ethernet controller or an embedded PROFINET controller. All Ethernetports are located on the front panel of the CPU.

The Ethernet controller Interface of RSTi-EP CPUs provides Transmission Control Protocol and Internet Protocol (TCP/IP) communications with other control systems, host computers running the Host Communications Toolkit or programmer software, and computers running the TCP/IP version of the programming software. These communications use the Service Request Transport Protocol (SRTP), Modbus TCP, and Ethernet Global Data (EGD) protocols over a four-layer TCP/IP (Internet) stack.

The RSTi-EP CPUs also embed an industry-standard PROFINET controller that allows them to connect to any type of PROFINET I/O solutions either from Emerson or any third party. It offers enhanced

productivity, flexibility, and performance advantages for virtually any type of control application in a range of industries. PROFINET supports a variety of I/O without compromising system performance and can operate in high-noise environments.

The RSTi-EP CPUs are secure by design, incorporating technologies such as Trusted Platform Modules (currently disabled) and verified boot. Centralized configuration allows encrypted firmware updates to be executed from a secure central location.

In addition to IEC-61131 programming languages, RST-EP CPUs also support native use of user-defined C-blocks

1.5 Migrating Series 90 Applications to PACSystems

- The PACSystems control system provides cost-effective expansion of existing systems. Support for existing Series 90 modules, expansion racks, and remote racks protects your hardware investment. You can upgrade your timetable without disturbing panel wiring.
- The RX3i supports most Series 90-30 modules, expansion racks, and remote racks. For a list of supported I/O, Communications, Motion, and Intelligent modules, see the *PACSystems RX3i System Manual,* GFK-2314.
- The supports most existing Series 90-70 modules, expansion racks, and Genius networks. For a list of supported I/O, Communications, and Intelligent modules, see the *PACSystems Installation Manual,* GFK-2223.
- Conversion of Series 90-70 and Series 90-30 programs preserve existing development efforts.
- Conversion of VersaPro and Logicmaster applications to Machine Edition allows a smooth transition to PACSystems.

1.6 Documentation

1.6.1 PACSystems Manuals

PACSystems RX3i and RSTi-EP CPU Reference Manual	GFK-2222
PACSystems RX3i and RSTi-EP CPU Programmer's Reference Manual	GFK-2950
PACSystems RX3i and RSTi-EP TCP/IP Ethernet Communications User Manual	GFK-2224
PACSystems TCP/IP Ethernet Communications Station Manager User Manual	GFK-2225
C Programmer's Toolkit for PACSystems	GFK-2259
PACSystems Memory Xchange Modules User's Manual	GFK-2300
PACSystems RX3i & RSTi-EP PROFINET IO-Controller Manual	GFK-2571
PACSystems Hot Standby CPU Redundancy User Manual	GFK-2308
PACSystems Battery and Energy Pack Manual	GFK-2741
PAC Machine Edition Logic Developer Getting Started	GFK-1918
PAC Process Systems Getting Started Guide	GFK-2487
PACSystems RXi, RX3i Controller Secure Deployment Guide	GFK-2830

1.6.2 PACEdge Manuals

PACEdge 2.1 User Manual	GFK-3178
PACEdge Secure Deployment Guide	GFK-3197

1.6.3 RX3i Manuals

PACSystems RX3i System Manual	GFK-2314
PACSystems RX3i Rackless CPU w/PACEdge QSG	GFK-3053
PACSystems RX3i IC695ACC403 Rackless Energy Pack Quick Start Guide	GFK-3000
PACSystems RX3i IC695CPE330 1GHz 64MB CPU w/Ethernet Quick Start Guide	GFK-2941
PACSystems RX3i Sequence of Events User Manual	GFK-3050
PACSystems RX3i IC695ACC402 Energy Pack Quick Start Guide	GFK-2939
PACSystems RX3i IC695ACC400 Energy Pack Data Sheet	GFK-2724
DSM324i Motion Controller for PACSystems RX3i and Series 90-30 User's Manual	GFK-2347
PACSystems RX3i PROFIBUS Modules User's Manual	GFK-2301
PACSystems RX3i Max-On Hot Standby Redundancy User's Manual	GFK-2409
PACSystems RX3i Ethernet Network Interface Unit User's Manual	GFK-2439
PACMotion Multi-Axis Motion Controller User's Manual	GFK-2448
PACSystems RX3i PROFINET Scanner Manual	GFK-2737
PACSystems RX3i CEP PROFINET Scanner User Manual	GFK-2883
PACSystems RX3i Serial Communications Modules User's Manual	GFK-2460
PACSystems RX3i Genius Communications Gateway User Manual	GFK-2892
PACSystems RX3i DNP3 Outstation Module IC695EDS001User's Manual	GFK-2911
PACSystems RX3i IEC 104 Server Module IC695EIS001User's Manual	GFK-2949
PACSystems HART Pass-Through User Manual	GFK-2929

1.6.4 RSTi-EP Manuals

RSTi-EP User's Manual	GFK-2958
PACSystems DNP3 Outstation User Manual	GFK-3103
RSTi-EP CPE200 Series Quick Start Guide	GFK-3109

1.6.5 Series 90 Manuals

Series 90 Programmable Coprocessor Module and Support Software	GFK-0255
Series 90 PLC Serial Communications User's Manual	GFK-0582
Series 90-70 DLAN/DLAN+ Interface Module User's Manual	GFK-0729
Series 90-30 Genius Bus Controller User's Manual	GFK-1034

1.6.6 Distributed I/O Systems Manuals

Genius I/O System User's Manual	GEK-90486-1
Genius I/O Analog and Discrete Blocks User's Manual	GEK-90486-2

In addition to these manuals, datasheets and product update documents describe individual modules and product revisions. The most recent PACSystems documentation is available on Emerson's support website. (See link located at the end of this document.)

Section 2 CPU Features & Specifications

This section provides details on the hardware features of the PACSystems CPUs and their specifications.

- Common CPU Features
- RX3i CPU Features and Specifications
- RSTi-EP CPU Features and Specifications

2.1 Common CPU Features

2.1.1 Features are shared by all PACSystems CPU models

- Programming in Ladder Diagram, Function Block Diagram, Structured Text, and C
- Floating point (real) data functions
- Configurable data and program memory
- Non-volatile built-in flash memory for user data (program, configuration, register data, and symbolic variable) storage. Use of this flash memory is optional.
- Configurable RUN/STOP Mode switch
- Embedded serial and/or Ethernet communications (refer to comparison charts in RX3i CPU Features and Specifications and RSTi-EP CPU Features and Specifications)
- Up to 512 or 768 program blocks (model dependent). The maximum size for a block is 128KB.
- Auto Located Symbolic Variables, which allows you to create a variable without specifying a reference address.
- Bulk memory area accessed via reference table %W. The upper limit of this memory area can be configured to the maximum available user RAM.
- Larger reference table sizes, compared to Series 90* CPUs: 32Kbits for discrete %I and %Q and up to 32K words each for analog %AI and %AQ.
- Online Editing mode that allows you to easily test modifications to a running program. (For details on using this feature, refer to the programming software online help and *PACLogic Developer Getting Started,* GFK-1918.)
- The bit in word referencing allows you to specify individual bits in a WORD reference in retentive memory as inputs and outputs of Boolean expressions, function blocks, and calls that accept bit parameters.
- In-system upgradeable firmware for CPU
- The indirect mechanism for upgrading firmware in backplane modules via the CPU.

2.1.2 Features Shared by Certain PACSystems CPU Models

- RX3i CPE302, CPE305, CPE310, CPE330, CPE400, and CPL410 offer battery-less retention of user memory when each is connected to its compatible Energy Pack.
- RSTi-EP offers battery-less retention of user memory.
- All prior RX3i models have battery-backed RAM for user data (program, configuration, register data, and symbolic variable) storage and clocks.
- RX3i CPE302, CPE305, CPE310, CPE330, CPE400, CPL410, and RSTi-EP models have coin battery backup for their real-time clocks (elapsed time clock)
- RX3i CPE302, CPE305, CPE310, CPE330, and RSTi-EP CPE205, CPE210, CPE215, CPE220, and CPE240 models can upload and download data from a Removable Data Storage Device (RDSD). This feature is not yet available on RX3i CPE400, CPL410, and RSTi-EP CPE100/CPE115.
- RX3i CPE302, CPE305, CPE310, CPE330, CPE400, CPL410 and RSTi-EP models support OPC UA.
- RX3i CPE302, CPE305, CPE310, CPE330, CPE400, CPL410, and RSTi-EP models support Ethernet Global Data Class 1.
- RX3i CPE302, CPE305, CPE310, CPE330, CPE400 and CPL410, CPE115 from firmware version 9.97, and CPE205, CPE210, CPE215, CPE220, and CPE240 models support Simple Network Time Protocol (SNTP) Client, Coordinated Universal Time (UTC), and Daylight Savings Time (DST).
- RX3i CPE330, CPE400, CPL410, and RSTi-EP permit LAN2 to be configured as an Embedded PROFINET Controller. Refer to Section 2.1.6, Embedded PROFINET Controller.
- RX3i CPE330, CPE400, and CPL410 support up to 768 blocks including the _MAIN block with firmware release 9.70 or later. Note that PME 9.50 SIM 13 or later is also required for supporting a block count of up to 768. CPE205, CPE210, CPE215, CPE220, and CPE240 support up to 768 blocks.
- RX3i CPE302, CPE305, CPE310, CPE330, CPE400, CPL410, CPE205, CPE210, CPE215, CPE220, and CPE240 models monitor the internal temperature of the CPU hardware.

For a comparative review of CPU features, refer to RX3i CPU Features and Specifications and RSTi-EP CPU Features and Specifications. Note that each specific feature may require a corresponding firmware version of the CPU firmware.

2.1.3 Firmware Storage in Flash Memory

The CPU uses non-volatile flash memory for storing the operating system firmware. This allows firmware to be updated without disassembling the module or replacing EPROMs. The operating system firmware is updated by connecting to the CPU with a PC-compatible computer and running the software included with the firmware upgrade kit.

Each upgrade kit contains specific instructions for performing the upgrade. Depending on the CPU model and firmware version, the method employed is one of the following:

a) Use a serial port and the WinLoader utility (applies to CPU310, CPU315 & CPU320 models, and CPE305/CPE310 models containing firmware versions before v7.30)

b) Use a USB port and memory stick for CPE302-Axxx (with firmware version 9.40 and later) or CPE305-Axxx/CPE310-Axxx models (with firmware version 7.30 and later)

c) Use an Ethernet port and a Web-based mechanism for RXi CPUs, RX3i CPE302-Bxxx CPE305-Bxxx, CPE310-Bxxx, CPE330, CPE400, CPL410, and RSTi-EP CPE100/CPE115 and CPE205, CPE210, CPE215, CPE220, and CPE240.

2.1.4 Operation, Protection, and Module Status

The operation of the CPU can be controlled by the three-position RUN/STOP Switch or remotely by an attached programmer and programming software. Program and configuration data can be locked through software passwords. The status of the CPU is indicated by the CPU LEDs on the front of the module. For details, see *Indicators* for each PACSystems family.

Note: The RESET pushbutton is provided to support future features and has no effect on CPU operation in the current version

2.1.5 Ethernet Global Data

Note: Effective with RX3i firmware version 8.30[5], the CPE310/CPE305 CPUs also support EGD Class 1. Before that firmware version, EGD was only available in the RX3i via the RX3i Ethernet Interface Module (ETM001).

Effective with RX3i firmware version 8.60[6], CPE330 also supports EGD Class 1. This feature is available on all firmware versions of RX3i CPE400, CPL410, CPE302, and RSTi-EP CPE100/CPE115.

Each PACSystems CPU supports up to 255 simultaneous EGD pages across all Ethernet interfaces in the controller. EGD pages must be configured in the programming software and stored in the CPU. The EGD configuration can also be loaded from the CPU into the programming software. Both produced and consumed pages can be configured. PACSystems CPUs support the use of only part of a consumed EGD page, and EGD page production and consumption to the broadcast IP address of the local subnet.

The PACSystems CPU supports 2ms EGD page production and timeout resolution. EGD pages can be configured for a production period of 0, indicating the page is to be produced for every output scan. The minimum period for these "*as fast as possible*" pages are 2 ms.

During EGD configuration, PACSystems Ethernet interfaces are identified by their Rack/Slot location.

2.1.6 Embedded PROFINET Controller

The following CPUs support a feature that permits an Ethernet LAN to be configured for use as a PROFINET Controller:

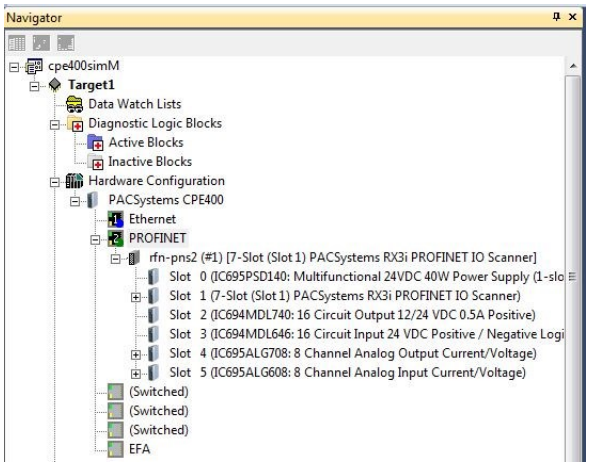
- RX3i CPL410
- RX3i CPE400[15]
- RX3i CPE330[16]
- RSTi-EP CPE100/CPE115[17]
- RSTi-EP CPE205, CPE210, CPE215, CPE220, CPE240

If the Embedded PROFINET Controller feature is to be configured, it must be configured on LAN2 for the CPUs listed above. In the case of RX3i CPE400, CPL410, and RSTi-EP CPE100/CPE115, which are Rackless CPUs, this will be its only PROFINET Controller. In the case of CPE330, the Embedded PROFINET Controller can co-exist with any rack-mounted PROFINET Controllers (IC695PNC001) present in its CPU rack.

To enable the Embedded PROFINET Controller in a project in PAC Machine Edition[18], select the RX3i CPE400, CPL410, CPE330, or RSTi-EP CPE100/CPE115 target in the *PME Navigator* (Figure 1) and open the Hardware Configuration. On the *Settings* tab, change the designated *LAN Mode* of the selected port to *PROFINET*. The *PROFINET Controller* node description then displays that a PROFINET node exists on the selected LAN.

For further details, refer to the PACSystems RX3i & RSTi-EP PROFINET IO-Controller User Manual, GFK-2571G or later.

Figure 1: Configuring an Embedded PROFINET Controller



[15] *CPE400 firmware version 9.00 or later is required for the embedded PROFINET Controller feature.*
[16] *CPE330 firmware version 8.90 or later is required for the embedded PROFINET Controller feature.*
[17] *PAC Machine Edition™ (PME) 9.50 SIM 4 or later is required in order to configure the MRP parameters for CPE100/CPE115.*
[18] *PAC Machine Edition Logic Developer PLC 8.60 SIM 13 or 9.00 SIM 4 or later is required for configuration of the Embedded PROFINET Controller function.*

A PROFINET configuration may be transferred between a PROFINET Controller module (IC695PNC001) and the target Embedded PROFINET IO-Controller using the *cut*/*copy*/*paste* or equivalent drag and drop functions in PAC Machine Edition.

Note: If the PME Project has PROFINET redundant devices, you must record any unique Secondary Target information and disable Redundancy before cut/copy/paste functions on PROFINET Controller modules will work. Then, re-enable Redundancy, mirror, and restore your unique Secondary Target information.

The Embedded PROFINET Controller may be configured as a Simplex PROFINET IO-Controller with support for up to:

- 8 I/O devices for the CPE100, CPE115, and CPE205
- 16 I/O devices for the CPE210 and CPE215
- 32 I/O devices for the CPE220, CPE240, CPE330, and CPE400
- 64 I/O devices for the CPL410

For update rates, loading, and other considerations, refer to the *PACSystems RX3i PROFINET IO Controller User Manual,* GFK-2571F or later.

The Embedded PROFINET Controller supports Media Redundancy Protocol (MRP) and may be used as either a Media Redundancy Manager (MRM) or Media Redundancy Client (MRC) on a redundant media ring. For details concerning the Media Redundancy Protocol, refer to the *PACSystems RX3i PROFINET IO Controller User Manual,* GFK-2571F or later.

The following CPUs support Hot Standby Redundancy with PROFINET IO, using the embedded PROFINET Controller (LAN2):

- CPE330 with firmware version 9.40 or later. The PROFINET Controller may be the embedded PROFINET Controller or a rack-mounted IC695PNC001.
- CPE400 with firmware version 9.30 or later. The PROFINET Controller is always the embedded PROFINET Controller.
- CPL410: The PROFINET Controller is always the embedded PROFINET Controller.

For embedded PROFINET Controllers, this feature permits control of up to:

- 32 devices, 32 of which may be redundant for the CPE330
- 32 devices, 20 of which may be redundant for the CPE400.
- 64 devices, 32 of which may be redundant for the CPL410.

For rack-mounted IC695PNC001, this feature permits control of up to 128 devices, all of which may be redundant.

Note that the host PLC CPU can support up to 255 redundant devices, which may be allocated across 2, 3, or 4 PROFINET Controllers (any combination of embedded PROFINET Controller and/or PNC001 modules).

2.1.7 OPC UA

All PACSystems supports Open Productivity and Connectivity Unified Architecture (OPC UA) Server communications on the embedded Ethernet port only.

Effective with CPE310/CPE305 firmware version 8.20, or CPE302 firmware version 9.40, the CPE embedded Ethernet port supports OPC UA Server.

Effective with CPE310/CPE305 firmware version 9.20, CPE330 firmware version 9.21, or CPE302 firmware version 9.40, OPC UA Server is configurable through PAC Machine Edition[19].

For more information on OPC UA support refer to *PACSystems RX3i and RSTi-EP TCP/IP Ethernet Communications User Manual,* GFK-2224 version M or higher.

2.1.8 Removable Data Storage Devices (RDSDs)

RX3i Backplane CPUs

The RX3i CPE302/CPE305/CPE310/CPE330[20] models provide the ability to transfer applications to and from Removable Data Storage Devices (RDSD). Typically, these are USB-compatible devices, such as a memory stick, smartphone, digital camera, or MP3 device. Once the data is copied to the RDSD, it can be written to other RX3i CPUs of the same type. To copy using RDSD, no PME programming software is needed. The RDSD interface requires a user-supplied flash memory device that complies with the USB 2.0 Specification.

The USB port must be enabled in the RX3i configuration to transfer data between the CPU and the RDSD. The compatible CPUs are shipped with the RDSD (USB) port enabled.

The RDSD load and store operations can include the following data:

- An entire application, including logic and configuration, reference table data, and cam files for Motion applications. (Motion files and local logic for DSM motion applications are supported.) Configuration can include Ethernet Global Data and Advanced User Parameters for the rack-based Ethernet interface. (Although a complete, unmodified application must be placed on the RDSD, you can use an *options.txt* file to download selected components of the application to the target CPU.)
- Passwords and OEM keys, if any, are encrypted and written to the RDSD when the project is loaded from the CPU. When the project is stored on a CPU that has no passwords or OEM keys, those are copied to the CPU.

With Legacy security, when the project is stored on a CPU that has passwords and/or OEM keys, the passwords must match, or the store will fail.

- Fault tables are written to the RDSD before and after a load to or store from the RDSD.
- If a hardware configuration that disables the USB port is successfully stored on the CPU, the fault tables will not be written to the RDSD after the store operation.

Notes:

- With Enhanced Security enabled, the RDSD update will fail if the RDSD source controller has Level 4 password protection and the destination controller is password-protected, regardless of whether the passwords match.
- The USB port is for the transfer of application data only. It is not intended for permanent connection. Do not leave RDSD devices connected during normal operation.
- When using RDSD, all programming software connections must be in the Offline state for the RDSD to function properly.
- CPE330 does not support Cfast memory cards as RDSD devices.
- CPE400/CPL410 does not support any RDSD devices.

[19] *PAC Machine Edition Logic Developer PLC 9.00 SIM 10, or 9.50 SIM 2, or later is required for OPC UA Server configuration.*
[20] *Not yet available on RX3i CPE400 and RSTi-EP CPE100/CPE115*

RSTi-EP Backplane CPUs

The RSTi-EP family backplane models, EPX CPE205/CPE210/CPE215/CPE220/CPE240, provide the ability to transfer applications to and from Removable Data Storage Devices (RDSD). Just as with the RX3i CPEs, USB-compatible devices can be used to copy data to the RDSD and write to other EPXCPE models without the need for PME programming software. The flash memory device must comply with the USB 2.0 specification.

Limitations: the RSTi-EP EPXCPE2XX models do *not* support microSD cards as RDSD devices.

EPXCPE RDSD Backup Generation

The EPXCPE models can perform an RDSD backup of the *PACS_folder*. The backup overwrites the *PACS_folder* if it exists on the USB disk or will create a backup in its absence.

Notes:

- .TAR file can be unzipped with 7-zip to restore the *PACS_folder* if needed.
- **A backup Error** fault is generated if the creation of a backup fails, stopping the RDSD operation.
- If a backup RDSD archive file exists on the USB disk, a new backup will not be created and the upload process will continue.

2.1.9 Uploading a Project from the CPU to the RDSD

Flash devices write in whole memory blocks and memory block sizes vary among devices. The amount of space used by a project may vary between RDSDs due to the differences in minimum block sizes and therefore the number of blocks used by a project. The minimum amount of memory required will be the size of the entire project plus an additional block for the *options.tx*t file if used.

1. Place the CPU that contains the project to be transferred in RUN Mode or STOP Mode.
2. If PME is online with the RX3i, either go Offline or select Monitor mode.
3. Insert the RDSD into the USB connector on the CPU. (After 1 – 2 seconds, the RDSD LED turns solid green.)
4. For CPE302/CPE305/CPE310, push the RDSD direction switch to the left (UPLOAD), then momentarily depress the START pushbutton. For CPE330, depress the RDSD UPLD pushbutton.
5. ***Do not*** remove the RDSD from the CPU during the transfer.

    - The RDSD LED blinks green during the transfer. This can take from 10 to 150 seconds, depending upon the size of the project data.
    - The RDSD LED should turn solid green, indicating that the transfer was completed successfully.
    - If the RDSD LED turns solid red, the transfer has failed. There will be a copy of the fault tables as they existed at the end of the attempted transfer on the RDSD. Insert the RDSD into a PC that has the PacsAnalyzer Utility software and select the *plcfaultafter.dat* file on the RDSD for fault table analysis by PacsAnalyzer. The PacsAnalyzer Utility software can be downloaded from the support website.
    - If the RDSD LED turns solid red, indicating an error, another RDSD operation cannot be initiated until the device is disconnected and then reconnected.

CAUTION

If the RDSD is removed during data transfer from the CPU, the integrity of the RDSD and the files on it cannot be guaranteed. The RDSD status LED may indicate an RDSD fault, and the CPU will abort the data transfer and remain in its current operating mode.

The project files, consisting of the entire contents of the *PACS_Folder* directory and all of its subdirectories, loaded on the RDSD must *not* be modified. If they are modified, the files transferred to the CPU will be invalid.

6. When the RDSD LED turns solid green, indicating the transfer has been completed, remove the RDSD from the CPU. The RDSD can now be used to transfer the application to other RX3i controllers of the same model type.

   You can copy the entire *applications* directory to another USB device and use that device as the source for downloads to CPE302/CPE305/CPE310/CPE330 CPUs, provided none of the files in that directory are changed in any way during the transfer.

Notes: Only one application project can be stored on the RDSD at a time. Before the RX3i writes the project to the RDSD, any previous application is removed; if a directory named PACS_Folder exists on the RDSD at the start of the upload, it is deleted with all of its contents.

RSTi-EP EPXCPE Upload Configuration File Instructions

To perform RDSD Upload (PLC->USB):

1. Insert a formatted (FAT or FAT32) USB stick into the powered-up CPU that has a stored configuration file.
2. When the FLT/RDSD LED is steady GREEN, it indicates that the RDSD Device is available for use.
3. Single-press the **PHYS** button.
4. The CPU will initiate an RDSD upload. This is indicated by FLT/RDSD LED **blinking AMBE**

RDSD Upload Status	LED Behavior
Success	FLT/RDSD LED steady **GREEN**
Failure	FLT/RDSD LED **blinking RED**

5. Remove the USB stick.

2.1.10 Downloading a Project from the RDSD to an RX3i CPU

To download a project to the RX3i, the RDSD must contain a valid project, consisting of the hardware configuration, application logic, and reference memory in a compiled format (originating from another RX3i controller). The project files, consisting of the entire contents of the *PACS_Folder* directory and all of its subdirectories, loaded on the RDSD must ***not*** be modified. If they are modified, the files transferred to the CPU will be invalid.

By default, all project components are stored on the CPU and are written to flash. You can change this operation by placing an *options.txt* file on the RDSD as described below.

1. Ensure that the RX3i is in STOP Mode
2. If PAC Machine Edition is online with the RX3i, either go Offline or select Monitor mode.
3. Connect the RDSD to the USB connector on the CPU that will be receiving the files. The RDSD LED turns solid green.
4. For CPE302/CPE305/CPE310, move the RDSD direction switch to the right (DOWNLOAD), then momentarily depress the START pushbutton. For CPE330, depress the RDSD DNLD pushbutton.
5. ***Do not*** remove the RDSD from the CPU during the transfer.
   - If the target name in the RDSD is different from the target name in the RX3i, the RDSD LED will blink red. If this is expected or acceptable, momentarily depress the START pushbutton again.
   - The RDSD LED blinks green during the transfer. This can take from 10 to 150 seconds, depending upon the size of the project data.
   - The RDSD LED should turn solid green, indicating that the transfer was completed successfully. Unless the RUN/STOP Switch has been disabled in the hardware configuration just stored, it can be used to place the RX3i into RUN Mode after the transfer.
   - If the RDSD LED turns solid red, the transfer has failed:
     - The target memory area(s) in the CPU are cleared. For example, if only the Logic is being downloaded from the RDSD and the store fails (e.g. stick pulled, a problem with the transfer of data), Logic memory is cleared following the failed RDSD download. If other memory areas were also queued up for transfer, those memory areas are also cleared as a result of the failure.
     - There will be a copy of the fault tables as they existed at the end of the attempted transfer on the RDSD. Insert the RDSD into a PC that has the PacsAnalyzer Utility software and select the *plcfaultafter.dat* file on the RDSD for fault table analysis by PacsAnalyzer Utility.
     - If the RDSD LED turns solid red, indicating an error, another RDSD operation cannot be initiated until the device is disconnected and then reconnected.

CAUTION

If the RDSD is removed during data transfer to the CPU, the RX3i controller will generate a fatal fault (sequence store fault) and SYS FLT LED will turn red. You will need to clear the fault tables through a programmer connection or by power cycling the CPU with the Energy Pack disconnected before attempting to download again. Each type of data being downloaded (logic, config, and/or data) is cleared within the target CPU.

6. When the RDSD LED turns solid green, indicating the transfer has been completed, remove the RDSD from the CPU.

The RUN/STOP Switch can be used to place the RX3i into RUN Mode after the transfer unless it has been disabled in the hardware configuration just stored. If the RUN/STOP switch is disabled, you will first need to connect with the programmer to place the RX3i in RUN Mode.

Downloading Configuration File Instructions for an EPXCPE

To perform RDSD Download (USB->PLC, ONLY in STOP Mode):

1. Place the PLC into STOP mode. With either no stored configuration or a stored valid configuration that has USB Enabled
2. Insert a formatted (FAT or FAT32) USB stick into the powered-up CPU that has a stored configuration file.
3. When the FLT/RDSD LED is steady GREEN, it indicates that the RDSD Device is available for use.
4. Double-press the **PHYS** button.
5. RDSD Download initiates and is shown by FLT/RDSD LED **blinking GREEN**
6. RDSD program name mismatch is shown by FLT/RDSD LED solid **RED**
7. Remove the USB stick.

RDSD Upload Status	LED Behavior
Success	FLT/RDSD LED steady **GREEN**
Failure	FLT/RDSD LED **blinking RED**

Using an Options.txt File to Modify Download Operation on an EPXCPE

An *options.txt* file can be used to modify the operation of the RDSD during a store to the RX3i. This is a plain-text file that can contain some or all of the following statements, in any order. The format of each option line is the option keyword, followed by a space, followed by either a capital Y or a capital N. The options keyword must be spelled exactly as indicated below. If an option statement is omitted from the file, the default value will be used.

If you want to use all of the default operations, *the options.txt* file is not necessary.

Options.txt File Format

Option Keyword	Default value	Description

Download_LogicAndCfg	Y (yes)	Logic and configuration are copied to the CPE302/CPE305/CPE310/CPE330 (including symbolic variables)
Download_Data	Y (yes)	Reference memory is copied to the CPE302/CPE305/CPE310/CPE330 (excluding symbolic variables)
Download_CamFiles	Y (yes)	CAM files are copied to the CPE302/CPE305/CPE310/CPE330
Write_Flash	Y (yes)	The downloaded CPE302/CPE305/CPE310/CPE330 contents (as specified by the above keywords) by default will be written to flash upon completion of the store
Perform_Ip_Config	N (no)	Will apply the IP configuration specified in ipConfig.xml

Sample options.txt File

If the following *options.txt* file is present on the RDSD, logic, configuration, and reference data are copied to the CPU, and files are written to flash. Cam files are not copied. RDSD Ip Configuration feature will be applied (see next section).

Download_LogicAndCfg Y
Download_Data Y
Download_CamFiles N
Write_Flash Y
Perform_Ip_Config Y

RSTi-EP EPXCPE RDSD IP Configuration on an EPXCPE

There is additional functionality to be able to specify a custom IP configuration for LAN1 and LAN2 that will overwrite the LAN1 and LAN2 configurations present on the configuration in the RDSD project. This feature is intended to alleviate the problem where the user wishes to apply the same hardware configuration and logic to all their controllers, however, they cannot have the same IP address. With this feature, the user can apply the same logic and configuration for each controller using a single RDSD project but overwrite their LAN1 and LAN2 configurations to prevent IP conflicts.

To use this feature, there must be a file titled "ipConfig.xml" present on the RDSD. This file should contain elements for the serial numbers of the devices whose LAN configurations they wish to overwrite. Each serial number has an associated LAN1 and LAN2. The option to enable this operation must be present in the options.txt (see the previous section). When the RDSD download operation occurs this file will be parsed, and if a serial number matching that of the controller is found, the LAN1 and LAN2 configuration will be applied, overwriting the configuration present in the project on the RDSD.

Figure 2: Example ipConfig.xml

```
<ipConfig>
 <config serial="04896031">
  <LAN1>
   <ip>10.219.33.30</ip>
   <subnet>255.255.254.0</subnet>
   <gateway>10.219.32.1</gateway>
  </LAN1>
  <LAN2>
   <ip>192.168.1.102</ip>
   <subnet>255.255.254.0</subnet>
   <gateway>0.0.0.0</gateway>
  </LAN2>
 </config>
  <config serial="04896030">
  <LAN1>
   <ip>192.168.2.2</ip>
   <subnet>255.255.254.0</subnet>
   <gateway>0.0.0.0</gateway>
  </LAN1>
  <LAN2>
   <ip>192.168.1.104</ip>
   <subnet>255.255.254.0</subnet>
   <gateway>0.0.0.0</gateway>
  </LAN2>
 </config>
</ipConfig>
```

If the example ipconfig.xml above was on the RDSD, the feature was enabled in options.txt, and the RDSD download was performed on two controllers with serial numbers “04896031” and “04896030”, then these two controllers would share the same RDSD project, however, they would have different configurations for LAN1 and LAN2.

Important Notes Regarding IP Configuration

- If the option is enabled in options.txt, the feature will be executed.
- If the ipConfig.xml is missing or the XML file is not properly formatted (see figure 2) then the RDSD download will fail.
- Each serial number must have a LAN1 and LAN2 config, and each LAN config must have an IP, subnet, and gateway.
- If the serial number of the device you are performing the download on is not present in ipConfig.xml, and the feature is enabled in the options.txt, then the operation will fail.
- The LAN1 and LAN2 configurations must be valid configurations that can be applied in PME, that is, if the user tries to apply a configuration that PME would reject with an error, the RDSD operation will fail.
- PME is not aware of the LAN1 and LAN2 configurations specified in the XML file. After this operation, if you upload to PME, it will state that the LAN1 and LAN2 configurations are that of the PROJECT ON RDSD, not the configuration from the XML file.

- Downloading from PME, clearing controller memory, or performing an RDSD download with the IP configuration off will all overwrite the LAN1 and LAN2 from the previous RDSD IP configuration.
- The IP Configuration is persistent after a power cycle and firmware update.

Security

When the application is written to the RDSD from a controller that has passwords and/or an OEM key defined, the passwords and OEM key are encrypted and stored on the RDSD. When the project is written from the RDSD to a CPE302/CPE305/CPE310/CPE330[21], the passwords and OEM key are copied to it.

If an OEM key is defined on the RDSD when a transfer is complete, the OEM protection will be enabled (locked). When an application is being stored to a CPE302/CPE305 that already has passwords and/or an OEM key defined, the passwords/key on the RDSD must match the passwords/key in the target CPE302/CPE305/CPE310/CPE330, or the transfer will fail.

RDSD Error Reporting

Errors are indicated when the RDSD LED becomes solid red (not blinking). All errors are reported in the Controller fault tables. If the Controller has faults in its fault tables before it receives a store, the fault tables are written to *plcfaultbefore.dat* and *iofaultbefore.dat* on the RDSD. If the Controller has faults in its fault tables after it receives a store, the fault tables are written to *plcfaultafter.dat* and *iofaultafter.dat* on the RDSD. Previous versions of these files are deleted before the transfer. If either fault table is empty, the corresponding file is not written and will not be present.

To read any of the *.dat* files mentioned above, open PacsAnalyzer Utility. In *settings*, enable a *file to analyze*. Then click the *file analyze* button on the main screen. Select as *Input File* the *.dat* file to be analyzed. Select as *Output File* the filename and folder into which you wish to deposit the resulting text. The text will be in English.

If a hardware configuration that disables the USB port is stored on the CPU, the fault tables will not be written to the RDSD after the store operation because the USB port will be disabled at the end of the store process.

[21] *Not implemented on CPE400 at time of publication.*

2.1.11 CPU Over-Temperature Monitoring and Behavior

RX3i CPE302, CPE305, CPE310, CPE330, CPE400, and CPL410 models monitor the internal temperature of the CPU.

- If the temperature rises to a near-critical level, these CPUs set the *CPU Over Temperature* Fault (refer to section 3.2.7). The actual temperature varies from CPU to CPU, as each has a different temperature specification.
- If the temperature continues to rise and reaches the specification limit, the CPU goes into a firmware-controlled reset.
- Uniquely, the CPE400 and CPL410's TEMP LED is lit **amber**.
- Following reset, the CPU continues to monitor the internal temperature.
- If the temperature falls sufficiently (i.e. by 10°C or 18°F), the CPU will automatically attempt to restart.
- If the CPU is manually restarted before the temperature drops to the automatic restart level, the CPU will attempt to restart and will monitor its internal temperature as before.
- Upon successfully restarting, the Overtemp Fault will be recovered, providing the CPU has been connected to an Energy Pack. If there is no Energy Pack connected, or if the Energy Pack has discharged, the Overtemp Fault will be lost.
- The CPE400 and CPL410 always turn off their TEMP LED at power-up.

RSTi-EP EPXCPE2XX controllers monitor the internal temperature of its two CPUs to protect them from over-temperature. When either CPU (APU/RPU) reaches the warm set point (100), then a PLC fault is generated to indicate the PLC has reached the warm point and the OK LED is lit **amber**.

When either CPU reaches the hot set point (90), then a PLC fault is generated to indicate the PLC has reached the hot set point and the OK led is lit **red**, Next, the FW will perform an orderly shutdown before then removing the 5 V power. Finally, the PLC will monitor the temp and wait until the unit cools down before the 5 V power is re-applied.

OK LED Behavior for RSTi-EP EPX2XX Controllers

OK LED Behavior	Description
Green Solid	Normal Temperature (FW Controlled)
Amber Solid	Warm (FW Controlled)
Red Solid	Hot (FW Controlled)

2.2 RX3i CPU Features and Specifications

	CPU310	CPU315	CPU320/ CRU320[22]	CPE302[23]/ CPE305	CPE310	CPE330	CPE400/ CPL410
Lifecycle Phase	Discontinued - use CPE310	Discontinued - use CPE310 or CPE330	Discontinued - use CPE330	Active	Active	Active	Active
Operating System	VxWorks	VxWorks	VxWorks	VxWorks	VxWorks	VxWorks	VxWorks
#RX3i Slots Occupied	2	2	2	1	2	2	N/A
Backplane	<----------Supports High-Speed PCI IC695* and Serial IC694* Modules ---------->						PACEdge CPE400/CPL410
Standard Temperature Range							
RX3i	0°C to 60°C	0°C to 60°C	0°C to 60°C	0°C to 60°C[24]	0°C to 60°C[15]	0°C to 60°C	-40°C to 70°C[25]
Power Requirements							
RX3i +3.3Vdc	1.25 A	1.0 A	1.0 A	1.0 A	1.0 A	0 A	N/A
RX3i +5 Vdc	1.0 A	1.2 A	1.2 A	1.0 A (up to 1.5 A if USB draws 0.5A)	1.0 A (up to 1.5 A if USB draws 0.5A)	0 A	N/A
RX3i +24Vdc Relay with Energy Pack				0.5 A at start-up; 0.1 A otherwise	0.5 A at start-up; 0.1 A otherwise	0.750 A	N/A
RX3i +24Vdc Relay w/o Energy Pack						0.625 A	N/A
Input Power (Max)							20 W
Input Voltage (Min)							18 Vdc
Input Voltage (Max)							30 Vdc
Memory Backup Mechanism[26]	Battery see GFK-2741	Battery see GFK-2741	Battery see GFK-2741	Energy Pack: IC695ACC400	Energy Pack: IC695ACC400	Energy Pack: IC695ACC402	Energy Pack: IC695ACC403
Display							
	LEDs	LEDs	LEDs	LEDs	LEDs	LEDs	LEDs & OLED

[22] For CRU-type CPUs, see Redundancy section at bottom of this table.
[23] Where different, CPE302 value is shown in parentheses (). Also note that first Firmware Version of CPE302 was FW 9.40.
[24] LT versions of the hardware are rated from -40°C to 60°C.
[25] The maximum operating temperature varies according to installation altitude: 70 °C at 0m to 2000m, 65 °C at 2000m to 3000m, and 60 °C at 3000m to 4000m.
[26] See Battery Compatibility and Memory Retention (Time in Days at 20°C) in GFK-2741

	CPU310	CPU315	CPU320/ CRU320[22]	CPE302[23]/ CPE305	CPE310	CPE330	CPE400/ CPL410
Firmware Upgrade[27]							
CPU Firmware Upgrade Mechanism	<----------WinLoader/Serial Port---------->			v7.30 & later: USB earlier: WinLoader/ Serial Port	v7.30 & later: USB earlier: WinLoader/ Serial Port	Web Interface Ethernet Port	Web Interface Ethernet Port
Indirect Backplane Module Upgrade	<----------WinLoader/Serial Port---------->					Web Interface Ethernet Port	N/A
Program Portability							
Direct Import (with limitations)[28]					CPU310, CPU315	CPU315, CPU320	N/A
RX3i PACSystems Applications using Family Type Conversion							Y
Program Security							
Secure Boot						N	Y
Trusted Platform Module (TPM)						Y	Y
Program Storage							
Battery-backed RAM	10 MB[29]	20 MB[29]	64 MB[29]	302/305-Axxx: 2/5 MB[30] 302/305-Bxxx: 2/6 MB[31]	10 MB[30]	64 MB[30]	64 MB[30]
Non-Volatile Flash	10 MB	20 MB	64 MB	302/305-Axxx: 2/5 MB 302/305-Bxxx: 2/6 MB	10 MB	64 MB	64 MB
Battery Life Expectancy, RAM Backup[26]	see GFK-2741	see GFK-2741	see GFK-2741				N/A
Life Expectancy, Energy Pack Capacitors				5 years	5 years	5 years	5 years

[27] Effective with RX3i firmware version 9.40, the Authorized Firmware Update feature was added: with it, user can set/change his own password.
[28] See corresponding IPI for target CPU.
[29] Battery-backed RAM.
[30] RAM backup with compatible Energy Pack attached.

	CPU310	CPU315	CPU320/ CRU320[22]	CPE302[23]/ CPE305	CPE310	CPE330	CPE400/ CPL410
Auxiliary Storage							
CFast						Inactive	N/A
Remote Data Storage Device (RDSD)				Y - USB	Y - USB	Y - USB	N/A
Micro SD						N/A	N/A
Programming Capabilities							
Max Number of Program Blocks[32]	512	512	512	512	512	768	768
Program Block Max Size	128 KB	128 KB	128 KB	128 KB	128 KB	128 KB	128 KB
Discrete Reference Memory (%I, %Q)[33]	32 Kbits	32 Kbits	32 Kbits	(16)/32 Kbits	32 Kbits	32 Kbits	32 Kbits
Analog Reference Memory (%AI, %AQ) [21]	32 Kwords	32 Kwords	32 Kwords	32 Kwords	32 Kwords	32 Kwords	32 Kwords
Bulk Reference Memory (%W) [21]	up to max user RAM	up to max user RAM	up to max user RAM	up to max user RAM	up to max user RAM	up to max user RAM	up to max user RAM
Managed Memory (Symbolic + I/O Variables)[21,34]	up to 10 MB	up to 20 MB	up to 64 MB	302/305-Axxx: 2/5 MB 302/305-Bxxx: 2/6 MB	up to 10 MB	up to 64 MB	up to 64 MB
Floating Point	y	y	y	y	y	y	y
Ladder Diagram (LD)	y	y	y	y	y	y	y
Function Block Diagram (FBD)	y	y	y	y	y	y	y
Structured Text (ST)	y	y	y	y	y	y	y
PID Built-In Function Block	y	y	y	y	y	y	y
"C" Language External Blocks	y	y	y	y	y	y	y

[32] Support for up to 768 blocks requires firmware release 9.70 or later and PME 9.50 SIM 13 or later.
[33] Note: Whenever the size of any reference memory is changed, the content of the corresponding reference memory is automatically cleared.
[34] For discussion of memory types and how they are managed, refer to PACSystems RX3i CPU Programmer's Reference Manual, GFK-2950 Section 3.

	CPU310	CPU315	CPU320/ CRU320[22]	CPE302[23]/ CPE305	CPE310	CPE330	CPE400/ CPL410
Communications							
Ethernet Non-Switched RJ45 (dedicated NIC)	N/A	N/A	N/A	N/A	N/A	10/100/1000 x1	10/100/1000 x1
PACEdge Ethernet RJ45 (dedicated NIC)	N/A	N/A	N/A	N/A	N/A	N/A	10/100/1000 x1
Ethernet Switched RJ45 (shared NIC)	N/A	N/A	N/A	10/100/1000 x2 (-Bxxx only) 10/100 x1 (-Axxx only)	10/100 x1	10/100/1000 x2	10/100/1000 x4 (2 pairs)
Ethernet Communications Platform	ETM001 only	ETM001 only	ETM001 only	Built-in and /or ETM001	Built-in and /or ETM001	Built-in and /or ETM001	Built-in
Advanced User Parameters (AUP file)	N/A	N/A	N/A	Y[35]	Y[23]	N[36]	N[24]
RS-232	9-pin D x1	9-pin D x1	9-pin D x1	RJ-25 x1	9-pin D x1	N/A	RJ-45 x1
RS-485	15-pin D x1	15-pin D x1	15-pin D x1	N/A	15-pin D x1	N/A	N/A
USB				USB-A 2.0 x1	USB-A 2.0 x1	USB-A 2.0 or USB-A 1.1 x1	USB 3.0 x 2 (inactive)
Time-of-Day Clock							
Time-of-Day Clock Accuracy (@60°C)	±2 secs/day	±2 secs/day	±2 secs/day	±2 secs/day	±2 secs/day	±2 secs/day	±2 secs/day
Elapsed Time Clock (internal timing) accuracy	±0.01% max	±0.01% max	±0.01% max	±0.01% max	±0.01% max	±0.01% max	±0.01% max
Simple Network Time Protocol (SNTP) accuracy to timestamp[37]	±2 ms using ETM001	±2 ms using ETM001	±2 ms using ETM001	±2 ms using ETM001	±2 ms using ETM001	±2 ms using ETM001 or embedded	±2 ms embedded only
RTC Battery Backup				Y	Y	Y	Y
RTC Battery Life expectancy				5 years	5 years	5 years	5 years

[35] Refer to PACSystems RX3i TCP/IP Ethernet Communications User Manual, GFK-2224M or later for supported AUPs.

[36] The Advanced User Parameters (AUP) feature has been incorporated into PME Hardware Configuration (HWC) effective with PME release 8.60 SIM5.

[37] Effective with CPE302/CPE305/CPE310/CPE400 firmware version 9.20, or CPE330 firmware version 9.21, SNTP is supported by the embedded CPU Ethernet interfaces. PAC Machine Edition Release 9.00 SIM 10, or 9.50 SIM 2, or later is required for SNTP Client, UTC, and DST support.

	CPU310	CPU315	CPU320/ CRU320[22]	CPE302[23]/ CPE305	CPE310	CPE330	CPE400/ CPL410
Protocols							
Modbus RTU Slave	Y	Y	Y	Y	Y	N/A	N/A
SNP Slave	Y	Y	Y	Y	Y	N/A	N/A
Serial I/O	Y	Y	Y	Y	Y	N/A	Y[38]
SRTP (# simultaneous server conns)				up to 32	up to 32	up to 48	up to 48
Modbus TCP (# simultaneous server connections)				up to 16[39]	up to 16[39]	up to 16[39]	up to 16[39]
SRTP Channel or Modbus TCP Client (# simultaneous)				up to 16[39]	up to 16[39]	up to 32[39]	up to 32[39]
Ethernet Global Data (EGD)				FW 8.30[23,40]	FW 8.30[40]	FW 8.60[40]	Y
Number of EGD Exchanges (max)[41]				255	255	255	255
Selective Consumption of EGD				Y	Y	Y	Y
PROFINET[42]				N	N	FW 8.90	Y
OPC UA Server[43]				FW 8.20[23,44]	FW 8.20[44]	Y[44]	Y[44]
Remote Station Manager over UDP				Y	Y	Y - limited	Y - limited
Station Manager over Serial Comm Port	via ETM001	via ETM001	via ETM001	via ETM001	via ETM001	via ETM001	N/A
DNP3 Outstation master/client support (# simultaneous)							Up to 8

[38] CPE400 Serial IO requires firmware version 9.40 or later.
[39] Sixteen clients are permitted: each may be SRTP or Modbus/TCP.
[40] EGD Class 1 only: supports up to 255 simultaneous Class 1 EGD exchanges.
[41] Limit is per target, so all producers and consumers in the CPU system are counted towards this limit.
[42] CPE400 and CPE330 (firmware version 8.90 or later) provide PROFINET support via an embedded PROFINET Controller: no external hardware is required. All other CPUs that support PROFINET require a rack-mounted PROFINET Controller (IC695PNC001). CPE330 may also host IC695PNC001 modules in the CPU rack. Refer to the PACSystems RX3i PROFINET IO-Controller Manual, GFK-2571F or later.
[43] For a discussion of OPC UA, refer to PACSystems RX3i TCP/IP Ethernet Communications User Manual, GFK-2224M Section 10.
[44] Supports up to 5 concurrent sessions with up to 10 concurrent variable subscriptions and up to 12,500 Variables.

	CPU310	CPU315	CPU320/ CRU320[22]	CPE302[23]/ CPE305	CPE310	CPE330	CPE400/ CPL410
Redundancy Features			**Model CRU320 only**			**Configurable in CPE330**	**Configurable in CPE400/ CPL410**
Memory Error Checking and Correction (ECC)			Single bit correcting & Multiple bit checking			Single bit correcting & Multiple bit checking	Single bit correcting & Multiple bit checking
Switchover Time (max)[45]			1 logic scan			1 logic scan	1 logic scan
Switchover Time (min) [31]			3.133 ms			3.133ms	300ms
Max data in the redundancy transfer list[46]			2 MB			2 MB	2 MB
Redundant Synchronized Links Supported			RMX128 x2 max RMX228 x2 max			RMX128 x2 max RMX228 x2 max	LAN3

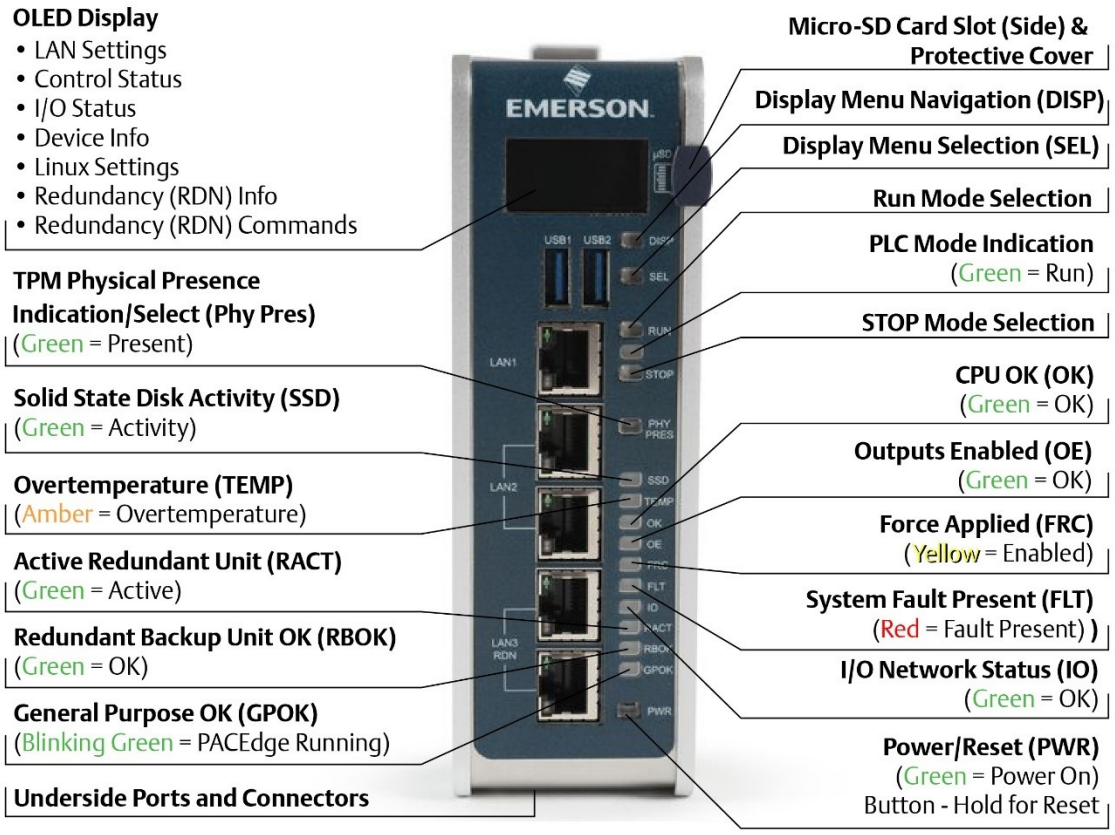
[45] Switchover time is defined as the time from failure detection until backup CPU is active in a redundancy system.
[46] Symbolic variable and Reference data can be exchanged between redundancy controllers, up to the stipulated limit.

2.2.1 CPE400 and CPL410

Introduction

The PACSystems™ RX3i CPL410/CPE400 Rackless Controllers ship with a fully licensed installation of Emerson's PACEdge, software suite, which is provided to enable data analytics and visualization at the Edge.

Figure 3: CPE400 and CPL410 Front View and Features



Features

- Users may program in Ladder Diagram, Structured Text, Function Block Diagram, or C.
- Contains 64vMB of configurable data and program memory.
- Supports auto-located Symbolic Variables that can use any amount of user memory.
- Reference table sizes include 32k bits for discrete %I and %Q and up to 32k words each for analog %AI and %vAQ. Bulk memory (%W) is also supported for data exchanges.
- Supports up to 768 program blocks[47]. The maximum size for a block is 128KB.
- Supports four independent 10/100/1000 Ethernet LANs. The three Ethernet ports located on the front panel, as shown in Figure 3, are exclusively assigned to the RX3i PLC. LAN1 attaches via the upper, dedicated RJ 45 connector. LAN2 and LAN3 each attach via a pair of internally-switched RJ 45 connectors. The fourth LAN, labeled ETH, is located on the underside and is exclusively used for PACEdge connectivity.
- The CPE400 is bundled with PACEdge + Movicon Connext Software. It also comes with the Movcion Connext Server pre-installed. For more details on PACEdge, see GFK-3178, *PACEdge 2.1 User Manual.* For help with Movicon, see http://www.movicon.info/HelpNExT3.4/en-US/PlatformNext.htm
- The CPL410 is bundled with PACEdge +Movicon WebHMI with 2,000 tags pre-licensed and pre-installed. For more details on PACEdge, see GFK-3178, *PACEdge 2.1 User Manual. For help with Movicon, see http://www.movicon.info/HelpNExT3.4/en-US/PlatformNext.htm*
- The embedded communications interface has dedicated processing capability, which permits the CPU to independently support LAN1 and LAN2 with:
  - up to 48 simultaneous SRTP Server connections;
  - up to 16 simultaneous Modbus/TCP Server connections;
  - 32 Clients are permitted; each may be SRTP or Modbus/TCP.
  - OPC UA Server with support for up to 5 concurrent sessions with up to 10 concurrent variable subscriptions and up to 12,500 variables;
  - up to 255 simultaneous Class 1 Ethernet Global Data (EGD) exchanges.
  - The embedded PLC may use one or both Ethernet LAN2 ports to support the embedded PROFINET I/O Controller. PROFINET supports up to 64 I/O devices with update rates of 1 – 512ms. I/O device update rates of 8ms and faster are possible with 16 or fewer devices. Update rates of 16ms and higher results whenever more than 16 devices are configured.
  - Media Redundancy Protocol (MRP) allows the CPL410 RX3i PLC to participate in a PROFINET I/O network with MRP ring technology. This eliminates the I/O network as a single point of failure. The RX3i PLC may be used as either a Media Redundancy Manager or Media Redundancy Client.
  - The CPE400/CPL410 RX3i PLC support Hot Standby Redundancy with PROFINET IO. In this configuration, LAN3 is used as a high-speed data synchronization link between the two redundant CPUs. Only the Primary and Secondary CPUs may be attached to LAN3. Two OLED menu items support Redundancy operation: RDN Info and RDN Command. The RACT and RBOK LEDs reflect the status of the Redundant CPUs.

[47] *Support for up to 768 blocks requires firmware release 9.70 or later and PME 9.50 SIM 13 or later.*

- The CPE400/CPL410 RX3i PLCs support two independent Redundant IP addresses, one for LAN1 and one for LAN2. LAN2 Redundant IP is supported when configured for Ethernet mode only. Redundant IP is supported by the SRTP Server, Modbus TCP Server, and EGD protocols. It is not possible to use Redundant IP with the OPC UA Server or with the Ethernet firmware update web page.
- The real-time part of CPE400/CPL410 is secure by design, incorporating technologies such as Trusted Platform Modules, secure boot, and encrypted firmware updates. It is neither accessible nor modifiable by customers nor intruders, thus guaranteeing the integrity of the controller. As the PACEdge part of the CPE400/CPL410 is open for user modifications, the same integrated security features cannot be provided for PACEdge. Customers must take the necessary steps to secure PACEdge to the degree necessary for their use case. Emerson provides a PACEdge Secure Deployment Guide (GFK-3197) to support customers in this task.
- Optional Energy Pack, IC695ACC403, allows the RX3i PLC of CPL410 to instantly save user memory to non-volatile storage in the event of loss of power.
- OPC UA Sweep Mode & Sweep Time: The RX3i PLC's sweep mode and sweep time are available through the OPC UA server. The Sweep Mode variable reports the controller's current mode: Stop Disabled, Run Enabled, Stop Enabled, Run Disabled, Stop Faulted, and Stop Halted. The Sweep Time variable reports the sweep time in seconds. These variables are located under Emerson Device Information -> PACSystems RX3i -> Controller.
- An OLED display that provides access to basic CPE400/CPL410 status and control information including each LAN's configured IP Address.
- Operating temperature range from -40°C to 70°C (-40°F to 158°F).
- Alternate panel-mount adaptor plate included.

Switches

All user-accessible switches are provided as pushbuttons on the front panel as described below.

Pushbutton	Function
DISP	Permits users to navigate menus in the OLED display.
SEL	Permits users to select the menu item on the OLED display.
RUN	Activates OLED Menu to select RUN/Enabled or RUN/Disabled Mode for the embedded PLC.
STOP	Activates OLED Menu to select STOP/Enabled or STOP/Disabled Mode for the embedded PLC.
PHY PRES	Not functional.
PWR	Hold down for a brief period to induce CPU Reset. Note that this does not turn unit power off, but only holds the unit in the Reset

Displays and Indicators (LEDs)

OLED Display

The monochrome organic light-emitting diode (OLED) display is used to display CPE400/CPL410 system menus. It interacts with the DISP pushbutton, which jogs the cursor from one menu item to the next, and with the SEL pushbutton, which activates the currently indicated menu item for further action.

The OLED display permits the user to:

- Display Ethernet LAN Settings: IPv4 address.
- Display the PLC firmware revision.
- Set/view PLC mode and view sweep time.
- Set the PLC mode to RUN/STOP with I/O Enabled/Disabled via the display.
  Note: the RUN and STOP pushbuttons activate the PLC Mode menu items per SectionView whether all, some, or none of the PROFINET I/O devices are connected.
- View HSB Redundancy Mode and State.
- Command an HSB Redundancy Role Switch.

Status Indicators (LEDs)

LED	*LED State*		*Operating State*
PLC MODE[48]	●	On Green	PLC is in RUN mode.
	○	Off	PLC is in STOP mode.
RUN	✱	Blinking in unison	CPU is updating an internal programmable hardware device.
OE	✱		

[48] *This LED is located between the RUN and STOP pushbuttons. It indicates the PLC Mode.*

LED	*LED State*		*Operating State*
PHY PRES	●	On Green	TPM Physical Presence (not functional).
	○	Off	
SSD	●	On Green	Activity detected on Solid State Disk.
	○	Off	No activity was detected on the Solid State Disk.
TEMP	●	On Red	CPU Overtemperature condition detected.
	○	Off	Overtemperature condition not detected.
OK	●	On Green	CPU has passed its power-up diagnostics and is functioning properly. *(Following initialization sequence.)*
	○	Off	Power is not applied or CPU has a problem.
	✸	Blinking; All other LEDs off	PLC in STOP/Halt state; possible watchdog timer fault. If the programmer cannot connect, cycle power with charged Energy Pack attached and refer to fault tables.
OK OE	✸ ✸	Blinking alternately	CPU encountered a Secure Boot Error.
OE	●	On Green	The output scan is enabled.
	○	Off	The output scan is disabled.
FRC	●	On Yellow	One or more Overrides were active in I/O Reference Table(s).
	○	Off	No Overrides are active in any I/O Reference Table.
FLT	●	On Red	PLC is in STOP/Faulted mode: a fatal fault has occurred.
	○	Off	No fatal faults were detected.
IO	●	On Green	PROFINET Connection Status = OK.
	○	Off	PROFINET Connection Status not OK.

LED	*LED State*	*Operating State*
RACT	On Green	Local Redundant CPU is Ready & Active.
	Off	Local Redundant CPU is not Ready.
RBOK	On Green	Remote Redundant CPU is Ready.
	Off	Remote Redundant CPU is not Ready.
GPOK	On Green	PACEdge Running.
	Off	PACEdge not running.
	Slow Blinking Green	Warning
	Fast Blinking Green	Error
PWR	On Green	CPU running.
	Blinking Green	Booting up – diagnostics in progress.
	On Red	Off
	Off	Reset / Power not detected.

Front Ethernet Indicators (LAN1, LAN2, LAN3 RJ45 Built-in LEDs)

LED	LED State	Operating State
Link Status (upper)	On Green	The corresponding link has been established.
	Blinking Green	Traffic is detected at the corresponding port.
	Off	No connection was established at the corresponding port.
Link Speed (lower)	On Green	Corresponding data speed is 1 Gbps or 100 Mbps.
	Off	The corresponding network data speed is 10 Mbps

Bottom Ethernet Indicators (ETH RJ45 Built-in LED)

LED	LED State	Operating State
Link Status (upper)	On Green	The corresponding link has been established.
	Blinking Green	Traffic is detected at the corresponding port.
	Off	No connection was established at the corresponding port.
Link Speed (lower)	On Green	The corresponding network data speed is 1 Gbps.
	On Yellow	GPOS port only: network data speed is 100 Mbps
	Off	The corresponding network data speed is 10 Mbps

USB Ports

On the front panel, the CPE400/CPL410 features two USB 3.0 ports, labeled USB1 and USB2.

- USB1 is assigned to PACEdge and can be used for keyboards, memory sticks, or other memory devices. For other USB devices, an appropriate PACEdge driver will need to be installed.
- USB2 is reserved for the Controller run time PACS.
- For USB port pinouts, refer to.

Note: In the first release, both USB ports are accessible by PACEdge. Do not implement PACEdge use cases relying on the availability of both USB ports, since USB2 will be assigned to the PLC/PACS in the future.

Front-Panel Ethernet Ports

All front panel Ethernet ports are exclusively assigned to the PLC component of the CPL410 and cannot be used by Linux.

LAN1 connects to the uppermost RJ45 connector. It is not switched.

LAN2 connects to the middle two RJ45 connectors. These two ports are switched internally.

LAN3 connects to the two lower RJ45 connectors. These two ports are switched internally. LAN3 may only be used to supply a high-speed synchronization link between the Primary and Secondary CPUs in Hot Standby Redundancy. Both ports are typically used, as described in the *PACSystems Hot Standby CPU Redundancy User Manual*, GFK-2308 (revision L or later).

Each of the embedded Ethernet interfaces automatically senses the data rate (10 Mbps or 100 Mbps or 1 Gbps), communications mode (half-duplex or full-duplex), and cabling arrangement (straight-through or crossover) of the attached link. LEDs embedded in each RJ45 connector provide indications per the table above.

LAN1 or LAN2 may be used to communicate with the PME programming software using the Service Request Transport Protocol (SRTP).

To establish Ethernet communications between the PME programming and configuration software and the CPU, you *first* need to know the target IP address. Use the OLED menu function to check the IP Address. The factory-shipped default settings are:

	CPL410 LAN1	CPL410 LAN2	CPL410 LAN3
IP Address:	192.168.0.100	10.10.0.100	N/A
Subnet Mask:	255.255.255.0	255.255.255.0	N/A
Gateway:	0.0.0.0	0.0.0.0	N/A

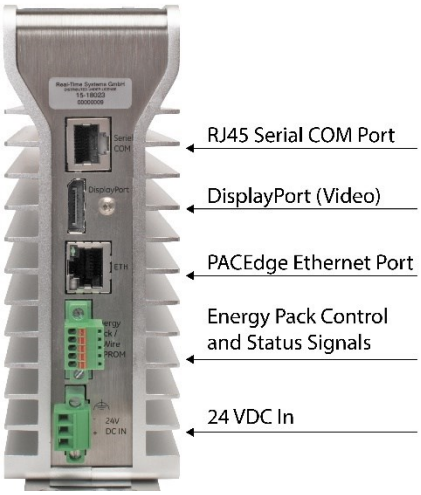
Serial COM Port

The RJ45 port, marked Serial COM, is located on the underside of the CPE400/CPL410, as shown in Figure 4. The serial port is exclusively assigned to the PLC. Its default settings are RS-232C, 115kBaud, 8 data bits, no parity, and 1 stop bit. For more general information on serial port usage, refer to Section 5.

This port supports Serial IO protocol.

Note that CPE400/CPL410 does not support PME connection via the serial port; use an Ethernet port

Figure 4: Underside Ports & Connectors



Video DisplayPort

The DisplayPort is located on the underside of the CPL410/CPE400, as shown in Figure 4. It provides signals for connecting either a suitable monitor or video adapter to the unit. **This port is not currently supported**.

PACEdge Ethernet Port

The RJ45 port, marked ETH, is located on the underside of the CPE400/CPL410, as shown in Figure 4. This Ethernet port is exclusively assigned to PACEdge. By default, this port is configured to use the DHCP protocol to receive a valid IP Address. Therefore, a DHCP server is needed to make first use of this port and to access PACEdge.

The GPOK LED, located on the front panel, indicates the status of the Linux interface. Green blinking indicates PACEdge is running and ready for login.

Energy Pack Connector

The CPE400/CPL410 compatible Energy Pack, IC695ACC403, is supplied with a purpose-built cable, IC695CBL003, which installs in the 24 Vdc In and Energy Pack Control & Status connectors. Use of the Energy Pack is optional. When used, it allows the RX3i PLC to save its current state upon loss of power. Refer to GFK-3000, *PACSystems RX3i Rackless Energy Pack IC695ACC403 Quick Start Guide,* for complete wiring and grounding instructions.

Note: Currently, there is no event to inform the PACEdge part of the CPL410 about a power loss. Therefore, PACEdge currently cannot take advantage of the Energy Pack.

Input Power Connector

Refer to GFK-3053, *RX3i IC695CPE400/IC695CPL410 1.2GHz 64MB Rackless CPU w/PACEdge Quick Start Guide*, Section 2.4.

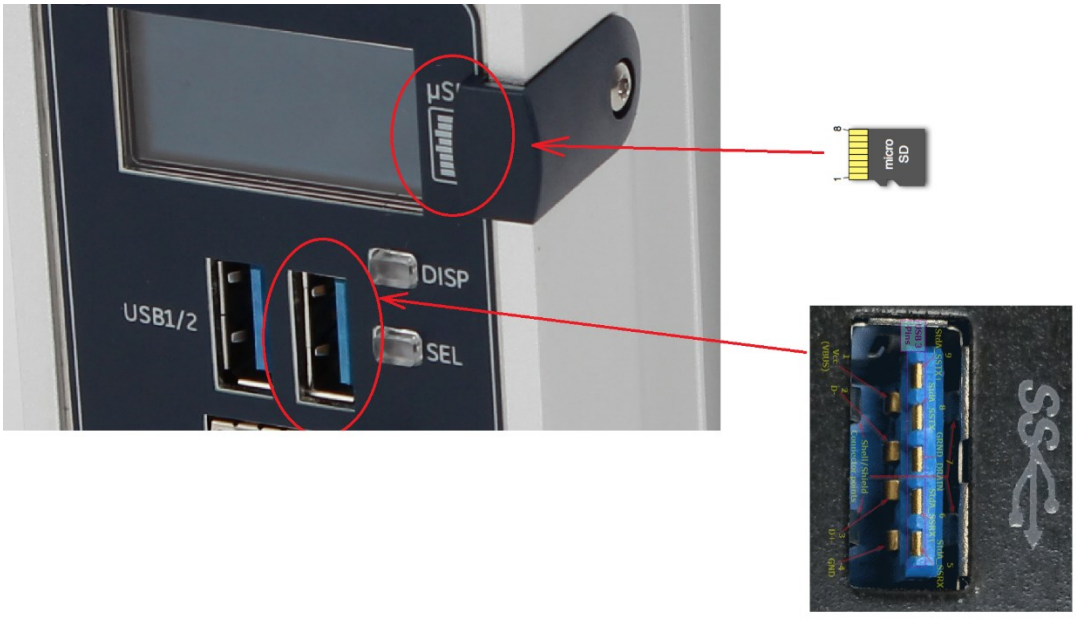
Micro-SD Card Interface

The Micro SD Card slot is located on the right-hand side of the equipment (Figure 5), level with the OLED display. This port is not functional at the time of publication. Once functional, it will support the Removable Data Storage Device features discussed in Section 2.1.8.

The interface supports SD, SDHC and SDXC µSD-Cards up to Version 3.0.

Insert the card into to slot, oriented as described below. Apply pressure until you feel some resistance. The card will latch into place and can then be read by the equipment. Apply pressure again to remove the card from the slot.

Figure 5: CPE400 Micro-SD & USB Connectors



Note:

The µSD-Card needs to be inserted in the slot with the correct orientation. The pins of the card need to face towards the front of the equipment (Figure 5).

- A cover and screw are provided. To minimize CPE400 susceptibility to electrical noise interference, keep the cover in place during normal operation.

Removable Data Storage Device

The CPL410/CPE400 is equipped with a micro-SD card slot. **Currently not supported.**

Replacement of Real-Time Clock Battery on CPE400/CPL410

Figure 6:Location of RTC Battery on CPE400/CPL410

The CPE400/CPL410 is shipped with a real-time clock (RTC) battery installed on an internal circuit board (Figure 6). This battery will need to be replaced periodically by a qualified service technician. Typically, no action is required during initial installation.

Should the RTC battery fail, the CPU date and time will be reset to 12:00 AM, 01-10-2000 at start-up. The CPU operates normally with a failed or missing RTC battery; however, the initial CPU time-of-day (TOD) clock information will be incorrect.

There are no diagnostics or indicators to monitor RTC battery status. The RTC battery has an estimated life of 5 years and must be replaced every 5 years on a preventative maintenance schedule.

To replace a depleted battery:

1. Power down the CPE400/CPL410.
2. Disconnect the external cables attached to the CPE400/CPL410, labeling each for later reconnection.
3. Remove the CPE400/CPL410 from its installed location.
4. Take the CPE400/CPL410 to a clean environment.
5. Remove the DIN-rail or panel-mount adaptor plate, as applicable.
6. Place the CPE400/CPL410 on a workbench so that the heat sink adjacent to the Emerson logo on the front panel is facing up.

7. With ESD protection in place, remove the four screws holding the upper side heat sink in place.
8. Remove the heat sink. This exposes the circuit board, connectors, and coin battery shown in Figure 6.
9. Take care to collect any thermal pads that may have been dislodged. These will be needed during reassembly.
10. While removing, or replacing the battery, take care not to damage the nearby ribbon cable (not shown).
11. Using non-conductive pliers, grip the battery and simultaneously hold back the retaining clip so it is clear of the battery.
12. Remove the depleted battery and dispose of it by an approved method.
13. Install the replacement battery so that the inscribed positive face is up.
14. Check that the retaining clip has engaged the edge of the newly installed battery.
15. Apply any dislodged thermal pads to the surface of the corresponding components on the circuit board.
16. Replace the heat sink.
17. Tighten all four retaining screws to 0.6 Nm.
18. Reattach the adaptor plate removed in step 5.
19. Restore the CPE400/CPL410 module to its original location and secure it in place.
20. Reconnect all cables to their original connectors.
21. Turn the power back on.
22. If needed, set the current date and time via PAC Machine Edition.

The replacement battery must be IC690ACC001 from Emerson, or an equivalent, such as Rayovac™ Lithium BR2032 Coin Cell 3V 190mAh -40°C to +85°C.

⚠ WARNING

The use of a different type of battery than that specified here may present a risk of fire or explosion.

The battery may explode if mistreated. Do not recharge, disassemble, heat above 100°C (212°F), or incinerate.

⚠ CAUTION

- To avoid damage from electrostatic discharge, use proper precautions when performing these procedures:
- Wear a properly functioning antistatic strap and be sure that you are fully grounded. Never touch the printed circuit board, or components on the board, unless you are wearing an antistatic strap.
- Any surface upon which you place the unprotected circuit board should be static-safe, facilitated by antistatic mats if possible.
- Extra caution should be taken in cold, dry weather when static charges can easily build up.

Setting the Real-Time Clock on CPE400/CPL410

The Real-Time Clock (RTRC) on the CPE400/CPL410 may be set by both the PACSystems Runtime and the Field Agent:

- The clock may be set from PACSystems using utilities in PAC Machine Edition.
- The clock may be set from PACEdge using the Cockpit The time source can be set manually or automatically using NTP servers.
- Regardless of whether the clock is set from the PACSystems Runtime or PACEdge, the CPE400/CPL410 must be rebooted after the clock is set so that the time change is applied across both applications.

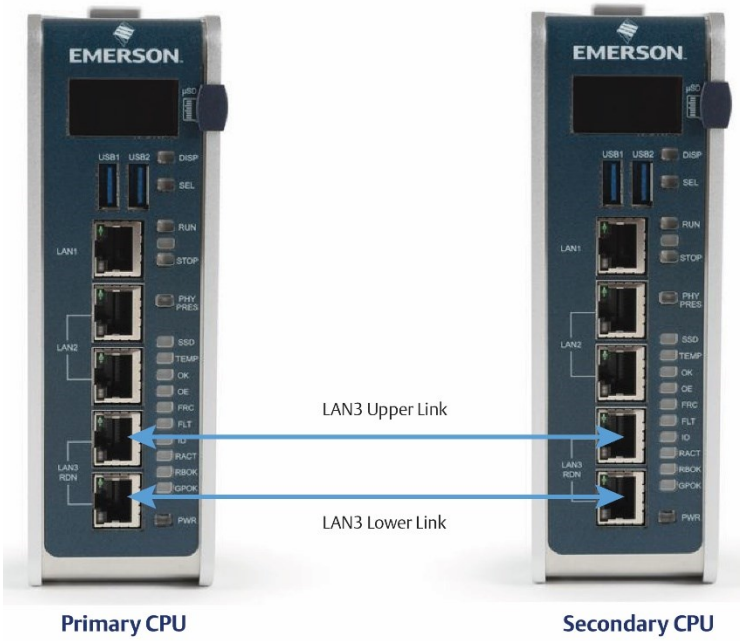
PROFINET Controller

An Embedded PROFINET Controller may be configured on LAN2. For additional details, refer to Section 2.1.6.

Hot Standby Redundancy

The CPE400/CPL410 can be configured as a Hot Standby Redundancy CPU with PROFINET IO. The two ports on LAN3 are used exclusively for this purpose: they provide a high-speed data synchronization link between the two CPUs. Connect the upper LAN3 port of the Primary CPU to the upper LAN3 port of the Secondary CPU and connect the lower LAN3 port of the Primary to the lower LAN3 port of the Secondary, as shown in Figure 7. Note that no additional hardware, other than the two redundant CPUs, may be connected to LAN3.

Figure 7: LAN3 Interconnects for Hot Standby Redundancy



To enable redundancy in a CPE400/CPL410 project, select the CPE400/CPL410 target in the *PME Navigator* and use the *Property Inspector* to change the *Enable Redundancy* target property to *True*.

Important: Set the *Background Window Timer* to a minimum of 5ms in both the Primary and Backup CPE400/CPL410 hardware configurations. The *Background Window Timer* setting may be found on the *Scan* Tab in the CPE400/CPL410's hardware configuration.

Once configured for HSB Redundancy, the RACT and RBOK LEDs become functional.

- RACT indicates the Local CPU is Ready & Active;
- RBOK indicates the Remote CPU is Ready.

These two LEDs are also reflected in the Status Data of the CPU and are presented as OPC UA Variables.

The OLED display includes two menu items used in conjunction with Redundancy:

RDN Info provides status information via the OLED display.

RDN Command permits the operator to perform a Role Switch.

To support Hot Standby operations, LAN2 is configured as a PROFINET IO Controller. For additional details, refer to section 2.1.6, Embedded PROFINET Controller.

For further details, refer to the *PACSystems Hot Standby CPU Redundancy User Manual*, GFK-2308 (rev L or later).

Redundant IP Addresses

The CPE400/CPL410 support two independent Redundant IP addresses, one for LAN1 and one for LAN2. LAN2 Redundant IP is supported when configured for Ethernet mode only.

Redundant IP is supported by the SRTP Server, Modbus TCP Server, and EGD protocols. It is not possible to use Redundant IP with the OPC UA Server or with the Ethernet firmware update web page.

For further details, refer to the *PACSystems Hot Standby CPU Redundancy User Manual*, GFK-2308 (rev L or later).

Error Checking and Correction

RX3i Redundancy CPUs provide error checking and correction (ECC), which results in slightly slower system performance, primarily during power-up, because it uses an extra 8 bits that must be initialized.

For details on ECC, refer to the PACSystems Hot Standby CPU Redundancy User Manual, GFK-2308.

Note: Multiple Recoverable Memory Error faults may be generated when a single-bit ECC error is detected. When a single-bit ECC error is detected, the value presented to the microprocessor is corrected. However, the value stored in RAM is not corrected until the next time the microprocessor writes to that RAM location.

Display Port

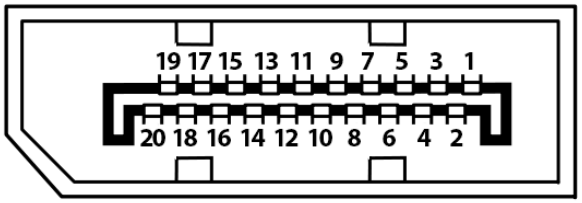
The Display port is not functional at the time of publication.

The Display Port is a DP++ video port located on the underside of the CPE440. It provides signals for connecting a suitable monitor or video adapter. Pinouts for the Display Port (Figure 8) are:

Pins						Signal Name
1			3			TxD0+/-
4			6			TxD1+/-
7			9			TxD2+/-
10			12			TxD3+/-
13						AUXSEL
15						CLK/AUX+
17						DAT/AUX-
18						HTPLG
14						NC
20						DP_VCC[49]
2	5	8	11	16	19	GND

[49] *DP_VCC is limited to 720mA by an electronic fuse. However, for normal operation do not exceed 500mA at this pin.*

Figure 8: Display Port Connector



Energy Pack Connector

The CPE400-compatible Energy Pack, IC695ACC403, is supplied with a purpose-built cable. Use of the Energy Pack is optional. Once charged up, the ACC403 allows the CPE400 to instantly save user memory to non-volatile storage in the event of loss of power. Upon restoration of power, with the ACC403 connected, the CPE400 PLC function can resume operations from the state saved at power-down.

Refer to the *PACSystems RX3i Rackless Energy Pack IC695ACC403 Quick Start Guide*, GFK-3000, for complete wiring and grounding instructions.

Input Power Connector

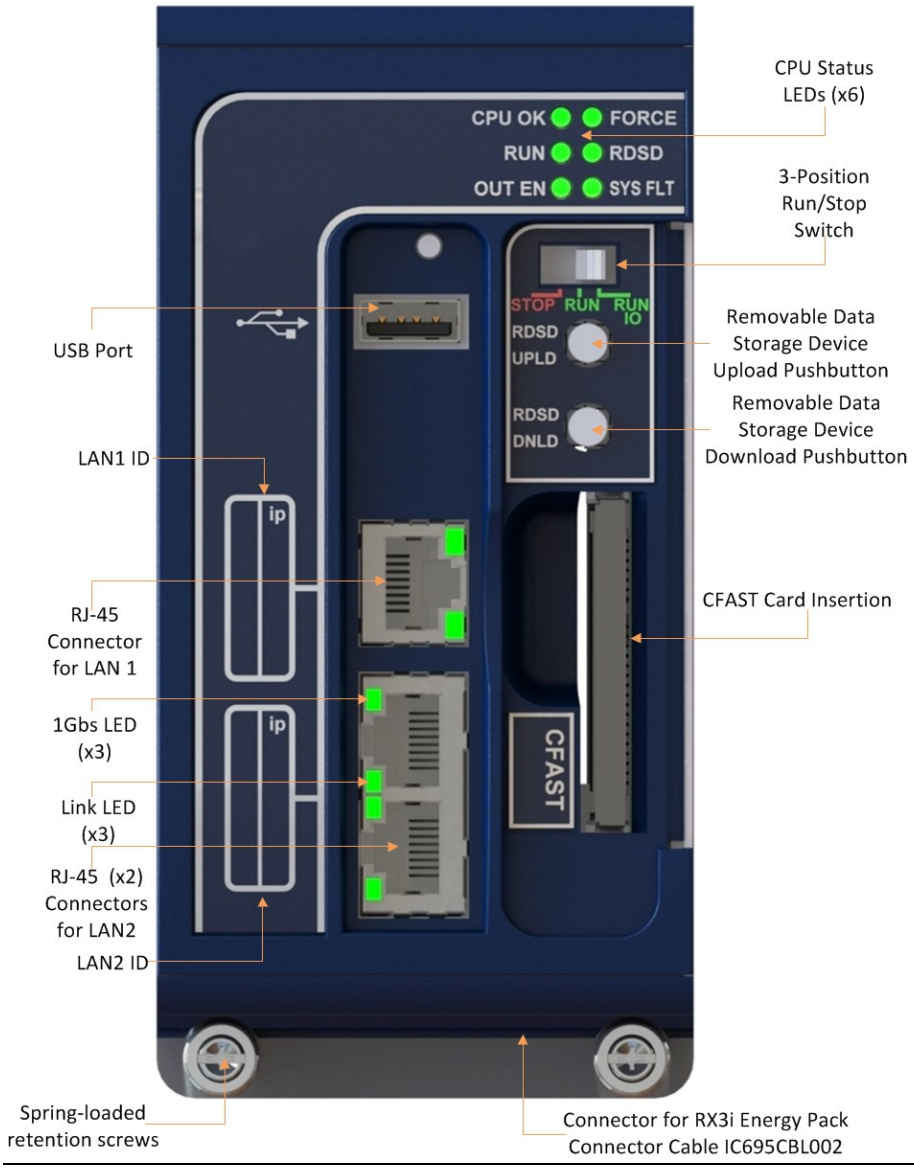
If no Energy Pack is to be connected, refer to Section 2.4 of the RX3i IC695CPE400/CPL410 Rackless CPUs w/PACEdge Quick Start Guide, GFK-3053. Otherwise, refer to the PACSystems RX3i Rackless Energy Pack IC695ACC403 Quick Start Guide, GFK-3000.

Backward Compatibility of CPE400/CPL410

In PAC Machine Edition, the CPE400/CPL410 is identified as a PLC within the *PACSystems RX3i Rackless* family. To convert an existing project which uses any other PLC, use the *Family Conversion* feature in PME. Be aware of the constraints involved. For instance, the first PROFINET Controller in an RX3i CPU320 application will be assigned to the Embedded PROFINET Controller feature (refer to section 2.1.6) of the CPE400.

2.2.2 CPE330

Figure 9: Front Display Port



Serial Ports CPE330

CPE330 is not equipped with a serial port. Use the embedded Ethernet ports for all communications with the CPU; use IC695CMM002 or IC695CMM004 modules for serial communications.

Ethernet Ports CPE330

CPE330 supports two independent 10/100/1000 Ethernet Local Area Networks (LANs).

- LAN1 connects to the uppermost RJ45 connector (Figure 9). It is not switched.
- LAN2 connects to the two lower RJ45 connectors. They are switched internally.

Space is provided beside each connector (Figure 9) to record the IP address used on each LAN.

Each of the embedded Ethernet interfaces automatically senses the data rate (10 Mbps or 100 Mbps or 1 Gbps), communications mode (half-duplex or full-duplex), and cabling arrangement (straight-through or crossover) of the attached link.

Any of the embedded Ethernet ports may be used to communicate with the PAC Machine Edition (PME) programming and configuration software using the Service Request Transport Protocol (SRTP).

For default, IP Address, and other details, refer to section 3.4.1, Establishing Initial Ethernet Communications.

Ethernet Network Configuration CPE330

The user must be careful when assigning IP Addresses and Subnet Masks for CPE330:

- Each LAN supports a unique IP Address
- LAN1 and LAN2 interfaces should not be configured for the same network.

By default, PME prohibits configuring both LAN interfaces on an overlapping IP subnet.

Care must also be taken when assigning IP Addresses and subnet masks to each LAN so that each network does not overlap any remote subnets in the network infrastructure:

- Subnets overlap with one another when the subnet portions of the IP Addresses are not unique
- Overlapping subnets may result in intermittent Ethernet communications or none at all. This would be due to packets being routed to the wrong LAN.
- Duplicate IP Addresses may also result in intermittent Ethernet communications or none at all. This is due to collisions on the LAN induced by two devices with the same IP Addresses communicating at the same time.

Ethernet Gateway Operation CPE330

The CPE330 allows the configuration of an Ethernet gateway on both LAN1 and LAN2. Since the CPE330 contains two LAN interfaces, each one supporting a unique IP Address, only one gateway is active at a time.

- Whenever a gateway is configured on only one of the two LAN interfaces and the other is not configured (0.0.0.0), the single gateway is shared by both interfaces;
- Whenever a gateway is configured on both LAN interfaces, the LAN1 gateway is given priority over the LAN2 gateway as long as LAN1 is functional. For example, in the event the LAN1 cable is disconnected, the CPE330 will use the LAN2 gateway as a backup.

PROFINET Controller

An Embedded PROFINET Controller may be configured on LAN2. For additional details, refer to Section 2.1.6 Embedded PROFINET Controller.

Sequence of Events

The CPE330 in combination with the IC695PNS101, IC694MDL655/IC694MDL660, and IC695HSC304/308 support a dedicated Sequence of Events monitoring system. Refer to PACSystems RX3i SoE User Manual, GFK-3050 for additional information.

Switches CPE330

Figure 10: CPE330 Run/Stop Switch and RDSD Switches

The RDSD and RUN/STOP Switches are located behind the protective door, as shown in Figure 11.

Refer to

RUN/STOP Switch Operation in Section 4.

The Reset pushbutton, located just above these switches, is currently not used.

RDSD Switch Operation CPE330

RDSD Pushbuttons	Function
RDSD UPLD	Loads user program or data from CPU to RDSD.
RDSD DNLD	Stores user program or data from RDSD to CPU.

Refer to Removable Data Storage Devices (RDSDs) for a full description of RDSD functionality.

Indicators CPE330

CPU Status Indicators

On Blinking Off

CPE330 LED	LED State	Operating State
CPU OK	On Green	CPU has passed its power-up diagnostics and is functioning properly. *(After the initialization sequence is complete.)*
	Off	Power is not applied or CPU has a problem, which may be indicated by the blink pattern.
	Blinking Other LEDs off	CPU in STOP-Halt state; possible watchdog timer fault. If PME cannot connect, cycle power with charged Energy Pack attached and refer to fault tables.
	On Amber	Whenever CPU OK and FORCE are both amber, CPU is indicating CPU320/CRU320 compatibility status. [50]
RUN OUT EN	Blinking in unison	CPU is updating an internal programmable hardware device.
RUN	On Green	CPU is in RUN Mode.
	Off	CPU is in STOP Mode.
OUT EN	On Green	Output scan is enabled.
	Off	Output scan is disabled.
FORCE	On Amber	Override is active on a bit of reference. Note: CPU OK is not simultaneously amber.[50]
	Off	No Overrides are active in I/O Reference Tables.
RDSD	On Green	USB or Cfast Device detected (No Activity)
	Blinking Green	Port activity detected on USB or Cfast Interface

[50] *Refer to.*
Communications

CPE330 LED	LED State	Operating State
	Off	No port activity was detected on USB or Cfast Interface
	On Red	RDSD Failure
	Blinking Red	Target name mismatch: Press the same RDSD pushbutton again to dismiss.
SYS FLT	On Red	CPU is in Stop/Faulted mode: a fatal fault has occurred.
	Off	No fatal faults were detected.

Ethernet Indicators CPE330 (embedded in RJ45 connectors)

On Blinking Off

LED	LED State	Operating State
LINK (upper)	On Green	The corresponding link is physically connected.
	Blinking Green	Traffic is detected at the corresponding port.
	Off	No connection was detected at the corresponding port.
1Gbps (lower)	On Amber (LAN1) or	The corresponding network data speed is 1 Gbps.
	On Green (LAN2)	
	Off	The corresponding network data speed is 100 Mbps or 10 Mbps.

Error Checking and Correction

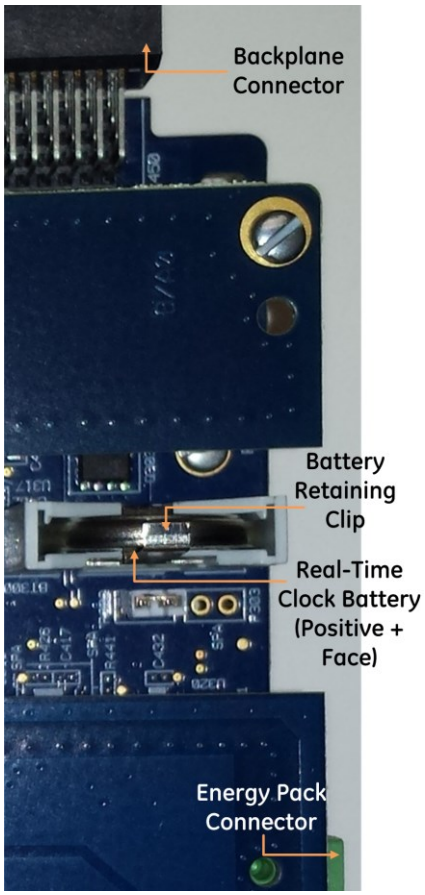
RX3i Redundancy CPUs provide error checking and correction (ECC), which results in slightly slower system performance, primarily during power-up, because it uses an extra 8 bits that must be initialized.

For details on ECC, refer to the PACSystems Hot Standby CPU Redundancy User Manual, GFK-2308.

Note: Multiple Recoverable Memory Error faults may be generated when a single-bit ECC error is detected. When a single-bit ECC error is detected, the value presented to the microprocessor is corrected. However, the value stored in RAM is not corrected until the next time the microprocessor writes to that RAM location.

Replacement of Real-Time Clock Battery on CPE330

Figure 11: Location and Orientation of Real-Time Clock Battery in CPE330



The CPE330 is shipped with a real-time clock (RTC) battery installed (see Figure 24). There is no isolation barrier between the battery and the circuit. This battery will need to be replaced periodically. Typically, no action is required during initial installation.

Should the RTC battery fail, the CPU date and time will be reset to 12:00 AM, 01-10-2000 at start-up. The CPU operates normally with a failed or missing RTC battery; however, the initial CPU time-of-day (TOD) clock information will be incorrect.

There are no diagnostics or indicators to monitor RTC battery status. The RTC battery has an estimated life of 5 years and must be replaced every 5 years on a preventative maintenance schedule.

To replace a depleted battery,

1. Power down the RX3i rack.
2. Disconnect cables attached to the CPE330 module, labeling each for later reconnection.
3. Remove the CPE330 module.
4. Take the CPE330 module to a clean environment.
5. Place the module on a workbench with the heat-sink side down.
6. With ESD protection in place, remove the four screws holding the upper side sheet metal in place.

7. Remove the sheet metal. This exposes the circuit board, connectors, and coin battery shown in Figure 24.
8. Using non-conductive pliers, grip the battery and simultaneously hold back the retaining clip so it is clear of the battery.
9. Remove the depleted battery and dispose of it by an approved method.
10. Install the replacement battery so that the inscribed positive face is towards the green connector (i.e. downwards as shown in Figure 24).
11. Check that the retaining clip has engaged the edge of the newly installed battery.
12. Replace the sheet metal cover.
13. Tighten all four retaining screws to 0.9Nm (8 in-lbs).
14. Restore the CPE330 module to its original location and secure it in place.
15. Reconnect all cables to their original connectors.
16. Turn the power on to the RX3i rack.

If needed, set the current date and time via PME or using SVC_REQ 7 (refer to PACSystems RX7i and RX3i CPU Programmer's Reference Manual, GFK-2950 Chapter 6).

Note: Battery replacement on CPE302/CPE305 & CPE310 is different: see Figure 29.

Replacement Real-Time Clock Battery

The replacement battery must be IC690ACC001 from Emerson, or an equivalent, such as Rayovac™ Lithium BR2032 Coin Cell 3V 190mAh -40°C to +85°C.

⚠ WARNING

The use of a different type of battery than that specified here may present a risk of fire or explosion.

The battery may explode if mistreated. Do not recharge, disassemble, heat above 100°C (212°F), or incinerate.

⚠ CAUTION

- To avoid damage from electrostatic discharge, use proper precautions when performing these procedures:
- Wear a properly functioning antistatic strap and be sure that you are fully grounded. Never touch the printed circuit board, or components on the board, unless you are wearing an antistatic strap.
- Any surface upon which you place the unprotected circuit board should be static-safe, facilitated by antistatic mats if possible.
- Extra caution should be taken in cold, dry weather when static charges can easily build up.

Backward Compatibility of CPE330 with CPU320, CRU320, or CPU315

The CPE330 may be interchanged with a corresponding CPU320/CRU320 with no upgrade to PAC Machine Edition (PME) software. Logic and configuration equality in PME is maintained when storing the same project to either a CPU320/CRU320 or a CPE330.

An *Extra Option Module* fault is logged in the Controller Fault Table on the CPE330's slot location after downloading a CPU320/CRU320 configuration to a CPE330. This indicates that the Embedded Ethernet interface did not receive a configuration. This fault is expected and does not interfere with normal controller operation.

Migration of CPU315 applications to the CPE330 is possible with no upgrade to PME by converting them to a CPU320 application and storing the project in the CPE330.

Versions of PME with native CPE330 support allow either a CPU320/CRU320 or a CPE330 configuration to be stored to the CPE330. When a CPE330 is configured as a CPU320/CRU320, Ethernet properties cannot be configured. However, the embedded Ethernet ports may be used with their previously configured IP Addresses or their default IP Addresses if the default IP Addresses were never changed.

Since CPE330 has no serial ports, any serial port activity associated with the previous CPU320/CRU320 application needs to be migrated to a suitable rack-based module (IC695CMM002 or IC695CMM004).

To download projects intended for the CPU320/CRU320 to a CPE330 with versions of PME that do not support the CPE330, you must change the CPE330's compatibility setting. To change the compatibility setting, perform the following operations:

1. Remove any USB stick that might be attached to the USB connector on the CPE330.
2. Place the RUN/STOP switch on the CPE330 in the STOP position.
3. Hold down the RDSD UPLD button and turn the power on to the CPE330. Continue to depress the RDSD UPLD button until the CPE330 powers up and displays one of the following patterns on the LEDs.

CPU Status Indicators

On Blinking Off

CPE330 LED	*LED State*	*Operating State*
CPU OK	On Amber	CPU320 Compatibility mode.
FORCE	On Amber	
RUN	Off	
RDSD	On Red	
OUT EN	Off	
SYS FLT	On Red	
CPU OK	On Amber	CRU320 Compatibility mode.
FORCE	On Amber	
RUN	On Red	
RDSD	Off	
OUT EN	Off	
SYS FLT	On Red	

1. To toggle the compatibility setting, press the RDSD DNLD button. The compatibility indication will toggle between the CPU320 compatibility and CRU320 compatibility patterns each time the RDSD DNLD button is pressed.
2. When the desired compatibility setting is displayed, press the RDSD UPLD button to save the setting and allow the CPE330 to continue its normal startup procedures with the new setting. The setting is maintained over a power cycle and firmware upgrade.

Note that with versions of PME that do not have native CPE330 support, only CPU320 projects can be stored to a CPE330 that is in CPU320 compatibility mode. Similarly, only CRU320 projects can be stored to a CPE330 that is in CRU320 compatibility mode.

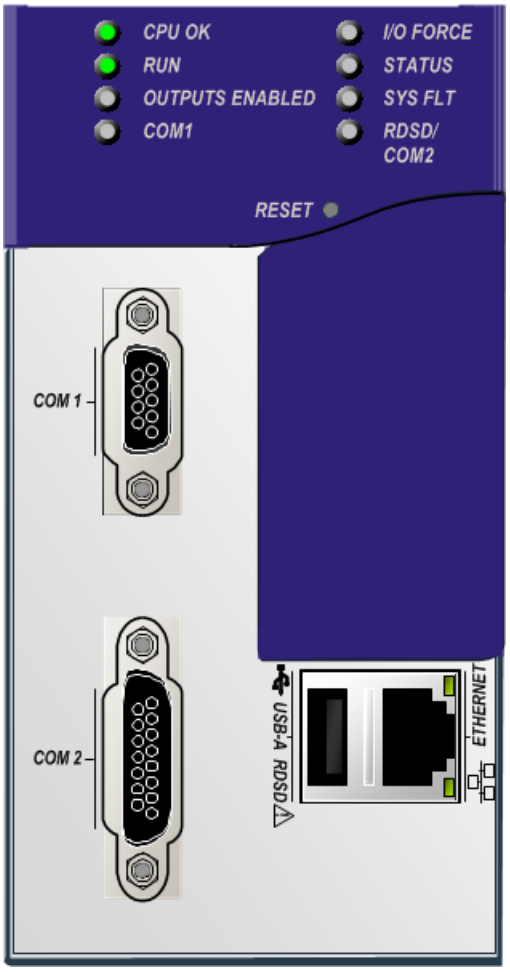
Users of a CPE330 with PME versions 8.60 SIM8 or later do not need to change this compatibility setting. PME versions 8.60 SIM8 or later allow for storing a CPU320 or CRU320 project without the need to change this setting. By factory default, the CPE330 identifies as a CPU320.

Note that CPE330s with firmware versions 8.45 through 8.60 support compatibility with the CPU320 and CPU315 only. The compatibility setting using the RDSD buttons (described above) is not supported for these firmware versions. Beginning with firmware version 8.70, CPE330s support compatibility with the CPU320, CPU315, and CRU320 using the RDSD buttons to set the compatibility setting.

2.2.3 CPE302[51], CPE305[51] and CPE310

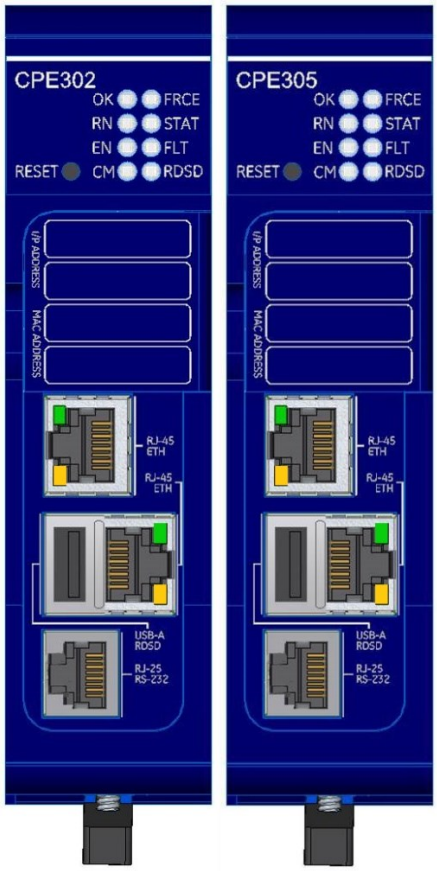
Figure 12: IC695CPE302/CPE305-Axxx Front View

Figure 13: IC695CPE310 Front View



[51] *Unless otherwise explicitly stated / differentiated all the statements are equally applicable to both the versions Axxx and Bxxx of these controllers.*

Figure 14: CPE302/305-Bxxx



Serial Ports

These ports provide serial interfaces to external devices. For serial port pin assignments, electrical isolation, and details on serial communications, refer to Section 5.

CPE302/CPE305: one RS-232 port (using RJ-25 connector).

CPE310: one RS-232 port (COM1) and one RS-485 port (COM2).

The RS-232 port does not supply the 5Vdc power offered by other RX3i and Series 90-30 CPUs.

Use cable IC693CBL316 to connect to the serial RJ-25 port on the CPE302/CPE305. This three-meter shielded cable provides a 9-pin D-connector on the other end.

Ethernet Port

For CPE302/305/310-Axxx, the embedded Ethernet interface provides one RJ45 Ethernet port that automatically senses the data rate (10 Mbps or 100 Mbps), communication mode (half-duplex or full-duplex), and cabling arrangement (straight-through or crossover) of the attached link.

For CPE302/305-Bxxx, the embedded Ethernet interface provides two Switched RJ45 Ethernet ports that automatically senses the data rate (10 Mbps or 100 Mbps or 1000 Mbps), communication mode (half-duplex or full-duplex), and cabling arrangement (straight-through or crossover) of the attached link.

The embedded Ethernet interface supports communications with the PAC Machine Edition (PME) programming and configuration software using the proprietary SRTP protocol. The CPE302/CPE305 /CPE310 CPUs provide two SRTP-server connections.

Refer to Section 3.4.1, Establishing Initial Ethernet Communications.

Switches CPE302/CPE305 & CPE310

The RDSD and RUN/STOP Switches are located behind the protective door, as shown in Figure 15 and Figure 16. Refer to

RUN/STOP Switch Operation in Section 4. The Reset pushbutton is not used.

Figure 15: External Features of CPE302/CPE305-Axxx

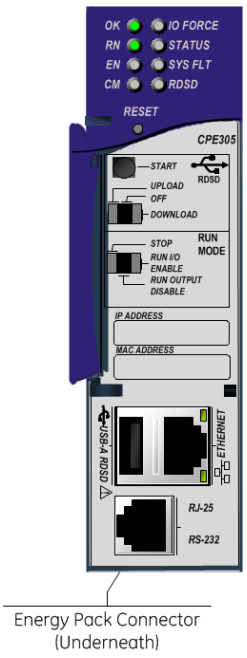


Figure 16: External Features of CPE310

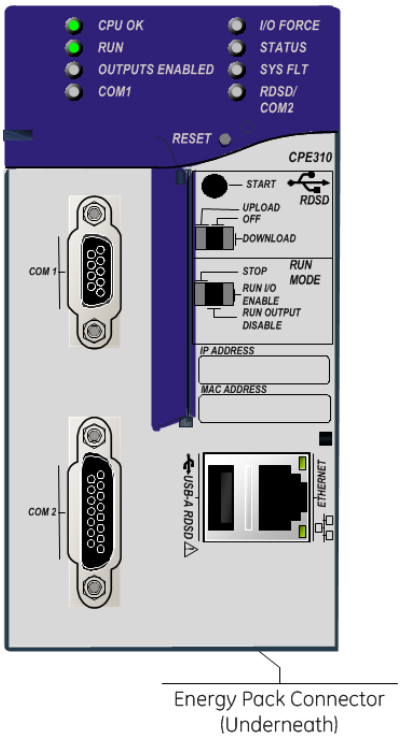
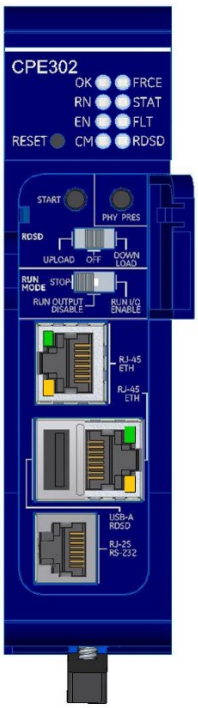


Figure 17: External Features of CPE302/CPE305-Bxxx



RDSD Switch Operation CPE302/CPE305 & CPE310

RDSD Switches	Function	
Start Pushbutton	Pressing this switch initiates RDSD data transfer. (The three-position switch must be or Download.)	
RDSD Switch	The RDSD Switch enables and disables RDSD data transfer and selects the direction	
	Upload	Loads application from CPU to RDSD.
	Off	Disables RDSD data transfer.
	Download	Stores application from RDSD to CPU.
PHY PRES Pushbutton	Used to reset the password on the firmware upgrade webpage.	
Run Mode Switch	Run I/O Enable*	The CPU runs with I/O sweep enabled.
	Run Output Disable*	The CPU runs with outputs disabled.
	Stop*	The CPU is not allowed to go into Run mode.

Indicators CPE302/CPE305 & CPE310

CPU Indicators

On Blinking Off

CPE302/ CPE305 LED	CPE310 LED	*LED State*	CPU Operating State
OK	CPU OK	On Green	CPU has passed its power-up diagnostics and is functioning properly. *(After the initialization sequence is complete.)*
		Off	CPU problem. RUN and OUTPUTS ENABLED LEDs may be blinking in an error code pattern, which can be used by technical support for troubleshooting. This condition and any error codes should be reported to your technical support representative.
		Blinking Other LEDs off	CPU in STOP-Halt state; possible watchdog timer fault. Refer to the fault tables. If PME cannot connect, cycle power with a charged Energy Pack attached and refer to fault tables.
OK EN	CPU OK OUTPUTS ENABLED	Blinking in unison	CPU is in boot mode and is waiting for a firmware update through a serial port.
RN	RUN	On Green	CPU is in RUN Mode.

CPE302/ CPE305 LED	CPE310 LED	*LED State*	CPU Operating State
		Off	CPU is in STOP Mode.
EN	OUTPUTS ENABLED	On Green	Output scan is enabled.
		Off	Output scan is disabled.
I/O FORCE (-Axxx) FRCE (-Bxxx)	I/O FORCE	On Amber	Override is active on a bit of reference.
STATUS (-Axxx) STAT (-Bxxx)	STATUS	Blinking Green	Energy Pack charging; not yet charged above the minimum operating voltage.
		On Red	Energy Pack circuit fault.
		Blinking Red	Energy Pack is near its end of life and should be replaced soon.
		On Green	Energy Pack is charged above its minimum operating voltage.
		Off	Energy Pack is not connected.
SYS FLT (-Axxx) (System Fault)	SYS FLT	On Red	CPU is in Stop/Faulted mode because a fatal fault has occurred.
CM	COM1	Blinking Green	Signals activity on serial port COM1.
		Off	No activity on serial port COM1.
N/A (-Axxx) RDSD (-Bxxx)	RDSD / COM2	Blinking Green	Signals activity on serial port COM2.
		Off	No activity on serial port COM2. (RDSD *not* attached)

RDSD Indicators CPE302/CPE305 & CPE310

On Blinking Off

CPE302/ CPE305 LED	CPE310 LED	LED State	RDSD Operating State
	SYS FLT	On Red	The RDSD has been removed during a store. The CPU must be power cycled to resume RDSD operations.
	RDSD / COM2	Off or Blinking Green	

CPE302/ CPE305 LED	CPE310 LED	LED State	RDSD Operating State
RDSD[52]	RDSD[52]/COM2	On Green	Valid RDSD connected or data transfer complete.
		Blinking Green	Data transfer is in progress.
		On Red	RDSD fault. Check for and correct the following conditions: • CPU type mismatch with the project on RDSD. • Data transfer error. • Corrupted or invalid USB file system. • Insufficient space on RDSD.
		Blinking Red	RDSD-Controller project name mismatch.
		Off	RDSD is not attached or the USB port is disabled.

Ethernet Indicators CPE302/CPE305 & CPE310

On Blinking Off

LED	LED State	CPU Operating State
100	On Green	The network data speed is 100 Mbps (-Axxx) or 1000 Mbps (-Bxxx).
	Off	The network data speed is 10 Mbps (-Axxx) or 10/100 Mbps (-Bxxx).
LINK	On Amber	The link is physically connected.
	Blinking Amber	Traffic is detected at the corresponding port.
	Off	No connection was detected.

[52] *RDSD active: RDSD attached to USB-A RDSD port.*

Error Checking and Correction

CPE302, CPE305, and CPE310 do not support error checking and correction (ECC).

Real-Time Clock Battery CPE302/CPE305-Axxx & CPE310

The CPE302, CPE305, and CPE310 are shipped with a real-time clock (RTC) battery (IC690ACC001) installed (Figure 18), and with an isolation barrier on the battery. Remove the isolation barrier via its pull tab before installing the CPE302, CPE305, or CPE310 module; otherwise, the battery will not function.

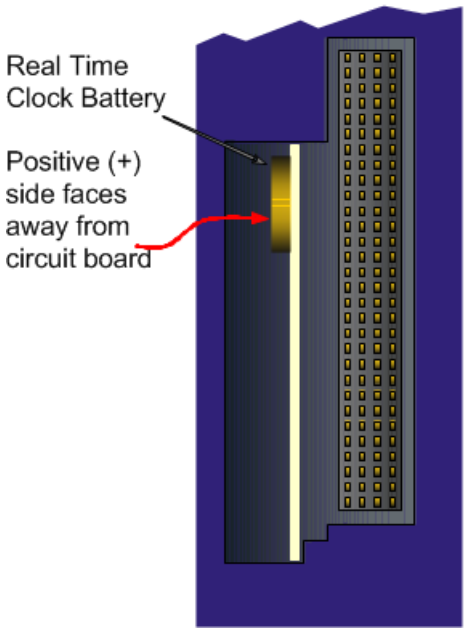
There are no diagnostics or indicators to monitor RTC battery status.

The RTC battery has an estimated life of 5 years and must be replaced every 5 years on a preventative maintenance schedule.

If the RTC battery fails, the CPU date and time are reset to 12:00 AM, 01-10-2000 at startup. The CPU operates normally with a failed or missing RTC battery; only the initial CPU TOD clock information will be incorrect.

Note: Battery replacement on CPE330 is different. Refer to Figure 12.

Figure 18: Accessing Real-Time Clock Battery (CPE302-Axxx, CPE305-Axxx, and CPE310



Replacing the Real-Time Clock Battery in CPE302/CPE305/CPE310

⚠ WARNING

The replacement battery must be IC690ACC001 from Emerson, or an equivalent, such as the use of a different type of battery than that specified here may present a risk of fire or explosion.

The battery may explode if mistreated. Do not recharge, disassemble, heat above 100°C (212°F), or incinerate.

⚠ CAUTION

- To avoid damage from electrostatic discharge, use proper precautions when performing these procedures:
- Wear a properly functioning antistatic strap and be sure that you are fully grounded. Never touch the printed circuit board, or components on the board, unless you are wearing an antistatic strap.
- Any surface upon which you place the unprotected circuit board should be static-safe, facilitated by antistatic mats if possible.
- Extra caution should be taken in cold, dry weather when static charges can easily build up.

Important

The following battery removal methods are for the CPE302/305-Axxx, CEP302/305-Bxxx and CPE330.

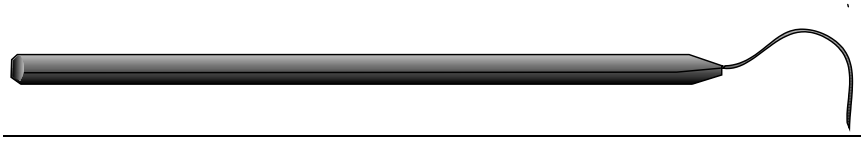
Battery Removal Method 1 for CPE302/305-Axxx and CPE330

1. Power down the rack and remove the CPU from the backplane.
2. Using a curved probe with a non-conducting surface, for example, a non-metallic dental pick, reach in from the back of the module and pull the battery out of its retaining clip. (You can use needle-nose pliers to grasp the battery and pull it the rest of the way out.)

Battery Removal Method 2

1. Power down the rack and remove the CPU from the backplane.
2. Squeeze both sides of the module and remove the front section of the plastic housing.
1. Lift the two clips on the side of the plastic housing to release the circuit board and pull the board out of the housing.
2. Pull the battery out of its retaining clip.

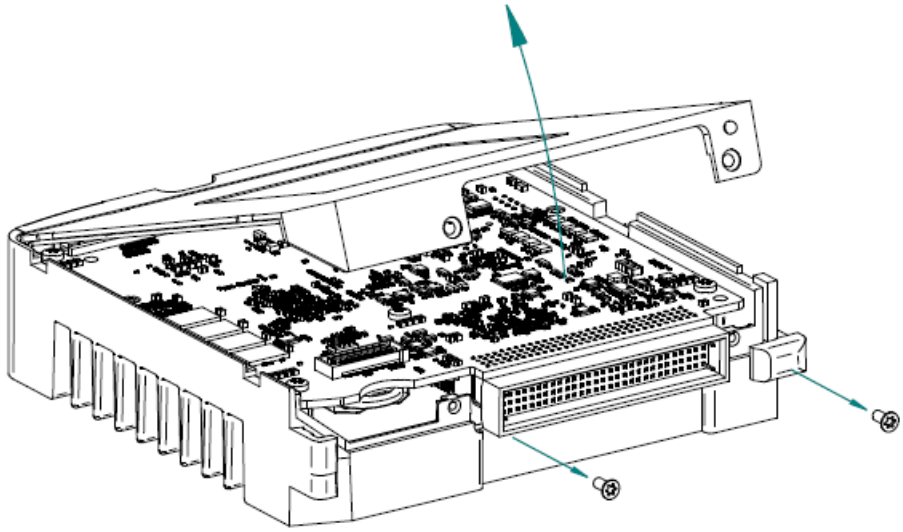
Figure 19: Sample Tool for Battery Removal



Battery Removal Method for CPE302/305-Bxxx

1. Power down the RX3i rack.
2. Disconnect the cables attached to the CPE302/305-Bxxx module, labeling each for later reconnection.
3. Remove the CPE302/305-Bxxx module from the RX3i rack.
4. Take the CPE302/305-Bxxx module to a clean environment.
5. Place the module on a workbench with the heat-sink side down.
6. With ESD protection in place and using a Torx-8 screwdriver, remove the two screws on the backside of the module as shown (Figure 20).

Figure 20: Remove Screws Before Opening Cover

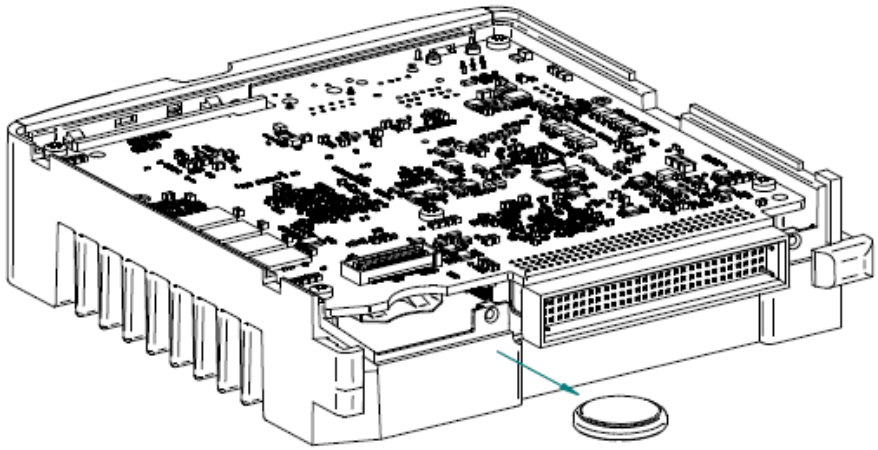


7. Open the module's cover to expose the circuit board and coin battery.

With non-conductive pliers, remove the battery (

8. Figure 21).
9. Dispose of the battery by an approved method.
10. Insert the new battery and ensure that inscribed positive face (+ve) faces away from the circuit board.
11. Replace and close the cover and tighten the screw to a torque of 0.6 Nm (5.3 lb-in).

Figure 21: Use Tweezers to Remove Battery.



12. Reconnect all the cables to their original connectors.
13. Restore power to the RX3i rack.

Installing a New RTC Battery

Install the battery with the positive (+) side up. That is, with the + side away from the board and toward the housing plastic.

Backward Compatibility of CPE310 with CPU310

The CPE310 may be swapped with a CPU310 with no upgrade to the PAC Machine Edition Logic Developer-PLC programming software. Logic and configuration equality in the programming software is maintained when storing the same project to either a CPU310 or a CPE310. PAC Machine Edition versions that recognize the CPE310 (7.0 SIM3 and newer), allow either a CPU310 configuration or a CPE310 configuration to be stored to the CPE310. For all programming software versions (both current and legacy) a CPU310 device can accept only a CPU310 configuration.

Legacy CPU310 Projects

The CPE310 supports CPU310 projects. PAC Machine Edition versions earlier than 7.10 SIM 3 interpret the CPE310 as a CPU310. The CPE310 can be configured as a CPU310 using PAC Machine Edition versions as old as 5.5, Service Pack 1.

RDSD Port

If a CPU310 configuration is stored to a CPE310, the RDSD port is enabled to allow you to transfer CPU310 projects to CPE310 models without using PAC Machine Edition.

Fault Behavior

Faults related to the embedded CPE310 Ethernet interface may be generated on power-up, as detailed in the following section.

Replacing a CPU310 with a CPE310

- A CPE310 that is configured as a CPU310 logs the following faults in the Controller fault table:
    - *A LAN Transceiver Fault* is generated because the RX3i system detects that the embedded Ethernet module does not have a network connection.
    - An *Extra Option Module* fault is generated because the embedded Ethernet module is detected as an unconfigured module.
    - If the Energy Pack capacitor pack is disconnected or fails, the legacy faults for a missing or failed battery are logged.
- When a CPE310 is configured as a CPU310, Ethernet properties cannot be configured and there should be no cable connected to the Ethernet port.
- When a CPE310 is configured as a CPU310, the Show Status dialog box in PAC Machine Edition displays *CPU310A*.

CPE310 versus CPU310 Performance Differences

The following differences should be considered when converting legacy applications or developing new applications.

- Some exceptionally lengthy CPE backplane operations, such as MC_CamTableSelect, Data Log, and Read Event Queue functions, will take longer to complete compared to other RX3i CPU models, and may delay backplane operations to IC695 modules.
- For example, when an MC_CamTableSelect function block is executed on the PMM335 module, the CPU's acknowledgment of the PMM355 module interrupt may be delayed. In this situation, you may see the following fault in the I/O Fault Table, even when the interrupt has not been dropped: *Error initiating an interrupt to the CPU.*
- Performance specifications for many features, such as power-up time, function block execution times, and I/O module sweep times have changed. For details, refer to Appendix A
- The RS-232 port on the CPE310 does not provide 5Vdc power on pin 5.

CPU305 Performance Differences vs. CPE310 and Legacy RX3i CPUs

The CPE302/CPE305 exhibits the same performance differences as listed above for the CPE310.

The CPE305 supports legacy CPU310 projects that fit within 5 MB[53] of user memory. The CPE302 supports legacy CPU310 projects that fit within 2 MB of user memory. ***The project configuration must be changed to support any such conversion***.

Because the CPE302/CPE305 has less user memory than the other RX3i CPUs, operations that involve transferring large files could fail.

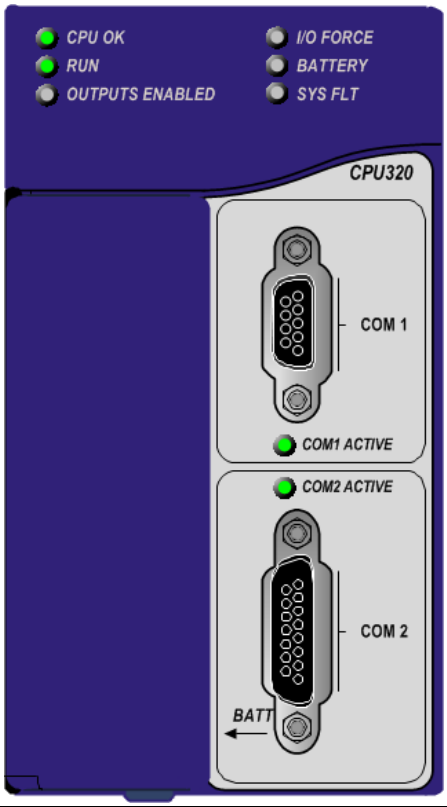
For example, depending on the number and sizes of Data Log files already stored, the Get_DL (Get Data Log) command could fail with a C10 hex (file transfer failure occurred while sending the data log file to the CPU) error. To correct this error

1. Upload the data logs to Machine Edition and delete the logs from the CPU.
2. Take steps to reduce the size of the log file, such as reducing the number of samples, the sample rate, or the number of parameters logged.

[53] *The CPE305-Bxxx has 6 MB of user memory.*

2.2.4 CPU315 and CPU320/CRU320

Figure 22: IC695CPU320 Front View



Serial Ports CPU315, CPU320 & CRU320

Each CPU has two independent, onboard serial ports, accessed by connectors on the front of the module. COM1 and COM2 provide serial interfaces to external devices. Either port can be used for firmware upgrades. For serial port pin assignments, electrical isolation, and details on serial communications, refer to Section 5.

Indicators CPU315, CPU320 & CRU320

Eight CPU LEDs indicate the operating status of various CPU functions.
Two Comm LEDs indicate activity on COM1 and COM2.

LED State ● On ✱ Blinking ○ Off		CPU Operating State
CPU OK	● On Green	CPU has passed its power-up diagnostics and is functioning properly. [54]
	○ Off	CPU problem. RUN and OUTPUTS ENABLED LEDs may be blinking in an error code pattern, which can be used by technical support for troubleshooting. This condition and any error codes should be reported to your technical support representative.
	✱ Blinking Green Other LEDs off	CPU in Stop/Halt state; possible watchdog timer fault. Refer to the fault tables. If PME cannot connect, cycle power with the battery attached and refer to fault tables.
CPU OK RUN OUTPUS ENABLED	✱ ✱ Blinking in unison ✱	CPU is in boot mode and is waiting for a firmware update through a serial port.
RUN	● On Green	CPU is in RUN Mode.
	○ Off	CPU is in STOP Mode.
OUTPUTS ENABLED	● On Green	Output scan is enabled.
	○ Off	Output scan is disabled.
I/O FORCE	● On Yellow	Override is active on a bit of reference.
BATTERY	○ Off	Normal battery[55]
	✱ Blinking Red	Battery low[55]
	● On Red	The battery has failed or is not attached[3]
SYSTEM FAULT	● On Red	CPU is in Stop/Faulted mode because a fatal fault has occurred.

[54] *After initialization sequence is complete.*

[55] *Low battery detection requires hardware revision –Fx or later and a smart battery. For details, refer to the PACSystems Battery and Energy Pack Manual, GFK-2741.*

LED State On Blinking Off		CPU Operating State
COM1 COM2	Blinking Green	Signals activity on a corresponding serial port.

Error Checking and Correction, IC695CRU320

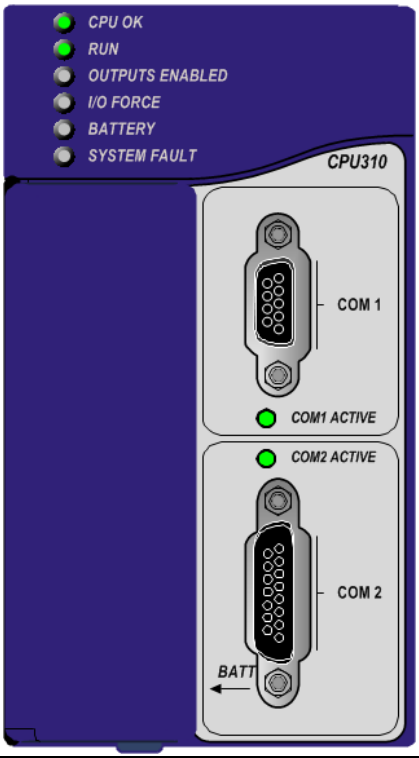
RX3i Redundancy CPUs provide error checking and correction (ECC), which results in slightly slower system performance, primarily during power-up, because it uses an extra 8 bits that must be initialized.

For details on ECC, refer to the PACSystems Hot Standby CPU Redundancy User Manual, GFK-2308.

Note: Multiple Recoverable Memory Error faults may be generated when a single-bit ECC error is detected. When a single-bit ECC error is detected, the value presented to the microprocessor is corrected. However, the value stored in RAM is not corrected until the next time the microprocessor writes to that RAM location.

CPU310

Figure 23: CPU310 Front View



Serial PortsCPU310

The CPU has two independent, onboard serial ports, accessed by connectors on the front of the module. COM1 and COM2 provide serial interfaces to external devices. Either port can be used for firmware upgrades. For serial port pin assignments and other details on serial communications, refer to Section 5.

The eight CPU LEDs indicate the operating status of various CPU functions.
The two Comm LEDs indicate activity on COM1 and COM2.

LED State ● On ✱ Blinking ○ Off		CPU Operating State
CPU OK	● On Green	CPU has passed its power-up diagnostics and is functioning properly.[56]
	○ Off	CPU problem. RUN and OUTPUTS ENABLED LEDs may be blinking in an error code pattern, which can be used by technical support for troubleshooting. This condition and any error codes should be reported to your technical support representative.
	✱ Blinking Green Other LEDs off	CPU in Stop/Halt state; possible watchdog timer fault. Refer to the fault tables. If PME cannot connect, cycle power with a battery attached and refer to fault tables.
CPU OK RUN OUTPUTS ENABLED	✱ Blinking in unison	CPU is in boot mode and is waiting for a firmware update through a serial port.
RUN	● On Green	CPU is in RUN Mode.
	○ Off	CPU is in STOP Mode.
OUTPUTS ENABLED	● On Green	Output scan is enabled.
	○ Off	Output scan is disabled.
I/O FORCE	● On Yellow	Override is active on a bit of reference.
BATTERY	○ Off	Normal battery[57]
	✱ Blinking Red	Battery low[57]
	● On Red	The battery has failed or is not attached[57]
SYSTEM FAULT	● On Red	CPU is in Stop/Faulted mode because a fatal fault has occurred.
COM1 COM2	✱ Blinking Green	Signals activity on a corresponding serial port.

[56] *After initialization sequence is complete.*
[57] *Low battery detection requires a smart battery. For details, refer to PACSystems Battery and Energy Pack Manual, GFK-2741.*

2.3 RSTi-EP CPU Features and Specifications

	EPSCPE100/CPE115	EPXCPE205	EPXCPE210/215/220	EPXCPE240
Lifecycle Phase	Active	Active	Active	Active
Backplane	Standalone	Y	Y	Y
Temperature Range				
RSTi-EP	-40 °C to 70 °C	-40 °C to 70 °C	-40 °C to 70 °C	-40 °C to 70 °C
Power Requirements				
RSTi-EP +3.3Vdc	N/A	N/A	N/A	N/A
RSTi-EP +5 Vdc	N/A	N/A	N/A	N/A
RSTi-EP +24Vdc Relay with Energy Pack	N/A	N/A	N/A	N/A
Input Power (Max)	6 W (250mA@ 24Vdc)	480 W (2 Rails, 10A @ 24Vdc)	480 W (2 Rails, 10A @ 24Vdc)	480 W (2 Rails, 10A @ 24Vdc)
Input Voltage (Min)	9Vdc	18Vdc	18Vdc	18Vdc
Input Voltage (Max)	30Vdc	30Vdc	30Vdc	30Vdc
Memory Backup Mechanism[58]	Internal super capacitor	NVRAM	NVRAM	NVRAM
Display				
	LEDs	LEDs	LEDs	LEDs
Firmware Upgrade				
CPU Firmware Upgrade Mechanism	Web Interface Ethernet Port	Web Interface Ethernet Port	Web Interface Ethernet Port	Web Interface Ethernet Port
Indirect Backplane Module Upgrade	N	Y[59]	Y[59]	Y[59]
Program Portability				
Direct Import (with limitations)[60]	N/A	N/A	N/A	N/A
RX3i PACSystems Applications using Family Type Conversion	Y	Y	Y	Y
Program Security				
Secure Boot	Y	Y	Y	Y
Trusted Platform Module (TPM)	Y (Disabled)	Y (Disabled)	Y (Disabled)	Y (Disabled)

[58] *See Battery Compatibility and Memory Retention (Time in Days at 20°C) in GFK-2741*
[59] *EPXCPE2XX controller can be updated in the backplane using the web interface*
[60] *See corresponding IPI for target CPU.*

	EPSCPE100/CPE115	EPXCPE205	EPXCPE210/215/220	EPXCPE240
Program Storage				
Battery-backed RAM	1 Mb[61]	448 Kb	CPE210: 1 Mb CPE215: 1.5 Mb CPE220: 2 Mb	4 Mb
Non-Volatile Flash	512 M	512 Kb	CPE210: 1 Mb CPE215: 1.5 Mb CPE220: 2 Mb	4 Mb
Battery Life Expectancy, RAM Backup[58]	N/A	N/A	N/A	N/A
Life Expectancy, Energy Pack Capacitors	15 Years if the ambient temp is 40°C	N/A	N/A	N/A
Auxiliary Storage				
CFast	N/A	N/A	N/A	N/A
Remote Data Storage Device (RDSD)	N/A	Y[62]	Y[60]	Y[60]
Micro SD	x 1 (Disabled)	Y	Y	Y
Programming Capabilities				
Max Number of Program Blocks	512	768	768	768
Program Block Max Size	128 KB	128 KB	128 KB	128 KB
Discrete Reference Memory (%I, %Q)[63]	2K Bits	1K Bits	CPE210/215: 2K Bits CPE220: 4K Bits	4K Bits
Analog Reference Memory (%AI, %AQ)[63]	32K Words	16K Words	32K Words	32K Words
Bulk Reference Memory (%W)[63]	up to max user RAM	229K Words	CPE210: 492K Words CPE215: 754K Words CPE220: 1,016K Words	2,064K Words

[61] *CPE200 does not use battery-backed RAM. It uses NonVolatile RAM (NVAM).*

[62] *MicroSD not supported for RDSD.*

[63] *Note: Whenever the size of any reference memory is changed, the content of the corresponding reference memory is automatically cleared.*

	EPSCPE100/CPE115	EPXCPE205	EPXCPE210/215/220	EPXCPE240
Managed Memory (Symbolic + I/O Variables)[63,64]	up to 1 MB	Discrete: 3,584K Bits Non-Discrete: 224 K Words	CPE210: Discrete: 7,680K Bits Non-Discrete: 480 K Words CPE215: Discrete: 11,776 K Bits Non-Discrete: 736K Words CPE220: Discrete: 15,872 K Bits Non-Discrete: 992K Words	Discrete: 32,256 K Bits Non-Discrete: 2,016K Words
Floating Point	Y	Y	Y	Y
Ladder Diagram (LD)	Y	Y	Y	Y
Function Block Diagram (FBD)	Y	Y	Y	Y
Structured Text (ST)	Y	Y	Y	Y
PID Built-In Function Block	Y	Y	Y	Y
"C" Language External Blocks	Y	Y	Y	Y
Communications				
Ethernet Non-Switched RJ45 (dedicated NIC)	10/100 x1	1-2-port configurable as 2 NICs or 1 NIC switch 100/1000	100/1000 x1	100/1000 x1
Ethernet Switched RJ45 (shared NIC)	10/100 x3	100/1000 x2	100/1000 x2	100/x1000 x2
10BaseT/100BaseT RJ45	Y	N	N	N
Ethernet Communications Platform	Built-in	Built-in	Built-in	Built-in
Advanced User Parameters (AUP file)	N[65]	N/A	N/A	N/A
RS-232	x 1	x 1	x 1	x 1
RS-485	x 1	N/A	N/A	N/A
USB	USB 2.0 x 1 (Disabled)	USB 2.0 x 2	USB 2.0 x 2	USB 2.0 x 2

[64] For discussion of memory types and how they are managed, refer to PACSystems RX3i and RSTi-EP CPU Programmer's Reference Manual, GFK-2950 Section 3.
[65] The Advanced User Parameters (AUP) feature has been incorporated into PME Hardware Configuration (HWC) effective with PME release 8.60 SIM5.

	EPSCPE100/CPE115	EPXCPE205	EPXCPE210/215/220	EPXCPE240
Time-of-Day Clock				
Time-of-Day Clock Accuracy (@60°C)	±2 secs/day	±3 secs/day	±3 secs/day	±3 secs/day
Elapsed Time Clock (internal timing) accuracy	±0.01% max	±0.01% max	±0.01% max	±0.01% max
Simple Network Time Protocol (SNTP) accuracy to timestamp	CPE100: N CPE115: Y	±2 ms embedded only	±2 ms embedded only	±2 ms embedded only
RTC Battery Backup	Y	Y	Y	Y
RTC Battery Life expectancy (at room temp)	10 years	10 years	10 years	10 years
Protocols				
Modbus RTU Slave	Y	Y	Y	Y
SNP Slave	N/A	N/A	N/A	N/A
Serial I/O	Y	Y	Y	Y
SRTP (# simultaneous server conns)	up to 16	up to 16	up to 16	up to 16
Modbus TCP (# simultaneous server connections)	up to 8	up to 16	up to 16	up to 16
SRTP Channel or Modbus TCP Client (# simultaneous)	up to 8	16 (Shared with SRTP)	16 (Shared with SRTP)	16 (Shared with SRTP)
Ethernet Global Data (EGD)	Y	Y	Y	Y
Number of EGD Exchanges (max) [66]	8	32	32	32
Selective Consumption of EGD	N/A	N/A	N/A	N/A
PROFINET[67]	Up to 8 devices	Up to 8 devices	CPE210/215: Up to 16 devices CPE240: Up to 32 devices	Up to 32 devices
OPC UA Server	Y	Y	Y	Y
Remote Station Manager over UDP	Y (limited)	N/A	N/A	N/A
Station Manager over Serial Comm Port	N/A	N/A	N/A	N/A
DNP3 Outstation master/client support[50] (# simultaneous)	CPE100: N CPE115: Up to 8	N/A	CPE210: N/A	Up to 8
			CPE215: Up to 8	
			CPE220: Up to 8	

[66] *Limit is per target, so all producers and consumers are counted towards this limit.*

[67] *CPE100/CPE115(firmware version 9.30 or later) provide PROFINET support with MRP via an embedded PROFINET Controller: no external hardware is required.*

	EPSCPE100/CPE115	EPXCPE205	EPXCPE210/215/220	EPXCPE240
Memory Error Checking and Correction (ECC)	N/A	N/A	N/A	N/A
Redundancy Features				
Switchover Time (max)[68]	N/A	N/A	N/A	N/A
Switchover Time (min)[68]	N/A	N/A	N/A	N/A
Max data in the redundancy transfer list[69]	N/A	N/A	N/A	N/A
Redundant Synchronized Links Supported	N/A	N/A	N/A	N/A

[68] *Switchover time is defined as the time from failure detection until backup CPU is active in a redundancy system.*
[69] *Symbolic variable and Reference data can be exchanged between redundancy controllers, up to the stipulated limit.*

2.3.1 CPE100/CPE115

Introduction

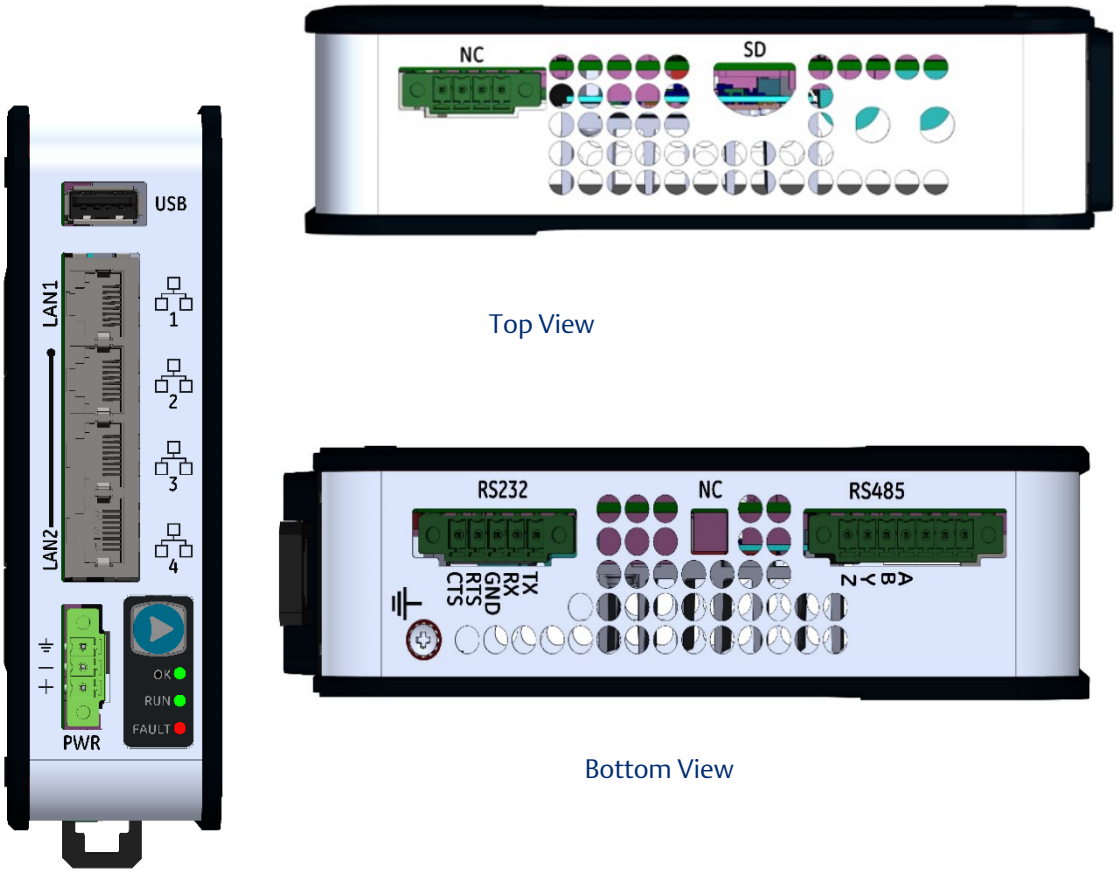
The EPSCPE100 and EPSCPE115 are the first standalone CPUs in the RSTi-EP family. Each is supported by two mounting options:

1. As shipped, it mounts onto a DIN rail using a DIN-rail adaptor plate.
2. Alternately, it mounts directly in a cabinet, using a panel-mount adaptor plate ICMFAACC001-AA.

The mounting instructions and power requirements are documented in the *Quick Start Guide*, GFK-3012, and are not replicated here.

The physical features of the CPE100/CPE115 are shown in Figure 24.

Figure 24: CPE100, Front, Top, and Bottom Views and Features



The PACSystems RSTi-EP EPSCPE100/CPE115 are enhanced performance standalone programmable controllers equipped with 1MB of user memory and four Ethernet ports to run real-time deterministic control applications. LAN1 is dedicated to high-speed Ethernet and LAN2 is comprised of three

switched ports configurable as either a second embedded Ethernet controller or an embedded PROFINET controller, which provides the PROFINET functionality and only supports a simplex mode of operation. It is a standalone PLC that supports distributed I/O.

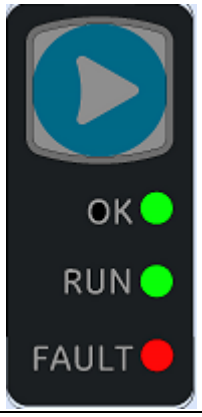
The CPE100/CPE115 are programmed and configured over Ethernet via Emerson's PAC Machine Edition (PME) software. Each is a standalone CPU with the following features:

- A built-in PACSystems RSTi-EP PLC CPU
  - Users may program in Ladder Diagram, Structured Text, or Function Block Diagram.
  - Contains 1 MB for CPE100 and 1.5MB for CPE115 of configurable data and program memory. This entire memory will be preserved between power cycles.
  - Supports auto-located Symbolic Variables that can use any amount of user memory.
  - Reference table sizes include 2k bits for discrete %I and %Q and up to 32k words each for analog %AI and %AQ. Bulk memory (%W) is also supported for data exchanges.
  - Supports up to 512 program blocks. The maximum block size is 128KB.
- Supports two independent 10/100 Ethernet LANs. LAN1 has only one port and is dedicated to high-speed Ethernet; whereas, LAN2 is comprised of three switched ports (labeled as 2, 3 & 4) which are configurable as either a second embedded Ethernet controller or an embedded PROFINET controller. All four ports are located on the front panel.
- The embedded communications interface has dedicated processing capability, which permits the CPU to independently support LAN1 and LAN2 with:
  - up to 16 combined SRTP Server and Modbus TCP Server connections out of which:
    - Modbus TCP cannot exceed more than 8 simultaneous connections.
    - SRTP server cannot exceed more than 16 simultaneous connections.
  - Up to 8 Clients are permitted. Each may be SRTP or Modbus TCP or a Combination of both.
  - Up to 8 simultaneous Class 1 Ethernet Global Data (EGD) exchanges.
- Ability to display the serial number and date code in PME Device Information Details.
- Media Redundancy Protocol (MRP) allows the CPE100/CPE115 to participate in a PROFINET I/O network with MRP ring technology. This eliminates the I/O network as a single point of failure. The CPE100/CPE115 may be used as either a Media Redundancy Manager or Media Redundancy Client.
- OPC UA Server supports up to two concurrent sessions with up to 4 concurrent variable subscriptions and up to 1000 variables.
- Modbus RTU Slave support on two serial ports i.e. RS-232 and RS-485 with both 2-wire and 4-wire interfaces. These ports are located on the underside of the controller and do not provide any type of isolation.
- CPE115 supports DNP3 outstation up to 8 concurrent master connections.
- Operating temperature range -40 °C to 70 °C (-40 °F to 158 °F).
- Supports 32-bit C blocks compiled with the C Toolkit Version 8.10 or later. All pre-existing C blocks must be recompiled before downloading.
- Supports Authorized Firmware Update feature. Users may now authorize access to firmware updates using a custom password. Details are included in the revised firmware update instructions.

- The PLC may use one, two, or three of the Ethernet ports of LAN2 to support the embedded Simplex PROFINET I/O Controller. PROFINET supports up to 8 I/O devices with update rates of 16 – 512 ms. It is not recommended to use update rates below 16 ms.
- The CPE100/CPE115 is secure by design, incorporating technologies such as secure boot, a trusted platform module (disabled), and encrypted firmware updates.
- Module LEDs on the faceplate provide basic status and control information of CPE100/CPE115.
- When shipped, the CPE100/CPE115 is configured only for DIN-rail mounting. An alternate panel-mount adaptor plate (ICMFAACC001-AA) is optional, but not included in the ship-set.

Membrane Run/Stop Pushbutton

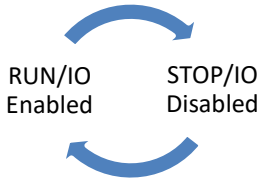
Figure 25: CPE100/CPE115 Membrane Pushbutton and Module Status LEDs



If the blue membrane pushbutton (Figure 25) is pressed while the CPE100/CPE115 is powering up, it restores the default IP address (192.168.0.100). It also erases the stored hardware configuration, logic, and contents of the backup RAM.

During normal operation, briefly pressing the membrane pushbutton changes the state of the CPU from its current Run/Stop state to its alternate state, as shown in the following state diagram:

Figure 26: State Diagram for CPE100/CPE115 Run/Stop Operation



The **Run/Stop** switch is enabled by default; it can be disabled in PME Hardware Configuration (HWC) settings.

LED Indicators (LEDs)

Ethernet Status Indicators

There are two LEDs (Yellow/Green) for each Ethernet port of LAN1 and LAN2, which are embedded in the RJ45 connectors. The green LED indicates an Ethernet connection has been established. The yellow LED indicates packet traffic.

Module Status Indicators

There are three LEDs and one membrane pushbutton on the front panel, as shown in Figure 25. The table below describes the behavior of each LED:

LED	*LED State*	*Operating State (after Power-Up)*
RUN	Blinking; All other LEDs off	This LED indicates the status of PLC during powering up. It starts blinking 6 seconds after applying power to the PLC and remains in this state for up to 15 seconds. After this, all LEDs turn off and will remain in this state until PLC is ready.
OK	On Green	PLC has passed its power-up diagnostics and is functioning properly
	Off	Power is not applied or PLC has a problem.
	Blinking; All other LEDs off	PLC in STOP/Halt state; possible watchdog timer fault. If the programmer cannot connect, cycle power and refer to the fault tables.
RUN	On Green	PLC is in RUN mode.
	Off	PLC is in STOP mode.
	Blinking; All other LEDs off	Indicates that PLC has encountered a fatal error and is blinking the error code.
Fault	On Red	PLC is in STOP/Faulted mode: a fatal fault has occurred.
	Off	No fatal faults were detected.

Ethernet Ports

CPE100/CPE115 provides two independent Fast Ethernet LANs. LAN1 has only one port and is dedicated to an embedded Ethernet controller and whereas LAN2 is comprised of 3 switched ports configurable either as a second embedded Ethernet controller or as an embedded PROFINET controller.

All the Ethernet ports of both the LAN1 and LAN2 are capable of automatically sensing the link data rate (10 Mbps or 100 Mbps), communications mode (half-duplex or full-duplex), and cabling arrangement (straight-through or crossover).

To establish Ethernet communications between the PME programming software and the CPE100/CPE115, you *first* need to set a valid IP address.

EPSCPE100/CPE115	**LAN1**	**LAN2**
Default IP Address	192.168.0.100	0.0.0.0
Subnet Mask	255.255.255.0	0.0.0.0
Gateway	0.0.0.0	0.0.0.0

Note:

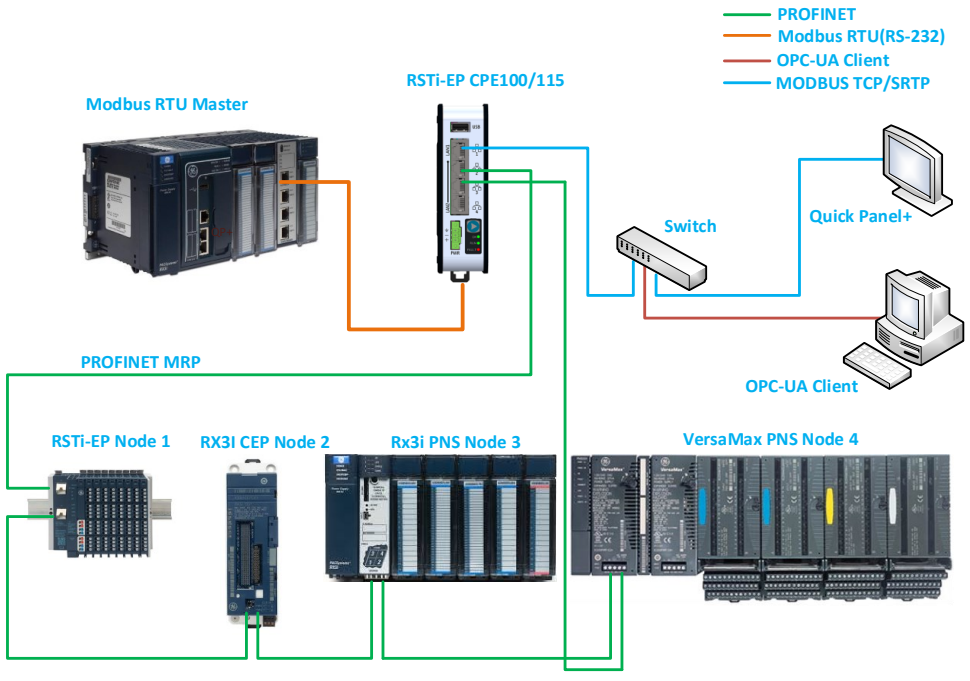
- LAN2 will not be operational unless it is configured by the programmer with a valid IP address.
- Care must be taken when assigning IP Addresses and subnet masks to each LAN so that an overlapping IP subnet is not created. Intermittent or no Ethernet communication may result if an overlapping IP subnet is created and the two interfaces are NOT connected (cabled) to the same physical network.
- By default, PME prohibits configuring both LAN interfaces on an overlapping IP subnet. (This may be changed by going to Controller General Options and changing the Multiple Embedded LANs on the Same Subnet to Show as Warning.)

The programming software 'PAC Machine Edition' uses SRTP (Service Request Transport Protocol), a proprietary protocol used primarily for communication with the controllers. The Ethernet port of LAN1 can be used to communicate with the PME software and is also a recommended option. Alternatively, any port of LAN2 can also be used but first, it should be configured with a valid IP address. Ethernet ports of LAN2 can also be configured to be used as either a second embedded Ethernet controller or as an embedded Simplex PROFINET I/O Controller.

Ethernet Topology

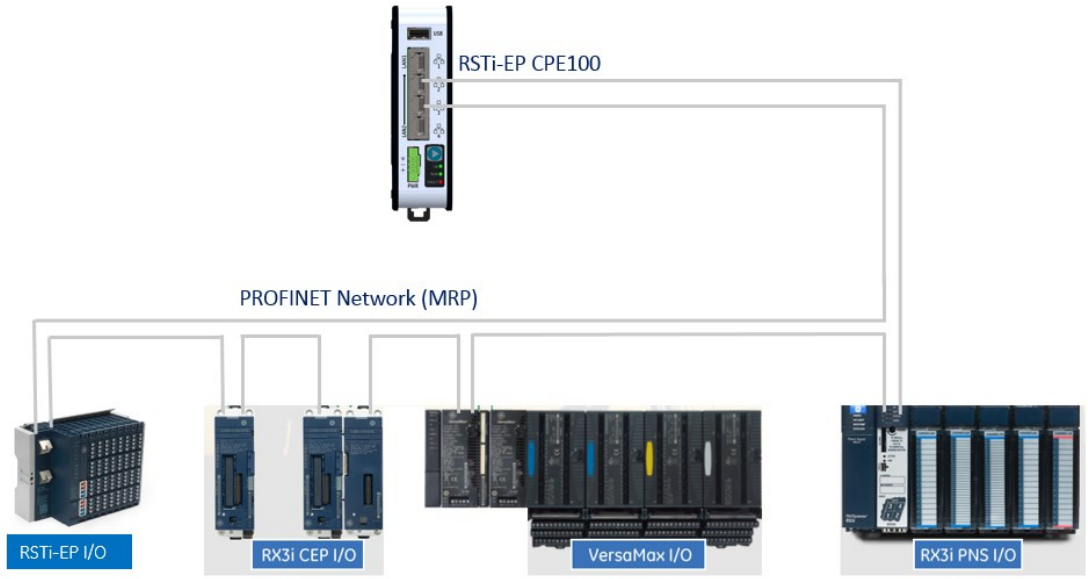
A typical application will take advantage of the two independent LANs. The dedicated LAN1 port will be used for communications with plant-level or supervisory layers. The switched LAN2 will be used to communicate with devices over PROFINET within the manufacturing cell or process.

Figure 27: Typical Multi-Tier LAN Application (Star/Bus Topology)



Whenever CPE100/CPE115 is configured for MRP only Ethernet Port2 & Port3 of LAN2 can be used to form a ring. Ethernet Port4 of LAN2 can still be used either to connect programmer, simplex PROFINET device or any other supported Ethernet protocols.

Figure 28: Typical Multi-Tier LAN Application (Ring Topology)



Super Capacitor

In the event of loss of system power, the internal supercapacitor maintains power long enough for the CPE100/CPE115 to write its user memory contents to non-volatile storage (flash) memory.

Operation

When the CPE100/CPE115 is powered up for the first time or is in a system that has been powered down long enough to completely discharge the internal supercapacitor, it may additionally require 70 to 75 seconds for it to charge to its operating level. The CPE100/CPE115 does not provide any status information about the state of the internal supercapacitor during power-up.

Life Expectancy

The super capacitor's life is computed based on the unit's ambient temperature and is given by the below estimates.

Surrounding Air Temperature near Capacitor	Typical Life Expectancy (in Years)
10°C	15
20°C	15
30°C	15
40°C	15
50°C	15
60°C	8.1
70°C	2.8

Product Limitations

This section lists the known limitations and features that are currently not supported by CPE100/CPE115:

1. SNTP is not supported by CPE100. CPE115 Supports SNTP.
2. RDSD is not supported.
3. Timed interrupt blocks are not supported.

Note: The above features may be supported in a subsequent firmware version. Refer to the datasheet for more information.

2.3.2 EPXCPE205, EPXCPE210, EPXCPE215, EPXCPE220, and EPXCPE240

Introduction

In addition to RSTi-EP's standalone controllers (EPSCPE100/EPSCPE115), Emerson also offers backplane models (EPXCPE205/210/215/220/240).

Features

Single/Dual LAN Operating Modes Configuration

This Operating Mode Configuration is specific to the EPXCPE205. Users can configure EPSCPE205 to Single LAN mode or Dual LAN operating mode configurations:

- *In Single LAN mode*: One 10/100/1000 Ethernet LAN, Port 1 and Port 2 attach to LAN2 through the internally switched RJ45 connectors. In this mode, the DUAL LAN LED will be OFF.
- *In Dual LAN mode:* Two independent 10/100/1000 Ethernet LANs. Port 1 attaches to LAN1 through a dedicated RJ45 connector. Port 2 attaches to LAN2 through an internally-switched RJ45 connector. In this mode, the DUAL LAN LED will be ON GREEN.

Operating Mode Configuration LED Indications

CPE205 Mode	LED Behavior
Single LAN Mode	DUAL LAN LED - **OFF**
Dual LAN Mode	DUAL LAN LED - solid **GREEN**

Project Upgradeability/Compatibility

RSTi-EP EPXCPE210

- Users can download EPSCPE100 projects to the EPXCPE210.
- Users are NOT able to download EPXCPE210 projects to EPSCPE100.

RSTi-EP EPXCPE215

- Users can download EPSCPE115 projects to the EPXCPE215
- Users are NOT to download EPXCPE215 projects to EPSCPE115.

Supports Retentive Memory

The RSTi-EP EPXCPE2XX controllers support PLC retentive variables as well as miscellaneous runtime operations. For more information, please see section *STOP-Halt Mode*.

Support for Run-Mode-Store to User Flash of User Program

Configuration change to enable a customer to indicate Persistence of RMS/WFW/Test & Edit of User Program to User Flash.

Ethernet Auto-Downshift Capability

The RSTi-EP EPXCPE controllers can downshift to the next highest available speed when a link fails to be established after several attempts.

Front Panel Description

Figure 29: EPXCPE205

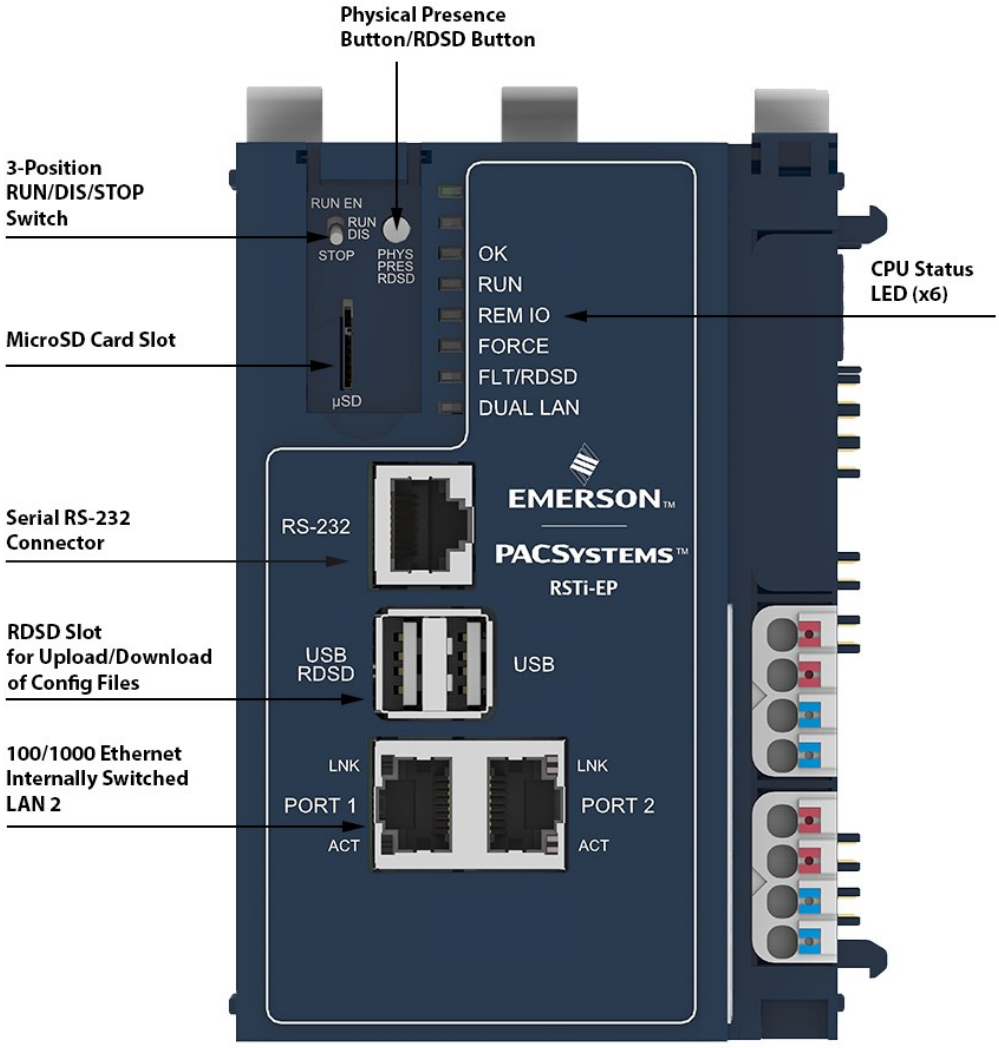
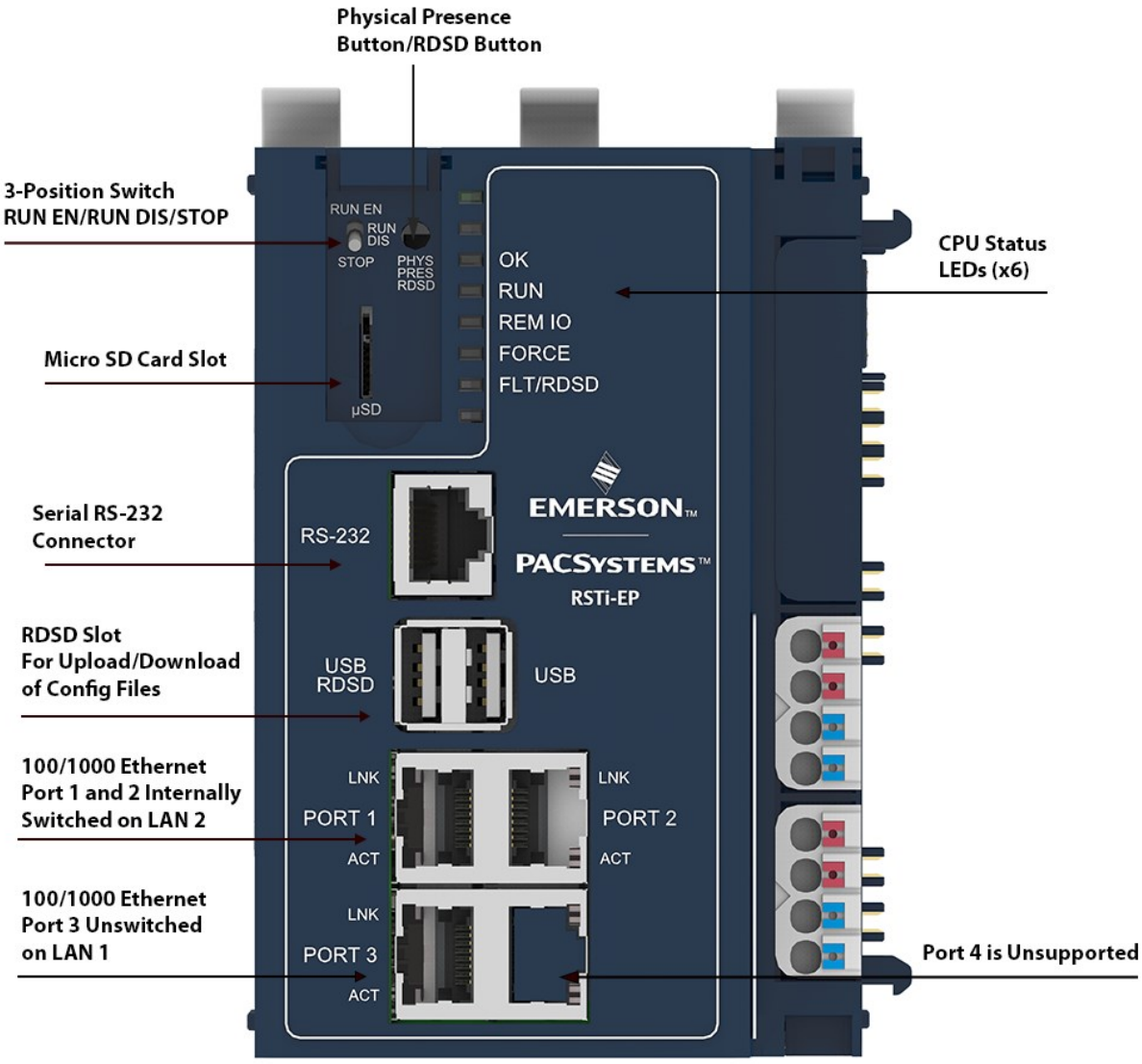


Figure 30: EPXCPE210/215/220/240



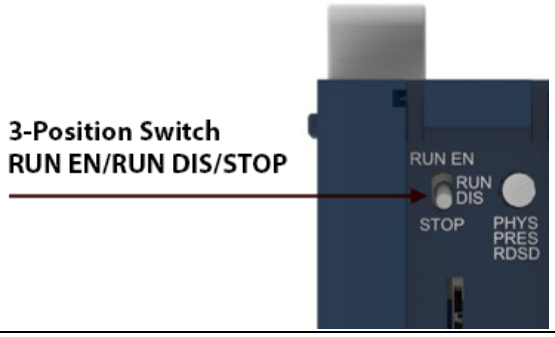
Displays and LED Indicators

Label	Function	Description/Notes
OK	LED	Muti-function PLC LED.
RUN	LED	Muti-function PLC LED.
REM IO	LED	Remote IO state LED. PLC LED that will indicate whether ALL or SOME PROFINET devices are connected.
FORCE	LED	PLC LED that indicates a “forced” value is present in User Logic.
FLT/RDSD	LED	PLC LED that will indicate a STOP-FAULT condition, IP Conflict Detected, and also the RDSD state.
DUAL LAN	LED	PLC LED that indicates CPE205 LAN mode configuration.
RUN EN	RUN/STOP Switch	PLC in Run Outputs Enabled.
RUN DIS	RUN/STOP Switch	PLC in Run Outputs Disabled.
STOP	RUN/STOP Switch	PLC in Stop.
PHY PRES RDSD	Physical Presence/RDSD	Button for Physical Presence actions (i.e. password change) and RDSD. Is used to initiate Reset to Factory Default Settings.
µSD	Micro SD Card	Port for the micro SD Card.
RS232	Serial Port	Port for serial communications. Only support RS-232.
USB RDSD	USB 2.0/RDSD	USB connection for RDSD, and potentially other features in the future.
PORT X	Ethernet Port	Ethernet Ports. X = 1, 2, 3, or 4.
LNK	Ethernet Port LED	LED that indicates Ethernet Link and Speed. Belongs to the Ethernet Connector.
ACT	Ethernet Port LED	LED that indicates Ethernet activity on the link. Belongs to the Ethernet Connector.

Ethernet Port LEDs

- The Ethernet Port LEDs are located in the PHY connector itself. They are controlled by the PHY
- There are two LEDs per port, one for Link/Speed (“LNK”) and one for Activity (“ACT”)
- **Ethernet LinkSpeed (“LNK”):**
  - GREEN indicates 1 Gbps
  - AMBER indicates 100 Mbps
  - OFF indicates 10 Mbps (currently not supported per spec)
- **Ethernet Activity (“ACT”):**
  - GREEN indicates Link
  - BLINKING GREEN indicates activity on the link

Figure 31: 3-Position Switch RUN EN/RUN DIS/STOP



Front Display LEDs

LED Label	LED State	Operating State
OK	Green (steady)	Powered up, the controller is up and ready.
	Amber (steady)	Thermal warning – a warm condition
	Red (steady)	Thermal shutdown - hot temperature condition
OK RUN	Green (flashing in unison)	Reset to factory defaults
OK RUN REM IO	Green (flashing in unison)	Programable parts update
RUN	Green (steady)	The controller is running with I/O outputs enabled
	Green (flashing)	Power up blink code
	Amber (steady)	The controller is running with I/O outputs disabled
REM IO	Green (steady)	All PROFINET devices connected
	Amber (steady)	Some PROFINET devices connected
FORCE	Amber (steady)	The controller has a forced value(s)
FLT/RDSD	Green (steady)	USB device present and RDSD is ready
	Green (steady)	RDSD Operation completed successfully
	Green (flashing)	RDSD download operation in progress
	Amber (flashing)	RDSD upload operation in progress
	Red (steady)	The controller is in a Stop-Fault state
	Red (flashing)	RDSD has failed
	Red (flashing)	If RDSD is not being performed, indicates IP conflict detected
DUAL LAN (EPXCPE205 only)	Green (steady)	Dual LAN operating mode
	Off	Single LAN operating mode

RUN/STOP Switch

The EPXCPE models all feature a 3-position switch, labeled: “STOP, RUN DIS, RUN EN” (Figure 1 and Figure 2).

Run/Stop Switch Position	CPU Sweep Mode
RUN EN	The CPU runs with I/O outputs enabled
RUN DIS	The CPU runs with I/O outputs disabled
STOP	The CPU is not allowed to go into RUN mode

Connectors and Ports

Ethernet Ports

The CPE210/215/220/240 controllers provide three internally switched Ethernet ports. (The CPE205 only provides two.)All the Ethernet ports are capable of automatically sensing the link data rate (100 Mbps or 1000 Mbps), communications mode (half-duplex or full-duplex), and cabling arrangement (straight-through or crossover).

To establish Ethernet communications between the PME programming software and the EPXCPE controllers the user will *first* need to set a valid IP address:

EPXCPE205 Single LAN Mode		
	Port 1	Port 2
Default IP Address	192.168.0.100 10.10.0.100	
Subnet Mask	255.255.255.0	
Gateway	0.0.0.0	

EPXCPE205 Dual LAN Mode		
	Port 1	Port 2
Default IP Address	192.168.0.100	10.10.0.100
Subnet Mask	255.255.255.0	
Gateway	0.0.0.0	

EPXCPE210/215/220/240				
	Port 1	Port 2	Port 3	Port 4
Default IP Address	10.10.0.100		192.168.0.100	N/A
Subnet Mask	255.255.255.0			
Gateway	0.0.0.0			

Note: Care must be taken when assigning IP Addresses and subnet masks to each LAN so that an overlapping IP subnet is not created. Intermittent or no Ethernet communication may result if an

overlapping IP subnet is created and the two interfaces are NOT connected (cabled) to the same physical network.

Serial COM Port (RS-232)

Pin No.	Signal Name	Description
1	DSR	Data Set Ready
2	DCD	Data Carrier Detect
3	DTR	Data Terminal Ready
4	GND	Signal Ground
5	RXD	Receive Data
6	TXD	Transmit Data
7	CTS	Clear to Send
8	RTS	Request to Send

Power Connectors

Supply		
Supply voltage for system and inputs	20.4V – 28.8V	
Supply voltage for outputs	20.4V – 28.8V	
Max. feed-in current for input modules (including current for RSTI-EP internal 5V DC --> 1.5A@24V)	8.5A + 1.5A	
Max. feed-in current for output modules	10 A	
Current consumption from system current path I_{SYS}	116 mA	
Connection data		
Type of connection	*Spring style*	
Conductor cross-section	Single-wired, fine-wired	0.14 – 1.5 mm^2 (AWG 26 – 16)

Connection LEDs	
PWR LED	**Green:** Supply voltage > 18Vdc **Red:** At least one current path < 18 V
	**1 -Green:** Input current path supply voltage > 17 V D
	2- Red: Input current path supply voltage < 17 V DC
	3
	4
	**1- Green:** Output current path supply voltage > 17 V DC
	2- Red: Output current path supply voltage < 17 V DC
	3
	4

Figure 3: Power Feed Minimum Current Draw

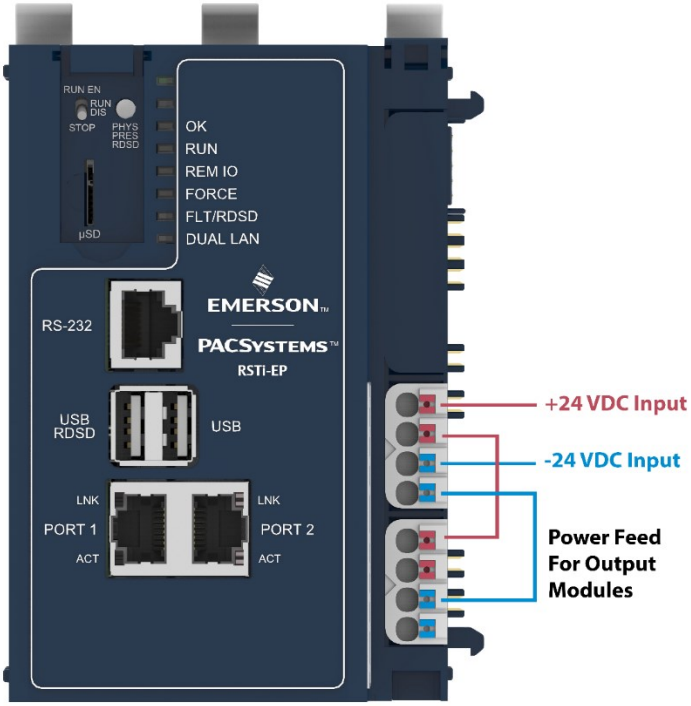
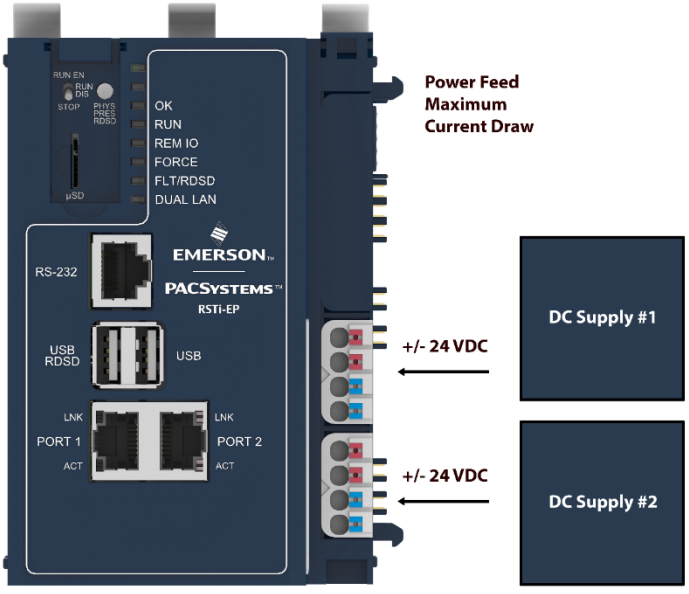


Figure 4: Power Feed Maximum Current Draw



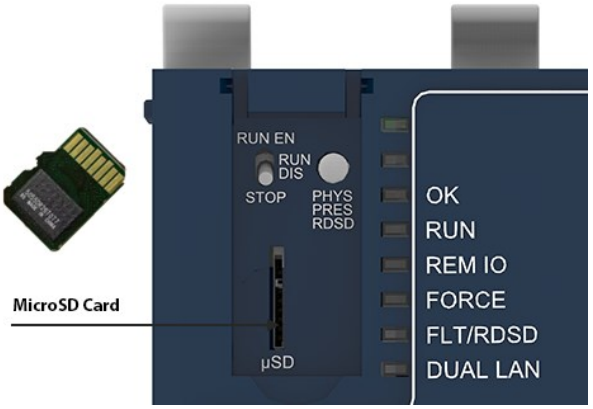
MicroSD Card Interface

The MicroSD card slot is located on the left-hand side of the module (Figure 32), behind the protective door . It supports ServReq 56/57 data retention/restore functionality

The interface supports SD, SDHC and SDXC µSD-Cards up to Version 3.0.

Insert the card into to slot, oriented as described below. Apply pressure until you feel some resistance. The card will latch into place and can then be read by the equipment. Apply pressure again to eject the card from the slot.

Figure 32: MicroSD



Note: the MicroSD Card needs to be inserted in the slot with the correct orientation. The pins of the card need to face towards the front of the equipment.

Network Configuration

EPXCPE205

The EPXCPE205 operating mode can be configured for Single LAN mode or Dual LAN mode. The default operating mode is Dual LAN mode.

Single LAN Mode	All protocols including MRP will be supported.
	Dual LAN LED will be **OFF**.
Dual LAN Mode	LAN 1 (port 2) All protocols are supported except PROFINET MRP.
	Dual LAN LED will turn solid **GREEN**.

Figure 6: Single-LAN Mode

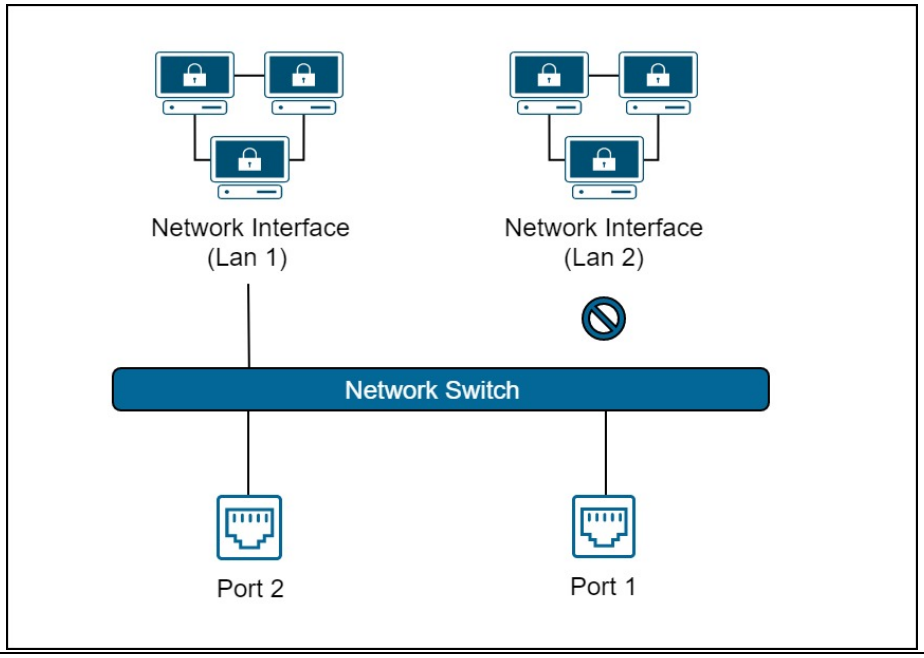
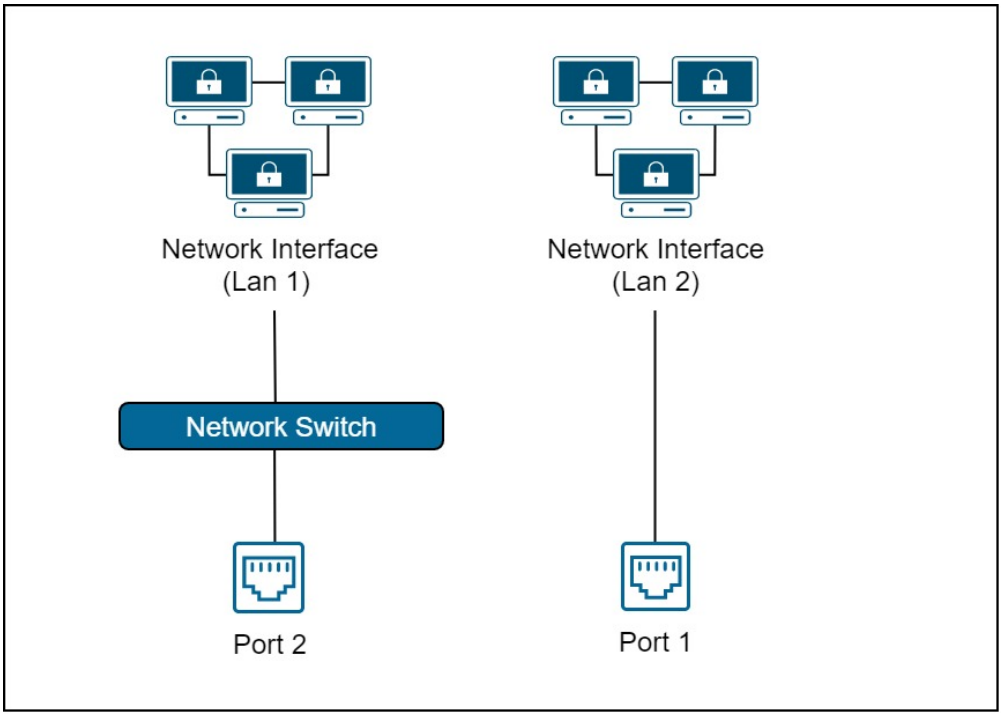


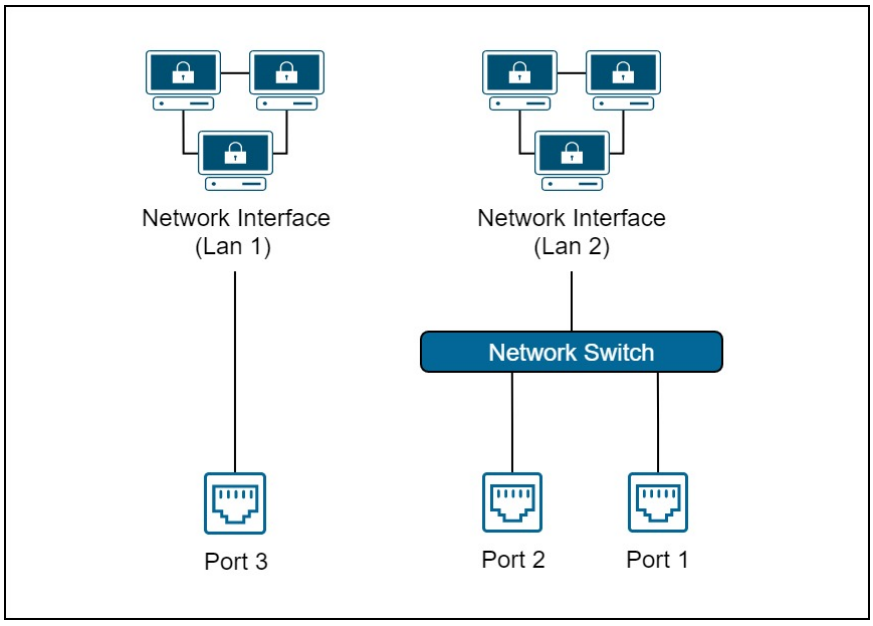
Figure 7: Dual-LAN Mode



EPXCPE210/215/220/240

LAN 1	Port 3. Unswitched. All protocols are supported except PROFINET.
LAN 2	Ports 1 and 2. Switched. All protocols are supported.

Figure 8: Network Configuration



EPXCPE205 Network Configuration Procedure

The EPXCPE205 supports two network operating modes: Single LAN and Dual LAN operating modes. To switch between the modes, the user must use the Set Operating Mode menu, which is accessible from the controller web page. (See steps below.)

IMPORTANT: Changing the operating modes will clear all user-stored controller logic. The IP addresses, controller, and IO faults will remain.

1. Begin by navigating to the controller web page by entering the IP address of the LAN you are connected to into the web browser.
2. At the top-right of the page, select the Operating Mode tab and select **Change Operating Mode**.
3. Once redirected to the change operating mode page, select the operating mode you wish to configure. **Note**: if the user selects the operating mode that they are currently in, nothing will happen.
4. To confirm the selection, press and hold the physical presence button on the controller until the webpage says to release it.
5. The controller will now reboot and come up in the new operating mode.

Update Firmware Webpage Password

Using a supported web browser, navigate to the PLC's IP address.

The webpage password can be updated while the PLC is in all modes (Run IO enabled, Run IO disabled, Stop, or Stop Fault).

Reset Webpage Password to Factory Default

Using a supported web browser, navigate to the PLC's IP address. Select **Administrator -> Reset Password** and follow the directions displayed on the web page. The physical presence button will need to be pressed during the password reset to ensure the password cannot be reset remotely.

Default password is `sierra`

Note: While the web page password reset is in progress, the push-button manager will be disabled. So the user will not be able to perform a reset to factory defaults or any other function dealing with the push button while resetting the webpage password.

Reset to Factory Defaults

IMPORTANT

Before beginning, ensure the following requirements have ben satisfied:

- **The controller is in STOP DISABLED mode.**
- **No USB stick is inserted.**
- **The Password Web Page is not waiting for input (password change).**

1. Begin by pressing the Physical Presence button (PHY PRES) for 10 seconds, The OK LED and RUN LEDs will blink GREEN continuously, indicating the Reset to Factory is occurring.
2. Release the PHY PRES button.
3. Blinking will cease and the board will be reset.

   The following events will occur :

   - Resets the IP addresses to their factory defaults (LAN1: 192.168.0.100 and LAN2: 10.10.0.100)
   - Resets the Web page firmware password to its default setting. (Password is `sierra`)
   - Deletes controller and IO faults
   - Clears NVRAM
4. After the CPU powers up, the status of the factory defaults can be observed in the PLC fault table as **Restart Due to User Initiated Reset to Factory Defaults** after successful completion. If it fails the user can observe the following message in the PLC fault table: **Reset to Factory Defaults Internal Failure** or **Reset to Factory Defaults Failed** or **Unable to enter reset to factory defaults**

Removable Data Storage Device (RDSD)

The EPXCPE user program and/or memory contents may be saved to a USB device via a single press of the pushbutton or loaded into the CPU from a USB device via a double press of the RDSD pushbutton. Similarly, data can be selected and saved. In this way, an application may be cloned and copied from one device to another

If the USB device in use already has a PACS_folder (user program) and a single press of the pushbutton occurs, a backup of the existing PACS_folder is archived (PACS_folder.tar) and then the upload occurs. If the archive already exists on the USB device, a new archive will not be created and the upload will take place.

If one needs to restore the archived version, delete the PACS_folder on the USB device and then extract the contents from PACS_folder.tar (7zip or equivalent) onto the root of the USB device.

Periodic Maintenance

Real-Time Clock Battery

The EPXCPE2xx is shipped with a real-time clock (RTC) battery pre-installed. Over time this battery will need to be replaced. No action is needed during the initial installation.

Should the RTC battery fail, the CPU date and time will reset to 12:00 AM, 01-01-2001 at start-up. The CPU operates normally with a failed or missing RTC battery; however, the initial CPU time-of-day (TOD) clock information will be incorrect.

To replace the RTC battery and reset RTC time, refer to the RX3i, & RSTi-EP CPU Reference Manual (GFK-2222) or later.

Battery required: Lithium 3v BR2032 or equivalent.

Section 3 CPU Configuration

The PACSystems CPU and I/O system is configured using PAC Machine Edition (PME) Logic Developer-PLC programming software.

The CPU verifies the physical module and rack configuration at power-up and periodically during operation. The physical configuration must be the same as the programmed configuration. Differences are reported to the CPU alarm processor for configured fault response. Refer to the *Machine Edition Logic Developer-PLC Getting Started Manual*, GFK-1918, and the online help for a description of configuration functions.

Note: A CPE020, CPE030, or CPE040 can be converted to the corresponding redundancy CPU (CRE020, CRE030, or CRE040) by installing different firmware and moving a jumper. Detailed instructions are included in the firmware upgrade kit for the redundancy CPU.

This section covers:

- Configuring the CPU
- Configuration Parameters
- Storing (Downloading) Hardware Configuration
- Configuring the Embedded Ethernet Interface

3.1 Configuring the CPU

To configure the CPU using the Logic Developer-PLC programming software, do the following:

In the Project tab of the Navigator, expand your PACSystems Target, the hardware configuration, and the main rack (Rack 0).

1. Right-click the CPU slot and choose Configure. The Parameter Editor window displays the CPU parameters.

Note: A double-wide RX3i CPU occupies two slots and can be installed in any pair of slots in Rack 0 except the two highest-numbered lots in the rack. The single-wide CPE302/CPE305 RX3i CPU requires one slot and can be installed in any slot in RX3i Rack 0, ***except*** the highest numbered slot or slot 0.

2. To edit a parameter value, click the desired tab, then click in the appropriate Values field. For information on these fields, refer to Configuration Parameters.
3. Store the configuration to the Controller so these settings can take effect. For details, see *Storing (Downloading) Hardware Configuration*.

3.2 Configuration Parameters

3.2.1 Settings Parameters

These parameters specify the basic operating characteristics of the CPU. For details on how these parameters affect CPU operation, refer to *PACSystems RX3i and RSTi-EP CPU Programmer's Reference Manual,* GFK-2950 Section 2.

Settings Parameters	
Passwords	Specifies whether passwords are Enabled or Disabled. Default: Enabled. ***Note:*** If Enhanced Security[70] is enabled in the target properties, the Passwords setting will be Enabled and read-only, and the *Access Control* tab appears. When passwords are disabled, they cannot be re-enabled without clearing PLC memory.
Legacy Client/Server Protocol Memory Access	Specifies whether non-programmer devices require a password login to access memory on the CPU when Privilege Level 2 password is set Choices: Authenticated, Unauthenticated Detault: Authenticated For details on setting the Legacy Client/Server Protocol Memory Access mode, refer to section "Legacy Client/Server Protocol Memory Access Privilege Levels" in this manual
Stop-Mode I/O Scanning	Specifies whether the I/O is scanned while the PLC is in STOP Mode. Default: Disabled. (Always Disabled for Redundancy CPU.) ***Note:*** This parameter corresponds to the I/O ScanStop parameter on a Series 90-70 PLC.

[70] *For availability, refer to the Important Product Information document for the CPU firmware version that you are using.*

Settings Parameters	
Watchdog Timer (ms)	(Denominated in ms, set in 10ms increments.) Requires a value that is greater than the program sweep time. The software watchdog timer is designed to detect *failure to complete sweep* conditions. The CPU restarts the watchdog timer at the beginning of each sweep. The watchdog timer accumulates time during the sweep. The software watchdog timer is useful in detecting abnormal operations of the application program, which could prevent the PLC sweep from completing within the watchdog time period. Valid range: 10 ms through 2550 ms, in increments of 10 ms. Default: 200. For details on setting the watchdog timer in a CPU redundancy system, refer to the *PACSystems Hot Standby CPU Redundancy User Manual,* GFK-2308.
Logic/ Configuration Power-up Source	Specifies the location/source of the logic and configuration data that is to be used (or loaded/copied into RAM) after each power-up. Choices: Always RAM, Always Flash, Conditional Flash. Default: Always RAM.
Data Power-up Source	Specifies the location/source of the reference data that is to be used (or loaded/copied into RAM) after each power-up. Choices: Always RAM, Always Flash, Conditional Flash. Default: Always RAM.

Settings Parameters	
RUN/STOP Switch	Enables or disables the physical operation of the RUN/STOP Switch. Choices: **Enabled:** Enables you to use the physical switch on the PLC to switch the PLC into STOP Mode or from STOP Mode into RUN Mode and clear non-fatal faults. **Disabled:** Disables the physical RUN/STOP Switch on the PLC. Default: Enabled. *Note:* If COM1 and COM2 are configured for any protocol other than RTU Slave or SNP Slave, the RUN/STOP switch should not be disabled without first must making sure that there is a way to stop the CPU, or take control of the CPU through another device such as an Ethernet interface. If the CPU can be set to STOP Mode, it will switch the protocol from Serial I/O to the STOP Mode protocol (default is RTU Slave). For details on STOP Mode settings, refer to COM1 and COM2 Parameters. This applies to COM1 on the CPE302/CPE305, which has only one serial port. This note does not apply to CPUs which have no serial ports.
Memory Protection Switch	Enables or disables the Memory Protect feature associated with the RUN/STOP Switch. Choices: **Enabled:** Memory Protect is enabled, which prevents writing to program memory and configuration and forcing or overriding discrete data. **Disabled:** Memory Protect is disabled. Default: Disabled.
Power-up Mode	Selects the CPU mode to be in effect immediately after power-up. Choices: Last, Stop, Run. Default: Last (the mode it was in when it last powered down). *Note:* If the battery or Energy Pack is missing or has failed and if Logic/ Configuration Power-up Source is set to *Always RAM*, the CPU powers up in STOP Mode regardless of the setting of the Power-up Mode parameter.

Settings Parameters	
Modbus Address Space Mapping Type	Specifies the type of memory mapping to be used for data transfer between Modbus TCP/IP clients and the PACSystems controller. Choices: **Disabled:** The *Disabled* setting is intended for use in systems containing Ethernet firmware that does not support Modbus TCP. **Standard Modbus Addressing:** Causes the Ethernet firmware to use the standard map, which is displayed on the Modbus TCP Address Map tab. Default: Disabled For details on the PACSystems implementation of the Modbus/TCP server, refer to *PACSystems & RX3i TCP/IP Ethernet Communications User Manual,* GFK-2224.
Universal Serial Bus	***RX3i CPE302/CPE305/CPE310/CPE330 and RSTi-EP CPE205/CPE210/CPE215/CPE220/CPE240 CPUs only.*** Enables or disables the USB port for use with RDSD (Removable Data Storage Devices). The USB port is enabled by default in the models listed in this section and their hardware configurations. If a CPU310 configuration is stored to a CPE310, the USB port will be enabled.
LAN1 Mode	***RX3i CPE330/CPE400/CPL410 and RSTi-EP CPE100/CPE115/CPE205/CPE210/CPE215/CPE220/CPE240 CPUs only.*** CPU LAN1 port mode. Choices: **Ethernet**: LAN port 1 is used for Ethernet communications. Default: Ethernet.

Settings Parameters	
LAN2 Mode	***RX3i CPE330/CPE400/CPL410 and RSTi-EP CPE100/CPE115/CPE205/CPE210/CPE215/CPE220/CPE240 CPUs only.*** CPU LAN2 port mode. Choices: **Ethernet:** LAN port 2 is used for Ethernet communications. This setting disables the embedded PROFINET controller. **PROFINET:** LAN port 2 is used by the embedded PROFINET controller for PROFINET communications. This setting enables the embedded PROFINET controller. **Disabled:** CPE400/CPL410 only. LAN port 2 is disabled. It will not require any configuration and cannot be used for any communications. Default: Ethernet for RX3i CPE330. PROFINET for RX3i CPE400, CPL410, and RSTi-EP CPE100/CPE115/***CPE205/CPE210/CPE215/CPE220/CPE240***.
LAN3 Mode	***RX3i CPE400/CPL410 CPUs only.*** CPU LAN3 port mode. Choices: **Redundancy:** Whenever redundancy is enabled in the hardware configuration, both LAN3 ports are used for Redundancy and Mode indicates Redundancy. The configuration is grayed out and is not editable. These ports may only be used as the high-speed data synchronization link between the Primary and Secondary CPUs in a Hot Standby Redundancy deployment. No additional hardware is permitted on LAN3. **Disabled:** Whenever redundancy is disabled in the hardware configuration, both LAN3 ports are disabled and Mode indicates Disabled. The configuration is grayed out and is not editable The LAN3 ports cannot be used for any communications. Default: Disabled.

Settings Parameters	
Network Time Sync	*RX3i CPE302/CPE305/CPE310/CPE330/CPE400/CPL410/CPE115/CPE205/CPE210/CPE215/CPE220/CPE240 CPUs only.* Activates Simple Network Time Protocol (SNTP) clock synchronization for the controller. Choices: **None:** SNTP is not active. **SNTP:** SNTP is active and configurable. Default: None.
Enable UTC Offset	*RX3i CPE302/CPE305/CPE310/CPE330/CPE400/CPL410/CPE115/CPE205/CPE210/CPE215/CPE220/CPE240 CPUs only.* Activates Coordinated Universal Time (UTC) settings for the controller. It allows you to select an appropriate local time zone for UTC. Choices: **Disabled:** UTC settings are not active. **Enabled:** UTC settings are active and configurable. Default: Disabled.
Day Light Savings Time (DST)	*RX3i CPE302/CPE305/CPE310/CPE330/CPE400/CPL410/CPE115/CPE205/CPE210/CPE215/CPE220/CPE240 CPUs only.* Activates Day Light Savings Time (DST) settings for the controller. Allows you to select appropriate local start and end times for Day Light Savings Time. Choices: **Disabled:** Day Light Savings Time settings are not active. **Enabled:** Day Light Savings Time settings are active and configurable. Default: Disabled.

3.2.2 Modbus TCP Address Map

This read-only tab displays the standard mapping assignments between the Modbus address space and the CPU address space. Ethernet modules and daughterboards in the PACSystems controller use Modbus-to-PLC address mapping based on this map.

Modbus Register	The Modbus protocol uses five reference table designations: 0xxxx Coil Table. Mapped to the %Q table in the CPU. 1xxxx Input Discrete Table. Mapped to the %I table in the CPU. 3xxxx Input Register Table. Mapped to the %AI register table in the CPU. 4xxxx Holding Register Table. Mapped to the %R table in the CPU. 6xxxx File Access Table. Mapped to the %W table in the CPU.
Start Address	Lists the beginning address of the mapped region.
End Address	Lists the ending address of the mapped region. For word memory types (%AI, %R, and %W) the highest address available is configured on the Memory tab.
PLC Memory	Lists the memory type of the mapped region.
Length	Displays the length of the mapped region.

3.2.3 SNTP

This tab displays the Simple Network Time Protocol configuration settings when SNTP is active.

SNTP Mode	SNTP Mode of operation. Specify the use of Multicast/Broadcast or Unicast settings to communicate to the time server. Choices: Multicast/Broadcast or Unicast. Default: Multicast/Broadcast.
Poll Interval	Interval, in seconds, at which new time requests are sent to the server. Only available when SNTP Mode is set to Unicast. Valid Range: 16 to 1024, even values only. Default: 32.
Primary IP Address	The IP address of the primary time server is in dotted decimal format. Valid Range: Any valid unicast IPv4 address. Default: 0.0.0.0.
Secondary IP Address	The optional IP address of the secondary time server is in dotted decimal format. Valid Range: Any valid unicast IPv4 address or 0.0.0.0 if unused. Default: 0.0.0.0.
Poll Count	The number of retransmissions that will be sent when no timely response is received from the server. Valid Range: 1 to 100. Default: 3.
Poll Timeout	The time, in seconds, to wait for a response from the server. Valid Range: 1 to 100. Default: 2.

3.2.4 Time

This tab displays the Coordinated Universal Time (UTC) and Day Light Savings Time (DST) configuration settings when UTC or DST is active.

UTC Offset	Local time zone offset with respect to UTC. Valid Range: Select the closest appropriate time zone for your location. Default: [UTC-5] Eastern Standard Time.
DST Offset	The offset between DST and standard time in hours and minutes. Minutes are limited to values of 0, 15, 30, and 45. Valid Range: 0:00 to 1:00. Default: 0:00.
DST Start Month	The month when DST starts. Valid Range: January to December. Default: January.
DST Start Day	The day when DST starts. Valid Range: Sunday to Saturday. Default: Sunday.
DST Start Week	The week of the month when DST starts. Valid Range: 1 to 5. * Default: 0.
DST Start Time	The time of day in hours and minutes when DST starts. Valid Range: 0:00 to 23:59. Default: 0:00.
DST Ref Zone	Indicates the time zone of reference for the DST Start and End times. Start and End times may be relative to either UTC or Local time. Choices: UTC, Local Time. Default: UTC.
DST End Month	The month when DST ends. Valid Range: January to December. Default: January.
DST End Day	The day when DST ends. Valid Range: Sunday to Saturday. Default: Sunday.
DST End Week	The week of the month when DST ends. Valid Range: 1 to 5. * Default: 0.
DST End Time	The time of day in hours and minutes when DST ends. Valid Range: 0:00 to 23:59. Default: 0:00.

* For European DST, enter 5 for start and end week to use the last Sunday of the month.

3.2.5 Scan Parameters

These parameters determine the characteristics of CPU sweep execution.

Scan Parameters	
Sweep Mode	The sweep mode determines the priority of tasks the CPU performs during the sweep and defines how much time is allotted to each task. The parameters that can be modified vary depending on the selection for sweep mode. The Controller Communications Window, Backplane Communications Window, and Background Window phase of the PLC sweep can be run in various modes, based on the PLC sweep mode. Choices: ▪ Normal mode: The PLC sweep executes as quickly as possible. The overall PLC sweep time depends on the logic program and the requests being processed in the windows and is equal to the time required to execute the logic in the program plus the respective window timer values. The window terminates when it has no more tasks to complete. This is the default value. ▪ Constant Window mode: Each window operates in a Run-to-Completion mode. The PLC alternates among three windows for a time equal to the value set for the window timer parameter. The overall PLC sweep time is equal to the time required to execute the logic program plus the value of the window timer. This time may vary due to sweep-to-sweep differences in the execution of the program logic. ▪ Constant Sweep mode: The overall PLC sweep time is fixed. Some or all of the windows at the end of the sweep might not be executed. The windows terminate when the overall PLC sweep time has reached the value specified for the Sweep Timer parameter.
Logic Checksum Words	The number of user logic words to use as input to the checksum algorithm each sweep. Valid range: 0 through 32760, in increments of 8. Default: 16.

Scan Parameters	
Controller Communication Window Mode	(Available only when Sweep Mode is set to *Normal.)* Execution settings for the Controller Communications Window. Choices: ▪ Complete: The window runs to completion. There is no time limit. ▪ Limited: Time sliced. The maximum execution time for the Controller Communications Window per scan is specified in the Controller Communications Window Timer parameter. Default: Limited. ***Note:*** This parameter corresponds to the Programmer Window Mode parameter on a Series 90-70 PLC.
Controller Communications Window Timer (ms)	(Available only when Sweep Mode is set to *Normal.* Read-only if the Controller Communications Window Mode is set to Complete.) The maximum execution time for the Controller Communications Window per scan. This value cannot be greater than the value for the watchdog timer. The valid range and default value depending on the Controller Communications Window Mode: ▪ Complete: There is no time limit. ▪ Limited: Valid range: 0 through 255ms. Default: 10. ***Note:*** This parameter corresponds to the Programmer Window Timer parameter on a Series 90-70 PLC.
Backplane Communication Window Mode	(Available only when Sweep Mode is set to *Normal.*) Execution settings for the Backplane Communications Window. Choices: Complete: The window runs to completion. There is no time limit. Limited: Time sliced. The maximum execution time for the Backplane Communications Window per scan is specified in the Backplane Communications Window Timer parameter. Default: Complete.

Scan Parameters	
Backplane Communications Window Timer (ms)	(Available only when Sweep Mode is set to *Normal*. Read-only if the Backplane Communications Window Mode is set to *Complete*.) The maximum execution time for the Backplane Communications Window per scan. This value can be greater than the value for the watchdog timer. The valid range and the default depend on the Backplane Communications Window Mode: ▪ Complete: There is no time limit. The Backplane Communications Window Timer parameter is read-only. ▪ Limited: Valid range: 0 through 255ms. Default: 255. (10ms for Redundancy CPUs.)
Background Window Timer (ms)	(Available only when Sweep Mode is set to *Normal*.) The maximum execution time for the Background Communications Window per scan. This value cannot be greater than the value for the watchdog timer. Valid range: 0 through 255 Default: 0 (5ms for Redundancy CPUs)
Sweep Timer (ms)	(Available only when Sweep Mode is set to *Constant Sweep*.) The maximum overall PLC scan time. This value cannot be greater than the value for the watchdog timer. Some or all of the windows at the end of the sweep might not be executed. The windows terminate when the overall PLC sweep time has reached the value specified for the Sweep Timer parameter. Valid range: 5 through 2550 ms, in increments of 5 ms. If the value entered is not a multiple of 5ms, it is rounded to the next highest multiple of 5ms. Default: 100.
Window Timer (ms)	(Available only when Sweep Mode is set to *Constant Window*.) The maximum combined execution time per scan for the Controller Communications Window, Backplane Communications Window, and Background Communications Window. This value cannot be greater than the value for the watchdog timer. Valid range: 3 through 255, in increments of 1. Default: 10.

Scan Parameters	
Number of Last Scans	(Available only for CPUs with firmware version 1.5 and greater.) The number of scans to execute after the PACSystems CPU receives an indication that a transition from RUN Mode to STOP Mode should occur. (Used for STOP and STOP-Fault, but not STOP-Halt.) Choices: 0, 1, 2, 3, 4, 5. Default: 0 when creating a new PACSystems target. 0 when converting a Series 90-70 target to a PACSystems target. 1 when converting a Series 90-30 target to a PACSystems target.

3.2.6 Memory Parameters

The PACSystems user memory contains the application program, hardware configuration (HWC), registers (%R), bulk memory (%W), analog inputs (%AI), analog outputs (%AQ), and managed memory.

Managed memory consists of allocations for symbolic variables and I/O variables. The symbolic variables feature allows you to create variables without having to manually locate them in memory. An I/O variable is a symbolic variable that is mapped to the inputs and outputs of a module in the hardware configuration. For details on using symbolic variables and I/O variables, refer to *PACSystems RX3i and RSTi-EP CPU Programmer's Reference Manual*, GFK-2950 Section 4.

The amount of memory allocated to the application program and hardware configuration is automatically determined by the actual program (including logic C data, and %L and %P), hardware configuration (including EGD and AUP), and symbolic variables created in the programming software. The rest of the user memory can be configured to suit the application. For example, an application may have a relatively large program that uses only a small amount of register and analog memory. Similarly, there might be a small logic program but a larger amount of memory needed for registers and analog inputs and outputs. Note that the content of reference memory is cleared any time the size of reference memory is changed.

0 provides a summary of items that count against user memory.

Calculation of Memory Required for Managed Memory

The total number of bytes required for symbolic and I/O variables is calculated as follows:

[((number of symbolic discrete bits) × 3) / (8 bits/byte)]

\+ [((number of I/O discrete bits) × Md) / (8 bits/byte)]

\+ [(number of symbolic words × (2 bytes/word)]

\+ [(number of I/O words) × (Mw bytes/word)]

Md = 3 or 4. The number of bits is multiplied by 3 to keep track of the force, transition, and value of each bit. If point faults are enabled, the number of I/O discrete bits is multiplied by 4.

Mw = 2 or 3. There are two 8-bit bytes per 16-bit word. If point faults are enabled, the number of bytes is multiplied by 3 because each I/O word requires an extra byte.

Calculation of Total User Memory Configured

The total amount of configurable user memory (in bytes) configured in the CPU is calculated as follows:

Total managed memory (bytes)

+ total reference words × (2 bytes/word)

+ *[if Point Faults are enabled]* (total words of %AI memory + total words of %AQ memory) × (1 byte/word)

+ *[if Point Faults are enabled]* (total bits of %I memory + total bits of %Q memory) / 8 bits/byte)

Note: The total number of reference points is considered system memory and is not counted against user memory.

Memory Allocation Configuration

Memory Parameters	
Reference Points	
%I Discrete Input, %Q Discrete Output, %M Internal Discrete, %S System, %SA System, %SB System, %SC System, %T Temporary Status, %G Genius Global	The upper limit for the range of each of these memory types. Read-only.
Total Reference Points	Read-only. Calculated by the programming software.
Reference Words	
%AI Analog Input	Valid range: 0 through 32,640 words. Default: 64
%AQ Analog Output	Valid range: 0 through 32,640 words. Default: 64
%R Register Memory	Valid range: 0 through 32,640 words. Default: 1024.
%W Bulk Memory	Valid range: 0 through maximum available user RAM. Increments of 2048 words. Default: 0.
Total Reference Words	Read-only. Calculated by the programming software.
Managed Memory	
Symbolic Discrete (Bits)	The configured number of bits is reserved for symbolic discrete variables. Valid range: 0 through 83,886,080 in increments of 32768 bits. Default: 32,768.
Symbolic Non-Discrete (Words)	The configured number of 16-bit register memory locations is reserved for symbolic non-discrete variables. Valid range: 0 through 5,242,880 in increments of 2048 words. Default: 65,536.

Memory Parameters	
I/O Discrete (Bits)	The configured number of bits is reserved for discrete IO variables. Valid range: 0 through 83,886,080 in increments of 32768 bits. Default: 0 For RSTi-EP CPE100/CPE115: Valid range: 0 through 4096 in increments of 2048 bits. Default: 0
I/O Non-Discrete (Words)	The configured number of 16-bit register memory locations is reserved for non-discrete IO variables. Valid range: 0 through 5,242,880 in increments of 2048 words. Default: 0
Total Managed Memory Required (Bytes)	Read-only. See *Calculation of Memory Required for Managed Memory*.
Total User Memory Required (Bytes)	Read-only. See *Calculation of Total User Memory Configured*.
Point Fault References	The Point Fault References parameter must be enabled if you want to use fault contacts in your logic. Assigning point fault references cause the CPU to reserve additional memory. When you download both the HWC and the logic to the PLC, the download routine checks if there are fault contacts in the logic, and if there are, it checks if the HWC to download has the Point Fault References parameter set to Enabled. If the parameter is Disabled, an error is displayed in the Feedback Zone. When you download only logic to the PLC, the download routine checks if there are fault contacts in the logic, and if there are, it checks if the HWC on the PLC has the Point Fault References parameter set to Enabled. If the parameter is Disabled, an error is displayed in the Feedback Zone.

3.2.7 Fault Parameters

You can configure each fault action to be either diagnostic or fatal.

A ***diagnostic fault*** does not stop the PLC from executing logic. It sets a diagnostic variable and is logged in a fault table.

A ***fatal fault*** transitions the PLC to the Stop Faulted mode. It also sets a diagnostic variable and is logged in a fault table.

Fault Parameters	
Loss of or Missing Rack	(Fault group 1.) When BRM failure or loss of power loses a rack or when a configured rack is missing, system variable #LOS_RCK (%SA12) turns ON. (To turn it OFF, fix the hardware problem and cycle power on the rack.) Default: Diagnostic.
Loss of or Missing I/O Controller	(Fault group 2.) When a Bus Controller stops communicating with the PLC or when a configured Bus Controller is missing, system variable #LOS_IOC (%SA13) turns ON. (To turn it OFF, replace the module and cycle power on the rack containing the module.) Default: Diagnostic.
Loss of or Missing I/O Module	(Fault group 3.) When an I/O module stops communicating with the PLC CPU or a configured module is missing, system variable #LOS_IOM (%SA14) turns ON. (To turn it OFF, replace the module and cycle power on the rack containing the module.) Default: Diagnostic.
Loss of or Missing Option Module	(Fault group 4.) When an option module stops communicating with the PLC CPU or a configured option module is missing, system variable #LOS_SIO (%SA15) turns ON. (To turn it OFF, replace the module and cycle power on the rack containing the module.) Default: Diagnostic.
System Bus Error	(Fault group 12.) When a bus error occurs on the backplane, system variable #SBUS_ER (%SA32) turns ON. (To turn it OFF, cycle power on the main rack.) Default: Fatal.
I/O Controller or I/O Bus Fault	(Fault group 9.) When a Bus Controller reports a bus fault, a global memory fault, or an IOC hardware fault, system variable #IOC_FLT (%SA22) turns ON. (To turn it OFF, cycle power on the rack containing the module when the configuration matches the hardware after a download.) Default: Diagnostic.

Fault Parameters	
System Configuration Mismatch	(Fault group 11.) When a configuration mismatch is detected during system power-up or a download of the configuration, system variable #CFG_MM (%SA9) turns ON. (To turn it OFF, power up the PLC when no mismatches are present or download a configuration that matches the hardware.) This parameter determines the fault action when the CPU is ***not running.*** If a system configuration mismatch occurs when the CPU is in RUN Mode, the fault action will be Diagnostic. This prevents the running CPU from going to STOP/FAULT mode. To override this behavior, see *Configuring the CPU to Stop Upon the Loss of a Critical* Module. Default: Fatal.
Fan Kit Failure	(Fault group 0x17.) When a fault is detected in the Smart Fan kit, system variable #FAN_FLT (%SA7) turns ON. (To turn a fan kit fault OFF, clear the Controller fault table or reset the PLC.) Default: Diagnostic.
Recoverable Local Memory Error	***Redundancy CPUs only.*** (Fault group 38) Determines whether a single-bit ECC error causes the CPU to stop or allows it to continue running. Choices: Diagnostic, Fatal. Default: Diagnostic. ***Note:*** When a multiple-bit ECC error occurs, a Fatal Local Memory Error fault (error code 169) is logged in the CPU Hardware Fault Group (group number 13).
CPU Over Temperature	(Fault group 24, error code 1.) When the operating temperature of the CPU exceeds the normal operating temperature, system variable #OVR_TMP (%SA8) turns ON. (To turn it OFF, clear the Controller Fault Table or reset the PLC.) Default: Diagnostic.
Controller Fault Table Size	(Read-only.) The maximum number of entries in the Controller Fault Table. The value is set to 64.
I/O Fault Table Size	(Read-only.) The maximum number of entries in the I/O Fault Table. A value set to 64.

Configuring the CPU to Stop Upon the Loss of a Critical Module

In some cases, you may want to override the RUN Mode behavior of the System Configuration Mismatch fault. A given module may be critical to the PLC's ability to properly control a process. In this case, if the module fails then it may be better to have the CPU go to STOP Mode, especially if the CPU is acting as a backup unit in a redundant system.

One way to cause the CPU to stop is to set the configured action for a Loss-of-Module fault to *Fatal* so that the CPU stops if a module failure causes a loss-of-module fault. The correct loss-of-module fault must be chosen for the critical module of interest: I/O controller, I/O module, and Option module. The Ethernet communications module is an example of an Option module.

This approach has a couple of disadvantages. First, it applies to all modules of that category, which may include modules that are not critical to the process. Second, it relies on the content of the fault table. If the table is cleared via program logic or user action, the CPU will not stop.

In systems that use Ethernet Network Interface Units (ENIUs) for remote I/O, a critical module of interest may be the Ethernet module that provides the network connection to the ENIU. Other techniques can be used to provide a more selective response to an Ethernet module failure than the Loss-of-Option module fault. One technique is to use application logic to monitor the Ethernet Interface Status bits, which are described in *Monitoring the Ethernet Interface Status Bits* in the *PACSystems RX3i and RSTi-EP TCP/IP Ethernet Communications User Manual,* GFK-2224. If the logic determined that a critical Ethernet module was malfunctioning, it could execute SVC_REQ #13 to stop the CPU.

Since the ENIU uses Ethernet Global Data to communicate with the PACSystems CPU, another selective technique is to monitor the Exchange Status Words to determine the health of individual EGD exchanges. For details on this status word, refer to *Exchange Status Word Error Codes* in *PACSystems RX3i and RSTi-EP TCP/IP Ethernet Communications User Manual,* GFK-2224. Because the types of errors indicated by the exchange status word may be temporary, stopping the CPU may not be an appropriate response for these errors. Nevertheless, the status could be used to tailor the response of the application to changing conditions in the EGD network.

In some cases, the critical module may reside in an expansion rack. In that case, in addition to the loss-of-module fault, it is recommended to set the Loss-of-Rack fault to Fatal. Then if the rack fails or loses power, the CPU will go to STOP Mode.

3.2.8 Redundancy Parameters (Redundancy CPUs Only)

These parameters apply only to redundancy CPUs or to those CPUs where the optional redundancy features have been activated. For details on configuring CPU for redundancy, refer to the *PACSystems Hot Standby CPU Redundancy User Manual,* GFK-2308.

3.2.9 Transfer List

These parameters apply only to redundancy CPUs. For details on configuring CPU for redundancy, refer to the *PACSystems Hot Standby CPU Redundancy User Manual,* GFK-2308.

3.2.10 COM1 and COM2 Parameters

These parameters configure the operating characteristics of the CPU serial ports. COM1 and COM2 have the same set of configuration parameters. The protocol (Port Mode) determines the parameters that can be set for each port.

Port Parameters	
Port Mode	The protocol to execute on the serial port. Determines the list of parameters displayed on the Port tab. Only the parameters required by the selected protocol are displayed. Choices: ▪ RTU Slave mode: Reserved for the use of the Modbus RTU Slave protocol. This mode also permits connection to the port by an SNP master, such as the WinLoader utility or the programming software. ▪ Message mode: The port is open for user logic access. This mode enables C language blocks to perform serial port I/O operations via the C Runtime Library functions. (CPE100/CPE115 does not support this feature) ▪ Available: The port is not to be used by the PLC firmware. (The CPE302/CPE305 ***does not*** support this selection.) ▪ SNP Slave: Reserved for the exclusive use of the SNP slave. This mode permits connection to the port by an SNP master, such as the WinLoader utility or the programming software. (CPE100/CPE115 does not support this feature) ▪ Serial I/O: Enables you to perform general-purpose serial communications by using COMMREQ functions. Default: RTU Slave. ***Note:*** If both serial ports are configured for any protocol other than RTU Slave or SNP Slave, the RUN/STOP switch should not be disabled without first making sure that there is a way to stop the CPU, or take control of the CPU through another device such as the Ethernet module. The Serial I/O protocol is only active when the CPU is in RUN Mode. If the CPU can be set to STOP Mode, it will switch the protocol from Serial I/O to the STOP Mode protocol (default is RTU Slave). If an SNP Master, such as the programming software in Serial mode, begins communicating on a port, the RTU protocol automatically switches to SNP Slave. As long as the CPU can be stopped, the protocol of the port can be auto-switched to one that enables serial programmer connection. Refer to *STOP Mode* protocols. If an Ethernet port is available, you can communicate with the CPU by connecting PME software via the Ethernet port.

Port Parameters	
Station Address	(RTU Slave only) ID for the RTU Slave. Valid range: 1 through 247. Default: 1. ***Note:*** You should avoid using station address 1 for any other Modbus slave in a PACSystems control system because the default station address for the CPU is 1. The CPU uses the default address in two situations: 1. If you power up without a configuration, the default station address of 1 is used. 2. When the Port Mode parameter is set to Message Mode, and Modbus becomes the protocol in STOP Mode, the station address defaults to 1. In either of these situations, if you have a slave configured with a station address of 1, confusion may result when the CPU responds to requests intended for that slave. ***Note:*** The least significant bit of the first byte must be 0. For example, in a station address of 090019010001, 9 is the first byte.
Data Rate	(All Port Modes, except *Available*.) Data rate (bits per second) for the port. Choices: 1200 Baud, 2400 Baud, 4800 Baud, 9600 Baud, 19.2k Baud, 38.4k Baud, 57.6k Baud, 115.2k baud. Default: 19.2k baud.
Data Bits	(Available only when Port Mode is set to Message mode or Serial I/O.) The number of bits in a word for serial communication. SNP uses 8-bit words. Choices: 7, 8. Default: 8.
Flow Control	(RTU slave, Message Mode, or Serial I/O.) Type of flow control to be used on the port. Choices: ▪ For Serial I/O Port Mode: None, Hardware, Software (XON/XOFF). ▪ For all other Port Modes: None, Hardware. Default: None. ***Note:*** The Hardware flow-control is RTS/CTS crossed.
Parity	(All Port Modes, except *Available*.) Parity is used in serial communication. Can be changed if required for communication over modems or with a different SNP master device. Choices: None, Odd, Even. Default: Odd.
Stop bits	(Available only when Port Mode is set to Message Mode, SNP Slave, or Serial I/O.) The number of stop bits for serial communication. SNP uses 1 stop bit. Choices: 1, 2. Default: 1.

Port Parameters	
Physical Interface	(All port modes except Available.) The type of physical interface that this protocol is communicating over. Choices: ▪ 2-wire: There is only a single path for receiving and transmitting communications. The receiver is disabled while transmitting. ▪ 4-wire: There is a separate path for receive and transmit communications and the transmit line is driven only while transmitting. ▪ 4-wire Transmitter There is a separate path for receive and transmit communications and the transmit line is driven continuously. Note that this choice is not appropriate for SNP multi-drop communications, since only one device on the multi-drop line can be transmitting at a given time. Default: 4-wire Transmitter On.
Turn Around Delay Time (ms)	(Available only when Port Mode is set to SNP Slave.) The Turn Around Delay Time is the minimum time interval required between the reception of a message and the next transmission. In 2-wire mode, this interval is required for switching the direction of data transmission on the communication line. Valid range: 0 through 2550ms, in increments of 10 ms. Default: 0.
Timeout(s)	(Available only when Port Mode is set to SNP Slave.) The maximum time that the slave will wait to receive a message from the master. If a message is not received within this timeout interval, the slave will assume that communications have been disrupted, and then it will wait for a new attach message from the master. Valid range: 0 through 60 seconds. Default: 10.
SNP ID	(Available only when Port Mode is set to SNP Slave.) The port ID is to be used for SNP communications. In SNP multi-drop communications, this ID is used to identify the intended receiver of a message. This parameter can be left blank if communication is a point to point. To change the SNP ID, click the values field and enter the new ID. The SNP ID is up to seven characters long and can contain the alphanumeric characters (A through Z, 0 through 9) or the underline (_).
Specify STOP Mode	(All port modes except *Available*.) Determines whether you accept the default STOP Mode or set it yourself. Choices: **No:** The default STOP Mode is used. **Yes:** The STOP Mode parameters appear and you can select the STOP Mode. If you set the STOP Mode to the same protocol as the RUN Mode, then the other STOP Mode parameters are read-only and are set to the same values as for the RUN Mode. Default: No.

<table>
<tr><th colspan="2">Port Parameters</th></tr>
<tr><td>STOP Mode</td><td>(Available only when Specify STOP Mode is set to Yes.)
The STOP Mode protocol to execute on the serial port. If you set the STOP Mode to the same protocol as for the RUN Mode, then the other STOP Mode parameters are read-only and are set to the same values as for the RUN Mode.
Choices and defaults are determined by the Port Mode setting.
■ SNP Slave: Reserved for the exclusive use of the SNP slave.
■ RTU Slave: Reserved for the exclusive use of the Modbus RTU Slave protocol.

If the STOP Mode protocol is different from the Port mode protocol, you can set parameters for the STOP Mode protocol.
If you do not select a STOP Mode protocol, the default protocol with default parameter settings is used.
<table>
<tr><th>Port (RUN) Mode</th><th>STOP Mode</th></tr>
<tr><td>RTU Slave</td><td>Choices: SNP Slave, RTU Slave
Default: RTU Slave.</td></tr>
<tr><td>Message Mode</td><td>Choices: SNP Slave, RTU Slave
Default: RTU Slave.</td></tr>
<tr><td>Available</td><td>Available
(Not supported on CPE302/CPE305)</td></tr>
<tr><td>SNP Slave</td><td>SNP Slave</td></tr>
<tr><td>Serial I/O</td><td>Choices: SNP Slave, RTU Slave
Default: RTU Slave.</td></tr>
</table>
Note: Setting the Port Mode to RTU Slave and the STOP Mode to SNP Slave may cause loss of programmer connection and delayed reconnection when the controller transitions from STOP to RUN Mode. To avoid this behavior, select SNP Slave for the Port Mode and do not specify a STOP Mode. For additional details, see RTU Slave/SNP Slave Operation with Programmer Attached.</td></tr>
<tr><td>Turn Around Delay Time (ms)</td><td>(Available only when STOP Mode is set to SNP Slave.) The Turn Around Delay Time is the minimum time interval required between the reception of a message and the next transmission. In 2-wire mode, this interval is required for switching the direction of data transmission on the communication line.
Valid range: 0 through 2550ms, in increments of 10 ms.
Default:
▪ When the STOP Mode is different from the Port Mode: 0ms.
▪ When the STOP Mode is the same as the Port Mode: the value is read-only and is set to the same value as the Turn-Around Delay Time for the Port Mode.</td></tr>
</table>

Port Parameters	
Timeout(s)	(Available only when STOP Mode is set to SNP Slave.) The maximum time that the slave will wait to receive a message from the master. If a message is not received within this timeout interval, the slave will assume that communications have been disrupted, and then it will wait for a new attach message from the master. Valid range: 0 through 60 seconds. Default: ▪ When the STOP Mode is different from the Port Mode: 10 seconds. ▪ When the STOP Mode is the same as the Port Mode: the value is read-only and is set to the same value as the Timeout for the Port Mode.
SNP ID	(Available only when STOP Mode is set to SNP Slave.) The port ID is to be used for SNP communications. In SNP multi-drop communications, this ID is used to identify the intended receiver of a message. This parameter can be left blank if communication is a point to point. To change the SNP ID, click the values field and enter the new ID. The SNP ID is up to seven characters long and can contain the alphanumeric characters (A through Z, 0 through 9) or the underline (_). Default: ▪ When the STOP Mode is different from the Port Mode: the default is blank. ▪ When the STOP Mode is the same as the Port Mode: the value is read-only and is set to the same value as the SNP ID for the Port Mode.
Station Address	(Available only when STOP Mode is set to RTU slave.) ID for the RTU Slave. Valid range: 1 through 247. Default: ▪ When the STOP Mode is different from the Port Mode: 1. ▪ When the STOP Mode is the same as the Port Mode: the value is read-only and is set to the same value as the Station Address for the Port Mode.

3.2.11 Scan Sets Parameters

You can create multiple sets of asynchronous I/O scans, with a unique scan rate assigned to each scan set. You can assign up to 31 scan sets for a total of 32. Scan set 1 is the standard scan set where I/O is scanned once per sweep. Each module is assigned to a scan set during the configuration of that module. Scan Set 1 is the default scan setting.

Scan Set Parameters	
Number	A sequential number from 1 to 32 is automatically assigned to each scan set. Scan set 1 is reserved for the standard scan set.
Scan Type	Determines whether the scan setting is enabled (as a fixed scan) or is disabled. Choices: Disabled, Fixed Scan. Default: Disabled.
Number of Sweeps	(Editable only when the Scan Type is set to Fixed Scan.) The scan rate of the scan set. Double-click the field, then select a value. A value of 0 prevents the I/O from being scanned. Valid range: 0 through 64. Default: 1.
Output Delay	(Editable only when the Number of Sweeps is non-zero.) The number of sweeps that the output scan is delayed after the input scan has occurred. Double-click on the field, then select a value. Valid range: 0 to (number of Sweeps - 1) Default: 0.
Description	(Editable only when the Scan Type is set to Fixed Scan.) Brief description of the scan set (32 characters maximum).

3.2.12 Power Consumption Parameters

The programming software displays the power consumed by the CPU (in Amps) for each voltage provided by the power supply.

3.2.13 Access Control

The Access Control List allows you to specify the reference address ranges that can be accessed by non-local devices such as HMIs and other controllers. To use this feature, Enhanced Security must be enabled in the properties of the target.

When Enhanced Security mode is enabled, any reference address range not defined ***cannot*** be accessed by other devices. External reads and writes that do not exist in the table are rejected by the firmware.

If overlapping memory ranges are defined, they must have the same Access level.

For symbolic variables, access control is specified by the *Publish* property of the variable, which includes a Read-Only and Read/Write setting.

Note: When requesting data from an external device, some drivers packetize data to optimize communication. If a request attempts to read a value that is not published, the entire packet will fail. A fault has been added to the fault table to help you understand a failed read/write. After addressing the fault, you must clear the fault to try again.

Access Control List Settings

Memory Area	The memory area in which the reference address range is defined. Default: Select an Area Choices: %AI Analog Input, %AQ Analog Output, %I Discrete Input, %G Genius Global, %M Internal Discrete, %Q Discrete Output, %R Register Memory, %S System, %SA System, %SB System, %SC System, %T Temporary Status, %W Bulk Memory.
Start	The starting offset of the reference address range. Default: 0 (not valid) Valid range: For %S, %SA, %SB and %SC, must be 1. All other memory types: 1 through the upper limit of the reference address range. Must be less than the End value.
End	The ending offset of the reference address range. Default: 0 (not valid) Valid range: For %S, %SA, %SB and %SC, must be 128. All other memory types: Any value greater than Start, through the upper limit of the reference address range. For word memory types (%AI, %R, and %W) the highest address available is configured on the Memory tab.
Access	Selects the type of external access allowed for the defined address range. Choices: Read-Only, Read/Write Default: Read-Only

Note: If the user receives the fault message **memory access rejected...**, it means that that the HMI or controllers are attempting to read a broader range than the specific data points allow.

Drivers will differ based on manufacturer, but with regard to how PAC Machine Edition handles encryption, %R memory is optimized by starting at the beginning of every 10 words (e.g. 1, 11, 21), and ends in ranges of 10. For example, if the user wants to specify access control for %R00334-%R00337, they would need to configure the Start at %R00331 and End at %R00340. For %M memory PAC Machine Edition packets data in 8 Bytes range. If the user wants to specify the range of %M00900-%M01020, then they would need to configure the access control list to Start at %M00897 and End at %M01024.

3.2.14 OPC UA Parameters

These parameters enable or disable the OPC UA Server.

OPC UA Parameters	
Server Enabled	Specify whether the CPU's OPC UA Server is enabled or not. Valid Range: True or False. Default: True.
UTC Offset	Local time zone offset with respect to UTC. (Read-Only: Controlled by the UTC Offset on the Time tab.)

3.3 Storing (Downloading) Hardware Configuration

A PACSystems control system is configured by creating a configuration file using the PME programming and configuration software, then transferring (downloading) the file from the programmer to the CPU via serial port COM1, serial port COM2, or an Ethernet port. If you use a serial port, it must be configured as RTU Slave (default) or SNP Slave.

The CPU stores the configuration file in its non-volatile RAM. After the configuration is stored, I/O scanning is enabled or disabled per the newly stored configuration parameters.

Before you can use an Ethernet Interface to store the hardware configuration to the PACSystems, you must first set the IP Address in the Ethernet Interface either by using the Set Temporary IP Address utility (refer to *Setting a Temporary IP Address*) or by downloading a hardware configuration through a serial connection.

1) In the programmer software, go to the Project tab of the Navigator, right-click the Target, and choose Go Online.

2) Right-click the Target and choose Online Commands, Set Programmer Mode. Make sure the CPU is in STOP Mode.

3) Right-click the Target node, and choose Download to Controller.

4) In the Download to Controller dialog box, select the items to download and click OK.

Note If you download to a PACSystems target that already has a project on it, the existing project is overwritten.

If I/O variables are configured, hardware configuration and logic cannot be stored independently. They must be stored at the same time.

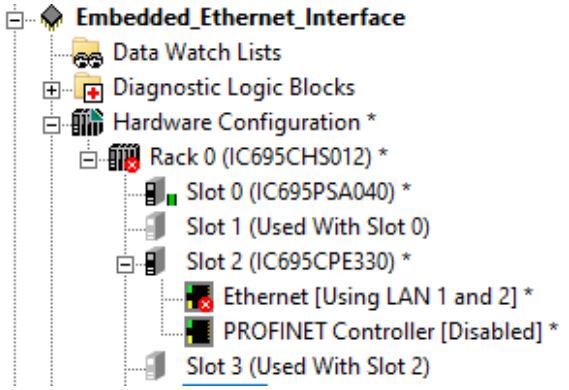
If passwords have been set, when you go online, you will be taken to the highest unprotected level. If no passwords have been set, you will go online with Privilege Level 4.

3.4 Configuring the Embedded Ethernet Interface

Before you can use the embedded Ethernet Interface, you must configure it using the programming software. To configure the embedded Ethernet interface:

1) In the Project tab of the Navigator, expand your PACSystems Target, the hardware configuration, and the main rack (Rack 0).
2) Expand the CPU slot (Slot 1). The Ethernet Interface daughterboard is displayed as *Ethernet*.
3) Right-click the daughterboard slot and choose Configure. The Parameter Editor window displays the Ethernet Interface parameters.

Figure 33: Embedded Ethernet Interface Configuration



Ethernet interface configuration includes the following additional procedures. For details on completing these steps, refer to the *PACSystems RX3i and RSTi-EP TCP/IP Ethernet Communications User Manual,* GFK-2224.

- Assigning an IP Address for initial network operation, such as connecting the programmer to download the hardware configuration, using the Set Temporary IP Address utility (refer to *Setting a Temporary IP Address*), or by downloading a hardware configuration through a serial connection.
- Configuring the characteristics of the Ethernet interface.
- Configuring Ethernet Global Data, if used.
- (Optional, not required for most systems). Setting up the RS-232 port for Local Station Manager Operation. This is part of the basic Ethernet Interface configuration.
- (Optional, not required for most systems). Configuring advanced user parameters. This requires creating a separate ASCII parameter file that is stored in the Controller with the hardware configuration. The Ethernet interface has a set of default Advanced User Parameter values that should be changed only in exceptional circumstances by experienced users.
- (Optional) Setting up the Controller for Modbus/TCP Server operation.
- ***Note:*** Whenever a CPE310 is configured as a CPU310, Ethernet properties cannot be configured.
- The embedded Ethernet interface is *not* supported when CPE310 is configured as a CPU310 and the Ethernet port should *not* be connected to any network because it may have adverse effects on the network and/or operation of the CPU.

Note: Whenever a CPE330 is configured as a CPU320, Ethernet properties cannot be configured. However, the embedded Ethernet ports may be used with their default IP Addresses.

3.4.1 Establishing Initial Ethernet Communications

To establish Ethernet communications between the PME programming and configuration software and the CPU, you *first* need to set an IP address. Use one of the following methods:

<table>
<tr><td>Default IP Addresses for RX3i CPE302/ CPE305/ CPE310/ CPE400/ CPL410[71] & RSTi-EP CPE100/ CPE115 CPE205

Embedded Ethernet</td><td>Initial Ethernet communication with the CPU may be established using the default IP addresses programmed at the factory:
<table>
<tr><th></th><th>RX3i CPE302/CPE305/ CPE310/CPE330/ CPE400/CPL410 and

RSTi-EP CPE100/CPE115/205/ 210/215/220/240</th><th>RX3i CPE330/ CPE400/CPL410

RSTi-EP 205/210/215/220/ 240</th><th>RSTi-EP CPE100/ CPE115</th></tr>
<tr><td></td><td>LAN1</td><td>LAN2</td><td>LAN2</td></tr>
<tr><td>IP Address:</td><td>192.168.0.100</td><td>10.10.0.100</td><td>0.0.0.0</td></tr>
<tr><td>Subnet Mask:</td><td>255.255.255.0</td><td>255.255.255.0</td><td>0.0.0.0</td></tr>
<tr><td>Gateway:</td><td>0.0.0.0</td><td>0.0.0.0</td><td>0.0.0.0</td></tr>
</table></td></tr>
<tr><td>Connecting to CPE302/CPE305/ CPE310 Embedded Ethernet when IP Addresses are not known</td><td>If the IP Address of the CPE302/CPE305/CPE310 embedded Ethernet interface is not known, communication may be established using one of these methods to set a permanent IP address:
• Connect to the CPE302/CPE305/CPE310 via its serial port and assign an IP Address to the embedded Ethernet interface by downloading a hardware configuration.
• Connect to the CPE302/CPE305/CPE310 with PME using an IC695ETM001 module with a known IP address and located in the same rack. Download a new hardware configuration with the desired IP address for the embedded Ethernet interface.</td></tr>
<tr><td>Connecting to CPE330 Embedded Ethernet when IP Addresses are not known</td><td>If the IP Addresses of the CPE330 embedded LAN1 and LAN2 Ethernet interfaces are not known, communication may be established using one of these methods to set new IP addresses:
• Setting a Temporary IP Address using the Set Temporary IP Address tool in PAC Machine Edition (PME). After setting the temporary address, connect to the selected CPE330 LAN using PME and download a new hardware configuration with the desired permanent IP addresses.
• Connect to the CPE330 with PME using an IC695ETM001 module with a known IP address and located in the same rack. Download a new hardware configuration with the desired permanent IP addresses for the CPE330 embedded Ethernet interfaces.</td></tr>
<tr><td>Connecting to CPE400/CPL410 Embedded Ethernet when IP Addresses are not known</td><td>Use the OLED display to read the IP Address of any LAN.
Note: Setting a Temporary IP Address tool is not available for CPE400 or CPL410.</td></tr>
</table>

[71] *CPE LAN3 IP Address is not configurable.*

Connecting to RSTi-EP CPE100/CPE115 and CPE205/210/215/220/240 Embedded Ethernet when IP Addresses are not known	The default IP address (192.168.0.100) of CPE100/CPE115 can be restored by powering up the module with the pushbutton pressed and waiting until the OK LED flashes twice. Te default IP address (192.168.0.100) of CPE205/210/215/220/240 cab be restored by pressing the Physical Presence button (PHY PRES) for 10 seconds, The OK LED and RUN LEDs will blink GREEN continuously, indicating the Reset to Factory is occurring. Releasing the button to complete the process. ***Note:*** *Setting a Temporary IP Address* tool is not available for these controllers. **CAUTION** This procedure also erases the stored hardware configuration, logic, and contents of the backup RAM.

3.4.2 Setting a Temporary IP Address

If supported by the host CPU[72], use the Set Temporary IP Address utility to specify an IP address in place of one that has been lost or forgotten.

The following restrictions apply when using the Set Temporary IP Address utility:

- To use the Set Temporary IP Address utility, the PLC CPU must not be in RUN Mode. IP address assignment over the network will not be processed until the CPU is stopped and is not scanning outputs.

The Set Temporary IP Address utility does not function if communications with the networked PACSystems target travel through a router. The Set Temporary IP Address utility can be used in communications with the networked PACSystems target travel across network switches and hubs.

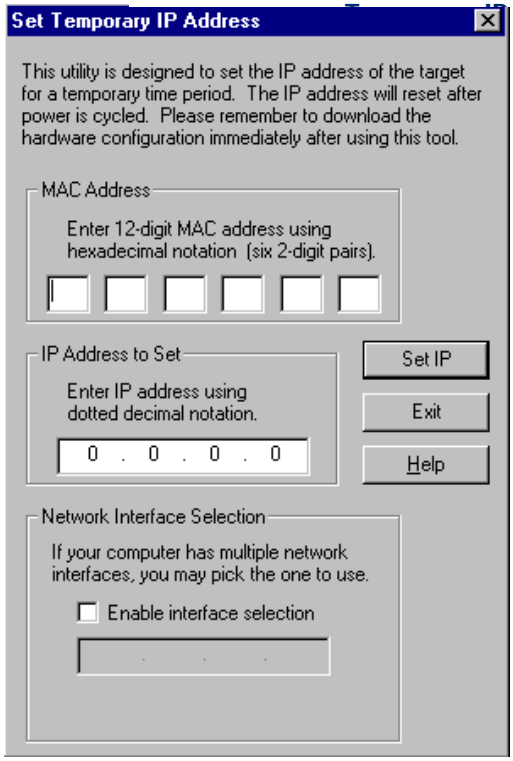
- The current user logged on to the computer running the Set Temporary IP Address utility must have full administrator privileges.
- The target PACSystems must be located on the same local sub-network as the computer running the Set Temporary IP Address utility. The sub-network is specified by the computer's subnet mask and the IP addresses of the computer and the PACSystems Ethernet Interface.

[72] *Not supported by RX3i CPE400 and RSTi-EP CPE100/CPE115/205/210/215/220/240*

Note: To set the IP address, you will need the MAC address of the Ethernet Interface to which PME will be connected.

1. Connect the PACSystems CPU LAN to the Ethernet network on which PME is communicating.
2. In the Project tab of the Navigator, right-click the PACSystems target, choose Offline Commands, and then choose Set Temporary IP Address. The Set Temporary IP Address dialog box (Figure 34) appears.
3. In the Set Temporary IP Address dialog box, do the following:
   - Key in the 12-digit hexadecimal MAC address (two digits per field).
   - In the *IP Address to Set* box, specify the temporary IP address you want to set for the PACSystems LAN.
   - If necessary, select the Enable Network Interface Selections check box and specify the IP address of the network interface on which the PACSystems is located.
4. When the fields are properly configured, click the Set IP button.
5. The IP Address of the specified PACSystems LAN will be set to the specified temporary address. This may take up to a minute.

Figure 34: Set



After the programmer connects over Ethernet, the permanent IP address for the Ethernet interface, which is set in the hardware configuration, will have been downloaded to the CPU.

The temporary IP address remains in effect until the Ethernet interface is restarted, power-cycled, or until the hardware configuration is downloaded or cleared.

CAUTION

The temporary IP Address set by the Set Temporary IP Address utility is not retained through a power cycle. To set a permanent IP Address, you must set the IP Address property of the target and download (store) HWC to the PACSystems.

The Set Temporary IP Address utility can assign a temporary IP Address even if the target Ethernet Interface has previously been configured to a non-default IP Address. (This includes overriding an IP Address previously configured by the programmer.)

Use this IP Address assignment mechanism with care.

Section 4 CPU Operation

This section describes the operating modes of a PACSystems CPU and describes the tasks the CPU carries out during these modes. The following topics are discussed:

- CPU Sweep
- Program Scheduling Modes
- Window Modes
- Data Coherency in Communications Windows
- Run/Stop Operations
- Flash Memory Operation

- Logic/Configuration Source and CPU Operating Mode at Power-Up
- Clocks and Timers
- System Security

- PACSystems I/O System

4.1 CPU Sweep

The application program in the CPU executes repeatedly until stopped by a command from the programmer, from another device, from the RUN/STOP Switch on the CPU module, or a fatal fault occurs. In addition to executing the application program, the CPU obtains data from input devices, sends data to output devices, performs internal housekeeping, performs communications tasks, and performs self-tests. This sequence of operations is called the ***sweep***.

The CPU sweep runs in one of three sweep modes:

Normal Sweep — In this mode, each sweep can consume a variable amount of time. The Logic Window is executed in its entirety each sweep. The Communications and Background Windows can be set to execute in Limited or Run-to-Completion mode.

Constant Sweep — In this mode, each sweep begins at a user-specified Constant Sweep time after the previous sweep began. The Logic Window is executed in its entirety each sweep. If there is sufficient time at the end of the sweep, the CPU alternates among the Communications and Background Windows, allowing them to execute until it is time for the next sweep to begin.

Constant Window — In this mode, each sweep can consume a variable amount of time. The Logic Window is executed in its entirety each sweep. The CPU alternates among the Communications and Background Windows, allowing them to execute for a time equal to the user-specified Constant Window timer.

Note: The information presented above summarizes the different sweep modes. For additional information, refer to CPU Sweep Modes.

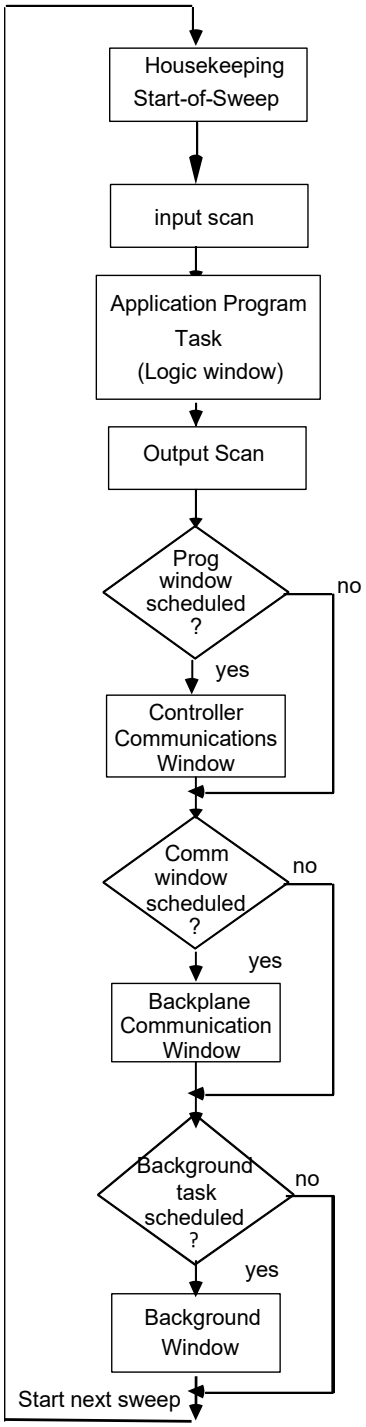
The CPU also operates in one of four RUN/STOP Modes (for details, refer to *Run/Stop Operations*):

- Run/Outputs Enabled
- Run/Outputs Disabled
- Stop/IO Scan
- Stop/No IO

4.1.1 Parts of the CPU Sweep

There are seven major phases in a typical CPU sweep as shown in the following figure.

Figure 35: Major Phases of a Typical CPU Sweep



Major Phases in a Typical CPU Sweep

Phase	Activity
Housekeeping	The housekeeping portion of the sweep performs the tasks necessary to prepare for the start of the sweep. This includes updating %S bits, determining timer update values, determining the mode of the sweep (Stop or Run), and polling expansion racks. Expansion racks are polled to determine if power has just been applied to an expansion rack. Once an expansion rack is recognized, the configuration of that rack and all of its modules are processed in the Controller Communications Window.
Input Scan	During the input scan, the CPU reads input data from the Genius Bus Controllers and input modules. If data has been received on an EGD page, the CPU copies the data for that page from the Ethernet interface to the appropriate reference memory. For details, see *PACSystems and RX3i TCP/IP Ethernet Communications User Manual,* GFK-2224. ***Note:*** The input scan is not performed if a program has an active Suspend I/O function on the previous sweep.
Application Program Task Execution (Logic Window)	The CPU solves the application program logic. It always starts with the first instruction in the program. It ends when the last instruction is executed. Solving the logic creates a new set of output data. For details on controlling the execution of programs, refer to *PACSystems RX3i and RSTi-EP CPU Programmer's Reference Manual,* GFK-2950. Interrupt-driven logic can execute during any phase of the sweep. For details, refer to *PACSystems RX3i and RSTi-EP CPU Programmer's Reference Manual,* GFK-2950 Section 2. A list of execution times for instructions can be found in Appendix A:.

Phase	Activity
Output Scan	The CPU writes output data to bus controllers and output modules. The user program checksum is computed. During the output scan, the CPU sends output data to the Genius Bus Controllers and output modules. If the producer period of an EGD page has expired, the CPU copies the data for that page from the appropriate reference memory to the Ethernet interface. The output scan is completed when all output data has been sent. If the CPU is in RUN Mode and it is configured to perform a background checksum calculation, the background checksum is performed at the end of the output scan. The default setting for the number of words to checksum each sweep is 16. If the words to checksum each sweep are set to zero, this processing is skipped. The background checksum helps ensure the integrity of the user logic while the CPU is in RUN Mode. The output scan is not performed if a program has an active Suspend I/O function on the current sweep.
Controller Communications Window	Services the onboard Ethernet and serial ports. In addition, reconfiguration of expansion racks and individual modules occurs during this portion of the sweep. The CPU always executes this window. The following items are serviced in this window: ▪ Reconfiguration of expansion racks and individual modules. During the Controller Communications Window, the highest priority is given to reconfiguration. Modules are reconfigured as needed, up to the total time allocated to this window. Several sweeps are required to complete the reconfiguration of a module. ▪ Communications activity involves the embedded Ethernet port and the two serial ports of the CPU. The time and execution of the Controller Communications Window can be configured using the programming software. It can also be dynamically controlled from the user program using Service Request function #3. The window time can be set to a value from 0 to 255 ms (the default is 10 ms). Note that if the Controller Communications Window is set to 0, there are two alternate ways to open the window: perform a power cycle without the battery (or Energy Pack) attached, or go to STOP Mode.

Phase	Activity
Backplane Communications Window	Communications with intelligent devices occur during this window. The rack-based Ethernet Interface module communicates in the Backplane Communications window. During this part of the sweep, the CPU communicates with intelligent modules such as the Genius Bus Controller and TCP/IP Ethernet modules. In this window, the CPU completes any previously unfinished request before executing any pending requests in the queue. When the time allocated for the window expires, processing stops. The Backplane Communications Window defaults to Complete (Run to Completion) mode. This means that all currently pending requests on all intelligent option modules are processed every sweep. This window can also run in Limited mode, in which the maximum time allocated for the window per scan is specified. The mode and time limit can be configured and stored to the CPU, or it can be dynamically controlled from the user program using Service Request function #4. The Backplane Communications Window time can be set to a value from 0 to 255ms (default is 255ms). This allows communications functions to be skipped during certain time-critical sweeps.
Background Window	CPU self-tests occur in this window. A CPU self-test is performed in this window. Included in this self-test is a verification of the checksum for the CPU operating system software. The Background Window time defaults to 0 ms. A different value can be configured and stored on the CPU, or it can be changed online using the programming software. Time and execution of the Background Window can also be dynamically controlled from the user program using Service Request function #5. This allows background functions to be skipped during certain time-critical sweeps.

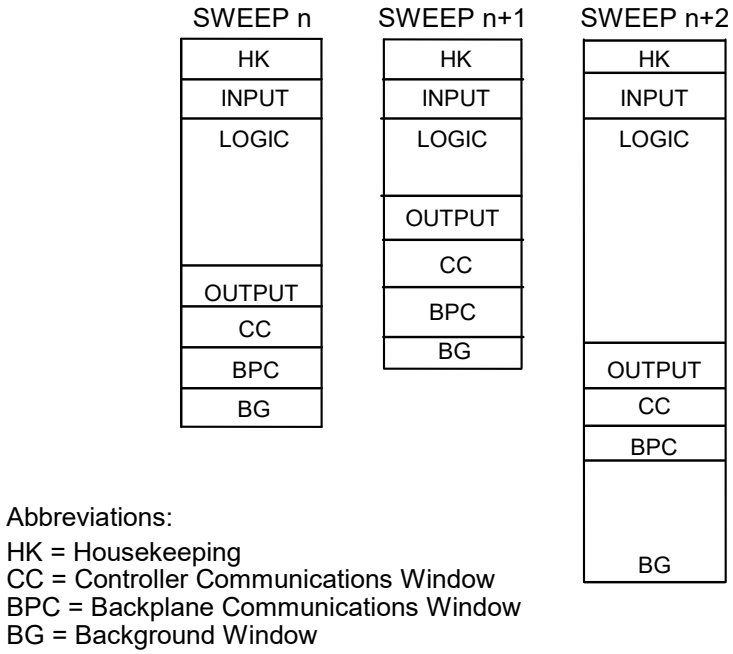
4.1.2 CPU Sweep Modes

Normal Sweep Mode

In Normal Sweep mode, each sweep can consume a variable amount of time. The Logic window is executed in its entirety each sweep. The Communications windows can be set to execute in a Limited or Run-to-Completion mode. Normal Sweep is the most common sweep mode used for control system applications.

The following figure illustrates three successive CPU sweeps in Normal Sweep mode. Note that the total sweep times may vary due to sweep-to-sweep variations in the Logic window, Communications windows, and Background window.

Figure 36: Typical Sweeps in Normal Sweep Mode



Constant Sweep Mode

In Constant Sweep mode, each sweep begins at a specified Constant Sweep time after the previous sweep began. The Logic Window is executed in its entirety each sweep. If there is sufficient time at the end of the sweep, the CPU alternates among the Controller Communications, Backplane Communications, and Background Windows, allowing them to execute until it is time for the next sweep to begin. Some or all of the Communications and Background Windows may not be executed. The Communications and Background Windows terminate when the overall CPU sweep time has reached the value specified as the Constant Sweep time.

One reason for using Constant Sweep mode is to ensure that I/O data are updated at constant intervals.

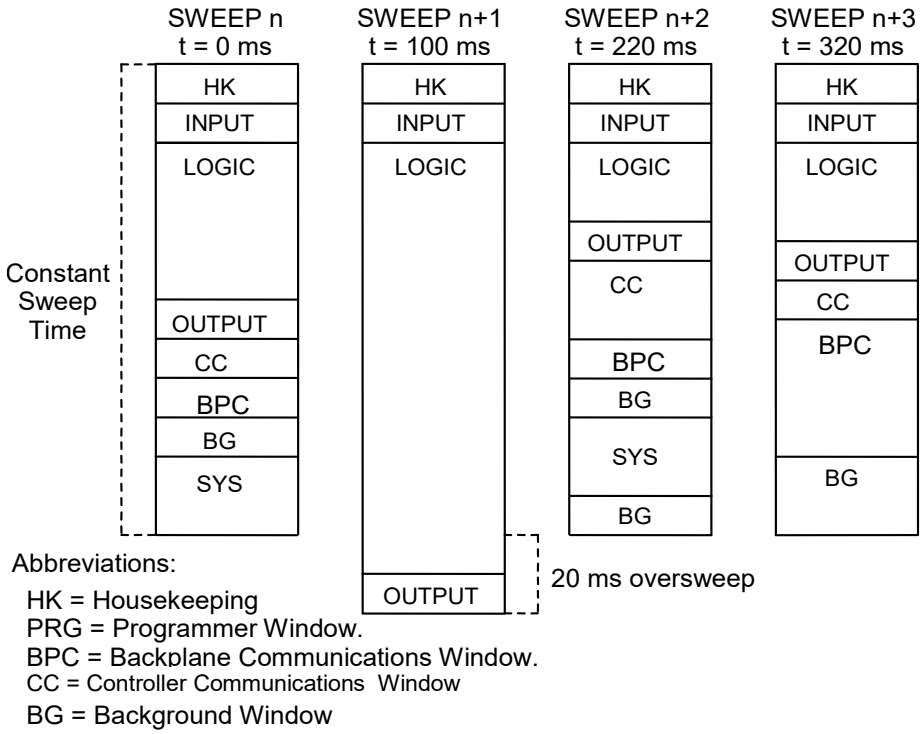
The value of the Constant Sweep timer can be configured to be any value from 5 to 2550 ms. The Constant Sweep timer value may also be set and Constant Sweep mode may be enabled or disabled by the programming software or by the user program using Service Request function #1. The Constant Sweep timer has no default value; a timer value must be set before or at the same time Constant Sweep mode is enabled.

The Ethernet Global Data[73] page, configured for either consumption or production, can add up to 1 ms to the sweep time. This sweep impact should be considered when configuring the CPU constant sweep mode and setting the CPU watchdog timeout.

If the sweep exceeds the Constant Sweep time in a given sweep, the CPU places an oversweep alarm in the CPU fault table and sets the OV_SWP (%SA0002) status reference at the beginning of the next sweep. Additional sweep time due to an oversweep condition in a given sweep does not affect the time given to the next sweep.

The following figure illustrates four successive sweeps in Constant Sweep mode with a Constant Sweep time of 100 ms. Note that the total sweep time is constant, but an oversweep may occur due to the Logic Window taking longer than normal.

Figure 37: Typical Sweeps in Constant Sweep Mode



[73] *For EGD configured on Embedded Ethernet interface of CPE302/CPE305/CPE310, refer to A.3.6 for Constant sweep impact.*

Constant Window Mode

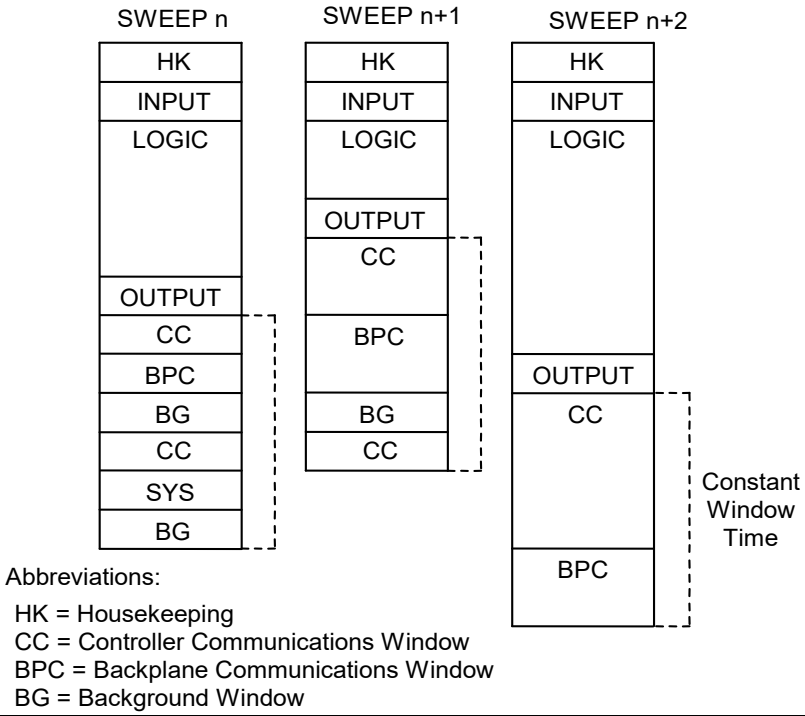
In Constant Window mode, each sweep can consume a variable amount of time. The Logic Window is executed in its entirety each sweep. The CPU alternates among the three windows, allowing them to execute for a time equal to the value set for the Constant Window timer. The overall CPU sweep time is equal to the time required to execute the Housekeeping, Input Scan, Logic Window, and Output Scan phases of the sweep plus the value of the Constant Window timer. This time may vary due to sweep-to-sweep variances in the execution time of the Logic Window.

An application that requires a certain amount of time between the Output Scan and the Input Scan, permitting inputs to settle after receiving output data from the program, would be ideal for Constant Window mode.

The value of the Constant Window timer can be configured to be any value from 3 to 255 ms. The Constant Window timer value may also be set by the programming software or by the user program using Service Request functions #3, #4, and #5.

The following figure illustrates three successive sweeps in Constant Window mode. Note that the total sweep times may vary due to sweep-to-sweep variations in the Logic Window, but the time is given to the Communications and Background Windows is constant. Some of the Communications or Background Windows may be skipped, suspended, or run multiple times based on the Constant Window time.

Figure 38: Typical Sweeps in Constant Window Mode



4.2 Program Scheduling Modes

The CPU supports one program scheduling mode: the Ordered mode. An ordered program is executed in its entirety once per sweep in the Logic Window.

4.3 Window Modes

The previous section describes the phases of a typical CPU sweep. The Controller Communications, Backplane Communications, and Background windows can be run in various modes, based on the CPU sweep mode. The following three window modes are available:

Mode Type	Description
Run-to-Completion	In Run-to-Completion mode, all requests made when the window has started are serviced. When all pending requests in the given window have been completed, the CPU transitions to the next phase of the sweep. (This does not apply to the Background window because it does not process requests.)
Constant	In Constant Window mode, the total amount of time that the Controller Communications window, Backplane Communications window, and Background window run is fixed. If the time expires while in the middle of servicing a request, these windows are closed, and communications will be resumed the next sweep. If no requests are pending in this window, the CPU cycles through these windows the specified amount of time polling for further requests. If any window is put in constant window mode, all are in constant window mode.
Limited	In Limited mode, the maximum time is fixed for the execution of the window. If time expires while in the middle of servicing a request, the window is closed, and communications will be resumed the next time that the given window is run. If no requests are pending in this window, the CPU proceeds to the next phase of the sweep.

4.4 Data Coherency in Communications Windows

When running in Constant or Limited Window mode, the Controller and Backplane Communications Windows may be terminated early in all CPU sweep modes. If an external device, such as CIMPLICITY HMI, is transferring a block of data, the coherency of the data block may be disrupted if the communications window is terminated before completing the request. The request will complete during the next sweep; however, part of the data will have resulted from one sweep and the remainder will be from the following sweep. When the CPU is in Normal Sweep mode and the Communications Window is in Run-to-Completion mode, the data coherency problem described above does not exist.

Note: External devices that communicate to the CPU while it is stopped will read information as it was left in its last state. This may be misleading to operators viewing an HMI system that does not indicate the CPU Run/Stop state. Process graphics will often indicate everything is still operating normally.

Also, note that non-retentive outputs do not clear until the CPU is transitioned from Stop to Run.

4.5 Run/Stop Operations

The PACSystems CPUs support four RUN/STOP Modes of operation. You can change these modes in the following ways: the RUN/STOP Switch, configuration from the programming software, LD function blocks, and system call from C applications. Switching to and from various modes can be restricted based on privilege levels, the position of the RUN/STOP Switch, passwords, etc.

Mode	Operation
Run/Outputs Enabled	The CPU runs user programs and continually scans inputs and updates physical outputs, including Genius and Ethernet outputs. The Controller and Backplane Communications Windows are run in Limited, Run-to-Completion, or Constant mode.
Run/Outputs Disabled	The CPU runs user programs and continually scans inputs, but updates to physical outputs, including Genius and Field Control, are not performed. Physical outputs are held in their configured default state in this mode. The Controller and Backplane Communications Windows are run in Limited, Run-to-Completion, or Constant mode.
Stop/IO Scan Enabled	The CPU does not run user programs, but the inputs and outputs are scanned. The Controller and Backplane Communications Windows are run in Run-to-Completion mode. The Background Window is limited to 10ms.
Stop/IO Scan Disabled	The CPU does not run user programs, and the inputs and outputs are not scanned. The Controller and Backplane Communications Windows are run in a Run-to-Completion mode. The Background Window is limited to 10ms. ***Note:*** STOP Mode I/O scanning is always disabled for redundancy CPUs.

Note: You cannot add to the size of %P and %L reference tables in RUN Mode unless the %P and %L references are the first of their type in the block being stored or the block being stored is new.

4.5.1 CPU STOP Modes

The CPU has four modes of operation while it is in STOP Mode. The two most common are:

STOP-I/O Enabled Mode

I/O Scan Enabled - the Input and Output scans are performed each sweep.

STOP-I/O Disabled Mode

I/O Scan Disabled - the Input and Output scans are skipped.

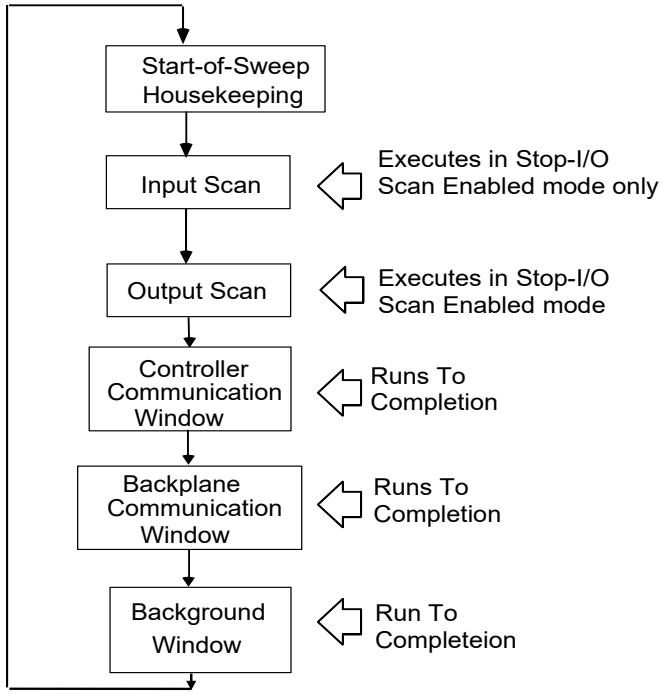
When the CPU is in STOP Mode, it does not execute the application program. You can configure whether the I/O is scanned during STOP Mode. Communications with the programmer and intelligent option modules continue in STOP Mode. Also, bus receiver module polling and rack reconfiguration continue in STOP Mode.

In both STOP Modes, the Controller Communications and Backplane Communications windows run in Run-to-Completion mode and the Background window runs in Limited mode with a 10 ms limit.

The number of last scans can be configured in the hardware configuration. Last scans are completed after the CPU has received an indication that a transition from Run to Stop or Stop Faulted mode should occur. The default is 0.

SVCREQ13 can be used in the application program to stop the CPU after a specified number of scans. All I/O will go to their configured default states, and a diagnostic message will be placed in the CPU Fault Table.

Figure 39: CPU Sweep in Stop-I/O Disabled and Stop-I/O Enabled Modes



STOP-Halt Mode

Recovering from STOP-Halt Mode (Firmware Versions 10.05 or Later)

PACSystems™ RX3i and RSTi-EP CPU firmware version 10.05 introduces new functionality to automatically recover from STOP-Halt mode for all CPU models. The secure remote STOP-Halt restart mechanism saves off pertinent debug, diagnostic, and fault information to retentive memory, and automatically resets the controller such that it restarts in STOP-Fault mode. Because the recovery process is automatic, there is no need to perform the model-specific STOP-Halt recovery procedures listed in the next section. The controller also logs the following fault in the Controller Fault Table identifying that an auto-recovery event occurred:

```
INFO_CPU_SOFTWR - CPU software event: Controller automatically recovered
from a fatal error. Error Code: 672. Group: 140.
```

There are some caveats to this feature that the user should be aware of: If the controller is configured to power up from RAM, an auto-recovery event will power up the controller with cleared Logic, Hardware Configuration, and Data/Reference memory, regardless of the presence of a battery or Energy Pack. However, if the controller is configured to power up from flash (always or conditionally), the RAM is still cleared such that Logic, Hardware Configuration, and Data/Reference memory are restored from flash as described in the *Flash Memory Operation* and the *Logic/Configuration Source and CPU Operating Mode at Power-up* sections below. Regardless of whether RAM is cleared or restored from flash, the controller powers up in STOP-Fault mode. For information regarding recovering the controller from STOP-Fault mode, see the *STOP-Fault Mode* section below.

Recovering from STOP-Halt Mode (Firmware Versions Earlier than 10.05, Only Supported on CPE400/CPL410)

The CPU will automatically go into STOP-Halt mode and suspend logic execution and I/O scanning for the following conditions:

- Software Watchdog timeout
- ECC Memory Check fault
- Illegal memory access from a C-Block
- Hardware Watchdog timeout. This condition resets the CPU and suspends backplane communications.

To recover from STOP-Halt mode, the CPU/CPE must be disconnected from its backup power source (battery or Energy Pack), powered off, then powered back on, after which the backup power source should be reconnected. The CPE400/CPL410 provides an alternative way to recover from STOP-Halt mode using the OLED display and without the need of removing the Energy Pack.

To enable backplane communications where they have been disabled in STOP-Halt mode, cycle power with its backup power source attached (battery or Energy Pack).

While the CPU is in STOP-Halt mode, the PacsAnalyzer Utility may be employed to examine the CPU's fault tables. The PacsAnalyzer Utility software is a tool that is embedded in PME. It can also be downloaded from Emerson's support website. (See link located at the end of this document.)

If backplane communications have been suspended, the PacsAnalyzer Utility must be directly connected to a serial or Ethernet port on the CPU. If backplane communications are operational, the PacsAnalyzer Utility may be connected via a communications or Ethernet module in the backplane, or to a CPU-embedded port.

CPE400/CPL4010 STOP-Halt Recovery Procedure

1. Collect a PacsAnalyzer trace before performing the rest of the steps. Once the recovery is applied, the CPU clears its Energy Pack memory. The User Flash memory is not automatically cleared.
2. Navigate to the **Controller Status** page in the OLED Display.
3. Select the **Recovery PLC** option.
4. Select the **Clear StopHalt** option.
5. Confirm the command by selecting **OK**.
    - The message **Please wait. Resetting in about 30 secs** appears in the display.
    - After about 30 seconds, the PLC restarts.
    - After the restart, the PLC will be in STOP mode. The STOP-Halt mode is gone.
    - The following faults are present in the Controller Fault table:
        - **User memory is not preserved. Error Code: 7. Group: 130**
        - **User-Initiated Recovery Action: Controller commanded to power up in Stop Mode. Error Code: 670. Group: 140**
6. Connect to the PLC and perform the necessary corrections. If the PLC is placed in RUN mode without fixing the offending code, the PLC will enter STOP-Halt and the procedure will have to be performed again.

Recovering from STOP-Halt mode RSTi-EP Controllers (Firmware Versions Earlier than 10.05)

To recover the controller from a Stop/Halt state, complete the following:

1. Connect a live Ethernet cable to LAN1.
2. Press and hold the membrane Run/Stop pushbutton and power down the controller.
3. Continue holding the Run/Stop push-button until power has drained completely (30 seconds) and then release the pushbutton. The LEDs for the LAN1 port will turn off completely. Note: Ethernet LEDs may blink slowly during the shutdown.
4. Reconnect power and power on the controller. Note: If the configuration and logic were downloaded into flash and the Power-up Mode parameter is also set as flash then, the only way to recover from Stop/Halt state is to perform a factory reset.

STOP-Fault Mode

In STOP-Fault Mode, logic execution and I/O Scanning cease after the number of last scans (configured by the user) has been exhausted. Client communications also cease at that time. Server communications are available, but with PLC data which has become static.

Within PME, the user can configure each fault action to be either *diagnostic* or *fatal*.

- A diagnostic fault does not stop the Controller from executing logic. It sets a diagnostic variable and is logged in a fault table.
- A fatal fault transitions the Controller to the STOP-Fault Mode. It also sets a diagnostic variable and is logged in a fault table.

Within PME, the user can also configure the number of last scans to be executed in the event of a fault (see PME *Scans* tab, *Number of Last Scans* parameter).

To recover from STOP-Fault Mode, resolve the underlying cause and clear the Controller Fault Table. This allows the CPU to transition to STOP-I/O Disabled Mode.

4.5.2 STOP-to-RUN Mode Transition

The CPU performs the following operations on Stop-to-Run transition:

- Validation of sweep mode and program scheduling mode selections
- Validation of references used by programs with the actual configured sizes
- Re-initialization of data areas for external blocks and standalone C programs
- Clearing of non-retentive memory

4.5.3 RUN Mode Store

Run Mode Store (RMS) consists in downloading logic to an Emerson Controller while it is executing. This can take the form of one of the following:

- **Writing word-for-word changes to the Emerson Controller**
  - Many changes to the program that do not modify the size of the program are considered word-for-word changes. Examples include changing the type of contact or coil or changing a reference address used for an existing function block. The following are word-for-word changes:
    - Switching between two symbolic variables
    - Switching between a symbolic variable and a mapped variable
    - Switching between a constant and a symbolic variable
- **Test Edit Store**
  - Small changes in a single block with the ability to test the changes before accepting them
- **Downloading to a running Emerson Controller**
  - Bigger changes as at this point the Controller and PME are not equal.
  - Changes include downloading EGD configuration information
- (For CPU Redundancy on a PACSystems RX7i or PACSystems RX3i CPU with firmware version 5.50 or later and with the Ethernet Global Data component.) Downloading EGD configuration information

For the EGD exchange Run Mode Store Enabled property for Machine Edition EGD exchanges:

- If set to True, then you can manually change any editable EGD exchange property for this exchange, and then download or RMS this exchange.

- If set to False, then:
    - The Controller must be stopped before you can download this EGD exchange. When performing an RMS, you cannot download any changes for this EGD exchange.

⚠ CAUTION

Caution: See produced or consumed Exchange ID.

(For PACSystems RX7i and PACSystems RX3i redundancy CPUs with firmware version 5.50 or later.) If you perform a Run Mode Store (RMS) and you select the **Do synchronized activation of redundant Controllers** option and both primary and secondary Controllers have been updated with new data, that is, the download of information to both Controllers is successful, then both Controllers begin to use the updated information. In this case, you must perform an RMS on both the primary and secondary Controllers.

When the symbolic variable is added and the user attempts to perform an RMS, the Run Mode Store dialog box will appear unless the Runmode Warning option is set to *False*. For help, click the Help button in the dialog box.

Note: (PAC Productivity Suite Systems only.) In the Run Mode Store dialog box, the set to the initial value of the associated variable option is selected by default. These initial values contain the block configurations.

Limitations

Not all Emerson Controllers support Run Mode Store (RMS).

During an RMS, you cannot download Controller supplemental files to your target.

In the following cases, RMS is not possible and you must stop the Controller before downloading:

- Adding or deleting an interrupt block.
- Changing the scheduling of a block.
- Adding too many %L variables to a PACSystems or Series 90-70 target, which exceeds the variables buffer size.

Note: The buffer size is automatically adjusted when downloading logic to a stopped Controller.

- Adding too many symbolic variables to a PACSystems target.

Note: If you delete symbolic variables and create new ones, the memory allocated for the deleted variables is NOT freed up for use by the new ones. This is to prevent potential memory corruption. The created variables might exceed the configured size for symbolic variables and prevent the RMS. The memory is freed up only when downloading logic and initial values to a stopped Controller or clearing logic.

- Modifying the value of the MFB Memory Allocation property
- Making various changes involving reference ID variables
- Making various changes to User-defined Data Types
    - Initial values are not stored during a RUN mode store
    - An RMS can be performed on a target that contains a variable of a UDT, unless:

  - An operation in the UDT editor modifies the offset or bitmask of an element that has the same name before and after the operation.
  - The size of the UDT definition increases.
  - Array length increases.
  - The memory type of the UDT definition changes.
  - There is a data type change in the UDT definition, except for the following interchangeable data types: WORD, INT, UINT, DWORD, DINT
  - The UDT definition is renamed.
- Modifying the data type or array dimensions of a symbolic variable. Exception: reducing the size of a one-dimensional array is supported during an RMS.
- Changes to PAC Productivity Suite function block configuration parameters while in offline mode are not processed when changing to online mode. However, creating a new function block instance will RMS the configuration parameter changes.
- Deleting, renaming, or moving a PACSystems user-defined function block input parameter, output parameter, or *member variable*. (See Setting a function block's input and output parameters and member variables.)

  Note: Adding an input or output parameter may also prevent RMS.
- Setting a C program's parameters (Series 90-70 targets, firmware version 6.00 or later.)
- Configuring the execution scheduling of a program (Series 90-70 targets, firmware version 6.00 or later.)

PME to PLC Communications During RMS

Communication between PME and the PLC is done over Ethernet using the Service Request Protocol (SRP) over TCP/IP.

SRP Protocol is an Emerson Proprietary protocol that is also used to communicate between subsystems inside an Emerson PACSystems Controller.

For low-level information, please consult PACSystems CPU Programmer's Reference Manual (GFK-2950). The section entitled *Service Request Function* has detailed information about this protocol.

The following sections will provide a high-level description of RMS functionality.

SRP Block Manage Request and User Logic Overview

Storing User Logic on the PACSystems PLC while the PLC is in RUN mode requires the use of Block Manage requests. If the PLC is in RUN mode when new User Logic is being stored, the newly stored User Logic will not be activated (to replace the previously running User Logic) until the end Block Manage request is received. This provides for a "bumpless" transition to the modified logic in run mode. In addition, if a failure should occur before the end Block Manage request, the PLC would at worst only lose what was being downloaded, and not would lose the currently executing user language. The following topics introduce logic store techniques.

Word-For-Word (WFW) Store

WFW is intended to store only a small amount of data in order so that the change can be performed very quickly. It is expected that the store service request used in the WFW change will allow for the modification of a file instead of the storing of a complete file. The source file(s) in the *Aux* directory must have the modified source indicated in some way so that a subsequent load of this folder will retrieve the correct source for the WFW modified logic. This may be accomplished with small changes to the existing source files or some form of a diff file added to the source filegroup. The details of the source file structure will be specified in the detailed sub-system designs.

All changes for a WFW must be made within the block manage sequence. The activation of the changes will be made atomically when the activation and cleanup request is received by the PLC.

Task

Provide PACSystems PLC with all WFW changed logic file sections for a folder.

Prerequisites

- Full Login established
- Privilege level 4 obtained
- OEM protection disabled
- PLC in RUN mode and logic was equal at the beginning of the logic edit.
- No other block manage in-process, is not allowed to perform any part of a WFW sequence during a run-mode-store or test-edit sequence.

Steps

1. Send a start **HOLD_BLOCKS BLOCK MANAGE SEQUENCE** request to PLC.
2. For each file at the block node that has changed from the PLC content, write a changed file section to the PLC with a path of */ram/<targetname>/logic/<program name>/<block name>/ <filename>.< extension >* making use of the **WRITE_ADDR_VAR** service request.

3. For each file at the logic node that has changed from the PLC content, write a changed file section to the PLC with a path of /*ram*/<*targetname*>/*logic*/<*file name*>.<*extension* > making use of the **WRITE_ADDR_VAR** service request.
4. For each source file at the Aux node that has changed from the PLC content, write a changed file section to the PLC with a path of /*ram*/<*targetname*>/*Aux*/<*file name*>.<*extension* > making use of the **WRITE_ADDR_VAR** service request.
5. Send an end **ACTIVATE_AND_CLEANUP BLOCK MANAGE SEQUENC**E request to PLC.

Results

On a successful WFW write of logic, the PACSystems PLC will contain the updated logic binary and source if the source option is selected. Activation of the new logic will take place after successfully receiving the ACTIVATE_AND_CLEANUP BLOCK MANAGE SEQUENCE request.

If an error should occur during the store of logic (inside the begin/end manage sequence), then the logic files currently active remain active.

Test-Edit Run-Mode Store

Task

Provide PACSystems PLC with changed logic files for a folder.

Prerequisites

- Full Login established
- Correct privilege level obtained – Level 4
- OEM protection disabled
- PLC in RUN mode

Steps

1. Send a start **HOLD_BLOCKS** BLOCK MANAGE SEQUENCE request to PLC.
2. For each file at the logic node that has changed from the PLC content, store it to the PLC with a path of /*ram*/<*targetname*>/*logic*.
3. For each file at one of the program nodes that have changed from the content in the PLC, store with a path of /*ram*/<*targetname*>/*logic*/<*program-name*>
4. For each file at an LD-Program block node that has changed from the content in the PLC, store with a path of /*ram*/<*targetname*>/*logic*/<*program-name*>/<*block-name*>
5. For each file in the original PLC logic that is not in the new modified logic, call the delete file Service Request. (The PLC needs to mark these files as to be deleted when the cleanup occurs.).
6. The Programmer then sends **ACTIVATE_AND_HOLD** BLOCK MANAGE SEQUENCE request to PLC, which activates the new logic data for testing and saves the old data in an inactive state.

   (**Optional Step 6**)

   The Programmer may now send **ACTIVATE_AND_HOLD** BLOCK MANAGE SEQUENCE request to PLC that restores the original logic data and saves the new data in an inactive state.

7. The Programmer then sends **CLEANUP_BLOCKS** BLOCK MANAGE SEQUENCE request to PLC that deletes any inactive logic data that is currently in the inactive state. This completes the Test-edit sequence and the resulting logic in the PLC will either be the original logic or the new tested logic depending on how many times the above optional step was executed.

Results

On a successful RMS Test-edit store of logic, the PACSystems PLC will contain the updated logic. Activation of the new logic will take place when successfully receiving the cleanup Manage Sequence Control request.

If an error should occur during the store of logic (inside the begin/activate manage sequence), then the logic files currently active remains active and the inactive logic is discarded.

If another HOLD_BLOCKS is issued after ACTIVATE_AND_HOLD and before CLEANUP_BLOCKS of a previous test-edit, the service request is rejected with an error message, and the state of Test-Edit is not changed.

If a power cycle occurs while in Test-Edit mode and there is currently inactive logic, the in-active logic will be retained on power-up. The programmer may obtain the status of an in-completed Test-Edit sequence by doing a Ret Control Service request. This should have a status field that indicates that there are inactive blocks in the PLC due to an interrupted Test-Edit.

Run-mode Store (without Test and Edit)

Task

Provide PACSystems PLC with all changed logic files for a folder.

Prerequisites

- Full Login established
- Correct privilege level obtained – Level 4
- OEM protection disabled
- PLC in RUN mode

Steps

1. Send **HOLD_BLOCKS** BLOCK MANAGE SEQUENCE request to PLC.
2. For each file at the logic node that has changed from the PLC content, store it to the PLC with a path of /ram/<targetname>/logic.
3. For each file at one of the program nodes that have changed from the content in the PLC, store with a path of /ram/<targetname>/logic/<program-name>
4. For each file at an LD-Program block node that has changed from the content in the PLC, store with a path of /ram/<targetname>/logic/<program-name>/<block-name>
5. For each file in the original PLC logic that is not in the new modified logic, call the delete file Service Request. (The PLC needs to mark these files as to be deleted when the cleanup occurs.) If the Programmer decides to abort the RMS store then send an **ABORT** BLOCK MANAGE SEQUENCE request to PLC
6. Otherwise
7. Send **ACTIVATE_AND_CLEANUP** BLOCK MANAGE SEQUENCE request to PLC to activate the new store logic data.

Results

On a successful RMS store of logic, the PACSystems PLC will contain the updated logic. Activation of the new logic will take place after successfully receiving the Activate_and_CLeanup request.

If an error should occur during the store of logic (inside the begin/activate manage sequence), then the logic files currently active remains active and the inactive logic is discarded. The PLC will be in RUN MODE and EQUAL to the Programmer.

PME Connection Loss During RMS to User Flash

Activate User Flash Files After Connection Loss

Method

1. With PME connected over Lan 1 and the debugger connected over Lan 2, perform a word-for-word-change.
2. Break during call to fsRmsFlashActivateInactiveFile() or any other functions occurring during STX_RMS_TO_USER_FLASH_ACTIVATE_FILE.
3. Disconnect Lan 1.
4. Reconnect PME after two minutes. (This is the default time it takes for the programmer lock to expire.)
5. PME will complain that the logic is not equal. Press **verify equality** logic is now equal.
6. Power cycle the board.
7. Re-connect PME. If the board is in run mode and the logic is equal, then no code needs to be written to recover from a connection loss.

Backup User Files After Connection Loss

Method

1. With PME connected over Lan 1and the debugger connected over Lan 2, perform a Word-for-Word-Store.
2. Break during backup to user flash files.
3. Disconnect Lan 1.
4. Reconnect PME after two minutes. (This is the default time it takes for the programmer lock to expire.)
5. PME will complain that the logic is not equal. Press **verify equality** logic is now equal.
6. Power cycle the board.
7. Re-connect PME. If the board is in run mode and the logic is equal, then no code needs to be written to recover from a connection loss.

Backup User Flash Files After Connection Loss

Method

1. With PME connected over Lan 1and the debugger connected over Lan 2, perform a Word-for-Word-Store.
2. Break during cleanup user flash files.
3. Disconnect Lan 1.
4. Reconnect PME after 2 minutes (default time it takes for programmer lock to expire)
5. PME will complain that the logic is not equal. Press **verify equality** logic is now equal.
6. Power cycle the board.
7. Re-connect PME. If the board is in run mode and the logic is equal, then no code needs to be written to recover from a connection loss.

Abort RMS to User Flash After Connection Loss

Method

1. PME connected over Lan1, debugger connected over lan2.
2. Perform a Word-for-Word change.
    a. Test and edit also work as well as loss of equality.
3. Break during abort user flash files (hardcoded a **returnValue** to fail).
4. Disconnect Lan 1.
5. Reconnect PME after 2 minutes (default time it takes for programmer lock to expire)
6. PME will complain that the logic is not equal. Press **verify equality** logic is now equal.
7. Power cycle the board.
8. Re-connect PME. If the board is in run mode and the logic is equal, then no code needs to be written to recover from a connection loss.

Store Inactive User Flash Files PME After Connection Loss

1. PME connected over Lan1, debugger connected over lan2.
2. Perform an RMS to User flash operation
3. Break during store inactive flash files state.
4. Disconnect Lan 1.
5. Reconnect PME after 2 minutes (default time it takes for programmer lock to expire)
6. PME says logic is not equal. Press 'verify equality' logic is now equal.
7. Power cycle the board.
8. Re-connect PME. If the board is in run mode and the logic is equal, then no code needs to be written to recover from a connection loss

4.5.4 RUN Mode Store to User Flash

PACSystems controllers are capable of using RUN mode Store to User Flash, which enables customers to store the Run Mode Store Operation done in User RAM into User Flash as well:

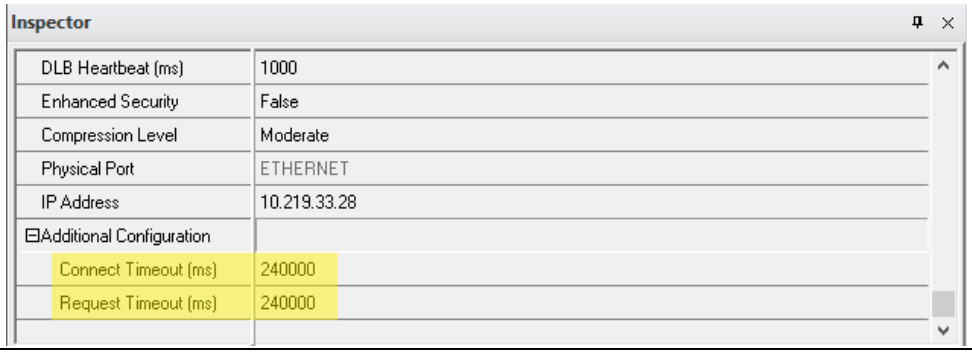
- This will keep User RAM and User Flash equal after Run Mode Store Operation.
- Current Run Mode Store Operations and its limitations stay as documented.

Note: RDSD will be deliberately disabled during RMS.

Restrictions and Assumptions

- Programmer operations will be prevented during the RUMS to User Flash
- Performing a large RMS using Flash can take up to 3 and a half minutes to complete depending on the size of the download and the sweep time. If performing a large download with a high sweep time the user must set the PME connection timeout and request timeout to their maximum 240 seconds to avoid a disconnect.

Figure 40: Connect Timeout and Request Timeout



4.6 Flash Memory Operation

The CPU stores the current configuration and application in user memory (either battery-backed RAM or non-volatile user memory, depending on the CPU model). You can also store the Logic, Hardware Configuration, and Reference Data in non-volatile flash memory. The PACSystems CPU provides enough flash memory to hold all of the userspace, all reference tables that aren't counted against userspace, and any overhead required. For details on which items count against user memory space, refer to Appendix B User Memory Allocation

By default, the CPU reads program logic and configuration, and reference table data from user memory at power-up. However, logic/configuration and reference tables can each be configured to always read from flash or conditionally read from flash. To configure these parameters in the programming software, select the CPU's Settings tab in Hardware Configuration.

If logic/configuration and/or reference tables are configured for conditional power-up from flash, these items are restored from flash to user memory when the user memory is corrupted or was not preserved (for example, the memory backup battery or Energy Pack is not installed or not operational). If logic/configuration and/or reference memory are configured for conditional power-up from flash and user memory has been preserved, no flash operation will occur.

If logic/configuration and/or reference tables are configured to always power up from flash, these items are restored from flash to user memory regardless of the state of the user memory.

Note: If ***any*** component (logic/configuration or reference tables) is read from flash, OEM-mode and passwords are also read from flash.

- In addition to configuring where the CPU obtains logic, configuration, and data during power-up, the programming software provides the following flash operations:
- Write a copy of the current configuration, application program, and reference tables (including overrides) to flash memory. Note that a write-to-flash operation causes all components to be stored to flash.
- Read a previously stored configuration and application program, and/or reference table values from flash into user memory.
- Verify that flash and user memory contain identical data.
- Clear flash contents.

Flash read and write operations copy the contents of flash memory or user memory as individual files. The programming software displays the progress of the copy operation and allows you to cancel a flash read or write operation during the copy process instead of waiting for the entire transfer process to complete. The entire user memory image must be successfully transferred for the flash copy to be considered successful. If an entire write-to-flash transfer is not completed due to canceling, power cycle, or some other intervention, the CPU will clear flash memory. Similarly, if a read-from-flash transfer is interrupted, user memory will be cleared.

4.6.1 RUN/STOP Switch Operation

The RUN/STOP switch is a 3-position switch that operates as follows:

Switch Position	CPU and Sweep Mode	Memory Protection
RUN I/O or RUN I/O Enable	The CPU runs with I/O sweep enabled.	User program memory is read-only.
RUN or RUN Output Disable	The CPU runs with outputs disabled.	User program memory is read-only.
STOP	The CPU is not allowed to go into RUN Mode.	User program memory can be written.

The RUN/STOP Switch can be disabled in the programming software HWC. The memory protection function of the switch can be disabled separately in HWC. The RUN/STOP switch is enabled by default. The memory protection functionality is disabled by default.

The Read Switch Position (Switch_Pos) function allows the logic to read the current position of the RUN/STOP Switch, as well as the mode for which the switch is configured. For details, refer to *PACSystems RX3i and RSTi-EP CPU Programmer's Reference Manual,* GFK-2950.

4.7 Logic/Configuration Source and CPU Operating Mode at Power-Up

Flash and user memory can contain different values for the Logic/Configuration Power-up Source parameter. The following tables summarize how these settings determine the logic/configuration source after a power cycle. CPU mode is affected by the Power-up Mode, the RUN/STOP Switch and Stop-Mode I/O Scanning parameters, the physical RUN/STOP Mode Switch position, and the Power Down Mode as shown in section 0.

Before Power Cycle		After Power Cycle	
Logic/Configuration Power-up Source in Flash	Logic/Configuration Power-up Source in RAM	Origin of Logic/Configuration	CPU Mode
Always Flash	Memory not preserved (i.e. no battery/Energy Pack, or memory corrupted)	Flash	See *CPU* Mode when Memory Not Preserved/Power-up Source is Flash
Always Flash	No configuration in RAM, memory preserved	Flash	See *CPU Mode when* Memory Preserved.
Always Flash	Always Flash	Flash	
Always Flash	Conditional Flash	Flash	
Always Flash	Always RAM	Flash	
Conditional Flash	Memory not preserved (i.e. no battery/Energy Pack or memory corrupted)	Flash	See *CPU* Mode when Memory Not Preserved/Power-up Source is Flash
Conditional Flash	No configuration in RAM, memory preserved	Uses default logic/configuration	Stop Disabled
Conditional Flash	Always Flash	RAM	See *CPU Mode when* Memory Preserved.
Conditional Flash	Conditional Flash	RAM	
Conditional Flash	Always RAM	RAM	
Always RAM	Memory not preserved (i.e. no battery/Energy Pack, or memory corrupted)	Uses default logic/configuration	Stop Disabled
Always RAM	No configuration in RAM, memory preserved	Uses default logic/configuration	Stop Disabled
Always RAM	Always Flash	Flash	See *CPU Mode when* Memory Preserved.
Always RAM	Conditional Flash	RAM	
Always RAM	Always RAM	RAM	
No Configuration in Flash	Memory not preserved (i.e. no battery/Energy Pack, or memory corrupted)	Uses default logic/configuration	Stop Disabled
No Configuration in Flash	No configuration in RAM, memory preserved	Uses default logic/configuration	Stop Disabled
No Configuration in Flash	Always Flash	RAM	See *CPU Mode when* Memory Preserved.
No Configuration in Flash	Conditional Flash	RAM	
No Configuration in Flash	Always RAM	RAM	

4.7.1 CPU Mode when Memory Not Preserved/Power-up Source is Flash

Configuration Parameters		RUN/STOP Switch Position	CPU Mode
Power-up Mode	RUN/STOP Switch		
Run	Enabled	Stop	Stop Disabled
Run	Enabled	Run Disabled	Run Disabled
Run	Enabled	Run Enabled	Run Enabled
Run	Disabled	N/A	Run Disabled
Stop	N/A	N/A	Stop Disabled
Last	Enabled	Stop	Stop Disabled
Last	Enabled	Run Disabled	Run Disabled
Last	Enabled	Run Enabled	Run Disabled
Last	Disabled	N/A	Run Disabled

4.7.2 CPU Mode when Memory Preserved

Configuration Parameters			RUN/STOP Switch Position	Power Down Mode	CPU Mode
Power-up Mode	RUN/STOP Switch	Stop-Mode I/O Scanning			
Run	Enabled	Enabled	Stop	N/A	Stop Enabled
Run	Enabled	Disabled	Stop	N/A	Stop Disabled
Run	Enabled	N/A	Run Disabled	N/A	Run Disabled
Run	Enabled	N/A	Run Enabled	N/A	Run Enabled
Run	Disabled	N/A	N/A	N/A	Run Enabled
Stop	N/A	Enabled	N/A	N/A	Stop Enabled
Stop	N/A	Disabled	N/A	N/A	Stop Disabled
Last	Enabled	Enabled	Stop	Stop Disabled	Stop Disabled
Last	Enabled	Enabled	Stop	Stop Enabled	Stop Enabled
Last	Enabled	Enabled	Stop	Run Disabled	Stop Enabled
Last	Enabled	Enabled	Stop	Run Enabled	Stop Enabled
Last	Enabled	Disabled	Stop	N/A	Stop Disabled
Last	Enabled	N/A	Run Disabled	Stop Disabled	Stop Disabled
Last	Enabled	Enabled	Run Disabled	Stop Enabled	Stop Enabled
Last	Enabled	Disabled	Run Disabled	Stop Enabled	Stop Disabled
Last	Enabled	N/A	Run Disabled	Run Disabled	Run Disabled
Last	Enabled	N/A	Run Disabled	Run Enabled	Run Disabled
Last	Enabled	N/A	Run Enabled	Stop Disabled	Stop Disabled
Last	Enabled	Enabled	Run Enabled	Stop Enabled	Stop Enabled
Last	Enabled	Disabled	Run Enabled	Stop Enabled	Stop Disabled
Last	Enabled	N/A	Run Enabled	Run Disabled	Run Disabled
Last	Enabled	N/A	Run Enabled	Run Enabled	Run Enabled
Last	Disabled	N/A	N/A	Stop Disabled	Stop Disabled
Last	Disabled	Enabled	N/A	Stop Enabled	Stop Enabled
Last	Disabled	Disabled	N/A	Stop Enabled	Stop Disabled
Last	Disabled	N/A	N/A	Run Disabled	Run Disabled
Last	Disabled	N/A	N/A	Run Enabled	Run Enabled

4.8 Clocks and Timers

Clocks and timers provided by the CPU include an elapsed time clock, a time-of-day clock, and software and hardware watchdog timers.

For information on timer functions and timed contacts provided by the CPU instruction set, refer to *Timers* in *PACSystems RX3i and RSTi-EP CPU Programmer's Reference Manual,* GFK-2950.

4.8.1 Elapsed Time Clock

The elapsed time clock tracks the time elapsed since the CPU powered on. The clock is not retentive across a power failure; it restarts on each power-up. This seconds count rolls over (seconds count returns to zero) approximately 100 years after the clock begins timing.

Because the elapsed time clock provides the base for system software operations and timer function blocks, it may not be reset from the user program or the programmer. However, the application program can read the current value of the elapsed time clock by using Service Request #16 or Service Request #50, which provides higher resolution.

4.8.2 Time-of-Day Clock

A hardware time-of-day clock maintains the time-of-day (TOD) in the CPU. The time-of-day clock maintains the following time functions:

- Year (two digits)
- Month
- Day of month
- Hour
- Minute
- Second
- Day of week

The TOD clock is battery-backed and maintains its present state across a power failure. The time-of-day clock handles month-to-month and year-to-year transitions and automatically compensates for leap years through the year 2036.

You can read and set the hardware TOD time and date through the application program using Service Request function #7. For details, refer to *PACSystems RX3i and RSTi-EP CPU Programmer's Reference Manual,* GFK-2950 Section 6.

High-Resolution Time of Day Software Clock

A high-resolution software TOD clock is implemented in firmware to provide nanosecond resolution. When the high-resolution software TOD clock is set, the hardware TOD clock is set with the YYYY: Mon: Day: Hr: Min: Sec fields in the POSIX[74] time, the RTC is read, and the delta between the POSIXtime and the value read from the RTC is computed and saved. Thus, if the 1-second resolution is desired the hardware TOD clock is read. Otherwise, the high-resolution software TOD clock is read to provide greater resolution. When the latter occurs, the hardware RTC is read and the saved delta is added to the value read.

[74] *RSTi-EP CPE100/CPE115 does not allow the Time-of-Day clock to be set older than 1st Jan, 2001 when POSIX format is used along with SVC_REQ 7.*

When the SNTP Time Transfer feature is implemented, all SNTP time updates received at the CPU will cause the high-resolution software TOD clock to be updated.

Synchronizing the High-resolution Time of Day Clock to an SNTP Network Time Server

In an SNTP system, a computer on the network (called an SNTP server) sends out a periodic timing message to all SNTP-capable Ethernet Interfaces on the network, which synchronize their internal clocks with this SNTP timing message. If SNTP is used to perform network time synchronization, the time-stamp information typically has ±10ms accuracy between controllers on the same network.

Synchronizing the CPU TOD clock to an SNTP server allows you to set a consistent time across multiple systems. Once the CPU TOD clock has been synchronized with the SNTP time, all produced EGD exchanges will use the CPU TOD current value for the time-stamp.

The CPU TOD clock is set with accuracy within ±2ms of the SNTP time-stamp.

TOD clock synchronization is enabled on an Ethernet module by the advanced user parameter (AUP), *ncpu_sync*. The CPU must also use a COMMREQ in user logic to select an Ethernet module as the time master. For additional information, refer to *Time-stamping of Ethernet Global Data Exchanges* in *PACSystems RX3i and RSTi-EP TCP/IP Ethernet Communications User Manual,* GFK-2224 Section 4.

4.8.3 Watchdog Timer

Software Watchdog Timer

A software watchdog timer in the CPU is designed to detect *failure to complete sweep* conditions. The timer value for the software watchdog timer is set by using the programming software. The allowable range for this timer is 10 ms to 2550 ms; the default value is 200 ms. The software watchdog timer always starts from zero at the beginning of each sweep.

The software watchdog timer is useful in detecting abnormal operation of the application program that prevents the CPU sweep from completing within the user-specified time. Examples of such abnormal application program conditions are as follows:

- Excessive recursive calling of a block
- Excessive looping (large loop count or large amounts of execution time for each iteration)
- Infinite execution loop

When selecting a software watchdog value, always set the value higher than the longest expected sweep time to prevent accidental expiration. For Constant Sweep mode, allowance for over sweep conditions should be considered when selecting the software watchdog timer value.

Refer to *EGD Sweep Impact for RX3i CPE302/CPE305/CPE310 and RSTi-EP CPE100/CPE115 Embedded Ethernet Interface*.

The watchdog timer continues during interrupt execution. Queuing of interrupts within a single sweep may cause watchdog timer expiration.

If the software watchdog timeout value is exceeded, the OK LED blinks, and the CPU goes to STOP-Halt mode[75]. Certain functions, however, are still possible. A fault is placed in the CPU fault table, and outputs go to their default state. The CPU will only communicate with the programmer; no other

[75] *For firmware versions 10.05 and later, all RX3i & RSTi-EP CPEs restart into STOP-Fault mode per the Secure Remote STOP-Halt Restart Mechanism.*

communications or operations are possible. To recover, power must be cycled on the rack or backplane containing the CPU.

To extend the current sweep beyond the software watchdog timer value, the application program may restart the software watchdog timer using Service Request function #8. However, the software watchdog timer value may only be changed from the configuration software.

Note that Service Request Function #8 does not reset the output scan timer implemented on the Genius Bus Controller.

Note: The hardware watchdog is not the same as the Software watchdog. A hardware watchdog is a timer circuit that is reset by a periodic high priority firmware task. It is different on all products and is part of the hardware design. Software watchdog is the only CPU parameter that is user-configurable, hardware watchdog is invisible to users.

Hardware Watchdog Timer

A backup circuit provides additional protection for the CPU. If this backup circuit activates, the CPU is immediately Reset. Outputs go to their default states, no communications of any kind are possible, and the CPU halts. The recovery procedure is documented below.

There are two basic forms of hardware watchdog:

1) for RX3i CPE302, CPE305, CPE310, CPE330, CPE400, CPL410 and RSTi-EP CPE100/CPE115/CPE205/CPE210/CPE215/CPE220/CPE240, a watchdog reset results in:
    - an automatic restart into STOP-Halt mode for firmware versions before 10.05.
    - an automatic restart into STOP-Fault mode per the secure remote STOP-Halt restart mechanism in firmware versions 10.05 and later.
2) for RX3i CPU310, CPU315, CPU320, and all CPUs, the watchdog reset holds the CPU in reset until the next power cycle. There is no automatic restart. If a charged battery is connected, the power cycle will result in a restart into STOP-Halt mode.

For both watchdog reset types, the CPU is power cycled after the energy pack (for the CPE models), or battery (for the other models listed) has been removed. This procedure gets the CPU out of STOP-Halt. The backup power source should then be reconnected.

RX3i CPU Response to a Hardware Watchdog Timeout:

The following responses to a hardware watchdog timeout are common to all RX3i CPU and CPE models:

- While the CPU/CPE is in STOP-Halt mode, you can connect the programmer software or PacsAnalyzer to view the fault tables, including any faults logged before the timeout. (See below for distinctions between CPU and CPE behavior.) The PacsAnalyzer software is a tool that is embedded in PME. It can also be downloaded from Emerson's support website.
- During startup following hardware watchdog reset, the CPU/CPE logs an *informational fault* with Error Code 446, which indicates a watchdog auto-reset occurred.

The following responses to a hardware watchdog timeout are different between RX3i CPU and RX3i CPE models:

- RX3i CPU310, CPU315, and CPU320 retain Controller and I/O Fault tables after a hardware watchdog timeout.

- RX3i CPE302, CPE305, CPE310, CPE330, CPE400, CPL410, and RSTi-EP CPE100/CPE115/CPE205/CPE210/CPE215/CPE220/CPE240 do not retain Controller and I/O Fault tables following a hardware watchdog timeout.

Note: PACSystems does not support Fatal Fault Retries.

4.9 System Security

PACSystems CPUs support two types of system security:

- Passwords/privilege levels
- OEM protection

CPU versions 7.80 and later support Enhanced Security (including merged password tables). This provides a more secure mechanism for setting and authenticating passwords and OEM keys versus the Legacy Security Mode. Refer to the *Important Product Information* document for the CPU model and firmware version that you are using for any additional information.

For Enhanced Security operation, see 4.9.3, Enhanced Security for Passwords and OEM Protection. A summary of operational differences between Enhanced and Legacy Security modes is provided in Section

Legacy/Enhanced Security Comparison.

4.9.1 Passwords and Privilege Levels - Legacy Mode

PACSystems CPUs are equipped with password management. Passwords are disabled by default, but can be enabled, disabled, or configured with PAC Machine Edition (PME). When in Online mode, PLCs can be password protected at varying privilege levels (1-4). (Passwords are not enabled when the PLC is in Offline mode.) Passwords are unique to the PLC (and subsequently each access level) but will be shared between users. Unique passwords cannot be assigned to individual users.

NOTE:

- Duplicate passwords may be used on different privilege levels. For example, Level 1 and Level 2 may share the same password.
- Passwords must be between one and seven ASCII characters in length.

After passwords have been configured, access to the CPU data will be restricted to the user's privilege level. Higher privilege levels will provide access to lower privilege levels. For example, users with a Level 4 Privilege Level will be able to access privileges at *all* privilege levels.

Note: The RUN/STOP Switch on the CPU overrides password protection. Even though the programmer may not be able to switch between RUN and STOP Mode, the switch on the CPU can do so.

Privilege Levels

There are four different privilege levels. Level 1 provides the least access and Level 4 provides the most access. The current privilege level is identified by the padlock icon on the bottom row of the PME screen. (For example, the icon will display that the PLC is in privilege Level 2). Please see Table 4-1 for a description of CPU Privilege Levels.

Table 4-1: CPU Privilege Levels

Level	Password	Access Description
4	Yes	Write to configuration or logic. Configuration may only be written in STOP Mode; logic may be written in STOP Mode or RUN Mode. Set or delete passwords for any level. *Note:* This is the default privilege for a connection to the CPU if no passwords are defined.
3	Yes	Write to configuration or logic when the CPU is in STOP Mode, including word-for-word changes, addition/deletion of program logic, and the overriding of discrete I/O.
2	Yes	Write to any data memory. This does not include overriding discrete I/O by applying a force. The CPU can be started or stopped. CPU and I/O Fault Tables can be cleared.
1	Yes	Read any CPU data except for passwords. This includes reading fault tables, performing datagrams, verifying logic/configuration, loading program, and configuration, etc. from the CPU. None of this data may be changed. At this level, RUN/STOP Mode transitions from the programmer are not allowed.

Note: If Enhanced Security is enabled, access to register memory space is restricted. See 4.9.3, Enhanced Security for Passwords and OEM Protection for more information.

Setting a Legacy Level 2 password may impact some PLC memory access operations. For instance, any attempts to write to the PLC memory via COMMREQ may no longer work since PME is unable to provide a password in a COMMREQ. Likewise, backplane modules and other remote protocol users may be similarly impacted when they attempt to access PLC memory due to a lack of the ability to provide a password for Level 2 permissions. For customers who wish to utlizie Privilege Level 2 passwords for programmer operations but still allow legacy devices unable of providing a password login, refer to the Legacy Client/Server Protocol Memory Access feature, which provides for allowing non-programmer connections to inherit Level 2 permissions so that they might perform certain Level 2 PLC operations on Data Memory without requiring a Level 2 password. See the “Legacy Client/Server Protocol Memory Access” Privilege Levels section below for more information.

Please refer to the table below for more information on the PLC operation restrictions in each privilege level (Yes – Allowed; No – Restricted).

Table 4-2: PLC Operation Categorized by PLC Privilege Level

PLC Operations	PLC mode	PLC Privilege Level			
		Level 1	Level 2	Level 3	Level 4
Write to Data Memory	RUN	No	Yes	Yes	Yes
	STOP	No	Yes	Yes	Yes
Read from Data Memory	RUN	Yes	Yes	Yes	Yes
	STOP	Yes	Yes	Yes	Yes
Discrete I/O override	RUN	No	No	Yes	Yes
	STOP	No	No	Yes	Yes
RUN/STOP transitions	RUN	No	Yes	Yes	Yes
	STOP	No	Yes	Yes	Yes
Verify Logic	RUN	Yes	Yes	Yes	Yes
	STOP	Yes	Yes	Yes	Yes
Verify Configuration	RUN	Yes	Yes	Yes	Yes
	STOP	Yes	Yes	Yes	Yes
Read Fault tables	RUN	Yes	Yes	Yes	Yes
	STOP	Yes	Yes	Yes	Yes
Clear fault tables	RUN	No	Yes	Yes	Yes
	STOP	No	Yes	Yes	Yes
Addition of program logic	RUN	No	No	No	Yes
	STOP	No	No	Yes	Yes
Deletion of program logic	RUN	No	No	No	Yes
	STOP	No	No	Yes	Yes
Download Logic	RUN	No	No	No	Yes
	STOP	No	No	Yes	Yes
Download Configuration	RUN	No	No	No	Yes
	STOP	No	No	Yes	Yes

Upload Logic	RUN	Yes	Yes	Yes	Yes
	STOP	Yes	Yes	Yes	Yes
Upload Configuration	RUN	Yes	Yes	Yes	Yes
	STOP	Yes	Yes	Yes	Yes
Set Passwords for any Level	RUN	No	No	No	Yes
	STOP	No	No	No	Yes
Delete Passwords for any Level	RUN	No	No	No	Yes
	STOP	No	No	No	Yes
Modbus Writes	RUN	No	No	Yes	Yes
	STOP	No	No	Yes	Yes
Modbus Reads	RUN	Yes	Yes	Yes	Yes
	STOP	Yes	Yes	Yes	Yes
SRTP Writes	RUN	No	Yes	Yes	Yes
	STOP	No	Yes	Yes	Yes
SRTP Reads	RUN	Yes	Yes	Yes	Yes
	STOP	Yes	Yes	Yes	Yes

“Legacy Client/Server Protocol Memory Access” Privilege Levels

This feature will be available on all current active models (CPE100/115, CPE205/CPE210/CPE215/CPE220/CPE240, CPE302/305/310, CPE330, and CPE400/CPL410). The parameter “Legacy Client/Server Protocol Memory Access” is available in PME in the CPU’s Hardware Config in the Settings tab.

When Legacy Client/Server Protocol Memory Access is configured as “Authenticated” (default value), the Controller’s operations do not change. All communications and connections operate as previously documented above.

When Legacy Client/Server Protocol Memory Access is configured as “Unauthenticated”, the Controllers operations change as follows:

- Enforces Level 2 Password for all programmer connections only
- Run/Stop Operations remain at Level 2 for programmer connections
- Allows non-programmer connections (backplane modules or other remote protocol users) to inherit Level 2 permissions
- Raises Run/Stop Operations to Level 3 for non-programmer connections

WhenLegacy Client/Server Protocol Memory Access is configured to “Unauthenticated,” non-programmer connections (backplane modules or other remote protocol users) automatically inherit

Level 2 permissions. This applies to EGD (EGD Class II CMDs), Modbus, SNP, SRTP, and DNP3, but does not change the operation of OPC UA.

This gives these other non-programmer connections permissions to perform the following PLC Operations without requiring a Level 2 password, for example:

- Write to Data Memory
- Modbus Writes
- SRTP Writes

Privilege Level Request from PAC Machine Edition

The CPU Privilege Level is configured by PME. In Legacy mode, upon connection to the CPU, PME will request the CPU to move to the highest non-protected level. The user can request a higher (or lower) privilege level by supplying the password for the desired privilege level in PME.

If the password sent by PME does not match the password stored in the CPU's password access table for the requested level, the privilege level change is denied, the current privilege level is maintained, and a fault is logged in the CPU fault table.

Note:

- A request to change to a privilege level that is not password protected is made by supplying the new level and a null password.

If no passwords are defined, the default privilege level for a connection to the CPU is Privilege Level 4.

If a password is defined for any privilege level, then the default access level for a connection to the CPU is set to one level below. (For example, if a password is defined for Level 4 then on the next connection, the PLC access level is set to Level 3, if a password is defined for Level 3 & Level 4, then on the next connection, the PLC Privilege Level is Level 2). Please refer the table Table 4-3 for details.

Table 4-3: Privilege Level After Password Config

The following table applies when passwords are defined with the following protocol conditions:

- Modbus TCP Protocol

Passwords Defined				Privilege Level on Connection
Level 1	Level 2	Level 3	Level 4	
No	Yes	Yes	Yes	Level 1
No	No	Yes	Yes	Level 2
No	No	No	Yes	Level 3
No	No	Yes	No	Level 2
No	Yes	No	No	Level 1

Table 4-4: Privilege Level After Password Config

The following table applies when passwords are defined with the following protocol conditions:

- PME Connection with SRTP Protocol

PME SRTP will connect and obtain the highest access level that has no password set.

Note: SRTP password levels are obtained on a per session basis. For example, obtaining Level 4 access on a PME connection does *NOT* automatically permit Level 4 access on any other SRTP session. Each SRTP Connection session must obtain its access level.

Passwords Defined				Privilege Level on Connection
Level 1	Level 2	Level 3	Level 4	
No	Yes	Yes	Yes	Level 1
No	No	Yes	Yes	Level 2
No	No	No	Yes	Level 3
No	No	Yes	No	Level 4
No	Yes	No	No	Level 4

Maintaining Passwords through a Power Cycle

Initial passwords are blank for a new controller or a controller that has its passwords cleared. For passwords to be maintained through power cycles, the controller must either:

- Store to RAM and use an Energy Pack or battery to maintain a memory.
- Store to User Flash with the configuration set up to load from Flash at power-up.

Disabling Passwords

The use of password protection is optional. Passwords can be disabled using the programming software.

Note: To enable passwords after they have been disabled, the CPU must be power cycled with the battery or Energy Pack removed.

4.9.2 OEM Protection – Legacy Mode

Original Equipment Manufacturer (OEM) protection provides a higher level of security than password levels 1 through 4. This feature allows a third-party OEM to create control programs for the CPU and then set the OEM-locked mode, which prevents the end-user from reading or modifying the program.

The OEM protection feature is enabled/disabled using a 1- to 7-character password, known as the *OEM key*. When OEM protection is enabled, all read and write access to the CPU program and configuration is prohibited: any store, load, verify, or clear user program operation will fail.

OEM Protection in Systems that Load from Flash Memory

For OEM protection, it is recommended to store the program to User Flash and set the configuration to always load from Flash. When setting up OEM protection it is important to download the user program to RAM and User Flash before enabling the OEM protection. For example, the following steps can be used to set up OEM protection.

1. Set OEM Key password (Must be at Access Level 4 to set OEM Key)
2. Download the program to both RAM and User Flash.
3. Set OEM Protection to the Locked state (see firmware note below).

If you are storing a non-blank OEM key to flash memory, you should be careful to record the OEM key for future reference. If disabling OEM protection, be sure to clear the OEM key that is stored in flash memory.

Note: In CPU firmware versions 7.80 or later which support Enhanced Security (with merged password tables), OEM Protection Lock must be explicitly set.

In earlier versions, the OEM Protection could be enabled in User Flash without explicitly setting the OEM Protection to Locked. With the earlier firmware, a non-blank OEM Key that is loaded from User Flash at power-up would result in an automatic OEM Lock. In CPU firmware versions 7.80 or later (i.e., with merged passwords), this is no longer supported.

In firmware versions earlier than 6.01, the OEM protection was not preserved unless a battery was attached.

4.9.3 Enhanced Security for Passwords and OEM Protection

Enhanced Security passwords provide a cryptographically secure password protocol between an SRTP client (for example, PAC Machine Edition) and a PACSystems controller. Enhanced Security passwords operate in a very similar fashion to the Legacy Security password operation that is supported by previous firmware versions.

Enhanced Security passwords are enabled in PME. PME requires a password to enable/disable the Enhanced Security mode of a target. This PME password restricts changes to the security mode used by a specific PME target and is independent of any passwords later configured on the controller. Basically, this password is only needed to enable/disable the Enhanced Security mode for a given target.

Upon enabling Enhanced Security (usage is optional), an Access Control List editor appears in the Hardware Configuration. This Access Control List allows you to modify the register set that is available for non-local consumption (HMI's, other controllers, etc.). When Enhanced Security mode is used, the default behavior is to make all register space **not** available for consumption.

- A user with an enhanced security project must publish the register space that needs to be accessed externally as Read Only or Read/Write.
- For Symbolic variables, the tags are added to the access control list via the Publish property, which has been extended to include a Read-Only and Read/Write setting.
- To allow logic block access to local memory (%L and %P), you must set the Local Memory Access property for each block to Read Only or Read/Write.
- External Read and Write requests to memory that is not specified in the access control list are rejected by the firmware.

Note: When requesting data from an external device, some drivers packetize data to optimize communication. If a request attempts to read a value that is not published, the entire packet will fail. A single fault is added to the fault table to help identify a failed read/write. After addressing the fault, you must clear the fault before attempting the request again.

Enabling Enhanced Security on a target does not force the controller to use only Enhanced Security. The controller supports both Legacy and Enhanced Security requests concurrently. For example, one PME target could be used to set initial passwords with Legacy security, and a different PME target with Enhanced Security could connect and authenticate with the same controller.

Passwords set with one password mechanism (Legacy or Enhanced Security) can be authenticated and changed using the other mechanism, as long as the password is 7 ASCII characters or fewer. Setting passwords with Enhanced Security that are more than 7 characters prevents access using the Legacy mechanism. For example, you could use Enhanced Security to set a 10-character ASCII password for Level 4 and Level 3 privileges but set a 7-character ASCII password for Level 2. In this case, a Legacy target could be used to obtain Level 2 privileges, but the Legacy target could never access Level 4 or Level 3 privileges because of the 7-character ASCII limit of the Legacy scheme.

Password and OEM Protection in Systems that Load from Flash Memory

CAUTION

Be careful when setting and loading passwords from User Flash on every power-up. In this situation, it is not possible to clear passwords back to a default state if the Level 4 password and OEM key are forgotten.

For a recommended procedure, see *OEM Protection in Systems that Load from Flash Memory*.

4.9.4 Legacy/Enhanced Security Comparison

Table 4-5: Legacy/Enhanced Security Comparison

Feature	Legacy (less secure)	Enhanced (more secure)
Level 2, 3, and 4 protection	Levels 2, 3, and 4 must be set or modified simultaneously. (If you only want to change one, you must enter all three.)	Passwords can be set individually or in a group. When changing passwords, the old password for that level is required to change.
Maximum password length	7 characters	31 characters
Clearing passwords	Passwords can be cleared back to initial blank password values.	Once a password is set, the Enhanced Security mode in PME will not allow it to be cleared back to a blank password. To revert to a blank password, the CPU memory must be cleared and power cycled.
Passwords ≤7 characters, set with either mode	Password verification and password changes are allowed.	Password verification and password changes are allowed.
Passwords >7 characters, set with Enhanced Security mode	Password verification and password change are *not* allowed.	Password verification and password changes are allowed.
Maximum OEM key length	7 characters.	31 characters.
OEM keys ≤7 characters, set with Enhanced Security	Can change OEM Protection Lock state *Cannot* change the OEM key.	Can change the OEM Protection Lock state and the OEM key.
OEM keys >7 characters, set with Enhanced Security	*Cannot* change OEM Protection Lock state or the OEM key.	Can change the OEM Protection Lock state and the OEM key.

4.10 PACSystems I/O System

The PACSystems I/O system provides the interface between the CPU and other devices. The PACSystems I/O system supports:

- I/O and Intelligent option modules.
- Ethernet Interface
- Motion modules (RX3i)

PROFINET:

- RX3i CPE330, CPE400, CPL410, and RSTi-EP CPE100/CPE115 all permit one of their LANs to be configured as an embedded PROFINET Controller (see Section 2.1.6). Alternately, the RX3i PROFINET Controller IC695PNC001 installs in the RX3i Main I/O Rack[76]. The embedded PROFINET Controller and the IC695PNC001 are used to control remote I/O drops. Refer to *PACSystems RX3i & RSTi-EP PROFINET IO-Controller Manual*, GFK-2571G or later. Some examples of remote drops are:
- Standard rack-mounted I/O modules in RX3i racks scanned by the PROFINET scanner IC695PNS001and IC695PNS101. Refer to *PACSystems RX3i PROFINET Scanner Manual*, GFK-2737.
- A mini-drop consisting of one or two I/O modules and supervised by the IC695CEP001. Refer to *PACSystems RX3i CEP PROFINET Scanner User Manual,* GFK-2883.
- A Genius Bus supervised by a Genius Communications Gateway (IC695GCG001). Refer to *PACSystems RX3i Genius Communications Gateway User Manual*, GFK-2892.

The Genius I/O System

- A Genius I/O Bus Controller (GBC) module provides the interface between the CPU and a Genius I/O bus. Refer to *Series 90-70 Genius Bus Controller User's Manual, GFK-2017.*
- RX3i: A Genius Communications Gateway (IC695GCG001) provides the interface between devices on the Genius I/O bus and a PROFINET Controller (IC695PNC001) which is installed in the RX3i Main I/O rack. Refer to *PACSystems RX3i Genius Communications Gateway User Manual*, GFK-2892.
- For information on Genius I/O, refer to Genius I/O System User's Manual, GEK-90486-1, and Genius I/O Analog and Discrete Blocks User's Manual, GEK-90486-2.

[76] *Note that RX3i CPE400 and RSTi-EP CPE100/CPE115 does not support IC695PNC001, since they are Standalone CPUs. Both may co-exist on a PROFINET LAN.*

4.10.1 I/O Configuration

Module Identification

In addition to the catalog number, the programming software stores a Module ID for each configured module in the hardware configuration that it delivers to the CPU. The CPU uses the Module ID to determine how to communicate with a given module.

When the hardware configuration is downloaded to the CPU (and during subsequent power-ups), the CPU compares the Module IDs stored by the programmer with the IDs of the modules physically present in the system. If the Module IDs do not match, a System Configuration Mismatch fault will be generated.

Because I/O modules of a similar type may share the same Module ID, it is possible to download a configuration containing a module catalog number that does not match the module that is physically present in the slot without generating a System Configuration Mismatch.

Certain discrete modules with both reference memory inputs and reference memory outputs will experience invalid I/O transfer if the incorrect configuration is stored from a similar mixed I/O module. No-fault or error conditions will be detected during the configuration store and the module will be operational, although not in the manner described by the configuration.

For example, a configuration swap between the IC693MDL754 output module and IC693MDL660 input module will not be detected as a configuration mismatch, but I/O data transfer between the module and the CPU reference memory will be invalid. If the input module (MDL660) is sent the configuration of the output module (MDL754) with the following parameters:

Reference Address: %Q601
Module Status Reference: %I33
Hold Last State Enable

It will receive inputs at the module status reference %I33 and the status of the module will be received at %Q601.

If the output module is sent the configuration of the input module with the following parameters:

Reference Address: %I601
Input Filter: Enable
Digital Filter Settings Reference: %I65

It will output values at the digital filter settings reference %I65 and the status of the module will be received at %I601.

Default Conditions for I/O Modules

Interrupts

Some input modules that can be configured tend to interrupt the application program. By default, this interrupt is disabled and the input filter is set to slow. If changed by the programming software, the new settings are applied when the configuration is stored and during subsequent power cycles.

Outputs

The output defaulting operation is affected by two types of configuration parameters. First, there is a head-end configuration that sets the outputs default operation of the entire backplane. This includes options for "All Off", "Substitute Values", or "Hold Last State." The selected action applies when the CPU transitions from RUN/Enabled to RUN/Disabled or STOP Mode, or experiences a fatal fault. If a module does not have a configurable output default mode, its output default mode is Off.

> **Note**: Some analog output modules can be configured with a jumper located on the removable terminal block of the module. The jumper may be set to cause outputs to either *default to zero* or *retain the last state*.

In addition, some output modules support a Substitute Value configuration parameter for the output channels of the module. This allows a specific value to be applied when Substitute Values are enabled. Output modules that do not support Substitute Values will always apply a 0 when outputs defaulted in the Substitute Value mode.

A module's Substitute Value parameters can be applied in several scenarios. In the case of Bus Comm Fail, the Substitute Values are applied regardless of the head-end configuration. For example, even if the head-end configuration sets "All Off" or "Hold Last State", then when a Bus Comm Failure occurs the Substitute Values are still applied. Firmware update reset of the PLC is one scenario that triggers Substitute Values. Substitute Values are also applied when a system is running normally and transitions from outputs enabled to outputs disabled state. For modules that support Substitute Value configuration, the configuration default is 0 (behaves like "All Off" unless a user changes the module's config).

A hardware configuration store or clear does not immediately change the current state of any module's output default value. To see an output default change following config store, one must transition to outputs enabled and back to outputs disabled. For example, if the configuration is currently in a "Hold Last State" mode with outputs disabled, then an output that is currently non-zero will not immediately change its value to 0 when a new configuration is downloaded with "All Off" mode. A user must transition to outputs enabled and back to disabled to see the new outputs' default behavior.

Head-end Config	Initial outputs state after power cycle or bus recovery	Outputs Enabled	Outputs Disabled (following Enabled)	Bus Comm Fail (includes PLC reset)
All Off	0	(output scan value)	0	Substitute Value if supported by module's config; otherwise 0.
Substitute Values	0	(output scan value)	Substitute Value if supported by module's config; otherwise 0.	
Hold Last State	0	(output scan value)	Last output scan value	

Inputs

Input modules that have a configurable input default mode can be configured to Hold Last State or to set inputs to 0. If a module does not have a configurable input default mode, its input default mode is Off. The selected action applies when the CPU transitions from RUN/Enabled to RUN/Disabled or STOP Mode, or experiences a fatal fault.

For details on the power-up and STOP Mode behavior of other modules, refer to the documentation for that module.

Multiple I/O Scan Sets

Up to 32 I/O scan sets can be defined for a PACSystems CPU. A scan set is a group of I/O modules that can be assigned a unique scan rate. A given I/O module can belong to one scan set. By default, all I/O modules are assigned to scan set 1, which is scanned every sweep.

For some applications, the CPU logic does not need to have the I/O information every sweep. The I/O scan set feature allows the scanning of I/O points to be more closely scheduled with their use in user logic programs. If you have a large number of I/O modules, you may be able to significantly reduce scan time by staggering the scanning of those modules.

A disadvantage of placing all modules into different scan sets appears when the CPU is transitioning from Stop to Run. In that case, scan sets with a programmed delay are not scanned on the first sweep. These modules' outputs are not enabled until the new data has been scanned to them, perhaps many scans later. Therefore, there is a period during which the user logic is executing and some modules' outputs are disabled. During that time, the outputs of those modules are in the module’s stop-mode state. Stop-mode behavior is module-dependent. Some modules zero their outputs, some hold their last scanned state (if any), and some force their outputs to a configured default value. When the module's outputs are enabled, the module uses the last scanned value, which will either be zero, or the contents of the register the module uses to hold the corresponding output values from the reference tables.

4.10.2 I/O System Diagnostic Data Collection

Diagnostic data in a PACSystems I/O system is obtained in either of the following two ways:

- If an I/O module has an associated bus controller, the bus controller provides the diagnostic data from that module to the CPU. For details on GBC faults, see *PACSystems Handling of GBC Faults*.
- For I/O modules not interfaced through a bus controller, the CPU's I/O Scanner subsystem generates the diagnostic bits based on data provided by the module.

The diagnostic bits are derived from the diagnostic data sent from the I/O modules to their I/O controllers (CPU or bus controller). Diagnostic bits indicate the current fault status of the associated module. Bits are set when faults occur and are cleared when faults are cleared.

Diagnostic data is not maintained for modules from other manufacturers. The application program must use the BUS Read function blocks to access diagnostic information provided by those boards.

Note: At least two sweeps must occur to clear the diagnostic bits: one scan to send the %Q data to the module and one scan to return the %I data to the CPU. Because module processing is asynchronous to the controller sweep, more than two sweeps may be needed to clear the bits, depending on the sweep rate and the point at which the data is made available to the module.

Discrete I/O Diagnostic Information

The CPU maintains diagnostic information for each discrete I/O point. Two memory blocks are allocated in application RAM for discrete diagnostic data, one for %I memory and one for %Q memory. One bit of diagnostic memory is associated with each I/O point. This bit indicates the validity of the associated I/O data. Each discrete point has a fault reference that can be interrogated using two special contacts: a fault contact (-[F]-) and a no-fault contact (-[NF]-). The CPU collects this fault data if enabled to do so by the programming software. The following table shows the state of the fault and no-fault contacts.

Condition	[FAULT]	[NOFLT]
Fault Present	ON	OFF
Fault Absent	OFF	ON

Analog I/O Diagnostic Data

Diagnostic information is made available by the CPU for each analog channel associated with analog modules and Genius analog blocks. One byte of diagnostic memory is allocated to each analog I/O channel. Since each analog I/O channel uses two bytes of %AI and %AQ memory, the diagnostic memory is half the size of the data memory.

The analog diagnostic data contains both diagnostics and process data with the process data being the High Alarm and Low Alarm bits. The diagnostic data is referenced with the -[F]- and -[NF]- contacts. The process bits are referenced with the high alarm (-[HA]- and low alarm (-[LA]-) contacts. The memory allocation for analog diagnostic data is one byte per word of analog input and analog output allocated by programming software. When an analog fault contact is referenced in the application program, the CPU does an Inclusive OR on all bits in the diagnostic byte, except the process bits. The alarm contact is closed if any diagnostic bit is ON and OFF only if all bits are OFF.

PACSystems Handling of GBC Faults

Defaulting of input data associated with failed/lost GBCs

When a GBC is missing, mismatched, or otherwise failed, the CPU applies the Input Default setting for each device on that Genius bus when defaulting the input data. If the device is configured for HOLD LAST STATE, the data is left alone. If the device is configured for OFF, the input data is set to 0. If a redundant GBC is operational, the input data is not affected.

Application of default input and diagnostic data for lost redundant blocks

When a GBC reports that a redundant block is lost, the CPU updates the input data tables and input diagnostic tables with the default data during the very next input scan. The output diagnostic data tables are updated during the very next output scan.

4.10.3 Power-Up and Power-Down Sequences

Power-Up Sequence

System power-up consists of the following parts:

- Power-up self-test
- CPU memory validation
- System configuration
- Intelligent option module self-test completion
- Intelligent option module dual-port interface tests
- I/O system initialization

Power-Up Self-Test

On system power-up, many modules in the system perform a power-up diagnostic self-test. The CPU module executes hardware checks and software validity checks. Intelligent option modules perform setup and verification of onboard microprocessors, software checksum verification, local hardware verification, and notification to the CPU of self-check completion. Any failed tests are queued for reporting to the CPU during the system configuration portion of the cycle.

If a low or failed battery (or Energy Pack fault) indication is present, a fault is logged in the CPU fault table.

CPU Memory Validation

The next phase of system power-up is the validation of the CPU memory. First, if the system verifies that user memory areas are still valid. A known area of user memory is checked to determine if data was preserved. Next, if a ladder diagram program exists, a checksum is calculated across the _MAIN ladder block. If no ladder diagram program exists, a checksum is calculated across the smallest standalone C program.

When the system is sure that the user memory is preserved, a known area of the bit cache area is checked to determine if the bit cache data was preserved. If this test passes, the Bit Cache memory is left containing its power-up values. (Non-retentive outputs are cleared on a transition from STOP Mode to RUN Mode.) If the checksum is not valid or the retentive test on the user memory fails, the bit cache memory is assumed to be in error and all areas are cleared. The CPU is now in a cleared state, the same as if a new CPU module were installed. All logic and configuration files must be stored from the programmer to the CPU.

System Configuration

After completing its self-test, the CPU performs the system configuration. It first clears all system diagnostic bits in the bit cache memory. This prevents faults that were present before power-down but are no longer present from accidentally remaining as faulted. Then it polls each module in the system for completion of the corresponding self-test.

The CPU reads information from each module, comparing it with the stored (downloaded) rack/slot configuration information. Any differences between the actual configuration and the stored configuration are logged in the fault tables.

Intelligent Option Module Self-Test Completion

Intelligent option modules may take a longer time to complete their self-tests than the CPU due to the time required to test communications media or other interface devices. As an intelligent option module completes its initial self-tests, it tells the CPU the time required to complete the remainder of these self-tests. During this time, the CPU provides whatever additional information the module needs to complete its self-configuration, and the module continues self-tests and configuration. If the module does not report back in the time specified, the CPU marks the module as faulted and makes an entry in one of the fault tables. When all self-tests are complete, the CPU obtains reports from the module as generated during that particular module's power-up self-test and places fault information (if any) in the fault tables.

Intelligent Option Module Dual Port Interface Tests

After completion of the intelligent option module self-test and results reporting, integrity tests are jointly performed on the dual-port interface used by the CPU and intelligent option module for communications. These tests validate that the two modules can pass information back and forth, as well as verify the interrupt and semaphore capabilities needed by the communications protocol. After dual-port interface tests are complete, the communications messaging system is initialized.

I/O System Initialization

If the module is an input module, no further configuration is required. If the module is an output module, the module is commanded to go to its default state. The output modules default to all outputs off at power-up and in failure mode unless configured otherwise.

A bus transmitter module is interrogated about what expansion racks are present in the system. Based on the bus transmitter module's response, the CPU adds those racks and their associated slots into the list of slots to be configured.

Finally, the I/O Scanner performs its initialization. The I/O Scanner initializes all the I/O controllers in the system by establishing the I/O connections to each I/O bus on the I/O controller and obtaining all I/O configuration data from that I/O controller. This configuration data is compared with the stored I/O configuration and any differences reported in the I/O Fault Table. The I/O Scanner then sends each I/O controller a list of the I/O modules to be configured on the I/O bus. After the I/O controllers have been initialized, the I/O Scanner replaces the factory default settings in all I/O modules with any application-specified settings.

Power-Down Sequence

System power-down occurs when the power supply detects that incoming power has dropped for more than 15ms.

Power Cycle Operation with an Energy Pack

Energy Packs offer distinct advantages over batteries:

a) significantly longer life cycles

b) they are more reliable

c) flammability during shipment is not an issue

d) in their end-of-life phase, their decline is a lot more gradual.

The system design includes the ability of the CPU and the Energy Pack to monitor each other in real-time. This permits the user to monitor alarms and thereby determine when to replace a capacitor pack. The capacitor pack is normally replaced while the CPU is powered on, giving it time to charge up before any subsequent loss of power. Users should target periods that are expected to be free from electrical events, such as thunderstorms, to carry out such work. Capacitor packs may also be replaced while power is off.

When power is lost, the Energy Pack supplies current and maintains voltage levels for a period of time sufficient to permit the connected CPU to save all dynamic memory to non-volatile memory.

When power is restored, the CPU will not start running its application until the Energy Pack signals that it is fully charged. The CPU will then resume operation using the contents of memory retained at the previous loss of power event. The Energy Pack charges continuously during normal operation.

The RX3i product lines encompass several different Energy Packs, so it is important to use compatible products:

CPU	IC695CPE400 IC695CPL410	IC695CPE330	IC695CPE302 IC695CPE305 IC695CPE310	ICRXICTL000
Energy Pack	IC695ACC403	IC695ACC402	IC695ACC400	ICRXIACCEPK01
Capacitor Pack	IC695ACC413	IC695ACC412	IC695ACC400	ICRXIACCCPK01
Connecting Cable	IC695CBL003	IC695CBL002	IC695CBL001	ICRXIACCCBL01
Documentation	GFK-3000	GFK-2939	GFK-2724	GFK-2741

User memory is preserved only if the compatible Energy Pack is connected (and charged) at power-down.

If the Energy Pack is connected at power-up, the CPU waits for it to charge up before beginning normal operations. For CPE330/CPE400/CPL410, this typically takes up to 90 seconds.

In the event the Energy Pack fails to charge up in a reasonable amount of time, or is absent, the CPU will time out the waiting period and will then commence operations without the Energy Pack. When this occurs, the CPU is vulnerable to loss of memory, should another power failure occur. It is critical to monitor the status bits shown in *Energy Pack Status Bit Operation* so that human intervention can be summoned.

Removing or reconnecting the Energy Pack while the connected CPU is powered off does not affect the preservation of user memory.

Note: Because the Time of Day (TOD) clock is powered by a separate Real Time Clock battery in CPE302/CPE305/CPE310/CPE330/CPE400/CPL410, the Energy Pack does not affect the CPU TOD value.

Energy Pack Status Bit Operation

As shown in the table below, the CPU application program can monitor the status of the attached Energy Pack via %S0014 (PLC_BAT) and %SA0011 (LOW_BAT). For more details, refer to the Section on Diagnostics in *PACSystems and RX3i CPU Programmer's Reference Manual,* GFK-2950.

PLC_BAT (%S0014)	LOW_BAT (%SA0011)	Energy Pack Status
0	0	Energy Pack connected and operational (may be charging)
1	1	Energy Pack is not connected or has failed
0	1	Energy Pack is nearing its end of life and should be replaced.

The LEDs on the Energy Pack also indicate its status. Refer to the documentation for each product for the corresponding LED status.

Energy Pack Replacement

If an Energy Pack fails, you can replace it while the CPU is in operation. Use a compatible new unit or a compatible replacement Cap Pack. Whenever an Energy Pack is replaced, the newly installed Cap Pack must build up its charge.

In the case of ACC402 attached to CPE330, the Energy Pack, when hot-swapped, draws minimal current to recharge: it may therefore take up to 10 minutes for ACC402 to charge completely. This is a normal operation. Similarly for ACC403/CPE400 and ACC403/CPL410.

If a loss of power occurs while the Energy Pack is disconnected, or before the capacitors are fully charged, memory loss may occur.

CPE330/ACC402 Status Detection & Fault Reporting

Both the CPE330 and ACC402 contain intelligence, allowing each to determine the status of the other. This permits the CPU to report various conditions to the user via the status bits discussed in *Energy Pack Status Bit Operation*.

Whenever the CPE330 detects any kind of issue with the ACC402 Energy Pack, it resumes normal operation and issues warnings or faults to the user. The table below details the various permutations possible at power-up. If the Cap Pack is removed during normal operation, this fault will be reported as a failed battery fault.

Energy Pack Base	Cap Pack	CPE330 power-up response
Not present (removed or failed)	Not present (removed or failed)	Detects missing ACC402 and boots up immediately but does not use any stored memory when resuming operations. Issues fail battery fault.
Not present (bad base)	Present (good cap pack)	Detects missing ACC402 and boots up immediately but does not use any stored memory when resuming operations. Issues fail battery fault.
Present (good base)	Not present (removed or failed)	When the CPU does not see a fully charged status within 90 seconds, it does not use any stored memory when resuming operations. Issues fail battery fault.
Present (good base)	Present (good cap pack)	The CPU will wait for a fully charged status within the timeout period and then resume operation using the contents of memory retained at the previous loss of power event
Present (suspicious)	Present (suspicious)	If the CPU does not see a fully charged status within 90 seconds, it does not use any stored memory when resuming operations. Issues fail battery fault.

CPE400/CPL410/ACC403 Status Detection & Fault Reporting

Both the CPU (CPE400 or CPL410) and ACC403 contain intelligence, allowing each to determine the status of the other. This permits the CPU to report various conditions to the user via the status bits discussed in *Energy Pack Status Bit Operation*.

Whenever the CPE400 or CPL410 detects any kind of issue with the ACC403 Energy Pack, it resumes normal operation and issues warnings or faults to the user. The table below details the various permutations possible at power-up. If the Cap Pack is removed during normal operation, this fault will be reported as a failed battery fault.

Energy Pack Base	*Cap Pack*	*CPE400/CPL410 power-up response*
Not present (removed or failed)	Not present (removed or failed)	Detects missing ACC403 and boots up immediately but does not use any stored memory when resuming operations. Issues fail battery fault.
Not present (bad base)	Present (good cap pack)	Detects missing ACC403 and boots up immediately but does not use any stored memory when resuming operations. Issues fail battery fault.
Present (good base)	Not present (removed or failed)	When the CPU does not see a fully charged status within 45 seconds, it does not use any stored memory when resuming operations. Issues fail battery fault.
Present (good base)	Present (good cap pack)	The CPU will wait for a fully charged status within the timeout period and then resume operation using the contents of memory retained at the previous loss of power event
Present (suspicious)	Present (suspicious)	If the CPU does not see a fully charged status within 45 seconds, it does not use any stored memory when resuming operations. Issues fail battery fault.

Retention of Data Memory Across Power Failure

The following types of data are preserved across a power cycle with an operational battery:

Note: RX3i CPE302, CPE305, CPE310, CPE330, CPE400, or CPL410 and RSTi-EP CPE100/115 require an attached energy pack.

The RSTI-EP CPE205/210/215/220/140 models are ale to preserve the following data without the need for an attached energy pack.

- Application program
- Fault tables and other diagnostic data
- Checksums on programs and blocks
- Override data
- Data in register (%R), local register (%L), and program register (%P) memory
- Data in analog memory (%AI and %AQ)
- State of discrete inputs (%I)
- State of retentive discrete outputs (%Q)
- State of retentive discrete internals (%M)

The following types of data are not preserved across a power cycle:

- State of discrete temporary memory (%T)
- %M and %Q memories used on non-retentive -()- coils
- State of discrete system internals (system bits, fault bits, reserved bits).
- CPL410 PACEdge data

Section 5 Communications

This Section describes the Ethernet and Serial communications features of the PACSystems CPU.

Ethernet communications may be handled by the embedded CPU Ethernet port(s) or by an IC695ETM001 module installed in an RX3i rack. Refer to *PACSystems RX3i TCP/IP Ethernet Communications User Manual,* GFK-2224.

Serial communications may be handled by the embedded CPU Serial port(s) or by an IC695CMM002 or IC695CMM004 module installed in an RX3i rack. Refer to *PACSystems RX3i Serial Communications Modules User's Manual,* GFK-2460.

This section contains the following information concerning the embedded CPU ports:

- Ethernet Communications
- Serial Communications

5.1 Ethernet Communications

For details on Ethernet communications for PACSystems, please refer to the following manuals:

PACSystems RX3i and RSTi-EP TCP/IP Ethernet Communications User Manual, GFK-2224
PACSystems TCP/IP Ethernet Communications Station Manager User Manual, GFK-2225.

5.1.1 Embedded Ethernet Interfaces

RX3i

RX3i CPE302, CPE305, CPE310, CPE330, CPE400, and CPL410 CPUs provide one or more embedded Ethernet interfaces. If used, each interface connects to a Local Area Network (LAN).

The corresponding RJ45 Ethernet port(s) automatically sense the data rate on the attached LAN (1 Gbps, 100 Mbps, or 10 Mbps), as well as the corresponding communication mode (half-duplex or full-duplex), and the corresponding cabling arrangement (straight-through or crossover). Automatic detection greatly simplifies installation procedures.

See RX3i CPU Features and Specifications to determine the complete list of Internet protocols supported by each CPU.

Some important protocols supported by all RX3i CPUs are:

- TCP/IP, which provides basic Internet capabilities;
- SRTP, which is proprietary and which provides the interface with the PME programming and configuration software and supports communications with certain control systems and supervisory computer layers in the factory;
- Modbus/TCP supports the Modbus messaging structure over the Internet.

On the CPE302/CPE305/CPE310-Axxx models, the same shared processor performs both Ethernet port processing and Controller logic processing.

On the CPE302/CPE305-Bxxx, CPE330, the dual-core CPU enables communication to be handled by one core while CPU logic and I/O scanning are handled by the second core. Furthermore, each LAN interface is controlled by a dedicated Network Interface Controller (NIC). In the CPE400 and CPL410, one of the four microprocessor cores is dedicated to handling the communications function (LAN1, LAN2, and LAN3).

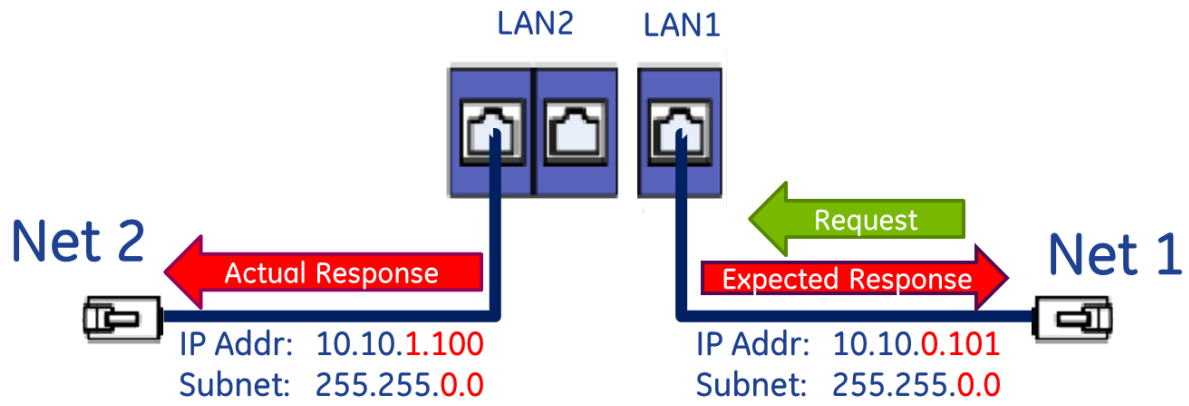
As a result of the hardware advances in the CPE330, CPE400, and CPL410, a higher level of processing power is provided in support of each LAN. This is especially important at higher data rates. It also offloads the handling of Ethernet-level activity from the processor core tasked with performing CPU logic and I/O scanning, permitting that core to run more efficiently.

Each interface on a LAN must have a unique IP Address *and* a non-overlapping IP subnet. This is configured in PME. Care must be taken to survey the entire connected network architecture to tabulate the IP addresses and IP subnets already in use, both on the local networks and any of its routed subnets connected with a gateway. Never assign a conflicting IP Address or configure duplicate IP subnets.

The following examples would be problematic:

Problem example #1:

Figure 41: CPE330 Overlapping Local IP Subnet Example



The issue demonstrated in Figure 41 is that requests entering one CPE330 interface can be routed out the other interface since both CPE330 Ethernet ports have been configured to be on the same network (255.255.0.0) but are physically connected to separate networks. Avoid this by assigning non-overlapping Subnets.

Problem example #2:

A user wishes to communicate through a routed network to an RX3i CPU with multiple network interfaces (CPE330, in this example). This remote IP device is configured with the following IP parameters:

IP	**192.168.0.5**
Subnet Mask	255.255.255.0
Gateway	192.168.0.250

LAN1 and LAN2 on the CPE330 are initially configured with the following problematic IP parameters:

	LAN1	LAN2
IP	10.10.0.1	192.168.0.1
Subnet Mask	255.255.255.0	255.255.255.0
Gateway	10.10.0.249	0.0.0.0

The user intends to communicate between the remote device and CPE330 LAN1 (Figure 42). IP Address routing allows the CPE330 to receive the remote IP requests through the respective gateways (192.168.0.250 for the remote node and 10.10.0.249 for CPE330 LAN1). However, since CPE330 LAN2 shares the same IP subnet as the remote network (192.168.0.x), responses may be routed to the local 192.168.0.x network rather than to the remote network (Figure 43).

The duplicate IP subnet in the example must be eliminated. One way to do this is simply to change the IP Address assigned to CPE330 LAN2 from 192.168.0.1 to 192.168.1.1 thereby creating a non-overlapping 192.168.1.x network. In short, consider the totality of the network when assigning IP subnets and IP Addresses.

Figure 42: Expected Response Path

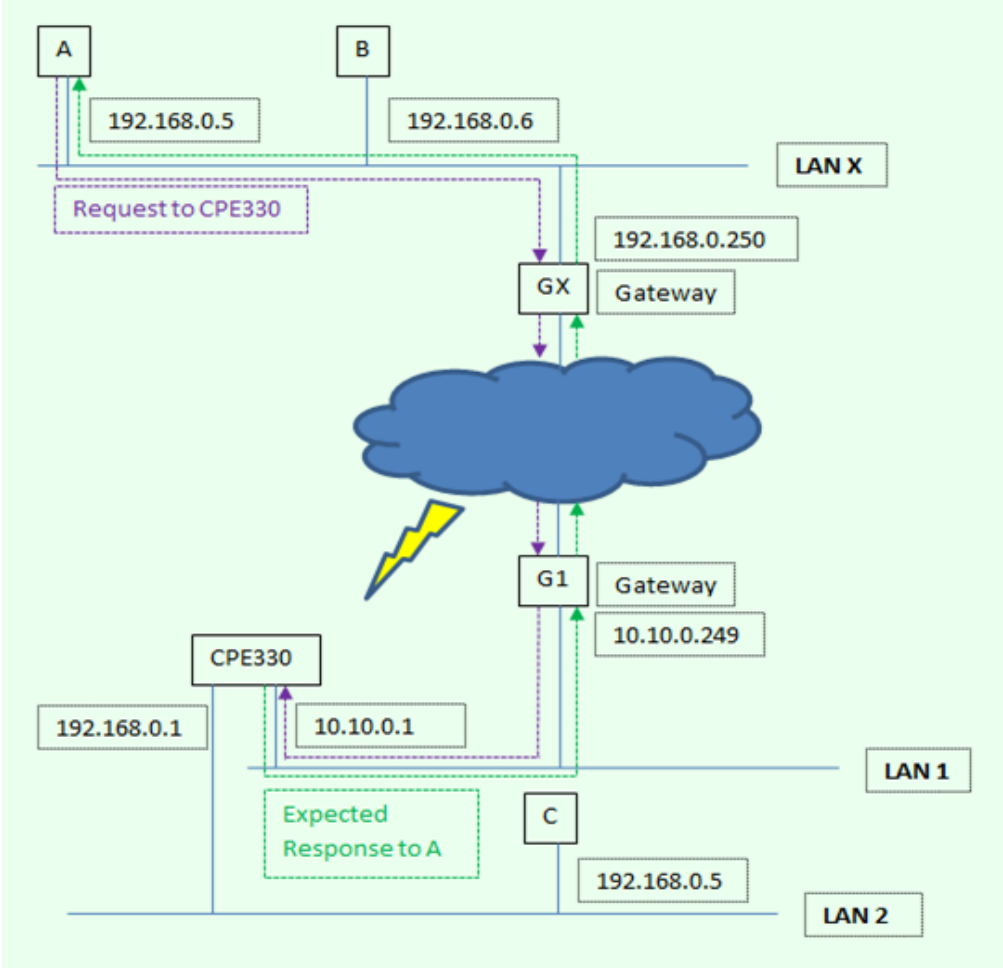
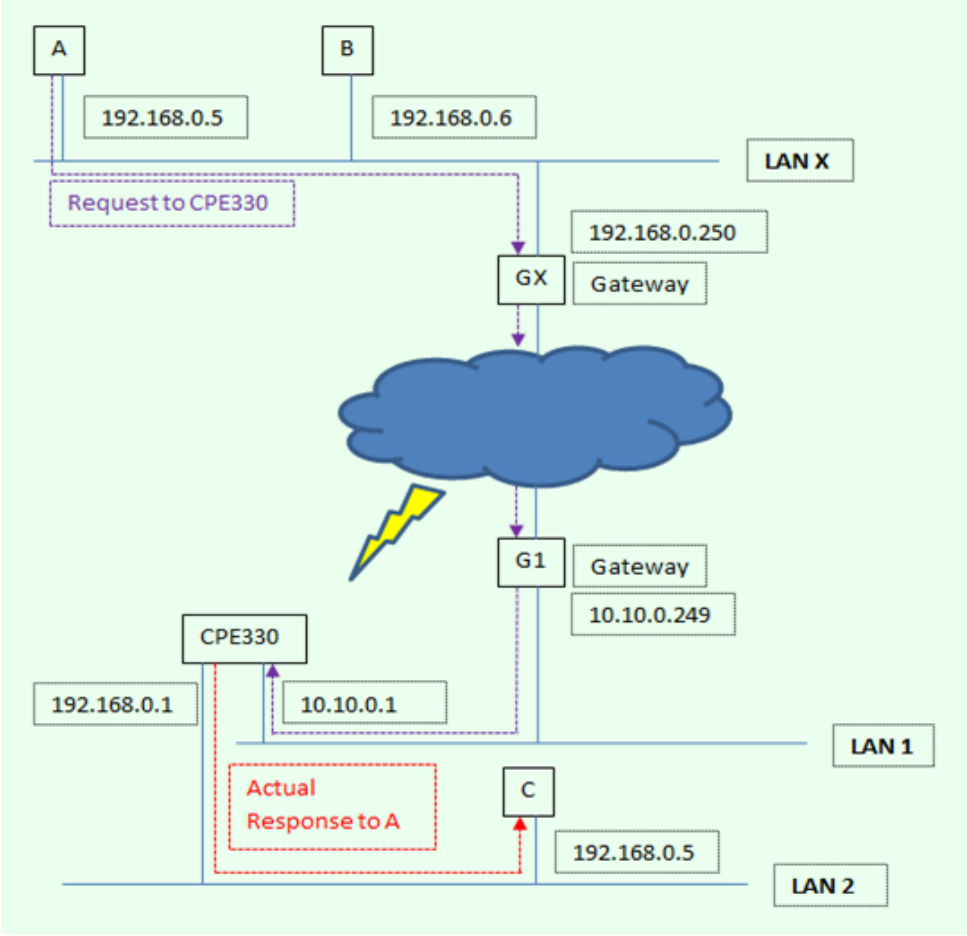


Figure 43: Actual Response Path



CAUTION

The two ports on the Ethernet Interface must *not* be connected, directly or indirectly to the same device. The hub or switch connections in an Ethernet network must form a tree; otherwise, duplication of packets may result.

RSTi-EP

RSTi-EP CPE100/CPE115/CPE205/CPE210/CPE215/CPE220/CPE240 provides one or more embedded Ethernet interfaces. If used, each interface connects to a Local Area Network (LAN).

The corresponding RJ45 Ethernet port(s) automatically sense the data rate on the attached LAN (100 Mbps or 10 Mbps for the CPE100/CPE115 or 1000 Mbps or 100 Mbps for the CPE205/CPE210/CPE215/CPE220/CPE240), as well as the corresponding communication mode (half-duplex or full-duplex), and the corresponding cabling arrangement (straight-through or crossover). Automatic detection greatly simplifies installation procedures.

See RSTi-EP CPU Features and Specifications to determine the complete list of Internet protocols

supported by each CPU.

Some important protocols supported by all RSTi-EP CPUs are: Some important protocols supported by all RSTi-EP CPUs are:

- TCP/IP, which provides basic Internet capabilities;
- SRTP, which is proprietary and which provides the interface with the PME programming and configuration software and supports communications with certain control systems and supervisory computer layers in the factory;
- Modbus TCP supports the Modbus messaging structure over the Internet.

On the CPE100/CPE115, the same shared processor performs both Ethernet port processing and Controller logic processing. On the CPE205/CPE210/CPE215/CPE220/CPE240, the multi-core CPU enables communication to be handled by one core while CPU logic and I/O scanning are handled by the second core.

Each interface on a LAN must have a unique IP Address *and* a non-overlapping IP subnet. This is configured in PME. Care must be taken to survey the entire connected network architecture to tabulate the IP addresses and IP subnets already in use, both on the local networks and any of its routed subnets connected with a gateway. Never assign a conflicting IP Address or configure duplicate IP subnets. For example, please refer to 5.1.1

10Base-T/100Base-Tx Port Pin Assignments

Pin assignments are the same for the RX3i and embedded Ethernet ports.

Pin Number	Signal	Description
1	TD+	Transmit Data +
2	TD-	Transmit Data -
3	RD+	Receive Data +
4	NC	No connection
5	NC	No connection
6	RD-	Receive Data -
7	NC	No connection
8	NC	No connection

Recovering a Lost IP Address

See Establishing Initial Ethernet Communications, Section 3.4.1.

5.1.2 Ethernet Interface Modules

In addition to Ethernet interfaces embedded in certain CPUs (see *RX3i CPU Features and Specifications)* and RX3i systems support rack-based Ethernet Interface modules. These modules are not interchangeable.

For details about the capabilities, installation, and operation of these modules, refer to PACSystems RX3i TCP/IP Ethernet Communications User Manual GFK-2224 and PACSystems TCP/IP Ethernet Communications Station Manager User Manual, GFK-2225.

Type	Catalog Number	Description
RX3i	IC695ETM001-Jx	Ethernet PCI Module
RX3i	IC695ETM001-Kxxx	Ethernet PCI Module (Version 7.0 or later)

5.2 Serial Communications

RX3i CPUs, except CPE330 and RSTi-EP CPE100/CPE115/CPE205/CPE210/CPE215/CPE220/CPE240, support one or more serial ports.

The independent onboard serial ports of the CPU are accessed via external connectors on the module. COM1 and COM2 provide serial interfaces to external devices. COM1 is also used for firmware upgrades.

5.2.1 Serial Port Communications Capabilities

COM1 and COM2 can each be configured for one of the following modes. For details on CPU configuration, refer to *Section 3*.

- RTU Slave – The port can be used for the Modbus RTU slave protocol. This mode also permits connection to the port by an SNP master, such as the WinLoader utility or the programming software. For details, refer to Section 6, *Serial I/O, SNP & RTU Protocols*.
- Message Mode – The port is available for access by user logic. This enables C language blocks to perform serial port I/O operations via C Runtime Library functions.
- Available – The port is not to be used by the CPU firmware.
- SNP Slave – The port can only be used for the SNP slave protocol. For details, refer to Section 6, *Serial I/O, SNP & RTU Protocols*.
- Serial I/O – The port can be used for general-purpose serial communication through the use of COMMREQ functions. For details, refer to Section 6, *Serial I/O, SNP & RTU Protocols*.

Features Supported

	RX3i		RSTi-EP	
Feature	Serial Port 1 (COM1)	Serial Port 2 (COM2)	Serial Port 1 (COM1)	Serial Port 2 (COM2)
RTU Slave protocol	Yes	Yes	Yes	Yes
SNP Slave	Yes	Yes	No	No
Serial I/O – used with COMMREQs	Yes[77]	Yes	No	No
Firmware Upgrade (WinLoader utility)	PLC in STOP/No IO mode	No	No	No
Message Mode –used only with C blocks (C Runtime Library Functions: serial read, serial write, sscanf, sprintf)	Yes	Yes	No	No
RS-232	Yes	No	Yes	No
RS-485	No	Yes	No	Yes

5.2.2 Configurable STOP Mode Protocols

You can configure the protocol to be used in STOP Mode, based on the configured serial port (RUN Mode) protocol. The Run/Stop protocol switching is independently configured for each serial port.

The RUN Mode protocol setting determines which choices are available for STOP Mode. If a STOP Mode protocol is not selected, the default STOP Mode protocol is used.

[77] *Serial IO is the only protocol supported by CPE400.*

5.2.3 Serial Port PIN Assngments

COM1 (RS-232, 9-pin Subminiature D Connector)

This port has a 9-pin, female, D-sub connector with a standard pinout. This is a DCE (data communications equipment) port that allows a simple straight-through cable to connect with a standard AT-style RS-232 port.

The CPE310 provides the DCD and RI signals to support point-to-point protocol (PPP).

COM1 RS-232 Signals

RX3i CPU, RX3i CRU, CPE Models			RX3i CPE310 Model		
Pin No.[78]	Signal Name	Description	Pin No.[78]	Signal Name	Description
1	NC	No Connection	1	DCD	Data Carrier Detect
2	TXD	Transmit Data	2	TXD	Transmit Data
3	RXD	Receive Data	3	RXD	Receive Data
4	DSR	Data Set Ready	4	DSR	Data Set Ready
5	0V	Signal Ground	5	COM	Signal Ground
6	DTR	Data Terminal Ready	6	DTR	Data Terminal Ready
7	CTS	Clear to Send	7	CTS	Clear to Send
8	RTS	Request to Send	8	RTS	Request to Send
9	NC	No Connection	9	RI	Ring Indicator

COM1(RS-232, Terminal Block Connector)

The CPE100/CPE115 provides RS-232 communication via a terminal block connector.

RSTi-EP CPE100/CPE115 Models	
Signal Name	Signal Name
TXD	TXD
RXD	RXD
0V	0V
RTS	RTS
CTS	CTS

[78] *Pin 1 is at the bottom right of the connector as viewed from the front of the module.*

COM1 (RS-232, RJ-25 Connector)

The CPE205/210/215/220/240 provides RS-232 communications via an RJ-25 connector and requires shielded cable IC693CBL316.

Pin No.	Signal Name	Description
1	DSR	Data Set Ready
2	DCD	Data Carrier Detect
3	DTR	Data Terminal Ready
4	GND	Signal Ground
5	RXD	Receive Data
6	TXD	Transmit Data
7	CTS	Clear to Send
8	RTS	Request to Send

COM1 (RS-232, RJ-25 Connector)

The CPE302/CPE305 provides RS-232 communications via an RJ-25 connector and requires shielded cable IC693CBL316.

CPE302/CPE305 COM1 RS-232 Signals

Pin No.	Signal Name	Description
1	CTS	Clear to Send
2	TXD	Transmit Data
3	0V	Signal Ground
4	0V	Signal Ground
5	RXD	Received Data
6	RTS	Request to Send

COM1 (RS232, RJ45 Connector)

The RJ45 Connector is provided for COM1 on CPE400[21] and CPL410 only. It has the following pinout.

Pin No.[79]	Signal Name	Description
1	NC	No Connection
2	NC	No Connection
3	TX	Transmit Data
4	RX	Receive Data
5	0V	Signal Ground
6	NC	No Connection
7	NC	No Connection
8	NC	No Connection

Figure 44: COM1 Port CPE400/CPL410

[79] *Pin 1 is the leftmost pin as shown in Figure 44.*

COM2 (RS-485, 15-pin Female D-sub Connector) – All RX3i CPU/CRU Models & RX3i CPE310

This is a DCE port that allows a simple straight-through cable to connect with a standard AT-style RS-232 port.

COM2 RS-485 Signals

Pin No.	Signal Name	Description
1	Shield	Cable Shield Located at the bottom right of the connector as viewed from the front of the module.
2	NC	No Connection
3	NC	No Connection
4	NC	No Connection
5	+5Vdc	Logic Power: Provides isolated +5Vdc power (300mA maximum) for powering external options.
6	RTS(A)	Differential Request to Send A
7	0V	Signal Ground
8	CTS(B‘)	Differential Clear To Send B
9	RT	Resistor Termination
10	RD(A‘)	Differential Receive Data A
11	RD(B‘)	Differential Receive Data B
12	SD(A)	Differential Send Data A
13	SD(B)	Differential Send Data B
14	RTS(B’)	Differential Request To Send B
15	CTS(A’)	Differential Clear To Send A

Station Manager RS-232 Signals

Pin No.[78]	Signal Name	Description
1	DCD	Data Carrier Detect
2	TXD	Transmit Data
3	RXD	Receive Data
4	DSR	Data Set Ready
5	0V	Signal Ground
6	DTR	Data Terminal Ready
7	CTS	Clear To Send
8	RTS	Request To Send
9	RI	Ring Indicator

COM2 (RS-485, Terminal Block Connector) – RSTi-EP CPE100, CPE115.

The CPE100/CPE115 provides RS-485 communication via a terminal block connector.

RSTi-EP CPE100/CPE115 Models	
RS-485 Pins	Signals
A	RX+
B	RX-
Y	TX+
Z	TX-

5.2.4 Serial Port Electrical Isolation

Some serial communication ports are isolated, while others are not, as indicated in the following table:

Family	Model	COM1	COM2	COM3
RX3i	CPU310	Non-Isolated	Non-Isolated	N/A
	CPU315	Non-Isolated	Non-Isolated	N/A
	CPU320/CRU320	Non-Isolated	Non-Isolated	N/A
	CPE302/CPE305	Non-Isolated	N/A	N/A
	CPE310	Non-Isolated	Non-Isolated	N/A
	CPE330	N/A	N/A	N/A
	CPE400/CPL410	Non-Isolated	N/A	N/A
RSTi-EP	CPE205	Non-Isolated	N/A	N/A
	CPE210	Non-Isolated	N/A	N/A
	CPE215	Non-Isolated	N/A	N/A
	CPE220	Non-Isolated	N/A	N/A
	CPE240	Non-Isolated	N/A	N/A
	CPE100	Non-Isolated	Isolated	N/A
	CPE115	Non-Isolated	Isolated	N/A

5.2.5 Serial Cable Lengths and Shielding

The connection from a CPU serial port to the serial port on a computer or other serial device requires a serial cable. Maximum cable lengths (the total distance from the CPU to the last device attached to the serial cable) are:

Port	Maximum Cable Length	Cable Type
COM1 (RS-232)	15 m (50 ft.)	Shielded cable ***required*** for RX3i

Note: For details on conformance to radiated emissions standards, refer to Appendix A in the following manuals: PACSystems RX3i System Manual, GFK-2314

5.2.6 Maximum Cable Lengths for RSTi-EP CPE205/210/215/220/240

To ensure that the PLC is optimized and limits electromagnetic interference, Emerson strongly advises the proper cable lengths listed in the table below.

Port Name	Maintenance Port?	Maximum Cable Length (Meters)	Cable Must be Shielded?
DC Input	No	30m	No
DC Field	No	30m	No
RS-232	No	15m	Yes
ETH Port 1	No	100m	No*
ETH Port 2	No	100m	No*
ETH Port 3	No	100m	No*
USB RDSD	Yes	3m	Yes
USB	Yes	3m	Yes

**These cables must be shielded if operating at 1 Gbps. If operating at or below 100 Mbps, unshielded cables can be used.*

5.2.7 Serial Port Baud Rates

Protocol	COM1 (RS-232)	COM2 (RS-485)	Station Manager - COM3 (RS-232)
RTU Slave	1200, 2400, 4800, 9600, 19.2K, 38.4K, 57.6K, 115.2K	1200, 2400, 4800, 9600, 19.2K, 38.4K, 57.6K, 115.2K	not supported
Firmware Upgrade via WinLoader	1200, 2400, 4800, 9600, 19.2K, 38.4K, 57.6K, 115.2K	Not supported	not supported
Message Mode	1200, 2400, 4800, 9600, 19.2K, 38.4K, 57.6K, 115.2K	1200, 2400, 4800, 9600, 19.2K, 38.4K, 57.6K, 115.2K	not supported
SNP Slave	1200, 2400, 4800, 9600, 19.2K, 38.4K, 57.6K, 115.2K	1200, 2400, 4800, 9600, 19.2K, 38.4K, 57.6K, 115.2K	not supported
Serial I/O	1200, 2400, 4800, 9600, 19.2K, 38.4K, 57.6K, 115.2K	1200, 2400, 4800, 9600, 19.2K, 38.4K, 57.6K, 115.2K	not supported

Communications Coprocessor Module (CMM)

PACSystems ***does not*** support the following with an IC697CMM711:

- Access to Symbolic variables
- WAIT mode COMMREQs.
- Connecting the programming software to the CPU through the CMM's serial ports.
- Permanent datagrams.

The following restrictions apply when using the IC697CMM711 with PACSystems:

- Access to %W references is partially supported. Only offsets 0—65535 of %W can be accessed via the CMM.
- The Program Name is currently always LDPROG1 for PACSystems.
- Reads and writes beyond currently configured reference table limits will report a minor code error of 90 (REF_OUT_OF_RANGE) instead of F4 (INVALID_PARAMETER) as reported on the Series 90-70.
- In the case of ERROR NACK, the Control Program number, privilege level, and other piggyback status data will be set to 0.
- PACSystems CPUs return the major/minor type of the 90-70 CPX935 (major type 12, minor type 35) to the CMM scratchpad memory when communicating with a CMM.
- Control Program Number will be returned as 01 in PACSystems instead of FF as reported on the Series 90-70.

Note: For details on the operation of the IC697CMM711, refer to the *Series 90 PLC Serial Communications User's Manual,* GFK-0582.

5.2.8 Programmable Coprocessor Module (PCM)

PACSystems ***does not*** support the following with an IC697PCM711:

- Connecting the programming software to the CPU through the serial ports on the PCM711.

Access to Symbolibc variables.

- WAIT mode COMMREQs.
- The following C functions are not supported:
- chk_genius_bus
- chk_genius_device
- get_cpu_type_rev
- get_memtype_sizes
- get_one_rackfault
- get_rack_slot_faults
- The C function write_dev will not write to *read-only* references (%S references, transition bits, and override bits). If this is attempted, the call will fail at run time and return an error code.
- The following restrictions apply when using the IC697PCM711 with PACSystems:
- Access to %W references is partially supported. Only offsets 0—65535 of %W can be accessed via the PCM.
- The Program Name is currently always LDPROG1 for PACSystems.
- In the case of ERROR NACK, the Control Program number, privilege level, and other piggyback status data will be set to 0.
- PACSystems CPUs return the major/minor type of the Series 90-70 CPX935 (major type 12, minor type 35) to the PCM scratchpad memory when communicating with a PCM.

Note: For details on the operation of the IC697PCM711, refer to Series 90 Programmable Coprocessor Module and Support Software, GFK-0255.

Section 6 Serial I/O, SNP & RTU Protocols

This section discusses the following topics related to communications on CPU serial ports COM1 and COM2:

- Configuring Serial Ports Using COMMREQ Function 65520
- Serial I/O Protocol
- RTU Slave Protocol
- SNP Slave Protocol

Details of the RTU and SNP protocol are described in the *Series 90 PLC Serial Communications User's Manual*, GFK-0582.

6.1 Configuring Serial Ports Using COMMREQ Function 65520

The Serial Port Setup COMMREQ function 65520 (FFF0 hex) may be used to activate a serial communication protocol for a serial port, overriding the protocol that was specified in the port settings of the CPU configuration. The COMMREQ installed protocol remains active as long as the CPU is in RUN Mode. When the CPU is STOPped, the COMMREQ installed protocol is removed, and the protocol settings from the CPU configuration are reactivated.

The COMMREQ requires that all its command data be placed in the correct order (in a *command block*) in the CPU memory before it is executed. The COMMREQ should be executed by a contact of a one-shot coil to prevent sending the data multiple times. For details on the operands and command block format used by the COMMREQ function, refer to *PACSystems RX3i CPU Programmer's Reference Manual,* GFK-2950 Section 4.

The COMMREQ uses the following TASKs to specify the port for which the operation is intended:

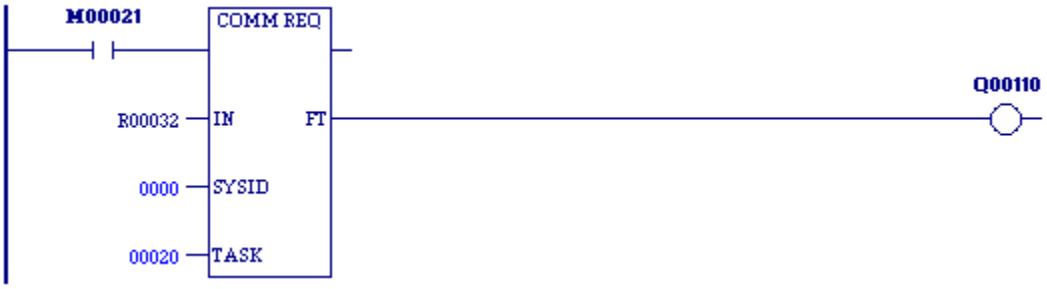
task 19 for COM1
task 20 for COM2

Note: Because address offsets are stored in a 16-bit word field, the full range of %W memory type cannot be used with COMMREQs.

6.1.1 COMMREQ Function Example

In the example, when %M0021 is ON, a Command Block located starting at %R0032 is sent to COM2 (communications task 20) of the CPU (rack 0, slot 0). If an error occurs processing the COMMREQ, %Q0110 is set.

Figure 45: COMMREQ Example



6.1.2 Timing

If a port configuration COMMREQ is sent to a serial port that currently has an SNP master (for example, the programmer) connected to it, the COMMREQ function returns an error code to the COMMREQ status word.

6.1.3 Sending Another COMMREQ to the Same Port

After sending a COMMREQ to configure a serial port, the application program should monitor the COMMREQ status word to determine when it can begin sending protocol-specific COMMREQs to that port. It is recommended that the application clear the COMMREQ status word before issuing the configuration change. The status word will be set to a nonzero value when the request has been processed.

6.1.4 Invalid Port Configuration Combinations

The PAC Machine Edition programming software safeguards against the download of some hardware configurations that would prevent the programmer from communicating serially with the CPU. In a system that does not have an embedded Ethernet module, if a rack-based Ethernet is not present, a serial connection is required for programmer communications.

For CPE302/CPE305/CPE310 CPUs, which have an embedded Ethernet port that, when configured, is available for programmer communications, the safeguards on serial port configurations are still enforced.

For CPE400 and CPL410, which only support the Serial I/O protocol, the only valid operation is to enter Serial I/O as the Protocol Selector (refer to Section 6.1.5).

6.1.5 COMMREQ Command Block Parameter Values

The following table lists common parameter values that are used within the COMMREQ command blocks for configuring a serial port. All values are in decimal.

Parameter	Values
Protocol Selector	1 = SNP 3 = RTU 5 = Serial I/O 7 = Message Mode
Data Rate	0 = 300 1 = 600 2 = 1200 3 = 2400 4 = 4800 5 = 9600 6 = 19200 7 = 38400 8 = 57600 9 = 115200
Parity	0 = None 1 = Odd 2 = Even
Flow Control	0 = Hardware [RTS / CTS] 1 = None 2 = Software [XON / XOFF] (Serial I/O only)
Bits Per Character	0 = 7 bits 1 = 8 bits
Stop Bits	0 = 1 stop bit 1 = 2 stop bits
Duplex Mode	0 = 2-wire 1 = 4-wire 2 = 4-wire transmitter always on
Turnaround Delay (SNP only)	0 = none 1 = 10ms 2 = 100ms 3 = 500ms
Timeout (SNP only)	0 = Long (8 sec) 1 = Medium (2 sec) 2 = Short (500ms) 3 = None (200ms)

6.1.6 Example COMMREQ Data Block for Configuring Message mode Protocol

Address	Values	Meaning
Address	12	Data Block Length
Address + 1	0 = No Wait (WAIT mode not supported)	WAIT/NOWAIT Flag
Address + 2	0008 = %R, register memory	Status Word Pointer Memory Type
Address + 3	A zero-based number that gives the address of the COMMREQ status word (for example, a value of 99 gives an address of 100 for the status word)	Status Word Pointer Offset
Address + 4	not used	Idle Timeout Value
Address + 5	not used	Maximum Communication Time
Address + 6	FFF0H	Command Word (serial port setup)
Address + 7	7 = Message Mode	Protocol
Address + 8	not used	Port Mode
Address + 9	See *COMMREQ Command Block Parameter* Values.	Data Rate
Address + 10	0 = None, 1 = Odd, 2 = Even	Parity
Address + 11	0 = Hardware, 1 = None, 2 = Software	Flow Control
Address + 12	not used	Turnaround Delay
Address + 13	not used	Timeout
Address + 14	0=7 bits, 1=8 bits	Bits per Character
Address + 15	0 = 1 stop bit, 1 = 2 stop bits	Stop Bits
Address + 16	not used	Interface
Address + 17	0 = 2-wire, 1 = 4-wire, 2 = 4-wire transmitter always on	Duplex Mode

6.1.7 Example COMMREQ Command Blocks for Serial Port Setup function

The following COMMREQ command blocks provide examples for configuring the various protocols. All values are in decimal unless followed by an H indicating hexadecimal.

Note that an example is not provided for Message Mode, but it can be set up with a command block similar to the one for Serial I/O, with a value of 7 for the protocol selector.

6.1.8 Example COMMREQ Command Block for Configuring SNP Protocol

	Values	Meaning
Address	16	Data Block Length
Address + 1	0 = No Wait (WAIT mode not supported)	WAIT/NOWAIT Flag
Address + 2	0008 = %R, register memory	Status Word Pointer Memory Type
Address + 3	A zero-based number that gives the address of the COMMREQ status word (for example, a value of 99 gives an address of 100 for the status word)	Status Word Pointer Offset
Address + 4	not used	Idle Timeout Value
Address + 5	not used	Maximum Communication Time
Address + 6	FFF0H	Command Word (serial port setup)
Address + 7	1 = SNP	Protocol
Address + 8	0 = Slave	Port Mode
Address + 9	See *COMMREQ Command Block Parameter* Values.	Data Rate
Address + 10	0 = None, 1 = Odd, 2 = Even	Parity
Address + 11	not used (SNP always chooses NONE by default)	Flow Control
Address + 12	0 = None, 1 = 10ms, 2 = 100ms, 3 = 500ms	Turnaround Delay
Address + 13	0 = Long, 1 = Medium, 2 = Short, 3 = None	Timeout
Address + 14	not used (SNP always chooses 8 bits by default)	Bits Per Character
Address + 15	0 = 1 Stop Bit, 1 = 2 Stop bits	Stop Bits
Address + 16	not used	Interface
Address + 17	not used (SNP always chooses 4-wire mode by default)	Duplex Mode
Address + 18	user-provided [80]	Device identifier bytes 1 and 2
Address + 19	user-provided [80]	Device identifier bytes 3 and 4
Address + 20	user-provided [80]	Device identifier bytes 5 and 6
Address + 21	user-provided [80]	Device identifier bytes 7 and 8

[80] *The device identifier for SNP Slave ports is packed into words with the least significant character in the least significant byte of the word. For example, if the first two characters are "A" and "B," the Address + 18 will contain the hex value 4241.*

6.1.9 Example COMMREQ Data Block for Configuring RTU Protocol

	Values	Meaning
Address	13, or 17	Data Block Length
Address + 1	0 = No Wait (WAIT mode not supported)	WAIT/NOWAIT Flag
Address + 2	0008 = %R, register memory	Status Word Pointer Memory Type
Address + 3	A zero-based number that gives the address of the COMMREQ status word (for example, a value of 99 gives an address of 100 for the status word)	Status Word Pointer Offset
Address + 4	not used	Idle Timeout Value
Address + 5	not used	Maximum Communication Time
Address + 6	FFF0H	Command Word (serial port setup)
Address + 7	3 = RTU	Protocol
Address + 8	0 = Slave	Port Mode
Address + 9	See *COMMREQ Command Block Parameter Values*.	Data Rate
Address + 10	0 = None, 1 = Odd, 2 = Even	Parity
Address + 11	0 = Hardware, 1 = None	Flow Control
Address + 12	not used	Turnaround delay
Address + 13	not used	Timeout
Address + 14	not used (RTU always chooses 8 bits by default)	Bits per Character
Address + 15	not used (RTU always chooses 1 stop bit by default)	Stop Bits
Address + 16	not used	Interface
Address + 17	0 = 2-wire, 1 = 4-wire, 2 = 4-wire transmitter always on	Duplex Mode
Address + 18	Station Address (1-247)	Device Identifier
Address + 19	Count of 100 µs units (0 = 3.5 character times)	End-of-frame timeout [81]
Address + 20	not used	
Address + 21	not used	
Address + 22	Count of 10 ms units (range 0-255)	Receive-to-transmit delay [81]

[81] *The End-of-frame timeout and Receive-to-transmit delay values were added in Release 6.70 for the RX3i. They are discussed in the RTU Slave Protocol section.*

6.1.10 Example COMMREQ Data Block for Configuring Serial I/O Protocol

	Values	Meaning
Address	12	Data Block Length
Address + 1	0 = No Wait (WAIT mode not supported)	WAIT/NOWAIT Flag
Address + 2	0008 = %R, register memory	Status Word Pointer Memory Type
Address + 3	A zero-based number that gives the address of the COMMREQ status word (for example, a value of 99 gives an address of 100 for the status word)	Status Word Pointer Offset
Address + 4	not used	Idle Timeout Value
Address + 5	not used	Maximum Communication Time
Address + 6	FFF0H	Command Word (serial port setup)
Address + 7	5 = Serial I/O	Protocol
Address + 8	not used	Port Mode
Address + 9	See *COMMREQ Command Block Parameter Values*.	Data Rate
Address + 10	0 = None, 1 = Odd, 2 = Even	Parity
Address + 11	0 = Hardware, 1 = None, 2 = Software	Flow Control
Address + 12	not used	Turnaround Delay
Address + 13	not used	Timeout
Address + 14	0=7 bits, 1=8 bits	Bits per Character
Address + 15	0 = 1 stop bit, 1 = 2 stop bits	Stop Bits
Address + 16	not used	Interface
Address + 17	0 = 2-wire, 1 = 4-wire, 2 = 4-wire transmitter always on	Duplex Mode

6.2 Serial I/O Protocol

Serial I/O protocol is a communication protocol that is driven entirely by the application program. Serial I/O protocol is active only when the CPU is in RUN Mode since it is driven completely by COMMREQ functions in the application program. Those COMMREQ functions are described in detail within this section.

When the CPU is stopped, a port configured for Serial I/O protocol will revert to a STOP Mode protocol as specified in the port settings of the CPU configuration. If a STOP Mode protocol was not specified, RTU slave protocol is used by default.

Serial I/O is the only protocol supported by CPE400 and CPL410. CPE400 requires firmware version 9.40 or later.

6.2.1 Calling Serial I/O COMMREQs from the CPU Sweep

Implementing a serial protocol using Serial I/O COMMREQs may be restricted by the sweep time. For example, if the protocol requires that a reply to a certain message from the remote device be initiated within 5ms of receiving the message, this method may not be successful if the sweep time is 5ms or longer, since the timely response is not guaranteed.

6.2.2 Compatibility

The COMMREQ function blocks supported by Serial I/O are not supported by other currently existing protocols (such as SNP slave and RTU slave). Errors are returned if they are attempted for a port configured for one of those protocols.

6.2.3 Status Word for Serial I/O COMMREQs

A value of 1 is returned in the COMMREQ status word upon successful completion of the COMMREQ. Any other value returned is an error code where the low byte is a major error code and the high byte is a minor error code.

Whenever an invalid memory type or offset or inaccessible CPU memory is provided, the CPU will not provide a status word instead it logs a diagnostic application alarm and COMMREQ FT is energized.

Major Error Code	Description	
01 (01h)	**Successful Completion** (this is the expected completion value in the COMMREQ status word).	
12 (0Ch)	**Local error** —Error processing a local command. The minor error code identifies the specific error.	
	02 (02h)	COMMREQ command is not supported.
	06 (06h)	Invalid CPU memory type specified.
	07 (07h)	Invalid CPU memory offset specified.
	08 (08h)	Unable to access CPU memory.
	12 (0Ch)	COMMREQ data block length is too small.
	14 (0Eh)	COMMREQ data is invalid.
	15 (0Fh)	Could not allocate system resources to complete COMMREQ.
13 (0Dh)	**Remote error** — Error processing a remote command. The minor error code identifies the error.	
	2 (02h)	A number of bytes requested to read is greater than the input buffer size OR the number of bytes requested to write is zero or greater than 250 bytes.
	3 (03h)	COMMREQ data block length is too small. String data is missing or incomplete.
	4 (04h)	Receive timeout awaiting serial reception of data
	6 (06h)	Invalid CPU memory type specified.
	7 (07h)	Invalid CPU memory offset specified.
	8 (08h)	Unable to access CPU memory.
	12 (0Ch)	COMMREQ data block length is too small.
	16 (10h)	Operating system service error. The operating system service used to perform the request has returned an error.
	17 (11h)	Port device error. The port device used to perform the service has detected an error. Either a break was received or a UART Error (parity, framing, overrun) occurred.
	18 (12h)	Request canceled. The request was terminated before it could complete.
	48 (30h)	Serial output timeout. The serial port was unable to transmit the string. (Could be due to missing CTS signal when the serial port is configured to use hardware flow control.)
14 (0Eh)	**Autodial Error** — An error occurred while attempting to send a command string to an attached external modem. The minor error code identifies the specific error.	

Major Error Code	Description	
	2 (02h)	The modem command string length exceeds the end of the reference memory type.
	3 (03h)	COMMREQ Data Block Length too small. Output command string data missing or incomplete.
	4 (04h)	Serial output timeout. The serial port was unable to transmit the modem autodial output.
	5 (05h)	The response was not received from the modem. Check modem and cable.
	6 (06h)	Modem responded with BUSY. The modem is unable to complete the requested connection. The remote modem is already in use; retry the connection request later.
	7 (07h)	Modem responded with NO CARRIER. The modem is unable to complete the requested connection. Check the local and remote modems and the telephone line.
	8 (08h)	Modem responded with NO DIALTONE. The modem is unable to complete the requested connection. Check the modem connections and the telephone line.
	9 (09h)	Modem responded with ERROR. The modem is unable to complete the requested command. Check the modem command string and modem.
	10 (0Ah)	The modem responded with RING, indicating that the modem is being called by another modem. The modem is unable to complete the requested command. Retry the modem command later.
	11 (0Bh)	Unknown response received from the modem. The modem is unable to complete the request. Check the modem command string and modem. The response should be CONNECT or OK.

6.2.4 Serial I/O COMMREQ Commands

The following COMMREQs are used to implement Serial I/O:

- Local COMMREQs - do not receive or transmit data through the serial port.
  - Initialize Port (4300)
  - Set Up Input Buffer (4301)
  - Flush Input buffer (4302)
  - Read port status (4303)
  - Write port control (4304)
  - Cancel Operation (4399)
- Remote COMMREQs - receive and/or transmit data through the serial port.
  - Autodial (4400)
  - Write bytes (4401)
  - Read bytes (4402)
  - Read String (4403)

6.2.5 Overlapping COMMREQs

Some Serial I/O COMMREQs must complete execution before another COMMREQ can be processed. Others can be left pending while others are executed.

COMMREQS that Must Complete Execution

- Autodial (4400)
- Initialize Port (4300)
- Set Up Input Buffer (4301)
- Flush Input buffer (4302)
- Read port status (4303)
- Write port control (4304)
- Cancel Operation (4399)
- Serial Port Setup (FFF0)

COMMREQs that can be Pending While Others Execute

The table below shows whether Write Bytes, Read Bytes and Read String COMMREQs can be pending when other COMMREQs are executed.

	NEW COMMREQ										
Currently-Pending COMMREQs	Autodial (4400)	Write Bytes (4401)	Initialize Port (4300)	Set Up Input Buffer (4301)	Flush Input Buffer (4302)	Read Port Status (4303)	Write Port Control (4304)	Read Bytes (4402)	Read String (4403)	Cancel Operation (4399)	Serial Port Setup (FFF0)
Write Bytes (4401)	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Read Bytes (4402)	No	Yes	Yes	No	No	Yes	Yes	No	No	Yes	No
Read String (4403)	No	Yes	Yes	No	No	Yes	Yes	No	No	Yes	No

6.2.6 Initialize Port Function (4300)

This function causes a reset command to be sent to the specified port. It also cancels any COMMREQ currently in progress and flushes the internal input buffer. RTS and DTR are set to inactive.

Example Command Block for the Initialize Port Function

	Value (decimal)	*Value (hexadecimal)*	*Meaning*
Address	0001	0001	Data block length
Address +1	0000	0000	NOWAIT mode
Address +2	0008	0008	Status word memory type (%R)
Address +3	0000	0000	Status word address minus 1 (%R0001)
Address +4	0000	0000	Not used
Address +5	0000	0000	Not used
Address +6	4300	10CC	Initialize port command

Operating Notes

Remote COMMREQs that are canceled due to this command executing will return a COMMREQ status word indicating request cancellation (minor code 12H).

CAUTION

If this COMMREQ is sent when a Write Bytes (4401) COMMREQ is transmitting a string from a serial port, transmission is halted. The position within the string where the transmission is halted is indeterminate. In addition, the final character received by the device to which the CPU is sending is also indeterminate.

6.2.7 Set Up Input Buffer Function (4301)

This function is provided for compatibility with legacy Serial I/O applications. In PACSystems releases 5.70 and later, the internal input buffer is always set to 2097 bytes. In earlier PACSystems implementations, the internal input buffer is set to 2K bytes.

The Set-Up Input Buffer function returns a success status to the COMMREQ status word, regardless of the buffer length specified in the command block.

As data is received from the serial port it is placed in the input buffer. If the buffer becomes full, any additional data received from the serial port is discarded and the Overflow Error bit in the Port Status word (See Read Port Status Function) is set.

Retrieving Data from the Buffer

Data can be retrieved from the buffer using the Read String or Read Bytes function. It is not directly accessible from the application program.

If data is not retrieved from the buffer in a timely fashion, some characters may be lost.

Example Command Block for the Set Up Input Buffer Function

	VALUE (decimal)	VALUE (hexadecimal)	MEANING
Address	0002	0002	Data block length
Address +1	0000	0000	NOWAIT mode
Address +2	0008	0008	Status word memory type (%R)
Address +3	0000	0000	Status word address minus 1 (%R0001)
Address +4	0000	0000	Not used
Address +5	0000	0000	Not used
Address +6	4301	10CD	Setup input buffer command
Address +7	0064	0040	Buffer length (in words)

6.2.8 Flush Input Buffer Function (4302)

This operation empties the input buffer of any characters received through the serial port but not yet retrieved using a read command. All such characters are lost.

Example Command Block for the Flush Input Buffer Function

	VALUE (decimal)	VALUE (hexadecimal)	MEANING
Address	0001	0001	Data block length
Address +1	0000	0000	NOWAIT mode
Address +2	0008	0008	Status word memory type (%R)
Address +3	0000	0000	Status word address minus 1 (%R0001)
Address +4	0000	0000	Not used
Address +5	0000	0000	Not used
Address +6	4302	10CE	Flush input buffer command

6.2.9 Read Port Status Function (4303)

This function returns the current status of the port. The following events can be detected:

1. A read request was initiated previously and the required number of characters has now been received or the specified time-out has elapsed.
2. A written request was initiated previously and transmission of the specified number of characters is complete or a time-out has elapsed.

The status returned by the function indicates the event or events that have been completed. More than one condition can occur simultaneously if both a read and a write were initiated previously.

Example Command Block for the Read Port Status Function

	VALUE (decimal)	VALUE (hexadecimal)	MEANING
Address	0003	0003	Data block length
Address +1	0000	0000	NOWAIT mode
Address +2	0008	0008	Status word memory type (%R)
Address +3	0000	0000	Status word address minus 1 (%R0001)
Address +4	0000	0000	Not used
Address +5	0000	0000	Not used
Address +6	4303	10CF	Read port status command
Address +7	0076	004C	Port status memory type (%M)
Address +8	0101	0065	Port status memory offset (%M101)

Port Status

The port status consists of a status word and the number of characters in the input buffer that have not been retrieved by the application (characters that have been received and are available).

word 1	Port status word (see below)
word 2	Characters available in the input buffer

Port Status Word Meanings

Bit	Name	Definition	Status	Meaning
15	RP	Read In progress	Set	Read Bytes or Read String invoked
			Cleared	Previous Read bytes or String have timed out, been canceled, or finished
14	RS	Read Success	Set	Read Bytes or Read String has completed
			Cleared	New Read Bytes or Read String invoked
13	RT	Read Time-out	Set	Receive timeout occurred during reading Bytes or Read String
			Cleared	New Read Bytes or Read String invoked
12	WP	Write-In progress	Set	New Write Bytes invoked
			Cleared	Previously invoked Write Bytes has timed out, been canceled, or finished
11	WS	Write Success	Set	Previously invoked Write Bytes has completed
			Cleared	New Write Bytes invoked
10	WT	Write Time-out	Set	Transmit timeout occurred during Write Bytes
			Cleared	New Write Bytes invoked
9	CA	Character Available	Set	Unread characters are in the buffer
			Cleared	No unread characters in the buffer
8	OF	Overflow error	Set	Overflow error occurred on the serial port or internal buffer
			Cleared	Read Port Status invoked
7	FE	Framing Error	Set	A framing error occurred on the serial port
			Cleared	Read Port Status invoked
6	PE	Parity Error	Set	Parity error occurred on the serial port
			Cleared	Read Port Status invoked
5	CTS	Clear to Send	Set	Clear to Send signal is active
			Cleared	Clear to Send signal is not active
4	DSR	Data Set Ready	Set	Data Set Ready signal is active
			Cleared	Data Set Ready signal is not active
3	RI	Ring Indicator	Set	The ring Indicator signal is active
			Cleared	The ring Indicator signal is not active
2	DCD	Data Carrier Detect	Set	Data Carrier Detect signal is active
			Cleared	Data Carrier Detect signal is not active
1-0	n/a	Not used	These bits are always set to 0	

Operating Notes

For reference, see the tables under in Section 5.

Support for the DSR status bit is provided for COM1 only, and on all RX3i models (except CPE302/CPE305), in versions 7.16 and later.

Support for the RP and DCD status bits is provided only for COM1 on the CPE310, in version 7.16 and later.

6.2.10 Write Port Control Function (4304)

This function controls output signals on the specified port:

Example Command Block for the Write Port Control Function

	VALUE (decimal)	VALUE (hexadecimal)	MEANING
Address	0002	0002	Data block length
Address +1	0000	0000	NOWAIT mode
Address +2	0008	0008	Status word memory type (%R)
Address +3	0000	0000	Status word address minus 1 (%R0001)
Address +4	0000	0000	Not used
Address +5	0000	0000	Not used
Address +6	4304	10D0	Write port control command
Address +7	xxxx	xxxx	Port control word

Port Control Word

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTS	DTR	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Port Control Word Meanings:

15	RTS	Commanded state of Request to Send signal 1 = Activates RTS 0 = Deactivates RTS
14	DTR	Commanded state of Data Terminal Ready signal 1 = Activates DTR 0 = Deactivates DTR
13-0	n/a	Unused (should be zero)

Operating Notes

For reference, see the tables in sectin 5.2.3, *Serial Port PIN Assngments*.

Support for the DTR output signal is provided for COM1 only, on all RX3i models (except CPE302, CPE305, CPE330, CPE400, and CPL410), in Rel 7.16 and later releases.

For CPU COM2 (RS-485), the RTS signal is also controlled by the transmit driver. Therefore, control of RTS is dependent on the current state of the transmit driver. If the transmit driver is not enabled, asserting RTS with the Write Port Control COMMREQ will not cause RTS to be asserted on the serial line. The state of the transmit driver is controlled by the protocol and is dependent on the current Duplex Mode of the port. For 2-wire and 4-wire Duplex Mode, the transmit driver is only enabled during transmitting. Therefore, RTS on the serial line will only be seen active on COM2 (configured for 2-wire or 4-wire Duplex Mode) when data is being transmitted. For point-to-point Duplex Mode, the transmit driver is always enabled. Therefore, in point-to-point Duplex Mode, RTS on the serial line will always reflect what is chosen with the Write Port Control COMMREQ.

6.2.11 Cancel COMMREQ Function (4399)

This function cancels the current operations in progress. It can be used to cancel both read operations and write operations.

If a read operation is in progress and there are unprocessed characters in the input buffer, those characters are left in the input buffer and available for future reads. The serial port is not reset.

Example Command Block for the Cancel Operation Function

	Value (decimal)	Value (hexadecimal)	Meaning
Address	0002	0002	Data block length (2)
Address +1	0000	0000	NOWAIT mode
Address +2	0008	0008	Status word memory type (%R)
Address +3	0000	0000	Status word address minus 1 (%R0001)
Address +4	0000	0000	Not used
Address +5	0000	0000	Not used
Address +6	4399	112F	Cancel operation command
Address +7	0001	0001	Transaction type to cancel 1 - All operations 2 - Read operations 3 - Write operations

Operating Notes

Remote COMMREQs that are canceled due to this command executing will return a COMMREQ status word indicating request cancellation (minor code 12H).

CAUTION

If this COMMREQ is sent in either Cancel All or Cancel Write mode when a Write Bytes (4401) COMMREQ is transmitting a string from a serial port, transmission is halted. The position within the string where the transmission is halted is indeterminate. In addition, the final character received by the device to which the CPU is sending is also indeterminate.

6.2.12 Autodial Function (4400)

This feature allows the CPU to automatically dial a modem and send a specified byte string.

To implement this feature, the port must be configured for Serial I/O. After the autodial function is executed and the modem has established a connection, other serial I/O functions (Write bytes, Set Up Input Buffer, Flush Input buffer, Read port status, Write port control, Read bytes, Read String, and Cancel Operation) can be used.

Example

Pager enunciation can be implemented by three commands, requiring three COMMREQ command blocks:

Autodial: 04400 (1130h)	Dials the modem.
Write Bytes: 04401 (1131h)	Specifies an ASCII string, from 1 to 250 bytes in length, to send from the serial port.
Autodial: 04400 (1130h)	It is the responsibility of the application program to hang up the phone connection. This is accomplished by reissuing the autodial command and sending the hang-up command string.

Autodial Command Block

The Autodial command automatically transmits an Escape sequence that follows the Hayes convention. If you are using a modem that does not support the Hayes convention, you may be able to use the Write Bytes command to dial the modem.

Examples of commonly used command strings for Hayes-compatible modems are listed below:

Command String	Length	Function
ATDP15035559999<CR>	16 (10h)	Pulse dial the number 1-503-555-9999
ATDT15035559999<CR>	16 (10h)	Tone dial the number 1-503-555-9999
ATDT9,15035559999<CR>	18 (12h)	Tone dial using outside line with pause
ATH0<CR>	5 (05h)	Hang up the phone
ATZ <CR>	4 (04h)	Restore modem configuration to internally saved values

Sample Autodial Command Block

This COMMREQ command block dials the number 234-5678 using a Hayes-compatible modem.

Word	Definition	Values
1	0009h	CUSTOM data block length (includes command string)
2	0000h	NOWAIT mode
3	0008h	Status word memory type (%R)
4	0000h	Status word address minus 1 (Register 1)
5	0000h	not used
6	0000h	not used
7	04400 (1130h)	Autodial command number
8	00030 (001Eh)	Modem response timeout (30 seconds)
9	0012 (000Ch)	Number of bytes in the command string
10	5441h	A (41h), T (54h)
11	5444h	D (44h), T (54h)
12	3332h	Phone number: 2 (32h), 3 (33h)
13	3534h	4 (34h), 5 (35h)
14	3736h	6 (36h), 7 (37h)
15	0D38h	8 (38h) <CR> (0Dh)

6.2.13 Write Bytes Function (4401)

This operation can be used to transmit one or more characters to the remote device through the specified serial port. The character(s) to be transmitted must be in a word reference memory. They should not be changed until the operation is complete.

Up to 250 characters can be transmitted with a single invocation of this operation. The status of the operation is not complete until all the characters have been transmitted or until a timeout occurs (for example, if hardware flow control is being used and the remote device never enables the transmission).

Example Command Block for the Write Bytes Function

	Value (decimal)	Value (hexadecimal)	Meaning
Address	0006	0006	Data block length (includes characters to send)
Address +1	0000	0000	NOWAIT mode
Address +2	0008	0008	Status word memory type (%R)
Address +3	0000	0000	Status word address minus 1 (%R0001)
Address +4	0000	0000	Not used
Address +5	0000	0000	Not used
Address +6	4401	1131	Write bytes command
Address +7	0030	001E	Transmit time-out (30 seconds). See note below.
Address +8	0005	0005	Number of bytes to write
Address +9	25960	6568	‘h’ (68h), ‘e’ (65h)
Address +10	27756	6C6C	‘l’ (6Ch), ‘l’ (6Ch)
Address +11	0111	006F	‘o’ (6Fh)

Although printable ASCII characters are used in this example, there is no restriction on the values of the characters that can be transmitted.

Operating Notes

Specifying zero as the Transmit time-out sets the time-out value to the amount of time needed to transmit the data, plus 4 seconds.

CAUTION

If an Initialize Port (4300) COMMEQ is sent or a Cancel Operation (4399) COMMREQ is sent in either Cancel All or Cancel Write mode while this COMMREQ is transmitting a string from a serial port, transmission is halted. The position within the string where the transmission is halted is indeterminate. In addition, the final character received by the device the CPU is sending to is also indeterminate.

6.2.14 Read Bytes Function (4402)

This function causes one or more characters to be read from the specified port. The characters are read from the internal input buffer and placed in the specified input data area. The function returns both the number of characters retrieved and the number of unprocessed characters still in the input buffer. If zero characters of input are requested, only the number of unprocessed characters in the input buffer is returned.

If insufficient characters are available to satisfy the request and a non-zero value is specified for the number of characters to read, the status of the operation is not complete until either sufficient characters have been received or the time-out interval expires. In either of those conditions, the port status indicates the reason for completion of the read operation. The status word is not updated until the read operation is complete (either due to timeout or when all the data has been received).

If the time-out interval is set to zero, the COMMREQ remains pending until it has received the requested amount of data, or until it is canceled.

If this COMMREQ fails for any reason, no data is returned to the input data area. Any data that has not been read from the internal input buffer remains and can be retrieved with a subsequent read request.

Example Command Block for the Read Bytes Function

	Value (decimal)	Value (hexadecimal)	Meaning
Address	0005	0005	Data block length
Address +1	0000	0000	NOWAIT mode
Address +2	0008	0008	Status word memory type (%R)
Address +3	0000	0000	Status word address minus 1 (%R0001)
Address +4	0000	0000	Not used
Address +5	0000	0000	Not used
Address +6	4402	1132	Read bytes command
Address +7	0030	001E	Read time-out (30 seconds)
Address +8	0005	0005	Number of bytes to read
Address +9	0008	0008	Input data memory type (%R).
Address +10	0100	0064	Input data memory address (%R0100)

Return Data Format for the Read Bytes Function

The return data consists of the number of characters read, the number of characters still available in the input buffer after the read is complete (if any), and the actual input characters.

Address	The number of characters reads
Address + 1	Number of characters still available in the input buffer, if any
Address + 2	first two characters (the first character is in the low byte)
Address + 3	third and fourth characters (the third character is in the low byte)
Address + n	subsequent characters

Operating Notes for Read Bytes

If the input data memory type parameter is specified to be a word memory type, and if an odd number of bytes is received, then the high byte of the last word to be written with the received data is left unchanged.

As data is received from the serial port it is placed in the internal input buffer. If the buffer becomes full, then any additional data received from the serial port is discarded and the Overflow Error bit in the Port Status word (See Read Port Status Function) is set.

6.2.15 Read String Function (4403)

This function causes characters to be read from the specified port until a specified terminating character is received. The characters are read from the internal input buffer and placed in the specified input data area.

The function returns both the number of characters retrieved and the number of unprocessed characters still in the input buffer. If zero characters of input are requested, only the number of unprocessed characters in the input buffer is returned.

If the terminating character is not in the input buffer, the status of the operation is not complete until either the terminating character has been received or the time-out interval expires. In either of those conditions, the port status indicates the reason for completion of the read operation.

If the time-out interval is set to zero, the COMMREQ remains pending until it has received the requested string, terminated by the specified end character.

If this COMMREQ fails for any reason, no data is returned to the input data area. Any data that has not been read from the internal input buffer remains, and it can be retrieved with a subsequent read request.

Example Command Block for the Read String Function

	Value (decimal)	Value (hexadecimal)	Meaning
Address	0005	0005	Data block length
Address +1	0000	0000	NOWAIT mode
Address +2	0008	0008	Status word memory type (%R)
Address +3	0000	0000	Status word address minus 1 (%R0001)
Address +4	0000	0000	Not used
Address +5	0000	0000	Not used
Address +6	4403	1133	Read string command
Address +7	0030	001E	Read time-out (30 seconds)
Address +8	0013	000D	Terminating character (carriage return): must be between 0 and 255 (0xFF), inclusive
Address +9	0008	0008	Input data memory type (%R)
Address +10	0100	0064	Input data memory address (%R0100)

Return Data Format for the Read String Function

The return data consists of the number of characters read, the number of characters still available in the input buffer after the read is complete (if any), and the actual input characters:

Address	The number of characters read
Address + 1	Number of characters still available in the input buffer, if any
Address + 2	first two characters (the first character is in the low byte)
Address + 3	third and fourth characters (the third character is in the low byte)
Address + n	subsequent characters

Operating Notes for Read String

If the input data memory type parameter is specified to be a word memory type, and if an odd number of bytes is actually received, then the high byte of the last word to be written with the received data is left unchanged.

As data is received from the serial port it is placed in the internal input buffer. If the buffer becomes full, then any additional data received from the serial port is discarded and the Overflow Error bit in the Port Status word (See Read Port Status Function) is set.

6.3 RTU Slave Protocol

RTU protocol is a query-response protocol used for communication between the RTU device and a host computer, which is capable of communicating using RTU protocol. The host computer is the master device and it transmits a query to an RTU slave, which responds to the master. The RTU slave device cannot query; it can only respond to the master. A PACSystems CPU can only function as an RTU slave.

The RTU data transferred consists of 8-bit binary characters with an optional parity bit. No control characters are added to the data block; however, an error check (Cyclic Redundancy Check) is included as the final field of each query and response to ensure the accurate transmission of data.

Note: You should avoid using station address 1 for any other Modbus slave in a PACSystems control system because the default station address for the PACSystems CPU is 1. The CPU uses the default address in two situations:

1. If you power up without a configuration, the default station address of 1 is used.
2. When the Port Mode parameter is set to Message Mode, and Modbus becomes the protocol in STOP Mode, the station address defaults to 1, unless you specify a STOP Mode for the serial port in the CPU configuration, and then change the station address to be used for STOP Mode.

In either of these situations, if you have a slave configured with a station address of 1, confusion may result when the PACSystems CPU responds to requests intended for that slave.

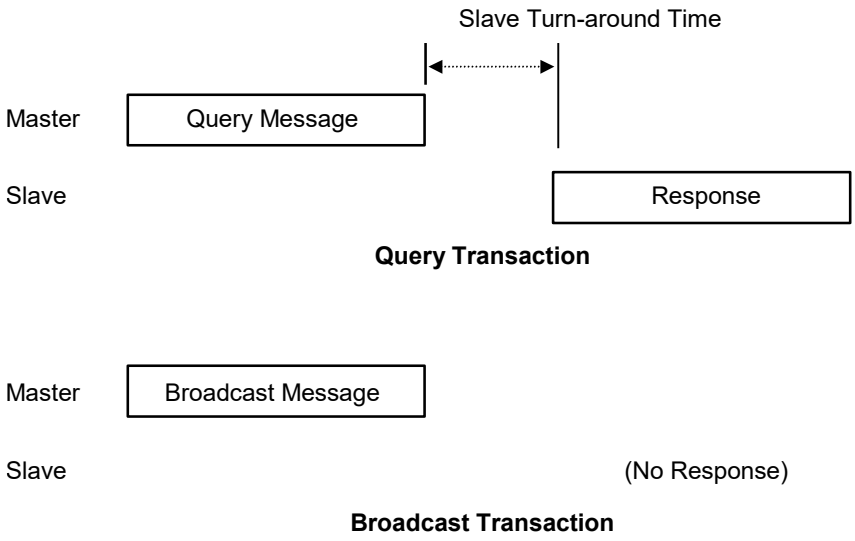
CPE400 and CPL410 do not support this protocol.

6.3.1 Message Format

The general formats for RTU message transfers are shown below:

RTU Message Transfers

Figure 46: RTU Message Transactions



The master device begins a data transfer by sending a query or broadcast request message. A slave completes that data transfer by sending a response message if the master sent a query message addressed to it. No response message is sent when the master sends a broadcast request.

RTU Slave Turnaround Time

The time between the end of a query and the beginning of the response to that query is called the slave turnaround time. The turnaround time of a PACSystems slave depends on the Controller Communications Window time and the sweep time of the PACSystems. RTU requests are processed only in the Controller Communications Window. In Normal sweep mode, the Controller Communications Window occurs once per sweep. Because the sweep time on PACSystems can be up to 2.5 seconds, the time to process an RTU request could be up to 2.5 seconds. Another factor is the Controller Communications Window time allowed in Hardware Configuration. If you configure a very small Controller Communications Window, the RTU request may not be completed in one sweep, causing RTU processing to require multiple sweeps. For details on CPU window modes, refer to *Window Modes* in Section 4.

Receive-to-Transmit Delay

Part of the RTU Slave Turnaround time is the receive-to-transmit delay. The RTU driver inserts this delay after a request from the master has been received, and before the response to the master is sent. Starting with Release 6.70 for the RX3i, the receive-to-transmit delay can be configured with the Serial Port Setup COMMREQ function 65520. The timeout is specified in units of 10 ms, with a range of

0–255 units (maximum delay is 2.55 seconds). If the specified time is less than 3.5 character times, then the delay is set to 3.5 character times.

Message Types

The RTU protocol has four message types: query, normal response, error response, and broadcast.

Query

The master sends a message addressed to a single slave.

Normal Response

After the slave performs the function requested by the query, it sends back a normal response for that function. This indicates that the request was successful.

Error Response

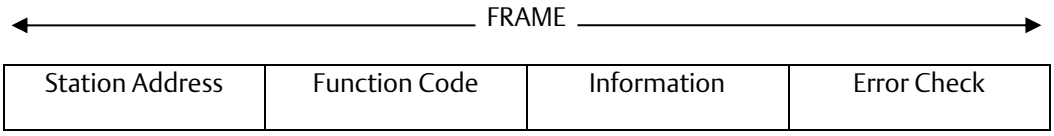
The slave receives the query, but cannot perform the requested function. The slave sends back an error response that indicates the reason the request could not be processed. (No error message will be sent for certain types of errors. For more information, refer to *Communication Errors* below.)

Broadcast

The master sends a message addressed to all the slaves by using address 0. All slaves that receive the broadcast message perform the requested function. This transaction is ended by a time-out within the master.

Message Fields

The message fields for a typical message are shown in the figure below and are explained in the following sections.



Station Address

The Station Address is the address of the slave station selected for this data transfer. It is one byte in length and has a value from 0 to 247 inclusive. An address of 0 selects all slave stations and indicates that this is a broadcast message. An address from 1 to 247 selects a slave station with that station address.

Function Code

The Function Code identifies the command being issued to the station. It is one byte in length and is defined for the values 0 to 255 as follows:

Function Code	Description
0	Illegal Function
1	Read Output Table
2	Read Input Table
3	Read Registers
4	Read Analog Input
5	Force Single Output
6	Preset Single Register
7	Read Exception Status
8	Loopback Maintenance
9-14	Unsupported Function
15	Force Multiple Outputs
16	Preset Multiple Registers
17	Report Device Type
18–21	Unsupported Function
22	Mask Write 4x Register
23	Read/Write 4x Registers
24–66	Unsupported Function
67	Read Scratch Pad Memory
68-127	Unsupported Function
128-255	Reserved for Exception Responses

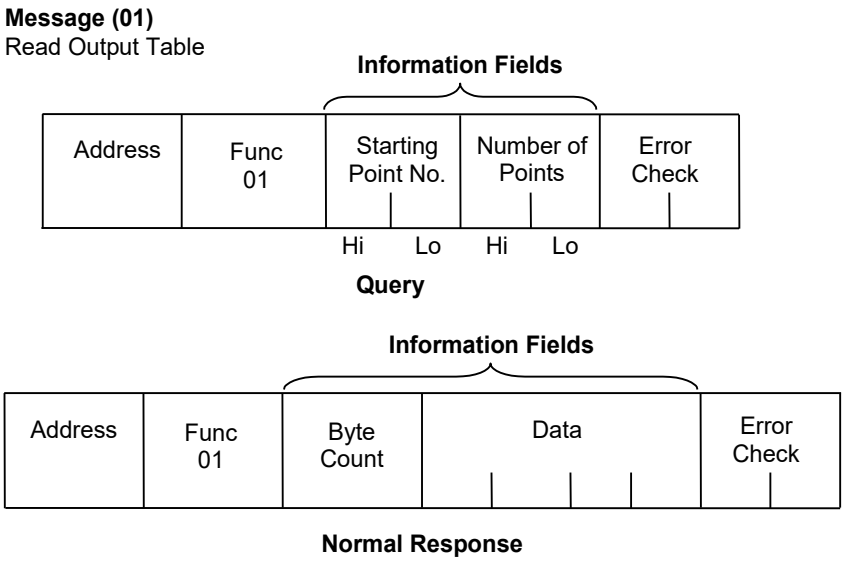
Information Fields

All message fields, other than the Station Address field, Function Code field, and Error Check field are called, generically, *information fields*. Information fields contain additional information required to specify or respond to a requested function. Different types of messages have different types or numbers of information fields. (Details on information fields for each message type and function code are found in *RTU Message Descriptions*. Some messages (Message 07 Query and Message 17 Query) do not have information fields.

Examples

As shown in the following figure, the information fields for the message *READ OUTPUT TABLE (01) Query* consist of the Starting Point No. field and Number of Points field. The information fields for the message *READ OUTPUT TABLE (01) Response* consist of the Byte Count field and Data field.

Figure 47: RTU Read Output Table Example



Some information fields include entries for the range of data to be accessed in the RTU slave.

Note: Data addresses are 0-based. This means you will need to subtract 1 from the actual address when specifying it in the RTU message. For message *(01) READ OUTPUT TABLE Query,* used in the example above, you would specify a starting data address in the Starting Point No. field. To specify %Q0001 as the starting address, you would place the address %Q0000 in this field. Also, the value placed in the Number of Points field determines how many %Q bits are read, starting with address %Q0001. For example:

- Starting Point No. field = %Q0007, so the starting address is %Q0008.
- The number of Points field = 16 (0010h), so addresses %Q0008 through %Q0023 will be read.

Error Check Field

The Error Check field is two bytes in length and contains a cyclic redundancy check (CRC-16) code. Its value is a function of the contents of the station Address, Function code, and Information field. The details of generating the CRC-16 code are described in *Cyclic Redundancy Check (CRC)*. Note that the Information field is variable in length. To properly generate the CRC-16 code, the length of the frame must be determined. To calculate the length of a frame for each of the defined function codes, see *Calculating the Length of Frame*.

Message Length

Message length varies with the type of message and the amount of data to be sent. Information for determining message length for individual messages is found in *RTU Message Descriptions*.

Character Format

A message is sent as a series of characters. Each byte in a message is transmitted as a character. The illustration below shows the character format. A character consists of a start bit (0), eight data bits, an optional parity bit, and a one-stop bit (1). Between characters, the line is held in the 1 state.

		MSB	Data Bits						LSB	
10	9	8	7	6	5	4	3	2	1	0
Stop	Parity (optional)									Start

Message Termination

Each station monitors the time between characters. When a period of three character times elapses without the reception of a character, the end of a message is assumed. The reception of the next character is assumed to be the beginning of a new message. The end of a frame occurs when the first of the following two events occurs:

1. The number of characters received for the frame is equal to the calculated length of the frame.
2. A length of 4 character times elapses without the reception of a character.

Timeout Usage

Timeouts are used on the serial link for error detection, error recovery, and to prevent the missing of the end of messages and message sequences. Note that although the module allows up to three character transmission times between each character in a message that it receives, there is no more than half a character time between each character in a message that the module transmits. After sending a query message, the master should wait an appropriate amount of time for the slave to turn around before assuming that the slave did not respond to the request. Slave turnaround time is affected by the Controller Communications Window time and the CPU sweep time, as described in *RTU Slave Turnaround Time*.

End-of-Frame Timeout

The End-of-frame timeout is a feature that compensates for message gaps that can occur due to the use of radio modems. The timeout is added to the amount of time allowed for receiving a message from the master. The timeout should be sized according to the maximum gap time that could be introduced by the master's transmitting equipment. Starting with Release 6.70 for the RX3i, the end-of-frame timeout can be configured with the Serial Port Setup COMMREQ function 65520. The timeout is specified in units of 100 µs. If the specified time is less than 3.5 character times, then the RTU driver sets the timeout to 3.5 character times.

6.3.2 Cyclic Redundancy Check (CRC)

The CRC is one of the most effective systems for checking errors. The CRC consists of two check characters generated at the transmitter and added at the end of the transmitted data characters. Using the same method, the receiver generates its own CRC for the incoming data and compares it to the CRC sent by the transmitter to ensure proper transmission. A complete mathematic derivation for the CRC is not given in this section. This information can be found in a number of texts on data communications. The essential steps that should be understood in calculating the CRC are as follows:

- The number of bits in the CRC multiplies the data bits that make up the message.

- The resulting product is then divided by the generating polynomial (using modulo 2 with no carries). The CRC is the remainder of this division.
- Disregard the quotient and add the remainder (CRC) to the data bits and transmit the message with CRC.
- The receiver then divides the message plus CRC by the generating polynomial and if the remainder is 0, the transmission was transmitted without error.

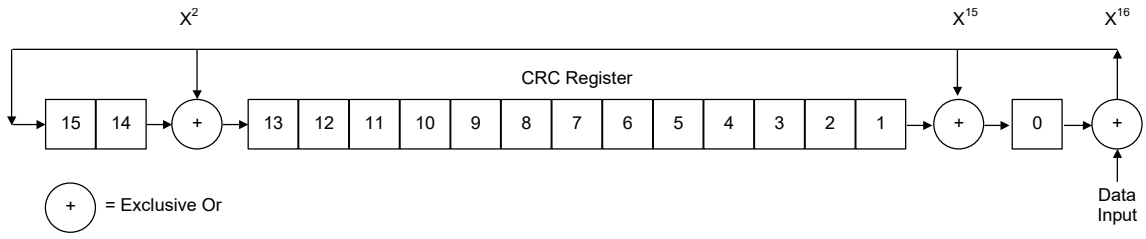
A generating polynomial is expressed algebraically as a string of terms in powers of X such as
$X_3 + X_2 + X_0$ (or 1)
which, in turn, can be expressed as the binary number 1101.

A generating polynomial could be any length and contain any pattern of 1s and 0s as long as both the transmitter and receiver use the same value. For optimum error detection, however, certain standard-generating polynomials have been developed. RTU protocol uses the polynomial
$X_{16} + X_{15} + X_2 + 1$ which in binary is 1 1000 0000 0000 0101. The CRC this polynomial generates is known as CRC-16.

The discussion above can be implemented in hardware or software. One hardware implementation involves constructing a multi-section shift register based on the generating polynomial.

Cyclic Redundancy Check Register

Figure 48: CRC Register Operation



To generate the CRC, the message data bits are fed to the shift register one at a time. The CRC register contains a preset value. As each data bit is presented to the shift register, the bits are shifted to the right. The LSB is XORed with the data bit and the result is: XORed with the old contents of bit 1 (the result placed in bit 0), XORed with the old contents of bit 14 (and the result placed in bit 13), and finally, it is shifted into bit 15. This process is repeated until all data bits in a message have been processed. Software implementation of the CRC-16 is explained in the section below.

Calculating the CRC-16

The pseudo-code for calculation of the CRC-16 is given below.

Preset byte count for data to be sent.

Initialize the 16-bit remainder (CRC) register to all ones.

XOR is the first 8-bit data byte with the high order byte of the 16-bit CRC register. The result is the current CRC.

INIT SHIFT: Initialize the shift counter to 0.

SHIFT: Shift the current CRC register 1 bit to the right.

Increment shift count.

Is the bit shifted out to the right (flag) a 1 or a 0?

If it is a 1, XOR is the generating polynomial with the current CRC.

If it is a 0, continue.

Is shift counter equal to 8?

If NO, return to SHIFT.

If YES, increment byte count.

Is the byte count greater than the data length?

If NO, XOR the next 8-bit data byte with the current CRC and go to INIT SHIFT.

If YES, add the current CRC to the end of the data message for transmission and exit.

When the message is transmitted, the receiver performs the same CRC operation on all the data bits and the transmitted CRC. If the information is received correctly the resulting remainder (receiver CRC) is 0.

Sample CRC-16 Calculation

The RTU device transmits the rightmost byte (of registers or discrete data) first. The first bit of the CRC-16 transmitted is the MSB. Therefore, in the example, the MSB of the CRC polynomial is to the extreme right. The X_{16} term is dropped because it affects only the quotient (which is discarded) and not the remainder (the CRC characters). The generating polynomial is therefore 1010 0000 0000 0001. The remainder is initialized to all 1s.

In this example, the CRC-16 is calculated for the RTU message, Read Exception Status 07. The message format is as follows:

Address	Function	CRC-16
01	07	

In this example, device number 1 (address 01) is queried. You need to know the amount of data to be transmitted and this information can be found for every message type in *Calculating the Length of Frame*. For this message, the data length is 2 bytes.

Transmitter CRC-16 Algorithm					
	MSB[83]		*LSB*[79]		*Flag*
Initial Remainder	1111	1111	1111	1111	
XOR 1st data byte	0000	0000	0000	0001	
Current CRC	1111	1111	1111	1111	
Shift 1	0111	1111	1111	1111	0
Shift 2	0011	1111	1111	1111	1
XOR Gen. Polynomial	1010	0000	0000	0001	
Current CRC	1001	1111	1111	1110	
Shift 3	0100	1111	1111	1111	0
Shift 4	0010	0111	1111	1111	1
XOR Gen. Polynomial	1010	0000	0000	0001	
Current CRC	1000	0111	1111	1110	
Shift 5	0100	0011	1111	1111	0
Shift 6	0010	0001	1111	1111	1
XOR Gen. Polynomial	1010	0000	0000	0001	
Current CRC	1000	0001	1111	1110	
Shift 7	0100	0000	1111	1111	0
Shift 8	0010	0000	0111	1111	1
XOR Gen. Polynomial	1010	0000	0000	0001	
Current CRC	1000	0000	0111	1110	
XOR 2nd data byte	0000	0000	0000	0111	
Current CRC	1000	0000	0111	1001	
Shift 1	0100	0000	0011	1100	1
XOR Gen. Polynomial	1010	0000	0000	0001	
Current CRC	1110	0000	0011	1101	
Shift 2	0111	0000	0001	1110	1
XOR Gen. Polynomial	1010	0000	0000	0001	
Current CRC	1101	0000	0001	1111	
Shift 3	0110	1000	0000	1111	1
XOR Gen. Polynomial	1010	0000	0000	0001	
Current CRC	1100	1000	0000	1110	
Shift 4	0110	0100	0000	0111	0
Shift 5	0011	0010	0000	0011	1
XOR Gen. Polynomial	1010	0000	0000	0001	
Current CRC	1001	0010	0000	0010	
Shift 6	0100	1001	0000	0001	0
Shift 7	0010	0100	1000	0000	1
XOR Gen. Polynomial	1010	0000	0000	0001	
Current CRC	1000	0100	1000	0001	
Shift 8	0100	0010	0100	0000	1
XOR Gen. Polynomial	1010	0000	0000	0001	
Transmitted CRC	1110	0010	0100	0001	
	E	2	4	1	

Receiver[82] *CRC-16 Algorithm*					
	MSB[79]		*LSB*[79]		*Flag*
Rcvr CRC after data	1110	0010	0100	0001	
XOR 1st byte Trns CRC	0000	0000	0100	0001	
Current CRC	1110	0010	0000	0000	
Shift 1	0111	0001	0000	0000	0
Shift 2	0011	1000	1000	0000	0
Shift 3	0001	1100	0100	0000	0
Shift 4	0000	1110	0010	0000	0
Shift 5	0000	0111	0001	0000	0
Shift 6	0000	0011	1000	1000	0
Shift 7	0000	0001	1100	0100	0
Shift 8	0000	0000	1110	0010	0
XOR 2nd byte Trns CRC	0000	0000	1110	0010	
Current CRC	0000	0000	0000	0000	
Shift 1-8 yields	0000	0000	0000	0000	
All errors for receiver final CRC-16 indicate transmission correct.					

Calculating the Length of Frame

To generate the CRC-16 for any message, the message length must be known. The length of all types of messages can be determined from the table below.

RTU Message Length

Function Code	*Name*	*Query or Broadcast Message Length Less CRC Code*	*Response Message Length Less CRC Code*
0		Not Defined	Not Defined
1	Read Output Table	6	3 + 3rd byte[84]
2	Read Input Table	6	3 + 3rd byte[84]
3	Read Registers	6	3 + 3rd byte[84]
4	Read Analog Input	6	3 + 3rd byte[84]
5	Force Single Output	6	6
6	Preset Single Register	6	6
7	Read Exception Status	2	3
8	Loopback/Maintenance	6	6
9-14		Not Defined	Not Defined
15	Force Multiple Outputs	7 + 7th byte[84]	6
16	Preset Multiple Registers	7 + 7th byte[84]	6
17	Report Device Type	2	8
18-21		Not Defined	Not Defined
22	Mask Write 4x Registers	8	8
23	Read/Write 4x Registers	13+byte 11[84]	5+byte 3[84]
24–66		Not Defined	Not Defined
67	Read Scratch Pad	6	3 + 3rd byte[84]
68-127		Not Defined	Not Defined
128-255		Not Defined	3

6.3.3 RTU Message Descriptions

[82] *The receiver processes incoming data through the same CRC algorithm as the transmitter. The example for the receiver starts at the point after all the data bits but not the transmitted CRC have been received correctly. Therefore, the receiver CRC should be equal to the transmitted CRC at this point. When this occurs, the output of the CRC algorithm will be zero indicating that the transmission is correct.*

The transmitted message with CRC would then be:

Address	*Function*	*CRC–16*	
01	*07*	*41*	*E2*

[83] *The MSB and LSB references are to the data bytes only, not to the CRC bytes. The CRC MSB and LSB order are the reverse of the data byte order.*

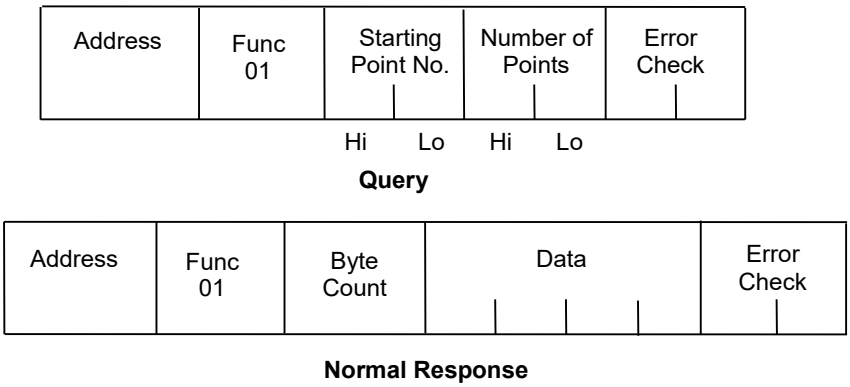
[84] *The value of this byte is the number of bytes contained in the data being transmitted.*

This section presents the format and fields for each RTU message.

Message (01): Read Output Table

Format

Figure 49: RTU Read Output Table Message Format



Query

- An address of 0 is not allowed because this cannot be a broadcast request.
- The function code is 01.
- The starting point number is two bytes in length and may be any value less than the highest output point number available in the attached CPU. The starting point number is equal to one less than the number of the first output point returned in the normal response to this request.
- The *number of points* value is two bytes in length. It specifies the number of output points returned in the normal response. The sum of the starting point value and the number of points value must be less than or equal to the highest output point number available in the attached CPU. The high order byte of the Starting Point Number and Number of Points fields is sent as the first byte. The low-order byte is the second byte in each of these fields.

Response:

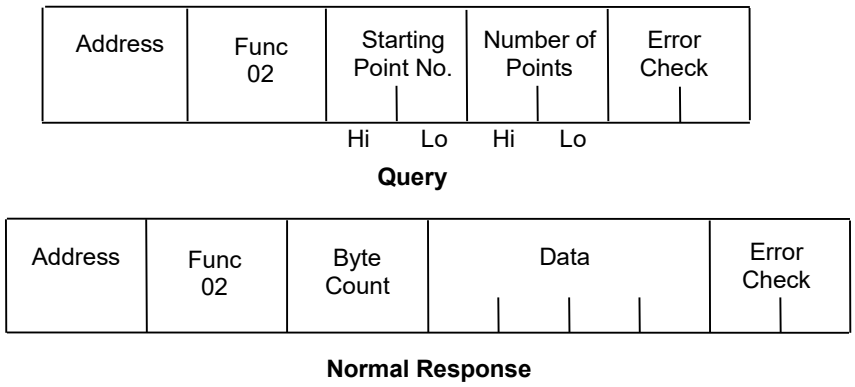
The byte count is a binary number from 1 to 256 (0 = 256). It is the number of bytes in the normal response following the byte count and preceding the error check.

The Data field of the normal response is packed with output status data. Each byte contains eight output point values. The least significant bit (LSB) of the first byte contains the value of the output point whose number is equal to the starting point number plus one. The values of the output points are ordered by number starting with the LSB of the first byte of the Data field and ending with the most significant bit (MSB) of the last byte of the Data field. If the number of points is not a multiple of 8, the last data byte contains zeroes in one to seven of its highest order bits.

Message (02): Read Input Table

Format

Figure 50: RTU Read Input Table Message Format



Query

- An address of 0 is not allowed as this cannot be a broadcast request.
- The function code is 02.
- The starting point number is two bytes in length and may be any value less than the highest input point number available in the attached CPU. The starting point number is equal to one less than the number of the first input point returned in the normal response to this request.
- The number of points value is two bytes in length. It specifies the number of input points returned in the normal response. The sum of the starting point value and the number of points value must be less than or equal to the highest input point number available in the attached CPU. The high order byte of the Starting Point Number and Number Of Bytes fields is sent as the first byte. The low-order byte is the second byte in each of these fields.

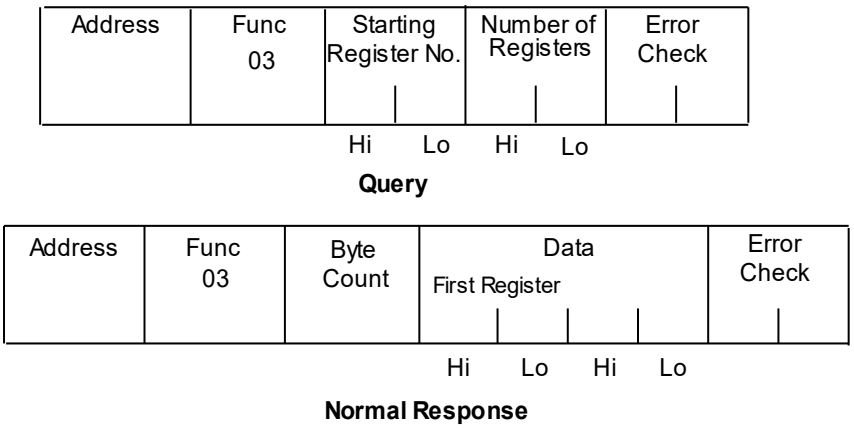
Response

- The byte count is a binary number from 1 to 256 (0 = 256). It is the number of bytes in the normal response following the byte count and preceding the error check.
- The Data field of the normal response is packed with input status data. Each byte contains eight input point values. The least significant bit (LSB) of the first byte contains the value of the input point whose number is equal to the starting point number plus one. The values of the input points are ordered by number starting with the LSB of the first byte of the Data field and ending with the most significant bit (MSB) of the last byte of the Data field. If the number of points is not a multiple of 8, then the last data byte contains zeroes in one to seven of its highest order bits.

Message (03): Read Registers

Format

Figure 51: RTU Read Registers Message Format



Query

- An address of 0 is not allowed as this request cannot be a broadcast request.
- The function code is equal to 3.
- The starting register number is two bytes in length. The starting register number may be any value less than the highest register number available in the attached CPU. It is equal to one less than the number of the first register returned in the normal response to this request.
- The number of registers values is two bytes in length. It must contain a value from 1 to 125 inclusive. The sum of the starting register value and the number of registers value must be less than or equal to the highest register number available in the attached CPU. The high order byte of the Starting Register Number and Number of Registers fields is sent as the first byte in each of these fields. The low-order byte is the second byte in each of these fields.

Response

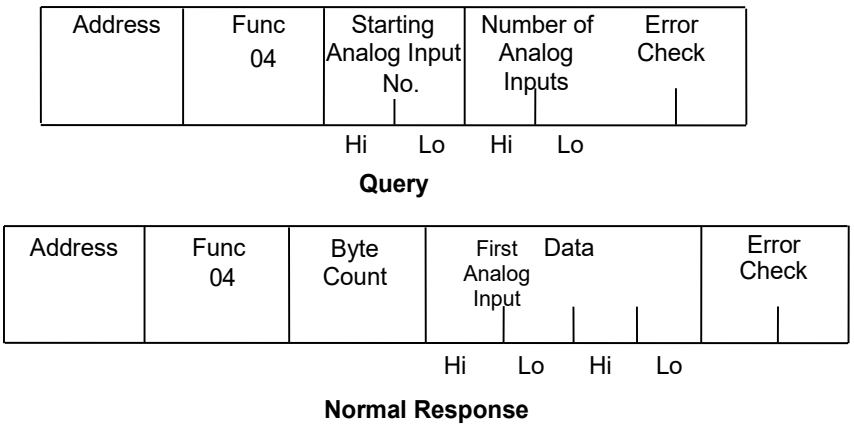
The byte count is a binary number from 2 to 250 inclusive. It is the number of bytes in the normal response following the byte count and preceding the error check. Note that the byte count is equal to two times the number of registers returned in the response. A maximum of 250 bytes (125) registers is set so that the entire response can fit into one 256-byte data block.

The registers are returned in the Data field in order of number with the lowest number register in the first two bytes and the highest number register in the last two bytes of the Data field. The number of the first register in the Data field is equal to the Starting Register Number plus one. The high order byte is sent before the low order byte of each register.

Message (04): Read Analog Inputs

Format

Figure 52: RTU Read Analog Inputs Message Format



Query

- An Address of 0 is not allowed as this request cannot be a broadcast request.
- The function code is equal to 4.
- The Starting Analog Input Number is two bytes in length. The Starting Analog Input Number may be any value less than the highest analog input number available in the attached CPU. It is equal to one less than the number of the first analog input returned in the normal response to this request.
- The Number Of Analog Inputs value is two bytes in length. It must contain a value from 1 to 125 inclusive. The sum of the Starting Analog Input value and the Number Of Analog Inputs value must be less than or equal to the highest analog input number available in the at-attached CPU. The high order byte of the Starting Analog Input Number and Number of Analog Inputs fields is sent as the first byte in each of these fields. The low-order byte is the second byte in each of these fields.

Response

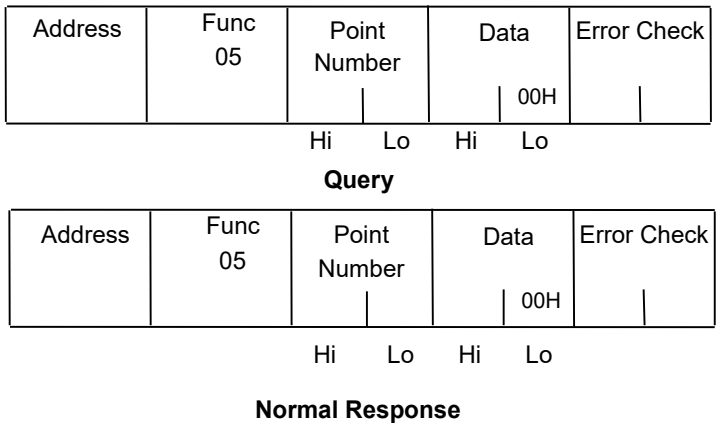
The Byte Count is a binary number from 2 to 250 inclusive. It is the number of bytes in the normal response following the byte count and preceding the error check. Note that the Byte Count is equal to two times the number of analog inputs returned in the response. A maximum of 250 bytes (125) of analog inputs is set so that the entire response can fit into one 256-byte data block.

The analog inputs are returned in the Data field in order of number with the lowest number of analog inputs in the first two bytes and the highest number of analog inputs in the last two bytes of the Data field. The number of the First Analog Input in the Data field is equal to the Starting analog input number plus one. The high order byte is sent before the low order byte of each analog input.

Message (05): Force Single Output

Format

Figure 53: RTU Force Single Output Message Format



Query

- An Address of 0 indicates a broadcast request. All slave stations process a broadcast request and no response is sent.
- The function code is equal to 05.
- The Point Number field is two bytes in length. It may be any value less than the highest output point number available in the attached CPU. It is equal to one less than the number of the output point to be forced on or off.
- The first byte of the Data field is equal to either 0 or 255 (FFH). The output point specified in the Point Number field is to be forced off if the first Data field byte is equal to 0. It is to be forced on if the first Data field byte is equal to 255 (FFH). The second byte of the Data field is always equal to zero.

Response

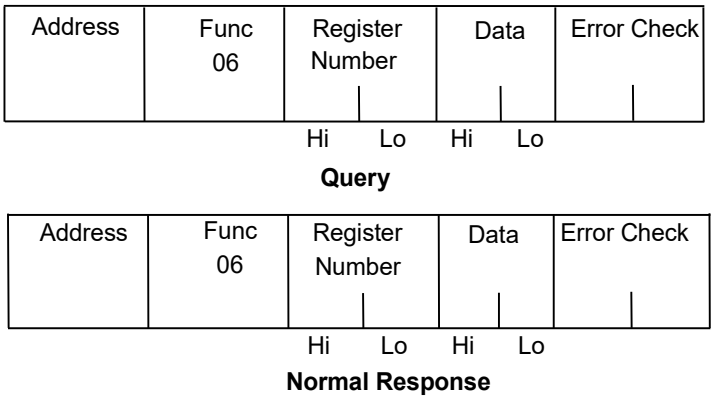
The normal response to a force single output query is identical to the query.

Note: The force single output request is not an output override command. The output specified in this request is ensured to be forced to the value specified only at the beginning of one sweep of the user logic.

Message (06): Preset Single Register

Format

Figure 54: RTU Preset Single Register Message Format



Query

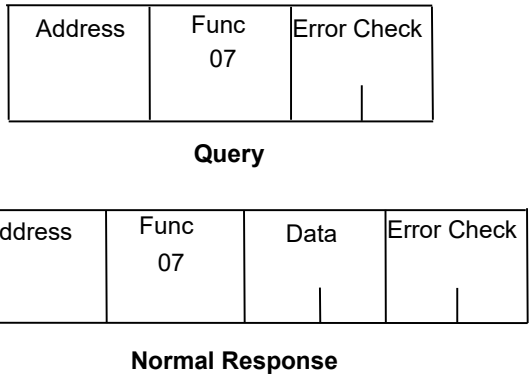
- An Address 0 indicates a broadcast request. All slave stations process a broadcast request and no response is sent.
- The function code is equal to 06.
- The Register Number field is two bytes in length. It may be any value less than the highest register available in the attached CPU. It is equal to one less than the number of the register to be preset.
- The Data field is two bytes in length and contains the value that the register specified by the Register Number Field is to be preset to. The first byte in the Data field contains the high order byte of the preset value. The second byte in the Data field contains the low order byte.

Response

The normal response to a preset single register query is identical to the query. Message (07): Read Exception Status

Format

Figure 55: RTU Read Exception Status Message Format



Query:

This query is a short form of a request to read the first eight output points.

- An Address of zero is not allowed as this cannot be a broadcast request.
- The function code is equal to 07.

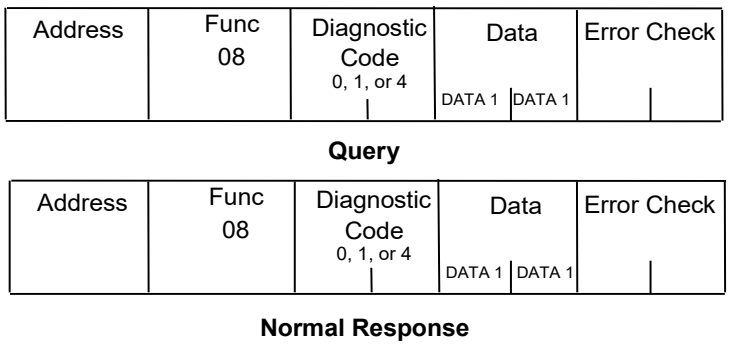
Response:

The Data field of the normal response is one byte in length and contains the states of output points one through eight. The output states are packed in order of number with output point one's state in the least significant bit and output point eight's state in the most significant bit.

Message (08): Loopback/Maintenance (General)

Format

Figure 56: RTU Loopback/Maintenance Message Format



Query

- The Function code is equal to 8.
- The Diagnostic Code is two bytes in length. The high order byte of the Diagnostic Code is the first byte sent in the Diagnostic-Code field. The low order byte is the second byte sent. The loopback/maintenance command is defined only for Diagnostic Codes equal to 0, 1, or 4. All other Diagnostic Codes are reserved.
- The Data field is two bytes in length. The contents of the two Data bytes are defined by the value of the Diagnostic Code. Response:
- See descriptions for individual Diagnostic Codes.
- Diagnostic Return Query Data Request (Loopback/Maintenance Code 00):
- An address of 0 is not allowed for the return query data request.
- The values of the two Data field bytes in the query are arbitrary.
- The normal response is identical to the query.
- The values of the data bytes in the response are equal to the values sent in the query.

Diagnostic Initiate Communication Restart Request (Loopback/Maintenance Code 01):

- An Address of 0 indicates a broadcast request. All slave stations process a broadcast request and no response is sent.
- This request disables the listen-only mode (enables responses to be sent when queries are received so that communications can be restarted).
- The value of the first byte of the Data field (DATA1) must be 0 or FF. Any other value will cause an error response to be sent. The value of the second byte of the Data field (DATA2) is always equal to 0.
- The normal response to an Initiate Communication Restart query is identical to the query.

Diagnostic Force Listen-Only Mode Request (Loopback/Maintenance code 04):

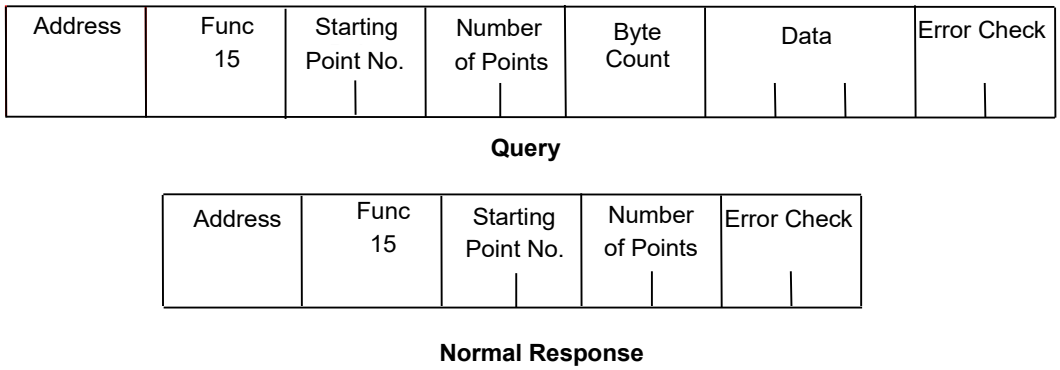
- An Address of 0 indicates a broadcast request. All slave stations process a broadcast request.
- After receiving a Force Listen-Only mode request, the RTU device will go into the listen-only mode, will not perform a requested function, and will not send either normal or error responses to any queries. The listen-only mode is disabled when the RTU device receives an Initiate Communication Restart request or when the RTU device is powered up.
- Both bytes in the Data field of a Force Listen-Only Mode request are equal to 0. The RTU device never sends a response to a Force Listen-Only Mode request.

Note: Upon power-up, the RTU device disables the listen-only mode and is enabled to continue sending responses to queries.

Message (15): Force Multiple Outputs

Format

Figure 57: RTU Force Multiple Outputs Message Format



Query

- An Address of 0 indicates a broadcast request. All slave stations process a broadcast request and no response is sent.
- The value of the Function code is 15.
- The Starting Point Number is two bytes in length and may be any value less than the highest output point number available in the attached CPU. The Starting Point Number is equal to one less than the number of the first output point forced by this request.
- The Number of Points value is two bytes in length. The sum of the Starting Point Number and the Number of Points value must be less than or equal to the highest output point number available in the attached CPU. The high order byte of the Starting Point Number and Number of Bytes fields is sent as the first byte in each of these fields. The low-order byte is the second byte in each of these fields.
- The Byte Count is a binary number from 1 to 256 (0 = 256). It is the number of bytes in the Data field of the force multiple outputs request.
- The Data field is packed with data containing the values that the outputs specified by the Starting Point Number and the Number of Points fields are to be forced to. Each byte in the Data field contains the values that eight output points are to be forced to. The least significant bit (LSB) of the first byte contains the value that the output point whose number is equal to the starting point number plus one is to be forced to. The values for the output points are ordered by number starting with the LSB of the first byte of the Data field and ending with the most significant bit (MSB) of the last byte of the Data field. If the number of points is not a multiple of 8, then the last data byte contains zeroes in one to seven of its highest order bits.

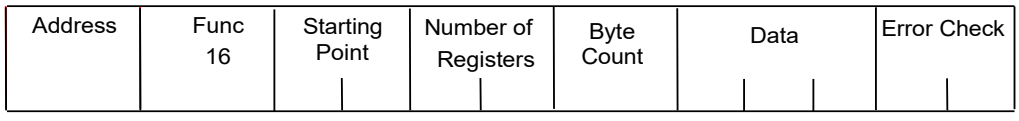
Response

- The descriptions of the fields in the response are covered in the query description.

Note: The force multiple outputs request is not an output override command. The outputs specified in this request are ensured to be forced to the values specified only at the beginning of one sweep of the user logic. Message (16): Preset Multiple Registers

Format:

Figure 58: RTU Preset Multiple Registers Message Format



Query

Normal Response

Query

- An Address of 0 indicates a broadcast request. All slave stations process a broadcast request and no response is sent.
- The value of the Function code is 16.
- The Starting Register Number is two bytes in length. The Starting Register Number may be any value less than the highest register number available in the attached CPU. It is equal to one less than the number of the first register preset by this request.
- The Number of Registers value is two bytes in length. It must contain a value from 1 to 125 inclusive. The sum of the Starting Register Number and the Number of Registers value must be less than or equal to the highest register number available in the attached CPU. The high order byte of the Starting Register Number and Number of Registers fields is sent as the first byte in each of these fields. The low-order byte is the second byte in each of these fields.
- The Byte Count field is one byte in length. It is a binary number from 2 to 250 inclusive. It is equal to the number of bytes in the data field of the preset multiple registers request. Note that the Byte Count is equal to twice the value of the Number of Registers.
- The registers are returned in the Data field in order of number with the lowest number register in the first two bytes and the highest number register in the last two bytes of the Data field. The number of the first register in the Data field is equal to the starting register number plus one. The high order byte is sent before the low order byte of each register.

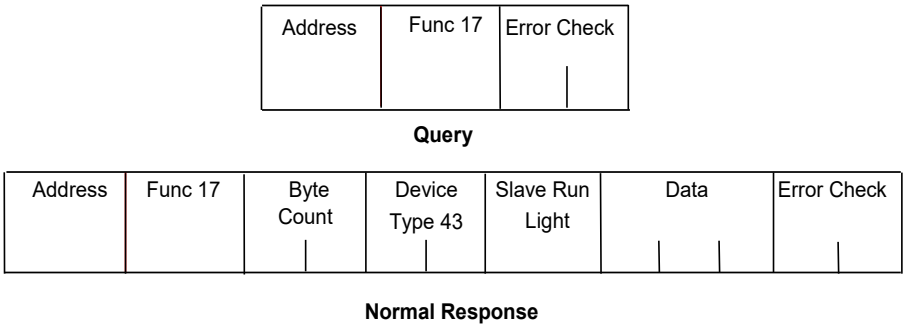
Response

The descriptions of the fields in the response are covered in the query description.

Message (17): Report Device Type

Format

Figure 59: RTU Report Device Type Message Format



Query:

The Report Device Type query is sent by the master to a slave to learn what type of programmable control or another computer it is.

- An Address of zero is not allowed as this cannot be a broadcast request.
- The Function code is 17.

Response

- The Byte Count field is one byte in length and is equal to 5.
- The Device Type field is one byte in length and is equal to 43 (hexadecimal) for PACSystems
- The Slave Run Light field is one byte in length. The Slave Run Light byte is equal to OFFH if the CPU is in RUN Mode. It is equal to 0 if the CPU is not in RUN Mode.
- The Data field contains three bytes. For PACSystems CPUs, the first byte is the Minor Type, and the remaining bytes are zeroes. The following table lists minor types.

Response Data (Minor Type)	CPU Model[85]
02 hex	IC698CPE010
04 hex	IC698CPE020
05 hex	IC698CRE020
06 hex	IC698CPE030
08 hex	IC698CPE040
0A hex	IC695CPE302 IC695CPE305 IC695CPU310
0C hex	IC695NIU001
10 hex	IC695CPU320
11 hex	IC695CRU320
12 hex	IC695CPE302 IC695CPE305
18 hex	IC695CPU315

[85] *Does not apply to CPE330, which has no serial ports.*

Message (22): Mask Write 4x Memory

Modifies the contents of a specified 4*x* register using a combination of an AND mask, an OR mask, and the register's current contents. The function can be used to set or clear individual bits in the register. Broadcast is not supported.

Query:

The query specifies the 4x reference to be written, the data to be used as the AND mask, and the data to be used as the OR mask.

The function's algorithm is:

Result = (Current Contents AND And_Mask) OR (Or_Mask AND And_Mask)

For example:

	Hex	Binary	
Current Contents	12	0001	0010
And_Mask	F2	1111	0010
Or_Mask	25	0010	0101
And_Mask	0D	0000	1101
Result	17	0001	0111

Note: If the Or_Mask value is zero, the result is simply the logical ANDing of the current contents and And_Mask. If the And_Mask value is zero, the result is equal to the Or_Mask value.

Note: The contents of the register can be read with the Read Holding Registers function (function code 03). They could, however, be changed subsequently as the controller scans its user logic program.

Example of a Mask Write to register 5 in slave device 17, using the above mask values:

Field Name	Example (Hex)
Slave Address	11
Function	16
Reference Address Hi	00
Reference Address Lo	04
And_Mask Hi	00
And_Mask Lo	F2
Or_Mask Hi	00
Or_Mask Lo	25
Error Check (LRC or CRC)	--

Response

The normal response is an echo of the query. The response is returned after the register has been written.

Message (23): Read Write 4x Memory

Performs a combination of one read and one write operation in a single Modbus transaction. The function can write new contents to a group of 4*x* registers, and then return the contents of another group of 4*x* registers. Broadcast is not supported.

Query

The query specifies the starting address and quantity of registers of the group to be read. It also specifies the starting address, the number of registers, and data for the group to be written. The Byte Count field specifies the number of bytes to follow in the Write Data field.

Here is an example of a query to read six registers starting at register 5, and to write three registers starting at register 16, in slave device 17:

Field Name	Example (Hex)
Slave address	11
Function	17
Read Reference Address Hi	00
Read Reference Address Lo	04
Quantity to Read Hi	00
Quantity to Read Lo	06
Write Reference Address Hi	00
Write Reference Address Lo	0F
Quantity to Write Hi	00
Quantity to Write Lo	03
Byte Count	06
Write Data 1 Hi	00
Write Data 1 Lo	FF
Write Data 2 Hi	00
Write Data 2 Lo	FF
Write Data 3 Hi	00
Write Data 3 Lo	FF
Error Check (LRC or CRC)	--

Response

The normal response contains the data from the group of registers that were read. The Byte Count field specifies the number of bytes to follow in the Read Data field.

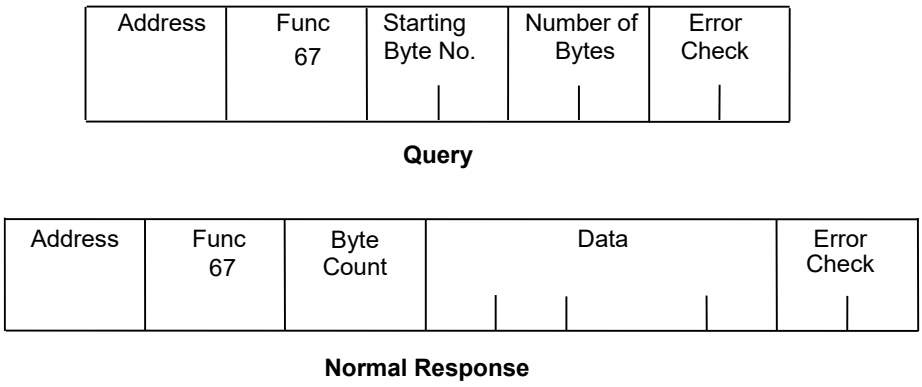
Here is an example of a response to the query:

Field Name	Example (Hex)
Slave Address	11
Function	17
Byte Count	0C
Read Data 1 Hi	00
Read Data 1 Lo	FE
Read Data 2 Hi	0A
Read Data 2 Lo	CD
Read Data 3 Hi	00
Read Data 3 Lo	01
Read Data 4 Hi	00
Read Data 4 Lo	03
Read Data 5 Hi	00
Read Data 5 Lo	0D
Read Data 6 Hi	00
Read Data 6 Lo	FF
Error Check (LRC or CRC)	--

Message (67): Read Scratch Pad Memory

Format:

Figure 60: RTU Read Scratch Pad Memory Message Format



Query

- An Address of 0 is not allowed as this cannot be a broadcast request.
- The Function Code is equal to 67.
- The Starting Byte Number is two bytes in length and may be any value less than or equal to the highest scratchpad memory address available in the attached CPU as indicated in the table below. The Starting Byte Number is equal to the address of the first scratchpad memory byte returned in the normal response to this request.
- The Number of Bytes value is two bytes in length. It specifies the number of scratchpad memory locations (bytes) returned in the normal response. The sum of the Starting Byte Number and the Number of Bytes values must be less than two plus the highest scratchpad memory address available in the attached CPU. The high order byte of the Starting Byte Number and Number of Bytes fields is sent as the first byte in each of these fields. The low-order byte is the second byte in each of the fields.

Response

- The Byte Count is a binary number from 1 to 256 (0 = 256). It is the number of bytes in the Data field of the normal response.
- The Data field contains the contents of the scratchpad memory requested by the query. The scratchpad memory bytes are sent in order of address. The contents of the scratchpad memory byte whose address is equal to the Starting Byte Number are sent in the first byte of the Data field. The contents of the scratchpad memory byte whose address is equal to one less than the sum of the starting byte number and number of bytes values are sent in the last byte of the Data field.

6.3.4 RTU Scratch Pad

The entire scratchpad is updated every time an external READ request is received by the PACSystems RTU slave. All scratchpad locations are *read-only*. The scratchpad is a byte-oriented memory type.

RTU Scratch Pad Memory Allocation

SP Address	Field Identifier	Bits							
		7	6	5	4	3	2	1	0
00	CPU Run Status	0	0	0	0	See note.[86]			
01	CPU Command Status	Bit pattern same as SP(00)							
02 03	CPU Type	Major[87] (in hexadecimal) Minor[88] (in hexadecimal)							
04 – 0B	CPU SNP ID	7 ASCII characters + termination character (00h)							
0C 0D	CPU Firmware Revision No.	Major (in BCD) Minor (in BCD)							
0E 0F	Communications Management Module (CMM) Firmware Revision No.	Major Minor							
10—11	Reserved	00h							
12	Node Type Identifier	PACSystems 43 (hexadecimal)							
13—15	Reserved	00h							
16	RTU Station Address	1—247 (decimal)							
17	Reserved	00h							
18—33[89]	Sizes of Memory Types								
18—1B	Register Memory	%R size (words)							
1C—1F	Analog Input Table	%AI size (words)							
20—23	Analog Output Table	%AO size (words)							
24—27	Input Table	%I size (bits)							
28—2B	Output Table	%O size (bits)							
2C—2F	Internal Discrete Memory	%M size (bits)							
30—33	User Program Code	The amount of program memory occupied by the logic program.							
34—FF	Reserved	00h							

6.3.5 Communication Errors

[86] *0000 = Run_Enabled 0100 = Halted 0001 = Run_Disabled 0101 = Suspended 0010 = Stopped 0110 = Stopped_IO_Enabled*

[87] *CPU Major Type Codes: PACSystems 0x43*

[88] *PACSystems Minor Types for CPU: refer to*

Message (17): Report Device Type

[89] *Four bytes hold the hexadecimal length of each memory type with the most significant word reserved for future expansion. For example, the default register memory size of 1024 words (0400h) would be returned in the following format:*

Word	*Least*	*Significant*	*Most*	*Significant*
SP Byte	*18*	*19*	*1A*	*1B*
Contains	*00*	*04*	*00*	*00*

Serial link communication errors are divided into three groups:

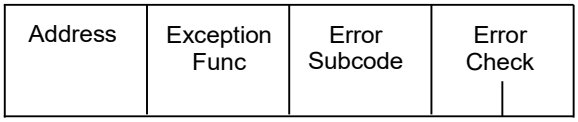
- Invalid Query Message
- Serial Link Time Outs
- Invalid Transaction

Invalid Query Message

When the communications module receives a query addressed to itself, but cannot process the query, it sends one of the following error responses:

	Subcode
Invalid Function Code	1
Invalid Address Field	2
Invalid Data Field	3
Query Processing Failure	4

The format for an error response to a query is as follows:



The address reflects the address provided on the original request. The exception function code is equal to the sum of the function code of the query plus 128. The error subcode is equal to 1, 2, 3, or 4. The value of the subcode indicates the reason the query could not be processed.

Invalid Function Code Error Response (1)

An error response with a subcode of 1 is called an invalid function code error response. This response is sent by a slave if it receives a query whose function code is not equal to 1- 8, 15, 16, 17, or 67.

Note: Starting with Release 6.70 for the RX3i, the invalid function code error response is not used. Instead, undefined and unsupported function codes are ignored, and no response is generated.

Invalid Address Error Response (2)

An error response with a subcode of 2 is called an invalid address error response. This error response is sent in the following cases:

1. The Starting Point Number and Number of Points fields specify output points or input points that are not available in the attached CPU (returned for function codes 1, 2, 15).
2. The Starting Register Number and Number of Registers fields specify registers that are not available in the attached CPU (returned for function codes 4, 16).
3. The Starting Analog Input Number and Analog Input Number fields specify analog inputs that are not available in the attached CPU (returned for function code 3).
4. The Point Number field specifies an output point not available in the attached CPU (returned for function code 5).
5. The Register Number field specifies a register not available in the attached CPU (returned for function code 6).
6. The Analog Input Number field specifies an analog input number not available in the at-attached CPU (returned for function code 3).
7. The Diagnostic Code is not equal to 0, 1, or 4 (returned for function code 8).
8. The starting Byte Number and Number of Bytes fields specify a scratchpad memory address that is not available in the attached CPU (returned for function code 67).

Invalid Data Value Error Response (3)

An error response with a subcode of 3 is called an invalid data value error response. This response is sent in the following cases:

The first byte of the Data field is not equal to 0 or 255 (FFh) or the second byte of the Data field is not equal to 0 for the Force Single Output Request (Function Code 5) or the initiate communication restart request (function code 8, diagnostic code 1). The two bytes of the Data field are not both equal to 0 for the Force Listen-Only request (Function Code 8, Diagnostic Code 4). This response is also sent when the data length specified by the Memory Address field is longer than the data received.

Query Processing Failure Error Response (4)

An error response with a subcode of 4 is called a query processing failure response. This error response is sent by an RTU device if it properly receives a query but communication between the associated CPU and the CMM fails.

Serial Link Timeout

The only cause for an RTU device to timeout is if an interruption to a data stream of 4 character times occurs while a message is being received. If this occurs the message is considered to have terminated and no response will be sent to the master. There are certain timing considerations due to the characteristics of the slave that should be taken into account by the master. After sending a query message, the master should wait an appropriate amount of time for the slave to turn around before assuming that the slave did not respond to the request. Slave turnaround time is affected by the Controller Communications Window time and the CPU sweep time, as described in *RTU Slave Turnaround Time*.

Invalid Transactions

If an error occurs during transmission that does not fall into the category of an invalid query message or a serial link time-out, it is known as an invalid transaction. Types of errors causing an invalid transaction include:

- Bad CRC
- The data length specified by the Memory Address field is longer than the data received
- Framing or overrun errors
- Parity errors

If an error in this category occurs when a message is received by the slave serial port, the slave does not return an error message; rather the slave ignores the incoming message, treating the message as though it was not intended for it.

6.3.6 RTU Slave/SNP Slave Operation with Programmer Attached

A port that has been configured for RTU Slave protocol can switch to SNP protocol if an SNP master such as a programmer begins communicating to the port. The programmer must use the same serial communications parameters (baud rate, parity, stop bits, etc.) as the currently active RTU Slave protocol for it to be recognized. When the CPU recognizes the SNP master, the CPU removes the RTU Slave protocol from the port and installs SNP Slave as the active protocol.

The SNP protocol that is installed in this case has the following fixed characteristics:

- The SNP ID is set to blank. Therefore, the SNP master must use a blank ID in the SNP attached message. This also means that this capability is only useful for point-to-point connections.
- The turnaround time is set to 0ms.
- The idle timeout is set to 10 seconds.

After the programmer is removed, there is a slight delay (equal to the idle timeout) before the CPU recognizes its absence. During this time, no messages are processed on the port. The CPU detects the removal of the programmer as an SNP Slave protocol timeout. Therefore, it is important to be careful when disabling timeouts used by the SNP Slave protocol.

When the CPU recognizes the programmer disconnect, it reinstalls the RTU Slave protocol unless a new protocol has been configured in the meantime. In that case, the CPU installs the new protocol instead.

Example

1. COM1 is running RTU Slave protocol at 9600 baud.
2. A programmer is attached to COM1. The programmer is using 9600 baud.
3. The CPU installs SNP Slave on COM1 and the programmer communicates normally.
4. The programmer stores a new configuration to COM1. The new configuration sets the port for SNP Slave at 4800 baud (it will not take effect until the port loses communications with the programmer).
5. When the CPU loses communication with the programmer, the new configuration takes effect.

6.4 SNP Slave Protocol

PACSystems CPUs can communicate with Machine Edition software through either COM1 or COM2 using SNP slave protocol.

CPU COM1 is wired as an RS-232 Data Communications Equipment (DCE) port and can be connected directly using a straight-through cable to one of the serial ports of a PC running Machine Edition or other SNP master software.

CPU COM2 is wired for RS-485. If the SNP master does not have an RS-485 port, an RS-485/RS-232 converter is required. The RX3i can use converter IC690ACC901, which uses +5Vdc from the serial port. The CPU COM2 does not support IC690ACC901 and requires an externally powered converter.

PACSystems provides the *break-free* version of SNP so that the SNP master does not need to issue a break signal as part of the SNP attach sequence. However, the CPU responds appropriately if a break signal is detected, by resetting the protocol to wait for another attach sequence from the master.

PACSystems supports both point-to-point connections (single master/single slave) and multi-drop connections (single master/multiple slaves).

For details on SNP protocol, refer to the *Series 90 PLC Serial Communications User's Manual,* GFK-0582.

CPE400 and CPL410 do not support this protocol.

6.4.1 Permanent Datagrams

Permanent datagrams survive after the SNP session that created them has been terminated. This allows an SNP master device to periodically retrieve datagram data from a number of different controllers on a multi-drop link, without the master having to establish and write the datagram each time it reconnects to the controller.

The maximum number of permanent datagrams that can be established is 32. When this limit is reached, additional requests to establish datagrams are denied. One or more of the permanent datagrams will need to be canceled before others can be established. Since the permanent datagrams are not automatically deleted when the SNP session is terminated, this limit prevents an inordinate amount of these datagrams from being established.

Permanent datagrams do not survive a power cycle.

6.4.2 Communication Requests (COMMREQs) for SNP

The PACSystems serial ports COM1 and COM2 currently do not provide SNP Master service, nor do they support COMMREQ functions for SNP commands. However, those COMMREQ functions can be used with PCM/CMM modules that are configured to provide SNP service. For more information, refer to the *Series 90 PLC Serial Communications User's Manual*, GFK-0582.

Appendix A: Performance Data

This appendix contains instructions and overhead timing collected for each PACSystems CPU module. This timing information can be used to predict CPU sweep times. The information in this appendix is organized as follows:

A.1: Boolean Execution Measurements (ms per 1000 Boolean executions)[90]

CPU Model	Boolean Category		
	Simple Address	Complex Address	Passed as Parameter
CPU310	0.253	1.371	0.467
CPE010	0.244	1.329	0.469
CPE020	0.095	0.543	0.198
CRE020	0.096	0.556	0.194
CPE030	0.087	0.450	0.183
CRE030	0.090	0.451	0.184
CPE040	0.029	0.150	0.061
CRE040	0.029	0.149	0.061

For more information on Execution Times (including Boolean Operation) for Ladder Diagram instructions, please see A.3:, RX3i & RSTi-EP Instruction Times.

.

[90] *Measured with CPU firmware version 7.18.*

A.2: Instruction Timing

Overview

The tables in this section list the execution and incremental times in microseconds (µs) for each function supported by the PACSystems CPUs. Two execution times are shown for each instruction.

Execution Time	Description
Enabled	Time in µs is required to execute the function or function block when power flows into the function with valid inputs.
Disabled	Time in µs is required to execute the function when it is not enabled.

Notes:

- All times represent typical execution time. Times may vary with input and error conditions.
- Enabled time is for single-length units of word-oriented memory.
- COMMREQ time was measured between CPU and Ethernet module with NOWAIT option.
- DOIO time was measured using a discrete output module.
- Timers are updated each time they are encountered in the logic by the amount of time consumed by the last sweep.
- Performance times for the BUS_ functions were measured on the RX3i using an RMX128 Redundancy Memory Xchange Module.
- Performance times for all redundancy (CRE and CRU) CPUs were measured with ECC enabled.
- Due to a change in caching, measured times for some instructions changed for release 6.0 as compared to releases 5.0/5.1. It was found that increases in some instructions were offset by decreases in other instructions so that no effective net change was observed.
- PLC Version Information

The instruction execution and incremental times were obtained by testing the following CPU versions:

	Model	*PLC Firmware Version*
All instructions except as listed below	RSTi-EP EPSCPE100	9.15
	RSTi-EP EPSCPE115	9.45
	IC695CPL410	9.55
	IC695CPE400	9.00
	IC695CPECPE310/CPE330	7.10
	IC695CPE302	9.40 (Axxx), 10.40 (Bxxx)
	IC695CPE305	7.10 (Axxx), 10.40 (Bxxx)
	IC695CPU310/CPU315	6.0
	IC695CPU320/IC695CRU320[91]	7.18
	IC698CPE010/CPE020	6.0
	IC698CRE020	6.0 (with ECC enabled)
	IC698CPE030/CPE040	6.0
	IC698CRE030/CRE040	6.0 (with ECC enabled)
MOVE_UINT	CPE010/020	3.5
	CRE020[91]	2.04 (with ECC enabled)
SVC_REQs for Redundancy	IC695CRU320[91]	6.0 (with ECC enabled)
TON, TOF, TP Instructions	CPU310/CPU315/CPU320, CRU320	5.7
	CPE010/CPE030/CPE040	3.6
	CRE030/CRE040[91]	3.6 (with ECC enabled)
Instructions for PACMotion	CPU315/CPU320	5.6
	CPU310	6.0

[91] *Due to Error Checking and Correction (ECC), Redundant CPU times are approximately 5% slower, on average, than the equivalent Non-Redundant CPU.*

A.3: RX3i & RSTi-EP Instruction Times

The following tables are intended to guide expected instruction execution times when using Ladder Diagram (LD) and C Language. For simplicity, the instructions are grouped as follows:

- Boolean Operation: this includes coil and contact operation (LD only).
- Word Operation applies to MOVE instruction for all basic types.
- Fixed-point math covers all math operations which are not floating-point (INT, UINT, DINT types).
- Floating-point math covers all math operations which are not fixed points (REAL and LREAL types).

Execution Times for LD Instructions							
Type of Operation	CPE100/CPE115	CPE200/210 215/220 240	CPE310/ CPE305/ CPE302 (Axxx)	CPE302/CPE 305 (Bxxx)	CPU320	CPE330	CPE400/ CPL410
Boolean Operation	127 ns	96 ns	87 ns	68 ns	46 ns	46 ns	46 ns
Word Operation	1276 ns	97 ns	722 ns	440 ns	363 ns	275 ns	363 ns
Fixed Point Math	1323 ns	1162 ns	778 ns	568 ns	413 ns	405 ns	413 ns
Floating Point Math	1328 ns	11621 ns	771 ns	576 ns	420 ns	410 ns	420 ns

Execution Times for C Instructions (Raw processor power)							
Type of Operation	CPE100/CPE115	CPE200/210 215/220 240	CPE310/ CPE305/CPE302 (Axxx)	CPE302/CPE 305 (Bxxx)	CPU320	CPE330	CPE400/ CPL410
Boolean Operation	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Word Operation	N/A	N/A	2 ns	4 ns	3 ns	2 ns	3 ns
Fixed Point Math	N/A	N/A	20 ns	5 ns	3 ns	3 ns	3 ns
Floating Point Math	N/A	N/A	24 ns	12 ns	9 ns	6 ns	7 ns

A.4: Overhead Sweep Impact Times

This section contains overhead timing information for the PACSystems CPUs. This information can be used in conjunction with the estimated logic execution time to predict sweep times for the CPUs. The information in this section is made up of a base sweep time plus sweep impact times for each of the CPU models. The predicted sweep time is computed by adding the sweep impact time(s), the base sweep, and the estimated logic execution time.

See sample calculation for estimating sweep times at

The following components make up the total sweep time:

- Programmer communications sweep impact
- I/O Scan and fault sweep impact
- Ethernet Global Data sweep impact
- Intelligent Option Module (LAN modules) sweep impact
- I/O interrupt performance and sweep impact
- Timed interrupt performance and sweep impact

A.5: Base Sweep Times

Base sweep time is the time for an empty _MAIN program block to execute, with no configuration stored and none of the windows active. The following table gives the base sweep times in microseconds (µs) for each CPU model.

Family	Model	Run I/O enabled (µs)	Run outputs disabled (µs)
RX3i	CPU310[92]	1086	1076
	CPU315 CPU320[92]	180	176
	CRU320[92]	198	194
	CPE302-Axxx CPE305-Axxx CPE310	426	424
	CPE302/CPE305 (Bxxx)	270	260
	CPE330	196	192
	CPE400 CPL410	193	189
RSTi-EP	CPE100	887	-
	CPE115	862	-

[92] *Base sweep time calculated with RUN/STOP switch, single ETM.*

Family	Model	Run I/O enabled (µs)	Run outputs disabled (µs)
	CPE205 CPE210 CPE215 CPE220 CPE240	508	503

The following diagram shows the differences between the full sweep phases and the base sweep phases.

A.6: Base Sweep vs. Full Sweep Phases

Base Sweep	*Full Sweep*
<START OF SWEEP>	<START OF SWEEP>
Sweep Housekeeping	Sweep Housekeeping
↓	↓
NULL Input Scan [93]	Input Scan [93]
↓	↓
Program Logic Execution	EGD Consumption Scan [94]
↓	↓
NULL Output Scan [93]	Program Logic Execution
↓	↓
↓	Output Scan [93]
↓	↓
↓	EGD Production Scans [94]
↓	↓
↓	Poll for Missing I/O Modules [95]
↓	↓
↓	Controller Communications Window
↓	↓
↓	Backplane Communications Window
<END OF SWEEP>	<END OF SWEEP>

[93] *If I/O is suspended, the input and output scans are skipped.*
[94] *If no Ethernet Global Data (EGD) exchanges are configured, the consumption and production scans are skipped.*
[95] *Polling for missing I/O modules only occurs if a Loss of … fault has been logged for an I/O module.*

For the base sweep, if there is no configuration, the input and output scan phases of the sweep are NULL (i.e., check for configuration and then end). The presence of a configuration with no I/O modules or intelligent I/O modules (GBC) has the same effect. The logic execution time is not zero in the base sweep. The time to execute the empty _MAIN program is included so that you only need to add the estimated execution times of the functions programmed. The base sweep also assumes no missing I/O modules. The lack of programmer attachment means that the Controller Communications Window is never opened. The lack of intelligent option modules means that the Backplane Communications Window is never opened.

A.7: What the Sweep Impact Tables Contain

In some tables, functions are shown as asynchronously impacting the sweep. This means that there is not a set phase of the sweep in which the function takes place. For instance, the scanning of all I/O modules takes place during either the input or output scan phase of the CPU's sweep. However, I/O interrupts are totally asynchronous to the sweep and will interrupt any function currently in progress.

The communication functions (except for the high-priority programmer requests) are all processed within one of the two windows in the sweep (the Controller Communications Window and the Backplane Communications Window). Sweep impact times for the various service requests are all minimum sweep impact times for the defined functions, where the window times have been adjusted so that no time slicing (limiting) of the window occurs in a given sweep. This means that, as much as possible, each function is completed in one occurrence of the window (between consecutive logic scans). The sweep impact of these functions can be spread out over multiple sweeps (limited) by adjusting the window times to a value lower than the documented sweep impact time. For the programmer, the default time is 10 ms; therefore, some of the functions listed in that section will naturally time slice over successive sweeps.

A.8: Programmer Sweep Impact Times

The following table shows nominal programmer sweep impact times in microseconds (µs).

Sweep Impact Item	*RX3i*			
	CPU310 (µs)	*CPU315 CPU320 CRU320 (µs)*	*CPE330 CPE400 CPL410 (µs)*	*CPE010 (µs)*
Programmer window	2.90	0.20	1.46	1.95
Reference table monitor	4.90	0.29	1.48	1.20
Editor monitor	4.10	0.31	1.41	1.41

Definitions:

I/O Scan	*Description*
Programmer window	The time required to open the Programmer Window but not process any requests. The programmer is attached through an Ethernet connection; no reference values are being monitored.
Reference table monitor	The sweep impact to refresh the reference table screen. (The %R table was used as the example.) Mixed table display impacts are slightly larger. The sweep impact may not be continuous, depending on the sweep time of the CPU and the speed of the host of the programming software.
Editor monitor	The sweep impact to refresh the editor screen when monitoring ladder logic. The times given in the table are for a logic screen containing one contact, two coils, and eleven registers. As with the reference table sweep impact, the impact may not be continuous.

A.9: I/O Scan and I/O Fault Sweep Impact

The I/O scan sweep impact has two parts, Local I/O and Genius I/O. The equation for computing I/O scan sweep impact is:

I/O Scan Sweep Impact = Local Scan Impact + Genius I/O Scan

A.10: Sweep Impact of Local I/O Modules

The I/O scan of I/O modules is impacted as much by the location and reference address of a module as it is by the number of modules. The I/O scan has several basic parts.

I/O Scan	*Description*
Rack Setup Time	Each expansion rack is selected separately because of the addressing of expansion racks on the VME bus. This results in a fixed overhead per expansion rack, regardless of the number of modules in that rack.
Per Module Setup Time	Each Local I/O module has a fixed setup scan time.
Byte Transfer Time	The actual transfer of bytes is much faster for modules located in the main rack than for those in expansion racks. The byte transfer time differences will be accounted for by using different times for I/O modules in the main rack versus expansion racks.

In addition, analog input expander modules (the same as Genius blocks) can be grouped into a single transfer as long as consecutive reference addresses are used for modules that have consecutive slot addresses. Each sequence of consecutively addressed modules is called a scan segment. There is a time penalty for each additional scan segment.

A.11: RX3i I/O Module Types

Type	*Part Numbers*
Discrete Input, 16-point	IC694MDL240, IC694MDL241, IC694MDL645, IC694MDL646
Discrete Input - Smart Digital Input, 16-point	IC695MDL664
Discrete Input, 32-point	IC694MDL654, IC694MDL655, IC694MDL654
Discrete Output, 8-point	IC694MDL330, IC694MDL732, IC694MDL930, IC694MDL940
Discrete Output, 16-point and 12-point	IC694MDL340, IC694MDL341, IC694MDL740, IC694MDL741
Discrete Output – Smart Digital Output. 16-point	IC695MDL765
Discrete Output, 32-point	IC694MDL350, IC694MDL340, IC694MDL742, IC694MDL752, IC694MDL753, IC694MDL940
Discrete Output, 32-point	IC694MDL758
Discrete In/Out, 8-point	IC693MDR390, IC693MAR590
Analog Input, 4-channel	IC695ALG220, IC694ALG221
Analog Input, 6-channel	IC695ALG106
Analog Input, 12-channel	IC695ALG112
Analog Input, 16-channel	IC694ALG222, IC694ALG223
Analog Output, 2-channel	IC694ALG390, IC694ALG391
Analog Mixed Input/Output	IC694LG442
Analog Input with Diagnostics	IC694ALG232, IC694ALG233
Analog Mixed Input/Output with Diagnostics	IC694ALG542

A.12: RX3i I/O Module Sweep Impact Times[96]

The following table provides sweep impact times for modules in the Main rack and an expansion (Exp) rack. The base case provides the overhead for a single module in the rack. The increment (Inc) refers to the overhead for each similar module that is added to the same rack. To estimate sweep impact for modules in a remote rack, multiply the time in the main rack by 6: main rack base time × 6 = approximate sweep impact in a remote rack

	CPU310 (µs)				*CPU315/CPU320 (µs)*			
	Main Rack		*Exp*		*Main Rack*		*Exp*	
	Base	*Inc*	*Base*	*Inc*	*Base*	*Inc*	*Base*	*Inc*
Discrete Input 16-point	57.1	41.4	87.6	74.4	37.4	34.6	68.2	66.3
Discrete Input 16-point (Smart Digital Input – IC695MDL664)	24.6	21.6	NA	NA	–	–	NA	NA
Discrete Input 32-point	78.4	59.7	105.9	96.1	56.2	55.3	86.1	85.7
Discrete Output 8-point	61.0	40.3	84.3	74.9	35.6	34.7	64.5	65.5
Discrete Output 16-point	61.5	38.9	87.0	74.4	35.4	34.5	65.2	64.9
Discrete Output 16-point (Smart Digital Output – IC695MDL765)	24.8	21.4	NA	NA	–	–	NA	NA
Discrete Output 32-point	79.7	57.0	101.8	90.6	54.4	50.1	81.8	81.9
Discrete Output 32-point (IC694MDL758)	–	–	–	–	128.6	123.7	220.9	216.0
Discrete Mixed 8-point in/ 8-point out	104.5	85.7	167.0	151.7	72.2	68.9	132.3	131.2
Analog In/Out 4-channel	114.9	99.0	142.7	132.0	93.7	92.5	124.8	123.3
Analog Input 16-channel	427.7	407.1	538.8	538.0	385.3	378.8	499.9	499.3
Analog Output 2-channel	98.3	80.8	154.4	143.4	69.7	66.8	129.1	128.3
Analog Input 6-channel, IC695ALG106	92.9	73.4	N/A	N/A	51.6	51.0	N/A	N/A
Analog Input 12-channel, IC695ALG112	111.7	94.8	N/A	N/A	66.8	58.7	N/A	N/A
Universal Analog IC695ALG600	90.3	77.2	N/A	N/A	50.9	45.7	N/A	N/A
Analog Input 8-channel IC695ALG608	84.4	68.3	N/A	N/A	43.3	39.8	N/A	N/A
Analog Input 16-channel IC695ALG616	99.5	82.6	N/A	N/A	56.3	55.6	N/A	N/A
Analog Output 4-channel IC695ALG704	122.0	101.8	N/A	N/A	54.6	48.3	N/A	N/A
Analog Output 8-channel IC695ALG708	121.6	103.3	N/A	N/A	54.7	49.6	N/A	N/A

[96] *CPE302/305/310 data has been temporarily removed for validation. Data for the -Axxx and -Bxxx models will be available in the near future.*

A.13: Worksheet A: I/O Module Sweep Time

The following form can be used for computing the I/O module sweep impact. The calculation contains times for analog input expanders that are either grouped into the same scan segment as the preceding module or are grouped in a separate new scan segment. The sweep impact times of I/O Modules can be found at *RX3i I/O Module Sweep* Impact Times.

Number of expansion racks		_____	
Sweep impact per expansion rack	x	_____	= _____
Number of discrete I/O modules—main rack		_____	
Sweep impact per discrete I/O module—main rack	x	_____	= _____
Number of discrete I/O modules—expansion rack		_____	
Sweep impact per discrete I/O module—expansion rack	x	_____	= _____
Number of analog input base and output modules—main rack		_____	
Sweep impact per analog input base and output module—main rack	x	_____	= _____
Number of analog input expander modules (same segment)—main rack		_____	
Sweep impact per analog input expander module (same segment)—main rack	x	_____	= _____
Number of analog input expander modules (new segment)—main rack		_____	
Sweep impact per analog input expander module (new segment)—main rack	x	_____	= _____
Number of analog input base and output modules—expansion rack		_____	
Sweep impact per analog input base and output module—expansion rack	x	_____	= _____
The number of analog input base and output modules (same segment)—exp. rack		_____	
Sweep impact per analog input base and output module (same seg.)—exp. rack	x	_____	= _____
The number of analog input base and output modules (new segment)—exp. rack		_____	
Sweep impact per analog input base and output module (new seg.)—exp. rack	x	_____	= _____
		Predicted I/O Module Sweep Impact	_____

Note: If point faults are enabled, substitute the corresponding times for point faults enabled.

A.14: Sweep Impact of Genius I/O and GBCs

For the sweep impact of Genius I/O and Genius Bus Controllers (GBC), there is a sweep impact for each GBC, a sweep impact for each scan segment, and a transfer time (per word) sweep impact for all I/O data.

The GBC sweep impact has three parts:

1. Sweep impact to open the System Communications Window. This is added only once when the first intelligent option module (of which the GBC is one) is placed in the system.
2. Sweep impact to poll each GBC for background messages (datagrams). This part is an impact on every GBC in the system.

Note: Both the first and second parts of the GBC's sweep impact may be eliminated by closing the Backplane Communications Window (setting its time to 0). This should only be done to reduce scan time during critical phases of a process to ensure minimal scan time. Incoming messages will timeout and COMM_REQs will stop working while the window is closed.

Sweep impact to scan the GBC. This results from the CPU notifying the GBC that its new output data has been transferred, commanding the GBC to ready its input data, and informing the GBC that the CPU has finished another sweep and is still in RUN Mode. Scan segment for a Genius I/O block consists of consecutive memory locations starting from a particular reference address. A new scan segment is created for each starting input or output reference address. The time to process a single scan segment is higher for an input scan segment than it is for an output scan segment. The scan segment processing is the same for analog, discrete, and global data scan segments. Discrete data is transferred a byte at a time and takes longer to complete the transfer than analog data, which is transferred a word at a time. Global data should be counted as either discrete or analog, based on the memory references used in the source or destination.

A.15: Sweep Impact Time of Genius I/O and GBCs

Note: Functions in **bold type** impact the sweep continuously. All other functions impact the sweep only when invoked. Not all the timing information listed in the following table was available at the print time for this manual (the blank spaces).

	CPU310 (µs)	*CPE010 (µs)*	*CPE020 (µs)*	*CPE030 (µs)*	*CPE040 (µs)*
Genius Bus Controller					
open backplane communications window	30.0	24.0	4.0	4.0	1.0
per Genius Bus Controller polling for background messages	403.0	19.0	11.0	9.0	6.0
per Genius Bus Controller I/O Scan					
Genius Bus Controller in the main rack	469.0	1.0	1.0	1.0	1.0
Genius Bus Controller in the expansion rack	683.0	11.0	7.0	6.9	1.0
Genius I/O Blocks					
per I/O block scan segment	3.0	217.0	217.0	193.7	208.0
per I/O block scan segment w/point faults enabled	3.0	217.0	217.0	194.8	213.0
per byte discrete I/O data in the main rack	13.0	3.0	3.0	2.1	3.0
per byte discrete I/O data in expansion racks	16.0	8.0	5.0	4.2	4.0
per word analog I/O data in the main rack	24.0	5.0	4.0	4.0	5.0
per word analog I/O data in expansion racks	34.0	11.0	8.0	8.0	11.0

A.16: Worksheet B: Genius I/O Sweep Time

Use the following worksheet for predicting the sweep impact due to Genius I/O. The sweep impact times can be found in *Sweep Impact Time of Genius I/O and GBCs*.

Open backplane communications window	_____	= _____
GBC poll for background messages Number of GBCs	_____ x _____	= _____
GBC I/O scan for the main rack	_____	
Number of GBCs in the main rack	x _____	= _____
GBC I/O scan for the expansion rack	_____	
Number of GBCs in the expansion rack	x _____	= _____
Input block scan segments—number of I/O block scan segments—sweep impact	_____ x _____	= _____
Output block scan segments—number of I/O block scan segments—sweep impact	_____ x _____	= _____
Bytes of discrete I/O data on GBCs—main rack Sweep impact/bytes of discrete I/O data—main rack	_____ x _____	= _____
Bytes of discrete I/O data on GBCs—expansion racks Sweep impact/bytes of discrete I/O data—expansion racks	_____ x _____	= _____
Words of analog I/O data on GBCs—main rack Sweep impact/word analog I/O data—main rack	_____ x _____	= _____
Words of analog I/O data on GBCs—expansion racks Sweep impact/word analog I/O data—expansion racks	_____ x _____	= _____
Predicted Genius I/O Scan Impact		_____

A.17: Ethernet Global Data Sweep Impact

Depending on the relationship between the CPU sweep time and an Ethernet Global Data (EGD) exchange's period, the exchange's data may be transferred every sweep or periodically after some number of sweeps. Therefore, the sweep impact varies based on the number of exchanges that are scheduled to be transferred during the sweep. All of the exchanges must be taken into account when computing the worst-case sweep impact.

The Ethernet Global Data (EGD) sweep impact has two parts, Consumption Scan, and Production Scan:

EGD Sweep Impact	=	Consumption Scan	+	Production Scan

This sweep impact should be taken into account when configuring the CPU constant sweep mode and setting the CPU watchdog timeout.

Where the Consumption and Production Scans consist of two parts, exchange overhead and byte transfer time:

Scan Time	=	Exchange Overhead	+	Byte Transfer Time

A.18: Exchange Overhead

Exchange overhead includes the setup time for each exchange that will be transferred during the sweep. When computing the sweep impact, include overhead time for each exchange.

Note: The exchange overhead times in the table below were measured for a best-case scenario of 1400 bytes over 100 variables.

EGD Exchange Overhead Time			
CPU	**Activity**	**Embedded Ethernet Interface (µs)**	**Rack-based Ethernet Module (µs)**
CPU310/NIU001	Consume / READ	NA	233.6
	Produce / WRITE	NA	480.6
CPU315/CPU320	Consume / READ	NA	100.0
	Produce / WRITE	NA	195.1
CPE010	Consume / READ	184.3	238.2
	Produce / WRITE	342.0	452.0
CPE020	Consume / READ	87.7	117.8
	Produce / WRITE	187.9	257.5
CPE030	Consume / READ	85.1	114.1
	Produce / WRITE	191.8	253.5
CPE040	Consume / READ	35.08	47.12
	Produce / WRITE	75.16	103.0

A.19: Data Transfer Time

Note: This is the time required to transfer the data between the CPU module and the rack-based Ethernet module. EGD data transfer times do not increase linearly concerning data size. Please use the data values in the table below to estimate data transfer times.

Note: CPE modules do not need to use this table concerning their embedded Ethernet port, as there is no transfer of data across the backplane related to EGD traffic.

CPU	Data Size (Bytes)	Direction	Embedded Ethernet Interface (µS)	Rack-based Ethernet Module (µS)
CPU310 NIU001[97]	1	Consume / READ	NA	9.3
	100	Consume / READ	NA	51.8
	200	Consume / READ	NA	97.9
	256	Consume / READ	NA	123.8
	1	Produce / WRITE	NA	6.5
	100	Produce / WRITE	NA	14.1
	200	Produce / WRITE	NA	17.7
	256	Produce / WRITE	NA	19.3
CPU315 CPU320	1	Consume / READ	NA	6.2
	100	Consume / READ	NA	49.5
	200	Consume / READ	NA	96.4
	256	Consume / READ	NA	122.8
	1	Produce / WRITE	NA	3.4
	100	Produce / WRITE	NA	9.9
	200	Produce / WRITE	NA	14.9
	256	Produce / WRITE	NA	16.5
CPE010	1	Consume / READ	4.1	8.8
	100	Consume / READ	25.7	23.5
	200	Consume / READ	49.0	38.6
	256	Consume / READ	61.4	46.8
	1	Produce / WRITE	1.9	8.8
	100	Produce / WRITE	4.0	16.5
	200	Produce / WRITE	6.0	22.2
	256	Produce / WRITE	7.1	25.1
CPE020	1	Consume / READ	2.7	5.5
	100	Consume / READ	23.6	19.5
	200	Consume / READ	46.3	34.9
	256	Consume / READ	58.9	42.7
	1	Produce / WRITE	0.8	5.5
	100	Produce / WRITE	2.7	13.9
	200	Produce / WRITE	4.7	19.2
	256	Produce / WRITE	5.9	22.1

[97] *EGD performance is different on the IC695NIU001+ (versions-AAAA and later) compared to the IC695NIU001. In general, consumed data exchanges with a size greater than 31 bytes will result in contributing less of a sweep time impact and data exchanges with a size less than that will contribute slightly greater sweep impact. All produced exchanges on the IC695NIU001+ will appear to have a slightly greater sweep impact when compared to the IC695NIU001.*

CPU	Data Size (Bytes)	Direction	Embedded Ethernet Interface (µS)	Rack-based Ethernet Module (µS)
CPE030	1	Consume / READ	2.8	5.3
	100	Consume / READ	25.8	18.7
	200	Consume / READ	50.7	33.4
	256	Consume / READ	60.1	40.4
	1	Produce / WRITE	0.8	5.5
	100	Produce / WRITE	2.5	13.1
	200	Produce / WRITE	4.2	18.2
	256	Produce / WRITE	5.2	21.5
CPE040	1	Consume / READ	1.9	3.85
	100	Consume / READ	21.1	10.1
	200	Consume / READ	43.5	31.4
	256	Consume / READ	56.5	39.2
	1	Produce / WRITE	0.3	3.8
	100	Produce / WRITE	1.8	11.8
	200	Produce / WRITE	3.6	16.8
	256	Produce / WRITE	4.8	19.8

A.20: Worksheet C: Ethernet Global Data Sweep Time

Number of consumed exchanges		_____		
Sweep impact per exchange	x	_____	=	_____
Number of data bytes in all of the consumed exchanges		_____		
Sweep impact per consumed data byte	x	_____	=	_____
Number of produced exchanges		_____		
Sweep impact per exchange	x	_____	=	_____
Number of data bytes in all of the produced exchanges		_____		
Sweep impact per produced data byte	x	_____	=	_____
		Predicted EGD Sweep Impact		_____

A.21: EGD Sweep Impact for RX3i CPE330 and CPE400/CPL410

The CPE330 and CPE400 process Ethernet communications independently from logic and sweep execution. This architecture precludes the possibility of Ethernet communications causing the watchdog timer to time out. Consequently, the discussion below does not apply to CPE330 or CPE400.

A.22: EGD Sweep Impact for RX3i CPE302/CPE305/CPE310 and RSTi-EP CPE100/CPE115 Embedded Ethernet Interface

Each EGD production or consumption will take about 200 µs regardless of the size of the exchange. For Produced Exchanges on the embedded port, you can think of it as a timed interrupt block that takes a 200µs duration to execute each time it is triggered. For Consumed Exchanges on the embedded port, you can think of it as an I/O interrupt block that takes a 200 µs duration to execute each time the remote unit sends an exchange and it is received on the embedded port.

It is important to note that this 200µs per exchange is not a simple 'sweep impact' time, but rather per execution of that exchanging time, and depending on sweep time length and production period it may occur more than one time per sweep.

Users configuring systems with EGD on an embedded Ethernet port should take care to make sure that production and consumption time on the embedded Ethernet port is accounted for.

The impact of EGD Exchanges configured on the Embedded Ethernet Interface of RX3i CPE302/CPE305/CPE310 and RSTi-EP CPE100/CPE115 on the Controller sweep can be reflected in two parameters:

1. ***Total_EgdImpactPerWDT_ms*** **:** This is the total EGD impact per Watchdog period configured in milliseconds (ms).
2. ***EgdProcessorUtilization*** %: This is the percentage of EGD processor utilization.

The formula for calculating these two parameters is shown below:

$$Total\_EgdImpactPerWDT\_ms = \sum_{n=1}^{255} \left(\frac{Wdt\_ms}{Period\_ms_n} \times ExchangePresent_n \times 0.2\,00 \right)$$

$$EgdProcessorUtilization\_\% = \left(\frac{Total\_EgdImpactPerWDT\_ms}{Wdt\_ms} \right) \times 100$$

Wdt_ms – Watchdog time configured in ms

$Period\_ms_n$ – Exchange Period nth Exchange, as configured in ms (Production Period for production exchanges and Consumption timeout for consumption exchanges

$ExchangePresent_n$ – nth Exchange configured (value=1) or not configured (value=0)

It is recommended that the calculated ***EgdProcessorUtilization***_% for a given EGD exchange configuration and Watchdog period should be less than 50-55% for stable operation within the watchdog period without WDT elapse.

Note: The higher percentage of this parameter indicates that the EGD on the Embedded Ethernet interface could have a greater impact on CPU applications.

A.23: Example Calculation for EGD Utilization on RX3i CPE302/CPE305/CPE310 and RSTi-EP CPE100/CPE115

The watchdog time is configured to 200ms for calculations in the table below:

SN	EGD Exchange	Type of Exchange	Period	Total_EgdImpactPerWDT_ms[n]
1	EGD Exchange#1	Producer	25	1.600
2	EGD Exchange#2	Producer	25	1.600
3	EGD Exchange#3	Producer	30	1.333
4	EGD Exchange#4	Producer	30	1.333
5	EGD Exchange#5	Consumer	30	1.333
6	EGD Exchange#6	Consumer	50	0.800
7	EGD Exchange#7	Consumer	50	0.800
8	EGD Exchange#8	Consumer	50	0.800
Total_EgdImpactPerWDT_ms				**9.600**
EgdProcessorUtilization_%				**4.800**

A.24: Normal Sweep – EGD on RX3i CPE302/CPE305/CPE310 and RSTi-EP CPE100/CPE115 Embedded Ethernet Interface

The following table shows the chart for setting up EGD exchanges on Embedded Ethernet for RX3i CPE302/CPE305/CPE310 with a no sweep load and no network traffic. The table is a compilation of results based on testing with two RX3i CPE310 Systems in which one is acting as the EGD Producer and the other is acting as the EGD Consumer.

SN	Production Period [Consumption Timeout*] (ms)	Data size per Exchange (Bytes)	Maximum Number of EGD Exchanges (Recommended)
A	500	1400	254
B	500	200	255
C	500	10	255
D	300	1400	166
E	300	200	255
F	300	10	255
G	200	1400	109
H	200	200	255
I	200	10	255
J	100	1400	54
K	100	200	255
L	100	10	255
M	50	1400	27
N	50	200	127
O	50	10	230
P	30	1400	16
Q	30	200	75
R	30	10	136
S	20	1400	11
T	20	200	50
U	20	10	91

The following table shows the chart for setting up EGD exchanges on Embedded Ethernet for RSTi-EP CPE100/CPE115 with a no sweep load and no network traffic. The table is a compilation of results based on testing with two RSTi-EP CPE100/CPE115 Systems in which one is acting as the EGD Producer and the other is acting as the EGD Consumer.

SN	Production Period [= 50% Consumption Timeout] (ms)	Data size per Exchange (Bytes)	Maximum Number of EGD Exchanges (Recommended)
A	500	1400	8
B	300	1400	8
C	200	1400	8
D	100	1400	8
E	50	1400	8
F	30	1400	8
G	20	1400	8

Note: The *Consumption Timeout* is set at twice the *Production Period* on the other consuming RX3i CPE310 / RSTi-EP CPE100/CPE115 node. For example, for A, the *Production Period* for all the Producer exchanges is set to *500ms*. This indicates that the *Consumption Timeout* for all the consumer exchanges on the consuming node is set to *1000ms (*twice the production period).

The following are important points to be considered when configuring EGD exchanges on Embedded Ethernet Interface.

1. The recommended values in the given table should be used in conjunction with the recommended limit value for $\boldsymbol{EgdProcessorUtilization}$_% as per the watchdog time and sweep load of the application.
2. EGD Consumption and Production below 20ms are not recommended for Embedded Ethernet Interface with RX3i CPE302/CPE305/CPE310, RSTi-EP CPE100/CPE115.
3. It is advisable to limit the number of EGD exchanges or EGD load on the Embedded Ethernet Interface of the RX3i CPE302/CPE305/CPE310, RSTi-EP CPE100/CPE115 and use higher periods while defining the system and configuration, and take into account the sweep load for minimizing EGD sweep impact.

A.25: Constant Sweep - EGD on RX3i CPE302/CPE305/CPE310 and RSTi-EP CPE100/CPE115 Embedded Ethernet Interface

The EGD on Embedded Ethernet Interface can be treated as interrupt blocks. Therefore, EGD exchanges with Constant sweep may cause sweep overruns and should be avoided. It is also recommended that the Production period for Producer exchanges be set to multiples (3 and above) of the Constant sweep time set to avoid stale data being produced by the RX3i CPE302/CPE305/CPE310 and RSTi-EP CPE100/CPE115 system on Embedded Ethernet Interface. Some of the factors affecting the Constant sweep overruns with EGD on Embedded Ethernet Interface are Constant sweep time, No of Exchanges, Production period, and Exchange data size.

The Constant sweep with EGD exchanges configured on Embedded Ethernet Interface and timed or I/O interrupts will also cause constant sweep overruns and are not recommended.

A.26: Sweep Impact of Intelligent Option Modules

The tables in this section list the sweep impact times in microseconds (µs) for intelligent option modules. The fixed sweep impact is the sum of the polling sweep impact and the I/O scan impact. The opening of the Backplane Communications Window and the polling of each module have relatively small impacts compared to the sweeping impact of CPU memory read or write requests.

Intelligent option modules include GBCs being used for Genius LAN capabilities. The sweep impact for these intelligent option modules is highly variable.

A.27: Fixed Sweep Impact Times of RX3i Intelligent Option Modules

Sweep Impact Item	CPU310 (µs)				CPU315/CPU320 (µs)				NIU001+ (µs)			
	Main		Exp		Main		Exp		Main		Exp	
	Base	Inc	Base	Inc	Base	Inc	Base	Inc	Base	Inc	Base	Inc
IC694APU300B and earlier	1085	—	—	—	1109	—	—	—	—	—	—	—
IC694APU300-CA and later												
Classic	2759[98]	—	—	—	2043[99]	—	—	—	—	—	—	—
Enhanced	4074[98]	—	—	—	3276[99]	—	—	—	—	—	—	—
IC694BEM331[100]	See footnote				—	—	—	—	—	—	—	—
IC694DSM31	See footnote				—	—	—	—	See footnote			
IC695ETM001	199	—	NA	NA	188	51	NA	NA	—	—	NA	NA
IC695HSC304	208.7	173.9	NA	NA	136.4	131.0	NA	NA	—	—	NA	NA
IC695HSC308	282.4	256.5	NA	NA	202.6	200.3	NA	NA	—	—	NA	NA
IC695PBM300		—	NA	NA		—	NA	NA	—	—	NA	NA
No I/O	132				60							
100 bytes Input, 100 bytes Output	196				105							
100 bytes Input, 200 bytes Output	206				140							
200 bytes Input, 100 bytes Output	248				106							
IC695PNC001	NA	NA	NA	NA	See footnote	NA	NA	NA	NA	NA	NA	NA

[98] *CPU firmware version 7.13*
[99] *CPU firmware version 7.14*
[100] *See Sweep Impact Time of Genius I/O and GBCs*

A.28: PROFINET Controller (PNC001) and PROFINET I/O Sweep Impact[101]

The PLC CPU sweep impact for a PROFINET IO network is a function of the number of PNCs, the number of PROFINET devices, and the number of each PROFINET device's IO modules. The table below shows the measured sweep impact of the RX3i PROFINET Controller, supported VersaMax PROFINET devices, and I/O modules.

Sweep Impact Item	CPU315/CPU320 (µs)	CPE330 w/PNC001 (µs)	CPE330 w/Embedded PNC (µs)	CPE400/CPL410 w/Embedded PNC (µs)
RX3i PROFINET Controller (PNC)	50	43	16	12
RX3i Devices				
PROFINET Scanner (PNS) IC695PNS001	46	42	10	24
ALG442 Mixed Analog	54	38	25	20
ALG220 Analog Input	27	29	14	10
ALG390 Analog Output	24	12	13	10
MDL645 Discrete Input	23	18	15	10
MDL740 Discrete Output	22	9	14	9
VersaMax Devices				
PROFINET Scanner (PNS), IC200PNS001	40	38	15	24
Discrete Input Module (8/16/32 pt.)	23	23	10	10
Discrete Output Module (8/16/32 pt.)	18	24	13	12

[101] *CPE302/305/310 data has been temporarily removed for validation. Data for the -Axxx and -Bxxx models will be available in the near future.*

Sweep Impact Item	CPU315/CPU320 (µs)	CPE330 w/PNC001 (µs)	CPE330 w/Embedded PNC (µs)	CPE400/CPL410 w/Embedded PNC (µs)
Analog Input Module (15 channel)	59	67	10	11
Analog Output Module (12 channel)	21	27	12	14
CMM020 (64AI/64AQ)	204	188	18	18

To calculate the total expected PLC sweep impact for a PROFINET I/O network, add the individual sweep impact times for each PROFINET Controller, PROFINET Device, and PROFINET Device I/O module, using the times provided above.

For example, for a PROFINET I/O network that consists of one PNC and one VersaMax PROFINET Scanner, which has both an 8-point input and an 8-point output module:

```
Expected PLC Sweep Impact = 50 (PNC) + 40 (PNS) + 23 (8pt. Input) + 18 (8pt. Output)

=131 µs.
```

A.29: DSM314 Sweep Impact

No. of Axes Configured	Rx3i CPU310 Rack (µs)		Rx3i NIU001+ Rack (µs)	
	Main	Exp	Main	Exp
1	1535	2160	1830	2360
2	2018	2906	2304	3160
3	2500	6371	2840	3920
4	2990	4430	3350	4680

A.30: I/O Interrupt Performance and Sweep Impact

There are several important performance numbers for I/O interrupt blocks. The sweep impact of an I/O interrupt invoking an empty block measures the overall time of fielding the interrupt, starting up the block, exiting the block, and restarting the interrupted task. The time to execute the logic contained in the interrupt block affects the limit by causing the CPU to spend more time servicing I/O interrupts and thus reduce the maximum I/O interrupt rate.

The minimum, typical, and maximum interrupt response times reflect the time from when a single I/O module sees the input pulse until the first line of ladder logic is executed in the I/O interrupt block. Minimum response time reflects a 300 µs input card filter time + time from interrupt occurrence to the first line of ladder logic in the I/O interrupt block. The minimum response time can only be achieved when no intelligent option modules are present in the system and the programmer is not attached. Typical response time is the minimum response time plus a maximum interrupt latency of 2.0 ms. This interrupt latency time is valid, except when one of the following operations occurs:

- The programmer is attached.
- A store of logic, *RUN Mode Store*, or word-for-word change occurs.
- A fault condition (logging of a fault) occurs.
- Another I/O interrupt occurs.
- The CPU is transferring a large amount of input (or output) data from an I/O controller (such as a GBC). Heavily loaded I/O controllers should be placed in the main rack whenever possible.
- An event that has higher priority and requires a response occurs. An example of this type of event is clearing the I/O fault table.

Any one of these events extends the interrupt latency (the time from when the interrupt card signals the interrupt to the CPU to when the CPU services the interrupt) beyond the typical value. However, the latency of an interrupt occurring during the processing of a preceding I/O interrupt is unbounded. I/O interrupts are processed sequentially so that the interrupt latency of a single I/O interrupt is affected by the duration of the execution time of all preceding interrupt blocks. (The worst case is that every I/O interrupt in the system occurs at the same time so that one of them has to wait for all others to complete before it starts.)

The maximum response times shown below do not include the two unbounded events.

A.31: I/O Interrupt Block Performance and Sweep Impact Times

Sweep Impact Item	*CPE302 CPE305 CPE310 (µs)*	*CPU310 (µs)*	*CPU315/ CPU320 (µs)*	*CPE010 (µs)*	*CPE020 (µs)*	*CPE030 (µs)*	*CPE040 (µs)*
I/O interrupt sweep impact	-[102]	127.8	-	309.7	335	125.6	24.0
Minimum response time	—	151.7	326.1	392.4	334	330.6	315.2
Typical response time		175.0	327.3	396.1	336	331.5	315.5
Maximum response time		302.7	346.2	434.9	359	375.1	325.7

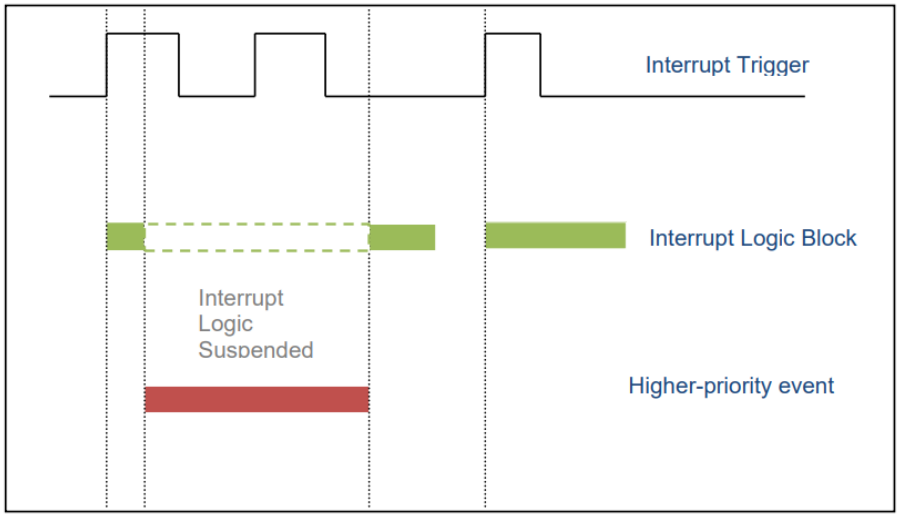
Note that the min, typical, and max response times include a 300 µs Input card filter time.

A.32: Dropped Interrupts

When multiple interrupts are triggered during the interrupt latency period, interrupt blocks may only be executed one time even though the interrupt trigger has occurred more than once. The likelihood of this occurring will increase if the system interrupt latency has increased due to the specific configuration and use of the system.

This will not cause the CPU to miss a given interrupt; just consolidate the number of times an interrupt block is executed even though the interrupt stimulus had occurred more than one time.

Figure 61: Interrupt Execution Considerations



[102] *Performance data not available for this release.*

A.33: Worksheet D: Programmer, IOM, I/O Interrupt Sweep Time

The following worksheet can be used to calculate the sweep impact times of programmer sweep impact, intelligent option modules, and I/O Interrupts. For time data, refer to the following tables:

Programmer Sweep Impact TimesSweep Impact Time of Genius I/O and GBCs

Programmer sweep impact		= _____
IOM—first module (open comm. window)	_____	
IOM—per module (polling)	+ _____	
LAN module I/O scan	+ _____	
Total IOM Sweep Impact		= _____
CPU memory access from IOMs		= _____
I/O interrupt sweep impact	_____	
I/O interrupts response time	+ _____	= _____
Predicted Sweep Time (Other)		_____

A.34: Timed Interrupt Performance

The sweep impact of a timed interrupt invoking an empty program block or timed program measures the overall time of fielding the interrupt, starting up the program or block, exiting the program or block, and restarting the interrupted task. The minimum, average, and maximum interrupt periods reflect the time period from when the first line of ladder logic is executed in the timed interrupt block.

Timed Interrupt Performance and Sweep Impact Times for a 0.001s Timed Interrupt Block

Sweep Impact Item	CPU310 (µs)	CPU315 CPU320 (µs)	CPE010 (µs)	CPE020 (µs)	CPE030 (µs)	CPE040 (µs)
Timed interrupt sweep impact	87.3	26.2	88.6	28.0	31.2	23.3
Minimum interrupt period	908.3	969.8	951.4	946.0	922.8	973.0
Average interrupt period	1000.0	1000.0	1005.5	999.7	1000.0	999.9
Maximum interrupt period	1081.2	1030.8	1056.6	1054.0	1077.0	1026.9

Appendix B: User Memory Allocation

User Memory Size is the number of bytes of memory available to the user for PLC applications.

Model	User Memory Size (MB)
IC695CPE302	2MB
IC695CPE305	5MB (Axxx), 6MB (Bxxx)
IC695CPU310, IC695CPE310, IC698CPE010, IC698CPE020, IC698CRE020	10MB
IC695CPU315	20MB
IC695CPL410, IC695CPE400, IC695CPE330, IC695CPU320, IC695CRU320	64MB
IC698CPE030, IC698CRE030 IC698CPE040, IC698CRE040	64MB
EPSCPE100	1MB
EPSCPE115	1.5MB

For a list of items that count against user memory, see below.

B.1: Items that Count Against User Memory

The following items count against the CPU memory and can be used to estimate the minimum amount of memory required for an application. Additional space may be required for items such as Advanced User Parameters, zipped source files, user heap, and published symbols.

Register Memory Size (%R)	Bytes = %R references configured × 2
Word Memory Size (%W)	Bytes = %W references configured × 2
Analog Inputs (%AI)	If point faults are enabled: Bytes = %AI references configured × 3 If point faults are disabled: Bytes = %AI references configured × 2
Analog Outputs (%AQ)	If point faults are enabled: Bytes = %AQ references configured × 3 If point faults are disabled: Bytes = %AQ references configured × 2
Discrete Point Faults	If point faults are enabled: Bytes = 3072
Managed Memory (Symbolic Variable and I/O Variable Storage)	The total number of bytes required for symbolic and I/O variables. Calculated as follows: [(number of symbolic discrete bits) × 3 / (8 bits/byte)] + [(number of I/O discrete bits) × Md / (8 bits/byte)] + [(number of symbolic words) × (2 bytes/word)] + [(number of I/O words) × (Mw bytes/word)] Md = 3 or 4. The number of bits is multiplied by 3 to keep track of the force, transition, and value of each bit. If point faults are enabled, the number of I/O discrete bits is multiplied by 4. Mw = 2 or 3. There are two 8-bit bytes per 16-bit word. If point faults are enabled, the number of bytes is multiplied by 3 because each I/O word requires an extra byte.
EGD (included in HWC)	Bytes = 0 if no Ethernet Global Data pages are configured
I/O Scan Set File (included in HWC)	Based on the number of scan sets used. ***Note:*** 32 bytes of user memory are consumed if the application scans all I/O every sweep (the default).
User Programs	Refer to B.2: User Program Memory Usage for details on user programs.

B.2: User Program Memory Usage

Space required for user logic includes the following items:

B.3: %L and %P Program Memory

%L and %P are charged against your userspace and sized depending on their use in your applications. The maximum size of %L or %P is 8192 words per block.

The %L and %P tables are sized to allow extra space for *RUN Mode Store*s per the following rules.

- If %L memory is not used in the block, the %L memory size is 0 bytes. If %L memory is used in the block, a buffer is added beyond the highest %L address actually used in logic or the variable table. The default buffer size is 256 bytes but can be changed by editing the Extra Local Words parameter in the block Properties.
- The same rules apply for the size of %P memory, but %P memory can be used in any block in the program.
- The buffer cannot make the %P or %L table exceed the maximum size of 8,192 words. In such a case, a smaller buffer is used.
- You can add, change, or delete %L and/or %P variables in your application and *RUN Mode Store* the application of these variables fit in the size of the last-stored %L/%P tables (where the *size* includes the previous buffer space), or if going from a zero to non-zero size.
- The size of the %L/%P tables is always recalculated for *STOP Mode Store*s.

B.4: Program Logic and Overhead

The data area for C (.gefelf) blocks are considered part of the user program and count against the user program size. Additional space is required for information internal to the CPU that is used for the execution of the C block.

The program block is based on overhead for the block itself plus the logic and register data being used (that is, %L).

Note: The program stack of the LD is not counted against the CPU's memory size.

Note: If your application needs more space for LD logic, consider changing some %P or %L references to %R, %W, %AI, or %AQ. Such changes require a recompilation of the program block and a *STOP Mode Store* to the CPU.

General Contact Information

Home link: http://www.emerson.com/industrial-automation-controls

Knowledge Base: https://www.emerson.com/industrial-automation-controls/support

Technical Support

Americas
Phone: 1-888-565-4155
1-434-214-8532 (If toll-free option is unavailable)

Customer Care (Quotes/Orders/Returns): customercare.mas@emerson.com
Technical Support: support.mas@emerson.com

Europe
Phone: +800-4444-8001
+420-225-379-328 (If toll-free option is unavailable)
+39-0362-228-5555 (from Italy - if toll-free 800 option is unavailable or dialing from a mobile telephone)\

Customer Care (Quotes/Orders/Returns): customercare.emea.mas@emerson.com
Technical Support: support.mas.emea@emerson.com

Asia
Phone: +86-400-842-8599
+65-6955-9413 (All other Countries)

Customer Care (Quotes/Orders/Returns): customercare.cn.mas@emerson.com
Technical Support: support.mas.apac@emerson.com

Any escalation request should be sent to: mas.sfdcescalation@emerson.com

Note: If the product is purchased through an Authorized Channel Partner, please contact the seller directly for any support.

Emerson reserves the right to modify or improve the designs or specifications of the products mentioned in this manual at any time without notice. Emerson does not assume responsibility for the selection, use, or maintenance of any product. Responsibility for proper selection, use, and maintenance of any Emerson product remains solely with the purchaser.

EMERSON