AMC131

MPC8641D Processor
AdvancedMC™ Module
Hardware Manual

205 Indigo Creek Drive | Rochester, NY 14626 | Phone +1.585.256.0200 | support@pt.com

www.pt.com

Document Revision History

Part Number	Date	Explanation of changes
126p0609.10	April 16, 2008	Initial Production Release
126p0609.11	June 3, 2008	Updated Battery Documentation. Updated LED definitions.
126p0609.12	October 20, 2008	Updated Table 5-1, “PMI/PICMG Command Subset Supported by the MMC Firmware.” Updated Web references.
126p0609.13	December 16, 2008	Expanded SW2 (now SW1) description.
126p0609.14	March 18, 2009	Added warning and caution about product handling in Chapter 9, “Agency Approvals” on page 103.
126p0609.15	June 10, 2009	Standardized “Module Management Controller (MMC)” on page 28 and Chapter 5, “System Monitoring and Alarms” on page 53. Updated “SDRAM” on page 25, “Memory Configuration” on page 35, “200-pin SDRAM Sockets (P6, P7)” on page 48, “Battery Socket (BT1)” on page 48, and “Battery Backup Characteristics” on page 101 to indicate that memory and batteries are not field servicable items. Changed name of SW2 (now SW1) to Multifunction Switch. Clarified “SW2-3: Spread Spectrum Generation” on page 40. Corrected board dimensions on Page 102.
126p0609.16	October 2, 2009	Standardized Chapter 9, “Agency Approvals” on page 103.
126p0609.17	May 11, 2010	Updated Table 2-4, “SERDES Port Mapping,” on page 27. Added “Working with the MiniSD Card” on page 35. Updated throughout for SRIO option. Updated “SW2-2: FCLKA Configuration - PCIe only” on page 39 to document the new default settings. Removed “Manual PCI Express Configuration” topic. Updated “Power Reset” on page 79.
126p0609.18	September 13, 2010	Updated throughout for 1.5 GHz version.
126p0609.19	January 13, 2011	Corrected “DC Operating Characteristics” on page 100 for 1.5 GHz version.

Errors and Omissions

Although diligent efforts are made to supply accurate technical information to the user, occasionally errors and omissions occur in manuals of this type. Refer to the Performance Technologies, Inc. web site to obtain manual revisions or current customer information:

http://www.pt.com.

Performance Technologies, Inc., reserves its right to change product specifications without notice.

Symbol Conventions in This Manual

The following symbols appear in this document:

Caution:
There is risk of equipment damage. Follow the instructions.

Warning:
Hazardous voltages are present. To reduce the risk of electrical shock and danger to personal health, follow the instructions.

Caution:
Electronic components on printed circuit boards are extremely sensitive to static electricity. Ordinary amounts of static electricity generated by your clothing or work environment can damage the electronic equipment. It is recommended that anti-static ground straps and anti-static mats are used when installing the board in a system to help prevent damage due to electrostatic discharge.

Additional safety information is available throughout this guide and in Chapter 9, “Agency Approvals” on page 103.

Contents

Tables

Figures

Chapter 1

About This Guide

This manual describes the operation and use of the AMC131 MPC8641D Processor AMC Module (referred to as AMC131 in this guide). The following outline describes the focus of each chapter.

Chapter 2, “Introduction,” on page 21 provides an overview of the AMC131 and includes information such as module features, functional block diagram, and a brief description of each block.

Chapter 3, “Getting Started,” on page 33 provides setup information such as unpacking the module, system requirements, and installation procedures.

Chapter 4, “Connectors,” on page 43 provides connector location, description, and pinout information.

Chapter 5, “System Monitoring and Alarms,” on page 53 describes the commands supported by the on-board Module Management Controller.

Chapter 6, “Reset Configuration,” on page 79 describes the AMC131 reset types with their respective sources.

Chapter 7, “Programmable Registers,” on page 83 describes the programmable registers used to configure and monitor AMC131 functionality.

Chapter 8, “Specifications,” on page 99 contains mechanical, electrical, and environmental specifications.

Chapter 9, “Agency Approvals,” on page 103 presents agency approvals and certification information.

Chapter 10, “Data Sheet Reference,” on page 107 provides information on data sheets, standards and specifications for the technology designed into the AMC131.

The AMC131 assembly should be used in conjunction with the PT software package that you have chosen, for example NexusWare® Core.

The most current documentation to support the additional components that you purchased from PT is available at www.pt.com under the product you are inquiring about.

For additional documentation references see Chapter 10, “Data Sheet Reference,” on page 107.

Text Conventions

Conventions in This Guide describes the text conventions that are used in this guide.

Conventions in This Guide

Convention	Used For
`Monospace font`	Monospace font represents sample code.
Bold font	Bold font represents: • paths • file names • UNIX commands • user input.
Italic font	Italic font represents: • notes that supply useful advice • supplemental information • referenced documents.

Customer Support and Services

PT offers a variety of standard and custom support packages to ensure customers have access to the critical resources that they need to protect and maximize hardware and software investments throughout the development, integration, and deployment phases of the product life cycle.

If you encounter difficulty in using this PT product, you may contact our support personnel by:

1. **EMAIL** (Preferred Method) – Email us at the addresses listed below or use our online email support form. Outline your problem in detail. Please include your return email address and a telephone number.
2. **TELEPHONE** – Contact us via telephone at the number listed below, and request Technical Support. Our offices are open Monday to Friday, 8:00 a.m. to 8:00 p.m. (Eastern Time).

PT Support Contact Information

	Embedded Systems and Software (Includes Platforms, Blades, and Servers)	SS7 Systems (Includes SEGway™)
Email	support@pt.com	ss7support@pt.com
Phone	+1 (585) 256-0248 (Monday to Friday, 8 a.m. to 8 p.m. Eastern Time)	+1 (585) 256-0248 (Monday to Friday, 8 a.m. to 8 p.m. Eastern Time)

If you are located outside North America, we encourage you to contact the local PT distributor or agent for support. Many of our distributors or agents maintain technical support staffs.

Customer Support Packages

Our configurable development and integration support packages help customers maximize engineering results and achieve time-to-market goals. To find out more about our Customer Support packages, visit http://www.pt.com/page/support/.

Other Web Support

Support for existing products including manuals, release notes, and drivers can be found on specific product pages at http://www.pt.com. Use the product search to locate the information you need.

Return Merchandise Authorization (RMA)

To submit a return merchandise authorization (RMA) request, complete the online RMA form available at http://pt.com/assets/lib/files/rma-request-form.doc and follow the instructions on the form. You will be notified with an RMA number once your return request is approved. Shipping information for returning the unit to PT will be provided once the RMA is issued.

Product Warranty

Performance Technologies, Incorporated, warrants that its products sold hereunder will at the time of shipment be free from defects in material and workmanship and will conform to Performance Technologies' applicable specifications or, if appropriate, to Buyer's specifications accepted by Performance Technologies in writing. If products sold hereunder are not as warranted, Performance Technologies shall, at its option, refund the purchase price, repair, or replace the product provided proof of purchase and written notice of nonconformance are received by Performance Technologies within 12 months of shipment, or in the case of software and integrated circuits within ninety (90) days of shipment and provided said nonconforming products are returned F.O.B. to Performance Technologies's facility no later than thirty days after the warranty period expires. Products returned under warranty claims must be accompanied by an approved Return Material Authorization number issued by Performance Technologies and a statement of the reason for the return. Please contact Performance Technologies, or its agent, with the product serial number to obtain an RMA number. If Performance Technologies determines that the products are not defective, Buyer shall pay Performance Technologies all costs of handling and transportation. This warranty shall not apply to any products Performance Technologies determines to have been subject to testing for other than specified electrical characteristics or to operating and/or environmental conditions in excess of the maximum values established in applicable specifications, or have been subject to mishandling, misuse, static discharge, neglect, improper testing, repair, alteration, parts removal, damage, assembly or processing that alters the physical or electrical properties. This warranty excludes all cost of shipping, customs clearance and related charges outside the United States. Products containing batteries are warranted as above excluding batteries.

THIS WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES WHETHER EXPRESS, IMPLIED OR STATUTORY INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS. IN NO EVENT SHALL PERFORMANCE TECHNOLOGIES BE LIABLE FOR ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES DUE TO BREACH OF THIS WARRANTY OR ANY OTHER OBLIGATION UNDER THIS ORDER OR CONTRACT.

Chapter 2

Introduction

This chapter provides a brief introduction to the AMC131 MPC8641D processor AdvancedMC® module. It includes a product definition, a list of product features, figures such as “AMC131 Front Panel” and "AMC131 Functional Blocks," and a description of each functional block. Information about unpacking, initial board configuration, and setup is provided in Chapter 3, “Getting Started,” on page 33.

Key topics in this chapter:

Product Definition

The AMC131 is a 32-bit AdvancedMC single board compute module with a Freescale™ MPC8641D dual core PowerPC® processor (referred to as MPC8641D in this guide).

The AMC131 is designed for use as a powerful compute element within MicroTCA® systems, such as the PT AMP5071, and mounted on a carrier board for use in an ATCA system, allowing system designers to build NEBS-compliant, highly reliable solutions that combine state-of-the-art management, the highest performance power and cooling, and integrated switching.

Ethernet connectivity is available through dual front panel RJ-45 10/100/1000Base-T Ethernet ports.

The AMC131 supports the PICMG sub-specifications AMC.1 (PCI Express), AMC.2 (Gigabit Ethernet), and AMC.4 (Serial Rapid IO), to ensure a comprehensive set of interconnecting capabilities to the carrier board.

The AMC131 onboard memory includes:

- SDRAM
- Global Storage
- Flash Memory
- Serial EEPROM

The AMC131 is compliant with the PICMG AMC.0 specification and adheres to mechanical, power, thermal, interconnect, and management functions defined in this specification.

Designed to run Linux applications, the AMC131 is the ideal processor for high-end packet processing or multi-threaded software applications found in IP Multimedia Subsystem (IMS), wireless, softswitch, defense, or any other compute-intensive application.

AMC131 Features

The AMC131 includes the following features:

- Freescale MPC8641D dual-core processor
- Dual front panel RJ-45 Ethernet (triple speed twisted pair)
- Dual DDR2 (64-bits + ECC controllers) support up to 2GB SDRAM
- 128MB Flash memory
- Mini Secure Digital (miniSD) (256 MB minimum)
- Real-time clock with battery backup (five year minimum)
- Processor-independent Watch Dog Timer
- RJ-11 console port on front panel with 15 kV Electrostatic Discharge (ESD) protection
- AMC transport build options:
  - PCI Express, AMC.1, one port of type 1, 2, 4, or 8 lanes at 2.5 Gb per lane only
  - PCI Express, AMC.1, one port of type 1, 2, or 4 lanes at 2.5 Gb per lane only and Serial Rapid IO, AMC.4, one port of 4 lanes at 3.125 Gbaud (2.5 Gb/s) per lane only
  - Serial Rapid IO, AMC.4, one port of 4 lanes with each lane at 3.125 Gbaud (2.5 Gb/s) only
- Ethernet AMC transport build option: AMC.2, Type E2
- Single width PICMG AMC.0 module. Standard height is mid-size. Full-size height option is available.
- NexusWare operating systems
- Front panel USB port

I/O Configurations

The AMC131 supports the I/O options provided in Table 2-1. Option 1 is the standard product offering.

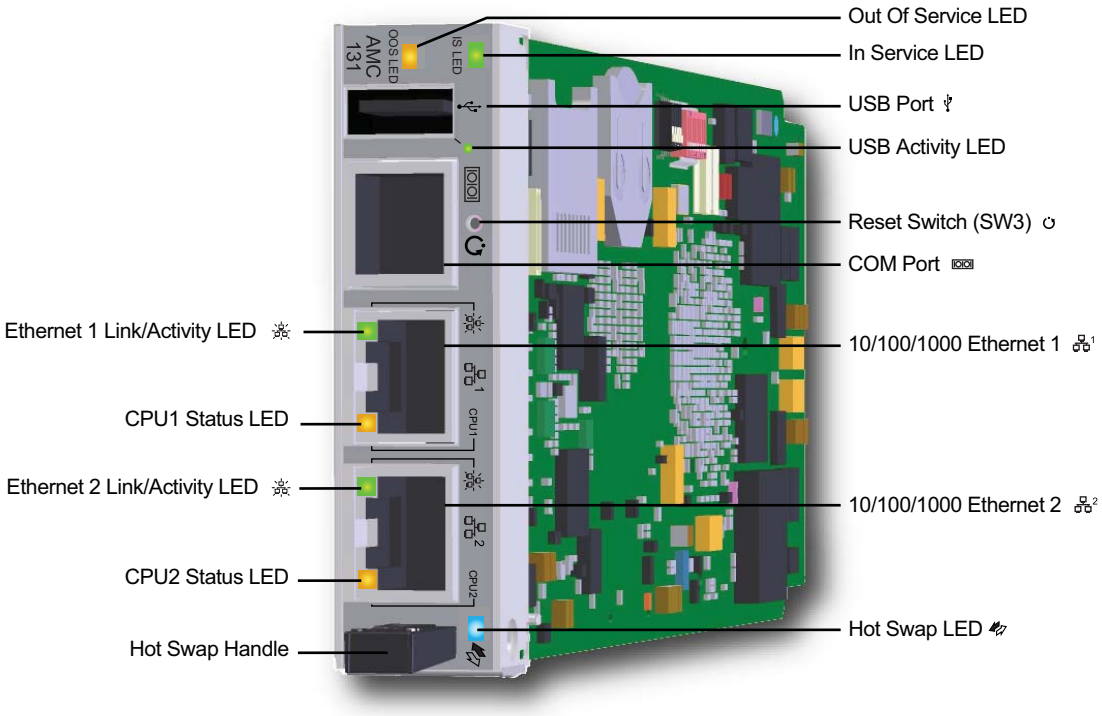
Table 2-1: AMC131 I/O Configurations

Option	Common Port 0	Common Port 1	Fat Pipe Port 4-7	Fat Pipe Ports 8-11
1	1 GbE	1 GbE	PCIe (x1, x2, x4, first half of x8)	PCIe (second half of x8)
2	Reserved			
3	1 GbE	1 GbE	SRIO x4	—
4	1 GbE	1 GbE	PCIe (x1, x2, x4)	SRIO x4

AMC131 Front Panel

Figure 2-1 shows the AMC131 mid-size front panel. For information about the various front panel components, see the appropriate sections in this guide.

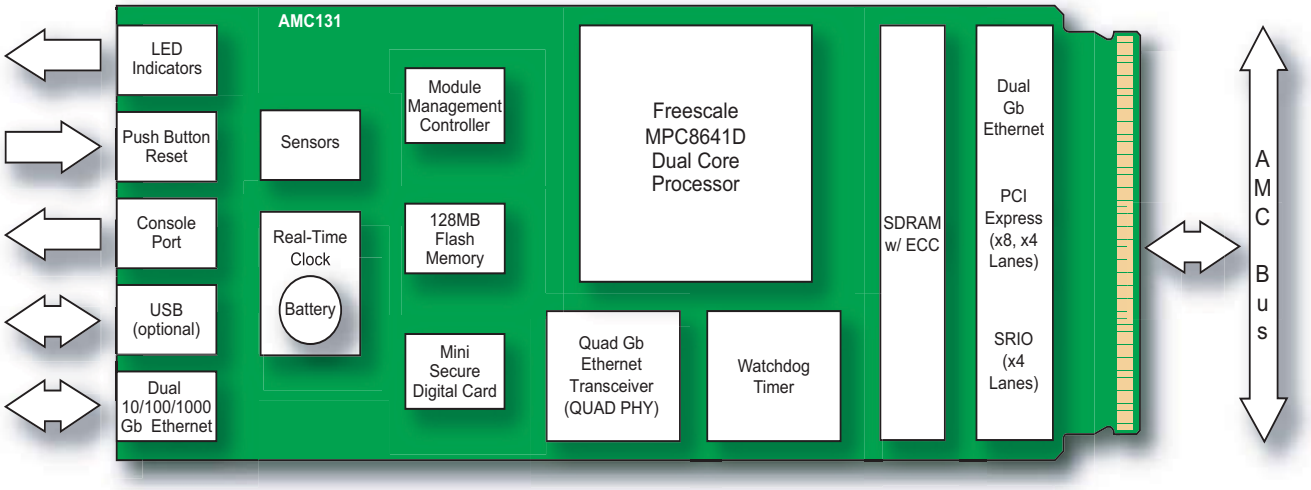
Figure 2-1: AMC131 Front Panel



Functional Blocks

Figure 2-2 is a functional block diagram of the AMC131. The topics following the figure provide overviews of the functional blocks.

Figure 2-2: AMC131 Functional Blocks



MPC8641D Dual-Core PowerPC Processor

The MPC8641D provides a dual-core architecture that delivers high bandwidth and advanced processing speed while addressing the power and heat constraints in an AMC form factor.

The MPC8641D integrates two PowerPC e600 processor cores that support high-speed interconnect technology to balance processor performance with I/O system throughput. The two e600 cores operate at 1.0 GHz, 1.25 GHz, 1.33 GHz, or 1.5 GHz. Each core has its own ECC protected 1 MB backside L2 cache. The AMC131 supports 1.0 GHz, 1.25 GHz, 1.33 GHz, or 1.5 GHz single- or dual-processor cores. Contact your PT sales representative for more information about these options.

The AMC131 design ensures that each PowerPC e600 processor has access to every resource on the module. Allocating management of multiple resources to the two processors requires several components. These components include:

- The IIC1 controller
- A common negotiating process

The IIC1 controller provides a communication mechanism that does not rely on any software managed resource.

The common negotiation process results in mutually exclusive management of a resource by a designated CPU. Resource management is accomplished by eight custom semaphore registers. For more information about these registers, see Chapter 7, “Programmable Registers,” on page 83.

The MPC8641D includes the following components:

- Two IIC Controllers
- Integrated Direct Memory Access Controller
- Dual Universal Asynchronous Receiver/Transmitter
- Ethernet Controllers
- Two SERDES controllers: One Serial Rapid IO (4 lanes) and one PCI Express (4 or 8 lanes)

Memory

AMC131 memory is comprised of the following components:

- SDRAM
- Flash Memory
- Global Storage
- Serial EEPROM

SDRAM

Two 200-pin sockets for DDR2 SDRAM are provided. This memory comprises 64 bits of data with 8 bits of ECC. Each socket on the module has 238M x 72 bits or 1GB for a total of 2 GB RAM. The built-in error correction code (ECC) helps ensure data integrity for reliable high-frequency operations. PT has qualified PC2-3200, 1GB, DDR2 SO-CDIMM for use with the AMC131. For more information, see “Memory Configuration,” on page 35.

For additional SDRAM density options, please contact your PT sales representative.

Memory is not a field serviceable item. Return the module to PT for memory replacement. See “Return Merchandise Authorization (RMA),” on page 19 for more information about returning merchandise.

Flash Memory

Onboard, non-volatile, programmable, Flash memory (128 Mb x 8) is provided for startup code and application storage. This memory is connected to the MPC8641D local bus and is accessible using the console port (see Figure 2-1, “AMC131 Front Panel,” on page 23).

Global Storage

A mini Secure Digital (miniSD) card is provided for additional non-volatile storage. This card provides a minimum of 256 MB memory for user code images. For additional miniSD density options, please contact your PT sales representative. See “Working with the MiniSD Card,” on page 35 for information about miniSD card installation and removal.

Serial EEPROM

Serial Electrically Erasable Programmable Read-only Memory (EEPROM, 256K x 1) is provided for boot initialization values.

AMC Interface

The AMC131 is compliant with the *AdvancedMC Advanced Mezzanine Card Base Specification, PICMG® AMC.0 R2.0*. It is designed to be hot swappable into a mid-size or full-size bay on a MicroTCA system or an AdvancedMC carrier board.

Note: *The 1.5 GHz option is available only in full-size.*

The AMC131 supports PICMG subsidiary specifications as shown in Table 2-2:

Table 2-2: AC131 Supported PICMG Subsidiary Specifications

	Types Supported		
	AMC.1 (PCIe)	AMC.2 (1GbE)	AMC.4 (SRIO)
Build Option 1	Type 1 - 1x on Port 4 Type 2 - 2x on Port 4,5 Type 4 - 4x on Port 4-7 Type 8 - 8x on Port 4-11	E1 and E2	N/A
Build Option 3	N/A	E1 and E2	Type 1 - 1x on Port 4 Type 3 - 1x on Port 6 Type 5 - 4x on Port 4-7
Build Option 4	Type 1 - 1x on Port 4 Type 2 - 2x on Port 4,5 Type 4 - 4x on Port 4-7	E1 and E2	Type 6 - 1x on Port 8 Type 8 - 1x on Port 10 Type 10 -4x on Port 8-11

Note: *Build option 2 is reserved.*

Quad Ethernet Port Architecture

Four 10/100/1000 Mbps RGMII Ethernet ports are supported by the onboard quad PHY. Two ports are implemented as BASE-T/TX connections to front panel RJ-45 connectors to support CAT5 cabling. Two ports are implemented as 1000 BASE-BX Ethernet connections to the AMC131 connector. See Figure 4-2, “AMC131 Connector Locations,” on page 47.

Two 1000 BASE-BX Ethernet connections are used on ports 0 and 1. Table 2-3 provides the Ethernet port mapping.

Table 2-3: Ethernet Port Mapping

MPC8641D RGMII Port	Quad PHY RGMII Port	Quad PHY Media Port	RJ45 Front Panel Port	AMC Common Options Port
TSEC{1}	TXRX{1}	TRD{1}	FP{1}	
TSEC{2}	TXRX{2}	TRD{2}	FP{2}	
TSEC{3}	TXRX{3}	SGIO{3}		TXRX{0}
TSEC{4}	TXRX{4}	SGIO{4}		TXRX{1}

For 10/100/1000 BASE-T RJ-45 and AMC131 connector pinout, see Chapter 4, “Connectors,” on page 43.

PCI Express

The PCI Express (PCIe) port is compatible with the *PCI Express Base Specification Revision 1.0a* and supports the following, with each lane at 2.5 Gbaud (2.0 Gb/s) only:

- x1, x2, x4, or x8 lane widths (build option 1)
- x1, x2 or x4 (build option 4)

The maximum supported packet payload size is 256 bytes. The interface supports virtual channel 0 (VC0) and traffic class 0 (TC0) only. The PCIe interface can be configured by the user as a PCIe Endpoint or as a Root Complex. PCIe is configured as an endpoint by default. Enabling the PCIe port is also configurable. The PCIe lanes are routed to the AMC fat pipe ports as shown in Table 2-4, “SERDES Port Mapping,” below.

For more information about configuring PCIe, see “SW2 (PCI Express/SRIO Configurations),” on page 39 and “Register 13,” on page 93.

Table 2-4: SERDES Port Mapping

AMC FatPipes Port	TXRX{4}	TXRX{5}	TXRX{6}	TXRX{7}	TXRX{8}	TXRX{9}	TXRX{10}	TXRX{11}
MPC8641D PCIe Port (Build option 1)	SD1{0} (PCIe)	SD1{1} (PCIe)	SD1{2} (PCIe)	SD1{3} (PCIe)	SD1{4} (PCIe)	SD1{5} (PCIe)	SD1{6} (PCIe)	SD1{7} (PCIe)
MPC8641D SRIO Port (Build option 3)	SD2{0} (SRIO)	SD2{1} (SRIO)	SD2{2} (SRIO)	SD2{3} (SRIO)	---	---	---	---
MPC8641D PCIe/SRIO Port (Build option 4)	SD1{0} (PCIe)	SD1{1} (PCIe)	SD1{2} (PCIe)	SD1{3} (PCIe)	SD2{0} (SRIO)	SD2{1} (SRIO)	SD2{2} (SRIO)	SD2{3} (SRIO)

Note: *Build option 2 was not implemented and is reserved.*

Serial Rapid IO

The Serial Rapid IO (SRIO) port is compatible with the Serial Rapid IO Specification, Revision 1.0. It supports x1, or x4 lane widths with each lane at 3.125 Gbaud (2.5 Gb/s) only. The SRIO interface can be configured by the user as an SRIO Agent or as a Host. SRIO is configured as an Agent by default. It can also be configured with either small system 8-bit source and destination IDs for up to 256 devices, or large system 16-bit IDs for up to 65,536 devices. For more information about configurig SRIO, see “SW1 (Serial Rapid IO Configurations),” on page 38.

The four SRIO lanes are routed to the AMC fat pipes ports as shown in Table 2-4.

Module Management Controller (MMC)

The AMC131 includes a Module Management Controller (MMC) based on the Atmel ATMEGA128L-8MU, which interfaces to the local Intelligent Platform Management bus (IPMB-L). The MMC monitors and controls the module's payload per the PICMG AMC.0 specification.

See Chapter 5, “System Monitoring and Alarms,” on page 53 for more information on MMC functionality, supported commands, AMC131 sensors, and the firmware upgrade process.

The AMC131 is compliant with standard *Intelligent Platform Management Interface v1.5 Specification* functionality. See “Module Management Controller,” on page 107, for information about this specification.

Sensors

The following sensors are monitored on the AMC131:

- Air intake temperature sensor
- MPC8641D built in temperature sensor
- MMC voltage monitoring (3.3V management power and 12V power to the AMC slot)
- MMC hot-swap switch sensor

For more information on AMC131 sensors, Chapter 5, “System Monitoring and Alarms,” on page 53.

Universal Serial Bus

The AMC131 has one Universal Serial Bus (USB) port on the front panel (see Figure 2-1, “AMC131 Front Panel,” on page 23). This USB 2.0 port is supported by the Philips ISP1160 dual-port host controller.

For USB pinout information, see “Console Port Connector (J2),” on page 45.

Console Port

The front panel console port (see Figure 2-1, “AMC131 Front Panel,” on page 23) is an RJ-11 connector with RS-232 signal levels and 15 KV ESD protection. This port is pinned out for Data Terminal Equipment (DTE) operation. Request-to-send (RTS), clear-to-send (CTS) and modem-control signals are not supported.

There are two options for connectivity through this connector

- Module Management Controller
- Payload module

Although two Universal Asynchronous Receiver/Transmitter (UARTS) are provided on the MPC8641D, only Port 0 is connected to the transceiver. Port 1 is used as a host interface for communication to/from the MMC.

When enabled (via Register 8), the user can toggle between the processor and the MMC console ports with a single break character. A double break (break-break), when enabled (via register 8), causes a hard reset to the processor.

When payload power is off, the console port transceiver is connected to the MMC. The console port transceiver is powered from the AMC Management Power (MP) pin. A hard reset leaves the console port focus unchanged. This is useful when access to time sensitive software queries is desired. A payload power reset always returns focus to the MMC.

For console port pinout information, see “Console Port Connector (J2),” on page 45.

Push Button Reset

The front panel push button (see Figure 2-1, “AMC131 Front Panel,” on page 23) provides a hard reset if pressed.

The AMC131 also supports power and soft reset options. For information about the AMC131 reset options, see Chapter 6, “Reset Configuration,” on page 79.

Real-time Clock with Battery Backup

The AMC131 features an I2C-based real time clock (RTC). This device contains the year, month, date, day, hours, minutes, and seconds. The clock operates in the 24-hour binary-coded decimal (BCD) format. Corrections are automatically made for 28, 29, 30 and 31 day months including leap year.

An alarm clock function is provided on the real-time clock, with one second minimum resolution that can be enabled to generate payload processor interrupts.

The AMC131 supports an external, industry standard, 2016 size, coin cell battery to ensure that the clock remains current when the computer is turned off. The battery life is five years.

Watchdog Timer

The watchdog timer function provides a timer independent of the MPC8641D that can be used to monitor system operation. Its duration is programmable, with a range of 26.1ms to 28.5 minutes. When enabled, the watchdog timer can be configured to provide an IRQ7 interrupt to the MPC8641D, a hard reset pulse to the board, or a kill function that causes the board to halt until power is cycled. The Watchdog configuration and operational functions are controlled through a set of custom registers in the onboard logic. The Watchdog function powers up in the disabled state and must be enabled under software control for all modes of operation.

The PT-supplied PTBMON debugger has the capability to utilize the Watchdog functions. For more information about PTBMON, see the *PTBMON Boot Monitor User Guide*.

LED Indicators

The AMC131 provides eight LED indicators (see Figure 2-1, “AMC131 Front Panel,” on page 23). Table 2-5 describes the AMC131 LEDs and their functions.

Table 2-5: AMC131 LED Architecture

LED Name	Color	Description
AMC LED 1 (OOS)*	Red	Out Of Service (geographic Option 1)
	Amber	Out Of Service (geographic Option 2)
AMC LED 2 (IS)*	Green	In Service (MMC Defined)
USB	Off	No USB Device Present
	Green	USB Device Present
Ethernet 1 Link/Activity	Off	No Link
	Green	Link Established but No Activity
	Blinking green	Link Established and Tx or Rx Activity
CPU1 Status	Off	LED Circuit Failure
	Amber	Power OK, CPU1 Can Boot
	Green	CPU1 In Service
	Red	CPU1 Fault
Ethernet 2 Link/Activity	Off	No Link
	Green	Link Established but No Activity
	Blinking green	Link Established and Tx or Rx Activity
CPU2 Status	Off	LED Circuit Failure
	Amber	Power OK, CPU2 Can Boot
	Green	CPU2 In Service
	Red	CPU2 Fault
Hot Swap	Blue	It is Safe to Extract the Module
	Blinking Blue	In Transition; Not Safe to Extract the Module
	Off	It is Not Safe to Extract the Module.

*The OOS and IS LEDs are used to indicate an “out of service” condition or an “in service” status, per the PICMG Advanced Mezzanine Card AMC.0 Specification R2.0. Although these LEDs are managed by the MMC, a carrier manager or shelf manager can override the MMC's local LED settings.

The OOS LED is activated to indicate that the payload is known to be out of service (payload power is off, held in reset, or faulted in a way that precludes operation). Otherwise the OOS LED is off. The health of the board cannot be inferred solely from the state of this LED. The default local color of the OOS LED is determined by FRU data and is configurable as either red or amber. Contact PT Customer Support for information about configuring this color.

The IS LED is activated when the OOS LED is turned off. It is never turned on when the OOS LED is on. The IS LED is green when all sensors are within the critical thresholds or amber when one or more sensors have exceeded a critical threshold.

Software

The AMC131 is designed to run Linux. The module software functions use proprietary PT NexusWare applications. For more information about NexusWare, see the NexusWare Core Reference Manual, available at the following URL:

http://pt.com/page/embedded/software/nexusware/nexusware-core

The PT boot monitor application, PTBMON, can be used to configure many aspects of module operation. PTBMON commands include the following:

- Commands for booting and loading programs into memory
- Debugging commands
- Shell commands, which cover a broad spectrum of functionality, such as providing information about the program and the actions that may be performed at the command-line prompt, setting and displaying the date, and performing volatile changes to the console's terminal settings
- Network commands for setting up the management port
- Memory manipulation commands
- Environment commands, which allow the creation, viewing, editing and deletion of environment variables
- PTI commands, which facilitate the initialization, modification and viewing of the boot data and prom data contained in FLASH, and commands that display PLD registers and networking related registers on the processor
- Miscellaneous commands that allow running a function when its memory address is known, flushing either the data or instruction caches, and erasing and programming the application flash, among other functions

For more information about PTBMON, see the *PTBMON Boot Monitor User Guide*.

As an alternative to PTBMON, the AMC131 may be provided with U-Boot, a flexible boot loader. Supporting a number of different computer architectures, U-Boot is closely integrated with embedded Linux, including support for flattened device trees.

Functionally similar to PTBMON, U-Boot provides much the same support for network management and serial communications ports, with the ability to load and boot programs into memory. Environment variables are used to store configuration data in FLASH. U-Boot is more flexible, with the ability to store commands and command sequences (scripts) in the board’s nonvolatile memory although it doesn't provide the same level of debug support, lacking the ability to manipulate cache from the command line.

The U-Boot software is released under the terms of the GNU General Public License. Visit http://www.denx.de/wiki/U-Boot/WebHome for more information about U-Boot.

For U-Boot loading options, please contact your PT sales representative.

Chapter 3

Getting Started

This chapter provides you with information about configuring the AMC131. It includes the following key topics:

Please read this chapter before attempting to use the board.

Unpacking

Inspect the packing container for any damage. If this container appears damaged, immediately contact the company responsible for the shipping and report the damage before opening and unpacking the container. It is recommended that you also notify PT (see “Customer Support and Services,” on page 18).

Caution:
To reduce the risk of damage to the AMC131, this equipment must be protected from electrostatic discharge (EMC) and physical shock. Never remove any of the socketed parts except in a static-free environment. Use the anti-static bag shipped with the product to handle the module.

Caution:
Avoid touching areas of integrated circuitry. Static discharge can damage these circuits.

System Requirements

The following topics provide information about the system requirements:

Compatibility

The AMC131 is a single-width module that is offered with either a mid-size or full-size front panel (see Figure 2-1, “AMC131 Front Panel,” on page 23).

The AMC131 is compliant with the *AdvancedMC Advanced Mezzanine Card Base Specification, PICMG® AMC.0 R2.0*. It is designed to be hot swappable into a mid-size or full-size bay on a MicroTCA system or an AdvancedMC carrier board.

Note: *The 1.5GHz option is available only in full-size.*

The AMC131 also supports the following PICMG subsidiary specifications:

- AMC.1 (PCIe), one port of type 1, 2, 4, or 8 at 2.5 Gb per lane only
- AMC.2 (1GbE) types E1and E2
- AMC.4 (SRIO):
  - Build option 3 supports AMC.4 types 1, 3, and 5 at 3.125 Gb per lane only
  - Build option 4 supports AMC.4 types 6, 8, and 10 at 3.125 Gb per lane only

Electrical and Environmental Requirements

Electrical specifications are presented in detail in “Electrical and Environmental Specifications,” on page 99.

The AMC131 operates between 0°C (32°F) and 55°C (132°F) ambient with a minimum of 250 LFM (1.27 meters per second) of external airflow.

It is the user’s responsibility to ensure that the AMC131 is installed in a chassis capable of supplying adequate airflow. The maximum power dissipation of the processor IC is 24W. External airflow *must* be provided at all times. See Chapter 8, “Specifications,” on page 99 for more details.

Warning:
Operating the AMC131 without adequate airflow will damage the processor.

The AMC131 may contain environmentally hazardous materials. You must make sure that you dispose of any such materials in accordance with your local rules and regulations. For disposal and recycling information, contact your local authorities or the Electronic Industries Alliance (EIA) at http://www.eiae.org/.

PT Compliance with RoHS and WEEE Directives statement is on page 106.

Memory Configuration

Each of the two 200-pin SODIMM sockets on the AMC131 is populated with DDR2 memory that is configured to run at PC2-3200 speed with a 1.0GHz processor. The 1.33 GHz and 1.5 GHz options are populated with PC2-5300 DDR2 memory that is configured to run at 266 MHz and 300 MHz respectively. A single socket can support a 2GB module, but the total memory loaded in the two sockets must not exceed 3GB.

Memory is not a field serviceable item. Return the module to PT for memory replacement. See “Return Merchandise Authorization (RMA),” on page 19 for more information about returning merchandise.

Figure 3-1 shows the memory addressing for the AMC131.

Figure 3-1: Memory Address Map

0000_0000 to 7FFF_FFFF	DDR2 memory
F7E0_0000 to F7E0_7FFF	Custom PAL register space (CS1)
F7E0_8000 to F7E0_8002	USB Controller
F7F0_0000 to F7FF_FFFF	CCSR Space
F800_0000 to FFE1_FFFF	Application code area in PROM (CS0)
FFE2_0000 to FFE2_07FF	Boot Data (CS0)
FFE4_0000 to FFE4_5BBB	PROM Data (CS0)
FFE6_0000 to FFE7_FFFF	Test Log Data (CS0)
FFE8_0000 to FFFF_FFFF	Boot Code (CS0)

Working with the MiniSD Card

MiniSD card support is provided via a miniSD card socket located on the solder side of the board. See Figure 3-2, “Installing a MiniSD Card,” on page 36 for the socket location.

Removing a MiniSD Card

To remove the miniSD card from the socket:

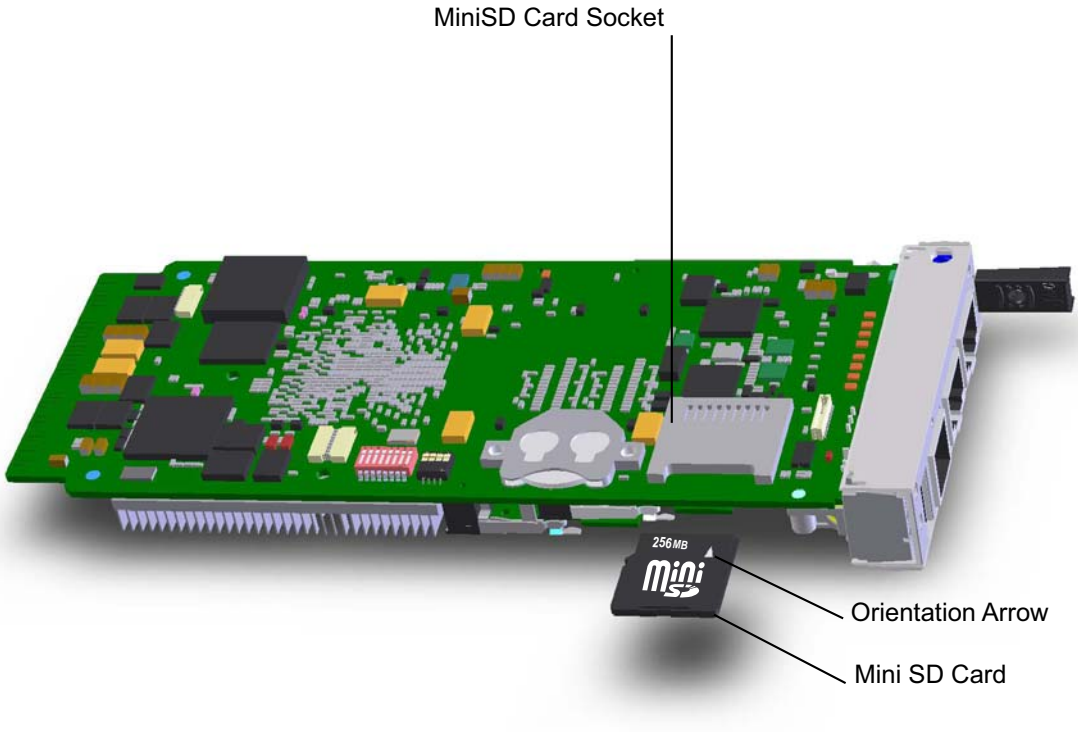
1. Push in on the miniSD card to unlatch it. The miniSD card springs out about 1/4" from the socket when it is properly unlatched.
2. Gently remove the miniSD card from the socket.

Installing a MiniSD Card

To install a miniSD card in the socket:

1. Orient the AMC131 and miniSD card as shown in Figure 3-2, with the miniSD card label away from the AMC131 PCB. Many miniSD cards have an arrow silkscreened onto their label side, as shown in Figure 3-2, indicating the end that should be inserted into the socket.
2. Insert the miniSD card into the socket until it clicks into place.

Figure 3-2: Installing a MiniSD Card



Connectivity

The AMC131 provides several connectors for interfacing with application-specific devices. Refer to Chapter 4, “Connectors,” on page 43 for complete connector descriptions and pinouts.

Switch Configuration

The AMC131 includes several options that tailor the operation of the module. Most of the options are selected through software; however, some options cannot be software controlled and are configured with switches. Closing or opening the desired switch selects each switch’s options.

Key topics in this section:

Switch Options and Locations

The AMC131 includes the following switches:

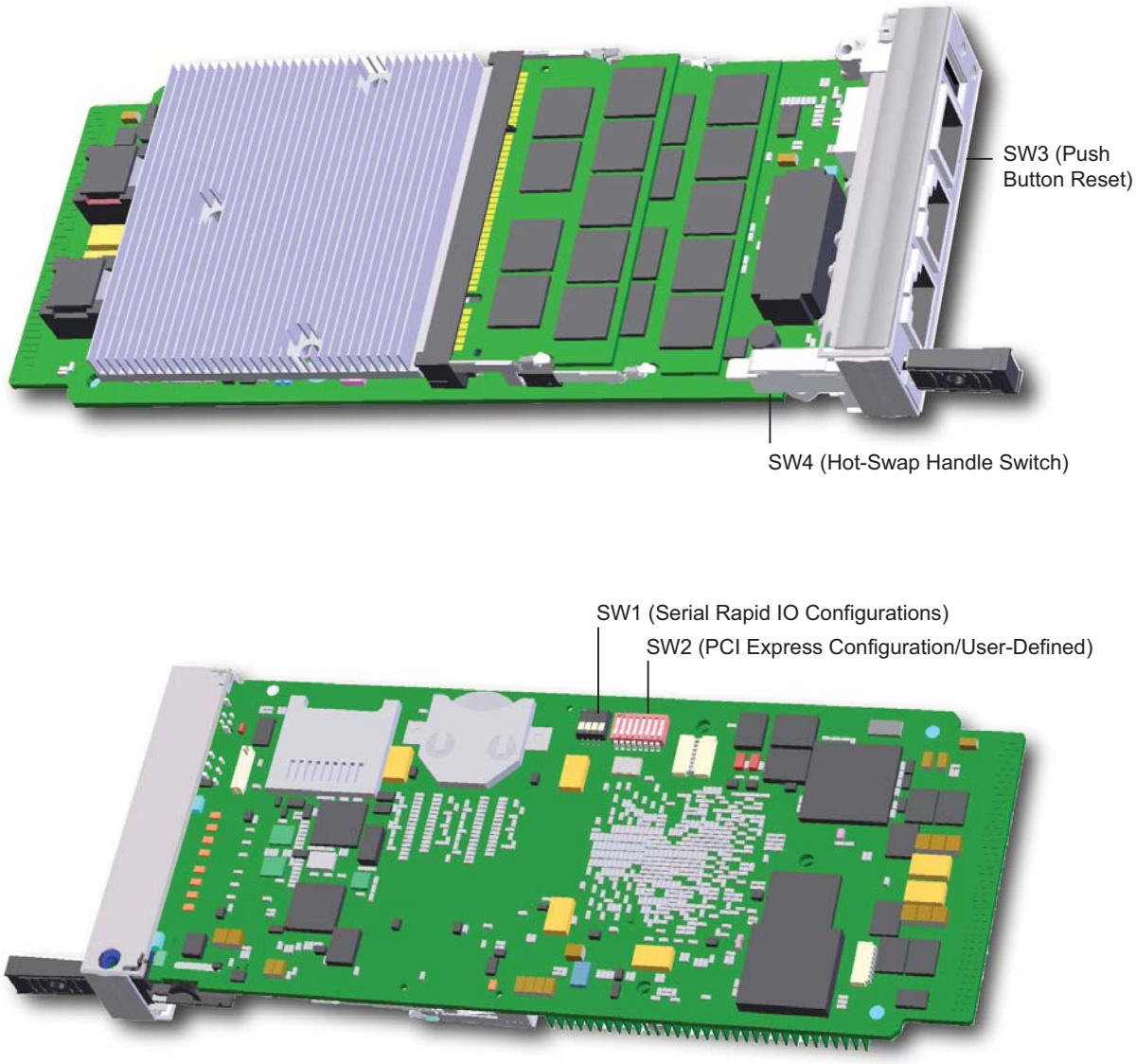
- DIP switches SW1 and SW2

Note: *PCIe-only AMC131 boards at revisions 120Q060960 and lower have only one DIP switch. The settings documented as SW2 in this document must be applied to the switch marked SW1 on those boards.*

- Push-button switch (SW3)
- Hot-swap handle switch (SW4)

Figure 3-3, “AMC131 Switch Locations,” shows the AMC131 switch locations on the top and bottom sides of the module.

Figure 3-3: AMC131 Switch Locations



Switch Descriptions

The following sections present the AMC131 switches in numerical order and provide a detailed description of each switch. Multiple-position switches are identified in the form of **SWx-N**, where **x** is the switch number and **-N** is the switch position. For example, SW2-3 means “switch number 2, position 3.”

Note: *PCIe-only AMC131 boards at revisions 120Q060960 and lower have only one DIP switch. The settings for SW2 in this document must be applied to the switch marked SW1 on those boards.*

SW1 (Serial Rapid IO Configurations)

Note: *All switches must be set prior to power-on reset.*

SW1-1: SRIO System Size

The Rapid IO Common Transport Specification defines two system sizes:

- The large size that uses 16-bit source and destination IDs, allowing for 65,536 devices
- The small size that uses 8-bit source and destination IDs, supporting 256 devices

SW1-1 sets the SRIO controller system size as shown in Table 3-1.

Table 3-1: Setting SRIO System Size

SW1-1	Function
OFF	Set the SRIO controller system size as large
ON (Default)	Set the SRIO controller system size as small

SW1-2 - SW1-4: SRIO Device IDs (small system only)

If large size is selected with SW1-1, SW1-2 - SW1-4 settings are ignored. If small size is selected, SW1-2 - SW1-4 are used to set configuration device ID bits 5 - 7 as shown in Table 3-2.

Table 3-2: Setting SRIO Device IDs

Switch	OFF	ON (Default)
SW1-2	Device ID bit 5 set to 0	Device ID bit 5 set to 1
SW1-3	Device ID bit 6 set to 0	Device ID bit 6 set to 1
SW1-4	Device ID bit 7 set to 0	Device ID bit 7 set to 1

If configured as a SRIO Host (see SW2-1: PCIe Root Complex/Endpoint - SRIO Host/Agent below) then the device ID is 0b0000_0||cfg_device_id[5:7].

If configured as a SRIO Agent:

- If cfg_dev_ID[7] input = 0, then the device ID is 0xFE and
- If cfg_dev_ID[7] input = 1, then the device is a generic, undiscovered endpoint with a device ID of 0xFF.

SW2 (PCI Express/SRIO Configurations)

The PCI Express is set up with some parameters set by default, some settings controlled by the MMC/shelf manager e-keying process, and some controlled by SW2. By default the PCI Express port is set to be in an on state with the maximum number of lanes enabled for the given build option. The remaining options are selected by using SW2 switches and by the MMC's interaction with the shelf manager. This is further explained in the following topics.

Note: *PCIe-only AMC131 boards at revisions 120Q060960 and lower have only one DIP switch. The settings documented as SW2 in this document must be applied to the switch marked SW1 on those boards.*

SW2-1: PCIe Root Complex/Endpoint - SRIO Host/Agent

This switch functions differently depending on the build option. For build option 4, which has both PCIe and SRIO ports active, the position of this switch applies to both PCIe and SRIO simultaneously. See Table 3-3.

Note: *This switch must be set prior to power-on reset.*

Table 3-3: Setting the PCIe and SRIO Port Functions

SW2-1	Build Option 1 - PCIe Interface Only	Build Option 3 - SRIO Interface Only	Build Option 4 - PCIE and SRIO Interfaces
OFF (Default)	Sets PCIe port as Endpoint	Sets SRIO port as Agent	Sets PCIe port as Endpoint *and* Sets SRIO port as Agent
ON	Sets PCIe port as Root Complex	Sets SRIO port as Host	Sets PCIe port as Root Complex *and* Sets SRIO port as Host

SW2-2: FCLKA Configuration - PCIe only

The choice of PCI Express clock source is normally made by the MMC through e-keying with the AMC/ATCA shelf controller. The e-keying process can set up the AMC131 to source the FCLKA, receive the FCLKA, or be isolated from the FCLKA. The AMC131 can also be forced to receive the FCLKA from the system. SW2-2 tells the MMC what action is desired for FCLKA selection. The settings for SW2-2 are shown in Table 3-4.

Table 3-4: Setting FCLKA

SW2-2	Function
OFF	Local clock source off. AMC PCIE FCLKA connected directly to MPC8641D PCIe Clock Input. Use this setting in AMC.1 R1.0 carriers.
ON (Default)	The MMC e-keys the FCLKA setting for the AMC131 in compliance with AMC.1 R2.0

If the AMC131 does not boot up when powered on, it is likely that FCLKA is not configured properly for the specific carrier. For more information about configuring FCLKA on this AMC, please contact PT Customer Support (see “Customer Support and Services” on page 18).

SW2-3: Spread Spectrum Generation

SW2-3 on enables the spread spectrum generation function on the on-board PCI Express clock generator as shown in Table 3-5:

Table 3-5: Setting Spread Spectrum

SW2-3	Function
OFF (Default)	Spread spectrum function disabled
ON	Spread spectrum function enabled

Off disables the spread spectrum function.

Note: *Enabling SSC is allowed only when FCLKA is not being received from the backplane. Therefore, to enable SSC, SW2-2 and SW2-3 must be set to ON and ekeying must not require the AMC131 to receive FCLKA.*

SW2-4 to SW2-8: Reserved

SW2-4 to SW2-8 are reserved for use by the factory. They should not be changed from the default setting of OFF.

SW3 (Push Button Reset)

The push button reset switch is accessible from the AMC131 front panel (see Figure 2-1, “AMC131 Front Panel,” on page 23). Pressing SW3 causes the MPC8641D processor to reset.

SW4 (Hot-Swap Handle Switch)

The AMC131 provides a hot swap handle on its front panel (see Figure 2-1, “AMC131 Front Panel,” on page 23). This handle is attached to a mechanical latching mechanism and to the hot swap switch. When this switch opens or closes it sends a request via the MMC to the carrier for a hot swap extraction or insertion. Its function and behavior is defined by the PICMG AMC.0 specification. The hot swap LED indicates the state of the module during extraction and insertion. See “LED Indicators,” on page 30 for more information.

Physical Installation

Before installing the AMC131, make sure the module is correctly configured for your application.

Installing the AMC131

The following instructions assume that chassis power is on and that the system supports hot-swap insertion. If the system does not support hot swap, power must be turned off prior to installation.

1. Unlock the ejector handle by gently pulling it away from the front panel.
2. Slide the AMC131 into an available slot in the system, aligning the board with the guides near the top of the slot. The module audibly snaps into place when properly inserted.
3. Press the handle toward the front panel to lock the module in the chassis. When the module card edge connector makes proper contact with the backplane connector, the blue hot swap LED turns ON and the hardware connection process begins. When the module is operational, the blue hot swap LED turns OFF.
4. Connect any cables from peripheral devices.

Removing the AMC131

To remove the module:

1. Unlock the module by pulling the handle away from the front panel. The blue hot swap LED blinks to indicate that the handle is open and the module is waiting to be deactivated. It is not yet safe to extract the module in this state. Wait until the blue LED stops blinking and remains illuminated to indicate that the board is ready for extraction.
2. When the blue LED stops blinking and remains illuminated, the module is quiesced and module payload power is disabled. It is now safe to extract the module. Gently pull on the handle to remove the module from the system.

Chapter 4

Connectors

The AMC131 has several connectors that interface with application-specific devices (see Figure 4-1, “AMC131 Front Panel Connectors,” on page 44 and Figure 4-2, “AMC131 Connector Locations,” on page 47).

Key topics in this chapter:

- “Front Panel Connectors,” on page 44
- “Internal Connectors,” on page 47
- “AdvancedMC Card Edge Connector (P5),” on page 50

A brief description of each connector is given in Table 4-1, “Connector Assignments.” A detailed description and pinout for each connector is given in the following topics.

Table 4-1: Connector Assignments

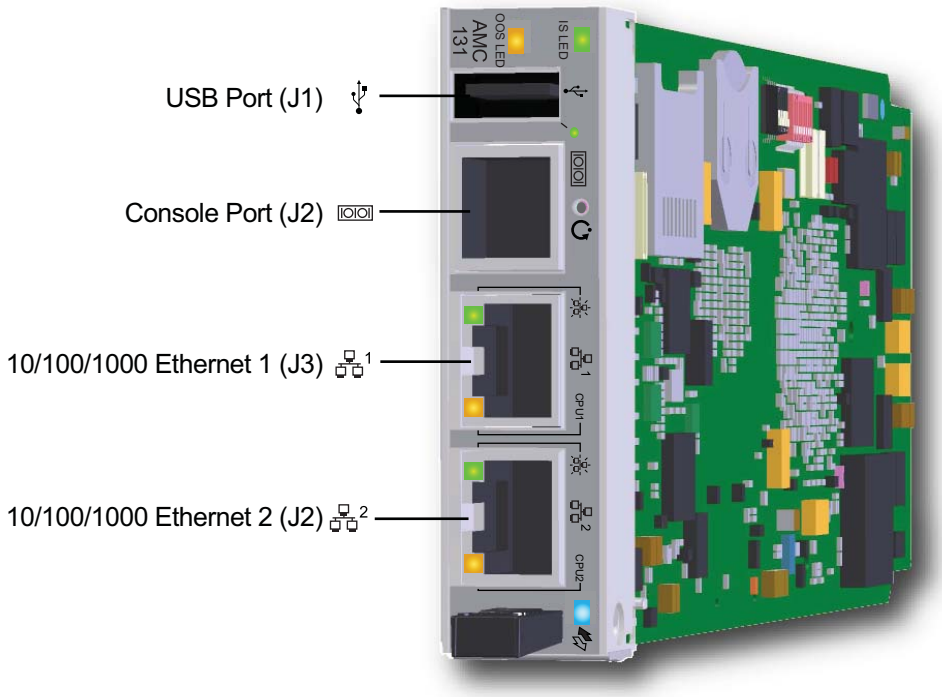
Function	Location
USB Connector (J1)	Front Panel
Console Port Connector (J2)	Front Panel
10/100/1000 Base-T Ethernet Port Connectors (J3, J4)	Front Panel
200-pin SDRAM Sockets (P6, P7)	Internal
Battery Socket (BT1)	Internal
MiniSD Card Connector (P1)	Internal
Payload Processor IMC Connector (P2, P3)	Internal
Reserved Connector (P4)	Internal
AdvancedMC Card Edge Connector (P5)	Card Edge

Front Panel Connectors

The AMC131 front panel connectors are as follows:

- USB port (optional) connector
- Console port connector
- 10/100/1000 Base-T Ethernet port (FP1 and FP2) connectors

Figure 4-1: AMC131 Front Panel Connectors



USB Connector (J1)

The AMC131 has a Type-A vertical USB connector on its front panel.

Table 4-2 shows the USB connector pinout.

Table 4-2: USB Connector Pinout

Type A Pin	Name
1	VBUS
2	D-
3	D+
4	GND

Console Port Connector (J2)

The front panel console port is an RJ-11 connector with RS-232 signal levels and 15kV ESD protection. It is pinned for data terminal equipment (DTE) operation.

Note: *The AMC131 does not support Request-to-Send (RTS), Clear-to-Send (CTS), and modem-control signals.*

The RJ-11 console port is multiplexed to support both the MPC8641D processor and the Module Management Controller (MMC). For more information about the console port functionality, see “Console Port,” on page 28.

Table 4-3 shows the console port pinout.

Table 4-3: Console Port Connector Pinout (RJ-11)

RJ-11 Pin	I/O	Name	Connection
1	NC	CTS	RTS
2	GND	GND	GND
3	Input	RXD	Rx Data
4	Output	TXD	Tx Data
5	GND	GND	GND
6	NC	RTS	CTS

Console Cable

A console cable is available to provide a male DB-9 connector. The cable provides a DTE connection (RS-232 levels), allowing connection to a modem (DCE) without the need of a null modem. A null modem (DB-9F to DB-9F) is provided to allow connection to a terminal or PC (DTE) directly. This cable is 3 feet in length (min.). The pinout of the DB-9 is described in Table 4-4.

Table 4-4: DB-9 Console Cable Pinout

DB-9	I/O
1	NC
2	RXD
3	TXD
4	DTR Looped to Pin 6
5	GND
6	DSR Looped to Pin 4
7	RTS Looped to Pin 8
8	CTS Looped to Pin 7
9	NC

10/100/1000 Base-T Ethernet Port Connectors (J3, J4)

Four Ethernet ports are supported by the AMC131. Two ports are implemented as 10/100/1000 BASE-T/TX connections to front panel RJ-45 connectors (FP1 and FP2), in order to support CAT5 cabling. These connectors are pinned for medium dependent interface (MDI) operation on DTE equipment according to IEEE 802.3- 2000, Table 23-6.

Both front panel Ethernet ports can be utilized as management ports to the MPC8641D processor. During the boot process, you are provided with a time period to choose to abort the normal process and proceed with a boot loading process using an Ethernet management port.

Table 4-5 shows the 10/100/1000 Base-T Ethernet port pinout.

Table 4-5: 10/100/1000 Base-T Ethernet Port Connector Pinout (RJ-45)

RJ-45 Pin	Name
1	TX_D1+
2	TX_D1-
3	RX_D2+
4	BI_D3+
5	BI_D3-
6	RX_D2-
7	BI_D4+
8	BI_D4-

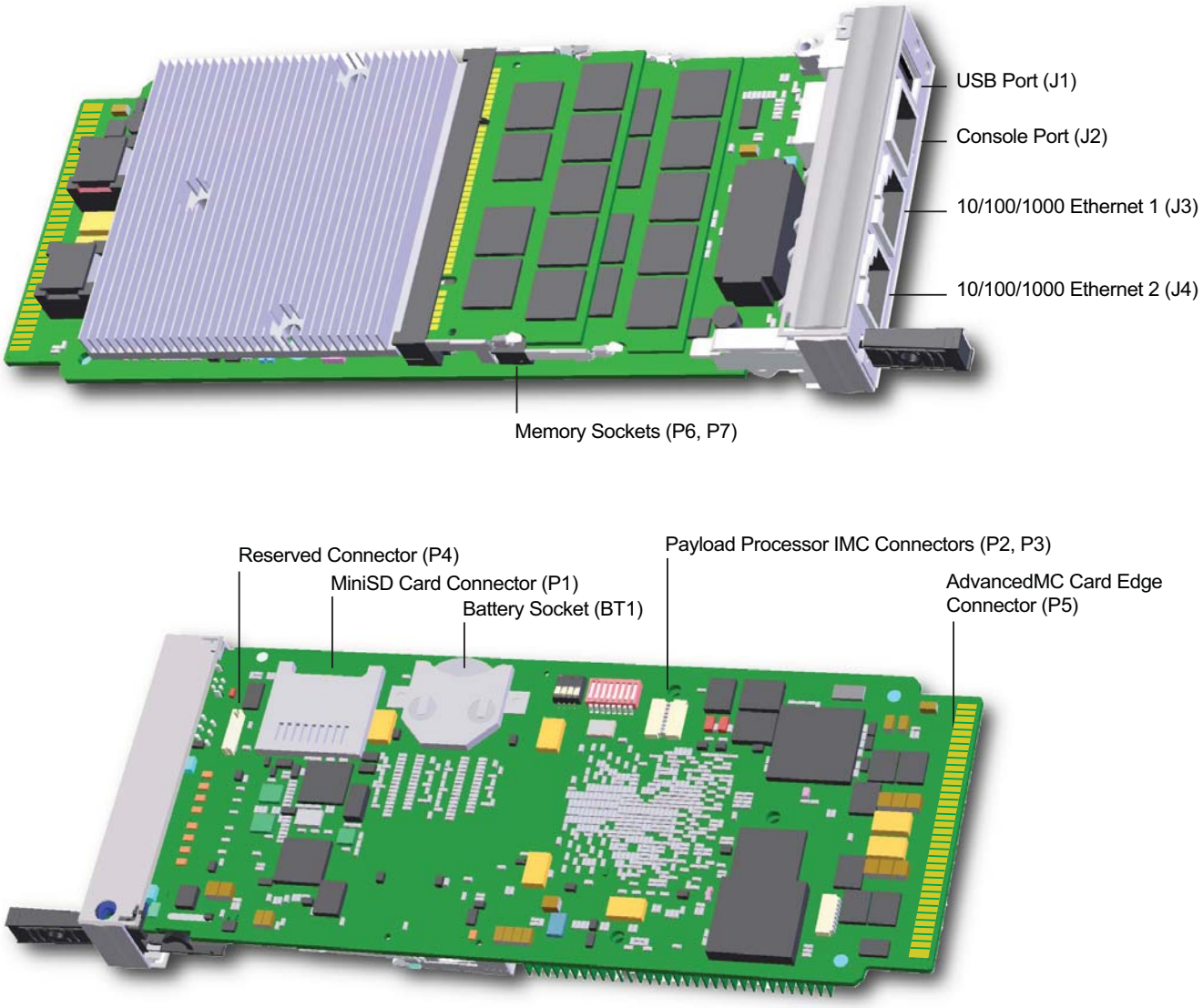
Internal Connectors

The AMC131 supports the following internal connectors:

- 200-pin SDRAM sockets
- Battery socket
- MiniSD card connector
- Payload processor in-circuit emulation (ICE) connector
- Module management controller (MMC) and in-circuit emulator (IMC) connector

See Figure 4-2, “AMC131 Connector Locations,” on page 47 for connector identification.

Figure 4-2: AMC131 Connector Locations



200-pin SDRAM Sockets (P6, P7)

For information on memory compatibility, see “SDRAM,” on page 25.

Note: *Memory is not a field serviceable item. Return the module to PT for memory replacement. See “Return Merchandise Authorization (RMA),” on page 19 for more information about returning merchandise.*

Battery Socket (BT1)

See “Real-time Clock with Battery Backup,” on page 29 for information on battery compatibility.

Note: *The battery is not a field serviceable item. Return the module to PT for memory replacement. See “Return Merchandise Authorization (RMA),” on page 19 for more information about returning merchandise.*

MiniSD Card Connector (P1)

MiniSD card support is provided via a miniSD card socket located on the board. See “Global Storage,” on page 25 for more information. See “Working with the MiniSD Card,” on page 35 for information about miniSD card installation and removal.

Payload Processor IMC Connector (P2, P3)

The payload processor in circuit emulator is accesses through connectors P2 and P3. Table 4-6 provides the pinout.

Table 4-6: Payload Processor ICE and JTAG PCB Connector Pinout

P3 PIN	NET	Pin Function on the Module
1	CPU1_TDO	CPU1 Test Data Out
2	CPU1_TDI	CPU1 Test Data In
3	COP_RUN_STOP_N	Common On-Chip Debugger (COP) Software Run/Stop
4	CPU1_TCK	CPU1 Test Clock
5	CPU1_TMS	CPU1 Test Mode Select
6	COP_SRST_N	Common On-Chip Debugger Software Reset
7	COP_HRST_N	COP Hardware Reset
8	CPU1_CKSTPO_N	CPU1 Checkstop Out
P2 PIN		
1	NC	No Connect
2	COP_TRST_N	COP Test Reset
3	V3P3_COP_VDD_SENSE	Ensures that the COP is aware whether 3.3V power is applied to the module
4	CPU1_CKSTPI_N	CPU1 Checkstop In
5	ICE_EN_N (connect to GND)	In Circuit Emulator Enable
6	GND	Logical Ground
8	NC	No Connect
8	GND	Logical Ground

Table 4-7 provides the cable pinout for the payload processor in circuit emulator.

Table 4-7: Payload Processor ICE Cable Connector Pinout

PIN	NET	Pin Function on the Module
1	COP_TDO	COP Test Data Out
2	-	-
3	COP_TDI	COP Test Data In
4	COP_TRST_N	COP Test Reset
5	COP_RUN_START_N	COP Run/Start
6	VDD_SENSE	Ensures that the COP is aware whether 3.3V power is applied to the module
7	COP_TCK	COP Test Clock
8	COP_CKSTP_IN_N	COP Checkstop In
9	COP_TMS	COP Test Mode Select
10	-	-
11	COP_SRST_N	COP Software Reset
12	GND	Logical Ground
13	COP_HRST_N	COP Hardware Reset
14	KEY	A key is a non-populated pin in the connector placed so you cannot plug it in incorrectly
15	COP_CKSTP_OUT_N	COP Checkstop Out
16	GND	Logical Ground

Reserved Connector (P4)

This connector is reserved for use by PT.

AdvancedMC Card Edge Connector (P5)

The AdvancedMC Connector provides the electrical interface between the AMC131 and the MicroTCA enclosure or ATCA carrier board. The AdvancedMC connector is fixed to the enclosure or carrier board and the card edge interface at the back of the AMC131 plugs into it. There are different styles of connectors for the different types of AdvancedMC bays and for different levels of connectivity. The card edge interface on the AMC131 is compatible with the extended, 170 pin B+ style connector.

Table 4-8 provides the pinout for the AMC131 card edge connector. NC indicates a pin that is not connected.

Table 4-8: AMC131 Connector Pinout

Pin	Signal	Driven By	Mating	Pin Function on the Module	Pin	Signal	Driven By	Mating	Pin Function on the Module
1	GND		First	Logic Ground	170	GND		First	Logic Ground
2	PWR	Carrier	First	Payload Power	169	TDI	Carrier	Second	JTAG Test Data Input
3	PS1#	Module	Last	Presence 1	168	TDO	Module	Second	JTAG Test Data Output
4	MP	Carrier	First	Management Power	167	TRST#	Carrier	Second	JTAG Test Reset Input
5	GA0	Carrier	Second	Geographic Addr. 0	166	TMS	Carrier	Second	JTAG Test Mode Select In
6	RSRV D6		Second	Reserved, n/a	165	TCK	Carrier	Second	JTAG Test Clock Input
7	GND		First	Logic Ground	164	GND		First	Logic Ground
8	RSRV D8		Second	Reserved, n/a	163	Tx20+		Third	Port 20 Tx +
9	PWR	Carrier	First	Payload Power	162	Tx20-		Third	Port 20 Tx -
10	GND		First	Logic Ground	161	GND		First	Logic Ground
11	Tx0+		Third	Port 0 Tx +	160	Rx20+		Third	Port 20
12	Tx0-		Third	Port 0 Tx -	159	Rx20-		Third	Port 20 Rx -
13	GND		First	Logic Ground	158	GND		First	Logic Ground
14	Rx0+		Third	Port 0 Rx +	157	Tx19+		Third	Port 19 Tx +
15	Rx0-		Third	Port 0 Rx -	156	Tx19-		Third	Port 19 Tx -
16	GND		First	Logic Ground	155	GND		First	Logic Ground
17	GA1	Carrier	Second	Geographic Addr. 1	154	Rx19+		Third	Port 19 Rx +
18	PWR	Carrier	First	Payload Power	153	Rx19-		Third	Port 19 Rx -
19	GND		First	Logic Ground	152	GND		First	Logic Ground
20	Tx1+		Third	Port 1 Tx +	151	Tx18+		Third	Port 18 Tx +
21	Tx1-		Third	Port 1 Tx -	150	Tx18-		Third	Port 18 Tx -
22	GND		First	Logic Ground	149	GND		First	Logic Ground
23	Rx1+		Third	Port 1 Rx +	148	Rx18+		Third	Port 18 Rx +
24	Rx1-		Third	Port 1 Rx -	147	Rx18-		Third	Port 18 Rx -
25	GND		First	Logic Ground	146	GND		First	Logic Ground

Table 4-8: AMC131 Connector Pinout (Continued)

Pin	Signal	Driven By	Mating	Pin Function on the Module	Pin	Signal	Driven By	Mating	Pin Function on the Module
26	GA2	Carrier	Second	Geographic Addr. 2	145	Tx17+		Third	Port 17 Tx +
27	PWR	Carrier	First	Payload Power	144	Tx17-		Third	Port 17 Tx -
28	GND		First	Logic Ground	143	GND		First	Logic Ground
29	Tx2+		Third	Port 2 Tx +	142	Rx17+		Third	Port 17 Rx +
30	Tx2-		Third	Port 2 Tx -	141	Rx17-		Third	Port 17 Rx -
31	GND		First	Logic Ground	140	GND		First	Logic Ground
32	Rx2+		Third	Port 2 Rx +	139	Tx16+		Third	Port 16 Tx +
33	Rx2-		Third	Port 2 Rx -	138	Tx16-		Third	Port 16 Tx -
34	GND		First	Logic Ground	137	GND		First	Logic Ground
35	Tx3+		Third	Port 3 Tx +	136	Rx16+		Third	Port 16 Rx +
36	Tx3-		Third	Port 3 Tx -	135	Rx16-		Third	Port 16 Rx -
37	GND		First	Logic Ground	134	GND		First	Logic Ground
38	Rx3+		Third	Port 3 Rx +	133	Tx15+		Third	Port 15 Tx +
39	Rx3-		Third	Port 3 Rx -	132	Tx15-		Third	Port 15 Tx -
40	GND		First	Logic Ground	131	GND		First	Logic Ground
41	ENABLE#	Carrier	Second	AMC Enable	130	Rx15+		Third	Port 15 Rx +
42	PWR	Carrier	First	Payload Power	129	Rx15-		Third	Port 15 Rx -
43	GND		First	Logic Ground	128	GND		First	Logic Ground
44	Tx4+		Third	Port 4 Tx +	127	Tx14+		Third	Port 14 Tx +
45	Tx4-		Third	Port 4 Tx -	126	Tx14-		Third	Port 14 Tx -
46	GND		First	Logic Ground	125	GND		First	Logic Ground
47	Rx4+		Third	Port 4 Rx +	124	Rx14+		Third	Port 14 Rx +
48	Rx4-		Third	Port 4 Rx -	123	Rx14-		Third	Port 14 Rx -
49	GND		First	Logic Ground	122	GND		First	Logic Ground
50	Tx5+		Third	Port 5 Tx +	121	Tx13+		Third	Port 13 Tx +
51	Tx5-		Third	Port 5 Tx -	120	Tx13-		Third	Port 13 Tx -
52	GND		First	Logic Ground	119	GND		First	Logic Ground
53	Rx5+		Third	Port 5 Rx +	118	Rx13+		Third	Port 13 Rx +
54	Rx5-		Third	Port 5 Rx -	117	Rx13-		Third	Port 13 Rx -
55	GND		First	Logic Ground	116	GND		First	Logic Ground
56	SCL_LIPMI Agent		Second	IPMB-L Clock	115	Tx12+		Third	Port 12 Tx +
57	PWR	Carrier	First	Payload Power	114	Tx12-		Third	Port 12 Tx -
58	GND		First	Logic Ground	113	GND		First	Logic Ground
59	Tx6+		Third	Port 6 Tx +	112	Rx12+		Third	Port 12 Rx +
60	Tx6-		Third	Port 6 Tx -	111	Rx12-		Third	Port 12 Rx -
61	GND		First	Logic Ground	110	GND		First	Logic Ground
62	Rx6+		Third	Port 6 Rx +	109	Tx11+		Third	Port 11 Tx +
63	Rx6-		Third	Port 6 Rx -	108	Tx11-		Third	Port 11 Tx -
64	GND		First	Logic Ground	107	GND		First	Logic Ground
65	Tx7+		Third	Port 7 Tx +	106	Rx11+		Third	Port 11 Rx +
66	Tx7-		Third	Port 7 Tx -	105	Rx11-		Third	Port 11 Rx -

Table 4-8: AMC131 Connector Pinout (Continued)

Pin	Signal	Driven By	Mating	Pin Function on the Module	Pin	Signal	Driven By	Mating	Pin Function on the Module
67	GND		First	Logic Ground	104	GND		First	Logic Ground
68	Rx7+		Third	Port 7 Rx +	103	Tx10+		Third	Port 10 Tx +
69	Rx7-		Third	Port 7 Rx -	102	Tx10-		Third	Port 10 Tx -
70	GND		First	Logic Ground	101	GND		First	Logic Ground
71	SDA LIPMI Agent		Second	IPMB-L Data	100	Rx10+		Third	Port 10 Rx +
72	PWR Carrier		First	Payload Power	99	Rx10-		Third	Port 10 Rx -
73	GND		First	Logic Ground	98	GND		First	Logic Ground
74	CLK1 +		Third	Sync Clock 1+	97	Tx9+		Third	Port 9 Tx +
75	CLK1-		Third	Sync Clock 1-	96	Tx9-		Third	Port 9 Tx -
76	GND		First	Logic Ground	95	GND		First	Logic Ground
77	CLK2 +		Third	Sync Clock 2+	94	Rx9+		Third	Rx +
78	CLK2-		Third	Sync Clock 2-	93	Rx9-		Third	Port 9 Rx -
79	GND		First	Logic Ground	92	GND		First	Logic Ground
80	CLK3 +		Third	Sync Clock 3+	91	Tx8+		Third	Port 8 Tx +
81	CLK3-		Third	Sync Clock 3-	90	Tx8-		Third	Port 8 Tx -
82	GND		First	Logic Ground	89	GND		First	Logic Ground
83	PS0#	Carrier	Last	Presence 0	88	Rx8+		Third	Port 8 Rx +
84	PWR	Carrier	First	Payload Power	87	Rx8-		Third	Port 8 Rx -
85	GND		First	Logic Ground	86	GND		First	Logic Ground

Note: *Shaded areas in table denote pins not used by AMC131*

Chapter 5

System Monitoring and Alarms

Introduction

The AMC131 performs system monitoring and alarming functions using the flexible, industry standard, Intelligent Platform Management Interface (IPMI). The AMC131 comes equipped with an on-board Module Management Controller (MMC) chip, IPMI and IPMB J-connector pinouts, and MMC firmware already installed on the board. The MMC firmware is based on Pigeon Point System's (PPS) MMC firmware. Some of the functions available on this board through the IPMI interface include:

- Monitoring of the CPU and board temperatures with critical and non-critical alerting
- Monitoring of the voltage rails with critical and non-critical alerting
- Remote reset and shutdown of the board
- Monitoring of ejector switches for hot swap functionality: PT NexusWare IPMI driver and firmware provide features for hot swap
- Monitoring and event reporting of critical errors
- Interface to IPMB line (IPMB-L)

In order to take advantage of the features provided by the firmware, IPMI aware applications must be developed. Information on IPMI v1.5 is provided at:

http://www.intel.com/design/servers/ipmi/spec.htm

Key topics in this chapter include:

MMC Functions

The MMC performs system monitoring and alarming functions using the flexible, industry standard, Intelligent Platform Management Interface (IPMI). The module comes equipped with an on-board MMC and IPMI v1.5 firmware already installed on the module. The MMC firmware is based on Pigeon Point System®'s (PPS) MMC firmware. Some of the functions available on the module through the IPMI interface include:

- Monitoring of the CPU and board temperatures with critical and non-critical alerting
- Monitoring of the voltage rails with critical and non-critical alerting
- Remote reset and shutdown of the module (hard and soft)
- Monitoring of ejector switches for hot-swap functionality (Performance Technologies' NexusWare IPMI driver and firmware provide additional payload features for hot swap)
- Monitoring and event reporting of critical errors
- Fabric and clock e-keying
- Interface to local IPMB (IPMB-L)

In order to take advantage of the features provided by the firmware, IPMI-aware applications must be developed. Information on IPMI v1.5 is provided at:

http://www.intel.com/design/servers/ipmi/spec.htm

Summary of Supported Commands

Table 5-1, "IPMI/PICMG Command Subset Supported by the MMC Firmware," lists all the commands supported by the MMC.

The **Spec Ref** column indicates where in the relevant specification a command is defined. IPMI references are to v1.5 unless indicated otherwise. The **MMC Req** column indicates if a particular command is required by the relevant specification (*AMC Specification* or *HPM.1 Specification*) or is optional. See the various notes under the table for more information.

Table 5-1: IPMI/PICMG Command Subset Supported by the MMC Firmware

Command	Spec Ref	NetFn	CMD	MMC Req
IPM Device "Global" Commands				
Get Device ID	17.1	App	01h	Mandatory
Cold Reset	17.2	App	02h	Optional
Warm Reset	17.3	App	03h	Optional
Broadcast "Get Device ID"[a]	17.9	App	01h	Mandatory
Messaging Commands				
Set BMC Global Enables	18.1	App	2Eh	Mandatory
Get BMC Global Enables	18.2	App	2Fh	Mandatory
Clear Message Flags	18.3	App	30h	Mandatory
Get Message Flags	18.4	App	31h	Mandatory
Get Message	18.6	App	33h	Mandatory
Send Message	18.7	App	34h	Mandatory

Table 5-1: IPMI/PICMG Command Subset Supported by the MMC Firmware (Continued)

Command	Spec Ref	NetFn	CMD	MMC Req
BMC Watchdog Timer				
Reset Watchdog Timer	21.5	App	22h	Mandatory
Set Watchdog Timer	21.6	App	24h	Mandatory
Get Watchdog Timer	21.7	App	25h	Mandatory
Event Commands				
Set Event Receiver	23.1	S/E	00h	Mandatory
Get Event Receiver	23.2	S/E	01h	Mandatory
Platform Event (a.k.a. “Event Message”)	23.3	S/E	02h	Mandatory
Sensor Device Commands				
Get Device SDR Info	29.2	S/E	20h	Mandatory
Get Device SDR	29.3	S/E	21h	Mandatory
Reserve Device SDR Repository	29.4	S/E	22h	Mandatory
Get Sensor Reading Factors	29.5	S/E	23h	Optional
Set Sensor Hysteresis	29.6	S/E	24h	Optional
Get Sensor Hysteresis	29.7	S/E	25h	Optional
Set Sensor Threshold	29.8	S/E	26h	Optional
Get Sensor Threshold	29.9	S/E	27h	Optional
Set Sensor Event Enable	29.10	S/E	28h	Optional
Get Sensor Event Enable	29.11	S/E	29h	Optional
Get Sensor Event Status	29.13	S/E	2Bh	Optional
Get Sensor Reading	29.14	S/E	2Dh	Mandatory
FRU Device Commands				
Get FRU Inventory Area Info	28.1	Storage	10h	Mandatory
Read FRU Data	28.2	Storage	11h	Mandatory
Write FRU Data	28.3	Storage	12h	Mandatory
AdvancedTCA Commands				
Get PICMG Properties	3-10	PICMG	00h	Mandatory
FRU Control	3-25	PICMG	04h	Mandatory
FRU Control Capabilities	3-24	PICMG	1Eh	Mandatory
Get FRU LED Properties	3-27	PICMG	05h	Mandatory
Get LED Color Capabilities	3-28	PICMG	06h	Mandatory
Set FRU LED State	3-29	PICMG	07h	Mandatory
Get FRU LED State	3-30	PICMG	08h	Mandatory
Get Device Locator Record ID[b]	3-35	PICMG	0Dh	Mandatory
AMC Commands				
Set AMC Port State	3-26	PICMG	19h	Optional/ Mandatory
Get AMC Port State	3-27	PICMG	1Ah	Optional/ Mandatory
Set Clock State	3-44	PICMG	2Ch	Optional/ Mandatory
Get Clock State	3-45	PICMG	2Dh	Optional/ Mandatory

Table 5-1: IPMI/PICMG Command Subset Supported by the MMC Firmware (Continued)

Command	Spec Ref	NetFn	CMD	MMC Req
HPM.1 Upgrade Commands (HPM.1)				
Get Target Upgrade Capabilities	3-3	PICMG	2Eh	Mandatory
Get Component Properties	3-5	PICMG	2Fh	Mandatory
Abort Firmware Upgrade	3-15	PICMG	30h	Optional
Initiate Upgrade Action[c]	3-8	PICMG	31h	Optional/ Mandatory
Upload Firmware Block	3-9	PICMG	32h	Mandatory
Finish Firmware Upload	3-10	PICMG	33h	Mandatory
Activate Firmware	3-11	PICMG	35h	Mandatory
Query Self-Test Results[d]	3-12	PICMG	36h	Optional/ Mandatory
Query Rollback Status[e]	3-13	PICMG	37h	Optional/ Mandatory
Initiate Manual Rollback[f]	3-14	PICMG	38h	Optional/ Mandatory

a. See "Device ID" below, for the device ID data retrieved in response to a **(Broadcast) Get Device ID** command for this module.
b. See "Device Locator Record" below, for the IPMB management controller device locator record retrieved in response to a **Get Device Locator Record ID** command for this module.
c. The HPM.1 **Initiate Upgrade Action** command is mandatory for an IPM Controller indicating that any of its implemented components supports preparation for Firmware Upgrade or comparison of the current firmware
d. The HPM.1 **Query Self-test Results** command is mandatory for IPM Controllers indicating self-test is supported in the Self-test capabilities field of the "Get target upgrade capabilities" response or the Self-test capabilities field of the Upgrade Image header.
e. The HPM.1 **Query Rollback Status** command is mandatory for IPM Controllers supporting automatic or manual Rollback.
f. The HPM.1 **Manual Firmware Rollback** command is mandatory for IPM Controllers indicating manual firmware Rollback is supported in the *Manual firmware Rollback* capabilities field of the "Get target upgrade capabilities" response.

Device Locator Record

The MMC firmware supports the *Get Device Locator Record ID* command for FRU device #0 (the only FRU device represented by an MMC). The MMC firmware obtains the ID of the IPMB management controller device locator record by scanning the SDR records embedded into the firmware.

Table 5-2 shows an example of an IPMB management controller device locator record (SDR type 0x12) describing the properties of the MMC:

Table 5-2: IPMB Management Controller Device Locator Record

Parameter	Value
Power State Notification	
ACPI System Power State notification required	NO
ACPI Device Power State notification required	NO
Global Initialization	
Controller logs Initialization Agent errors	NO
Log Initialization Agent errors accessing this controller	NO
Event Generation	Enable event message generation from controller
Device Capabilities	
Chassis Device	NO
Bridge	NO
IPMB Event Generator	YES
IPMB Event Receiver	NO
FRU Inventory Device	YES
SEL Device	NO
SDR Repository Device	NO
Sensor Device	YES
FRU Entity ID	0xC1
Entity Instance	(slot dependent)
OEM-specific	0
Device ID String Type/Length	8-bit ASCII with size of Device ID String (see below)
Device ID String	AMC131

Device ID

The MMC firmware provides the following device ID data in response to the (Broadcast) *Get Device ID* command:

Table 5-3: MMC Device ID

Parameter	Value
Device ID	0x00
Provides Device SDRs	YES
Device Revision Number	0x00
Device Available	YES
Firmware Revision	Changes with each release
IPMI Version	1.5
Additional Device Support	
Chassis Device	NO
Bridge	NO
IPMB Event Generator	YES
IPMB Event Receiver	NO
FRU Inventory Device	YES
SEL Device	NO
SDR Repository Device	NO
Sensor Device	YES
Manufacturer ID	0x000614
Product ID	0x0007
Auxiliary Firmware Revision Information	0x00000000

Sensors

Table 5-4 lists the sensors that are monitored by the MMC. Note that the sensor IDs are local to the MMC. The MMC's SDRs are inherited by the next level of management (MicroTCA MCMC or AMC carrier IPMC) and sensor IDs are reassigned.

Table 5-4: MMC Sensors

Sensor ID	Description	Lower Non-Recoverable Threshold	Lower Critical Threshold	Lower Non-Critical Threshold	Upper Non-Critical Threshold	Upper Critical Threshold	Upper Non-Recoverable Threshold
0	Hot Swap	N/A	N/A	N/A	N/A	N/A	N/A
1	3.3V MGMT	3.0V	3.068V	3.135V	3.465V	3.533V	3.6V
2	1.2V PHY1 AVDDL	1.13V	1.14V	1.15V	1.3V	1.31V	1.32V
3	1.05V/1.1V SVDD[1]	0.988V/ 1.05	1.0V/ 1.06	1.01V/ 1.07	1.09V/ 1.13	1.1V/ 1.14	1.112V/ 1.15
4	5.1V USB	N/A	N/A	4.375V	5.525V	N/A	N/A
5	2.5V PHY1	2.365V	2.38V	2.39V	2.61V	2.62V	2.635V
6	0.9V DDR2 VTT	0.791V	0.8V	0.81V	0.99V	1.0V	1.009V
7	12V	10.0V	10.4V	10.8V	13.2V	13.6V	14.0V
8	3.3V	2.96V	2.97V	2.98V	3.62V	3.63V	3.64V
9	1.8V DDR2 VDDQ	1.7V	1.71V	1.72V	1.88V	1.89V	1.9V
10	1.05V/1.1V Core[1]	0.988V/ 1.05	1.0V/ 1.06	1.01V/ 1.07	1.09V/ 1.13	1.1V/ 1.14	1.112V/ 1.15
11	1.05V/1.1V AVDD[1]	0.988V/ 1.05	1.0V/ 1.06	1.01V/ 1.07	1.09V/ 1.13	1.1V/ 1.14	1.112V/ 1.15
12	1.2V PHY1 DVDDL	1.13V	1.14V	1.15V	1.3V	1.31V	1.32V
13	5V REG1.8	4.49V	4.5V	4.51V	5.49V	5.5V	5.51V
14	BMC Watchdog	N/A	N/A	N/A	N/A	N/A	N/A
15	Version Change	N/A	N/A	N/A	N/A	N/A	N/A
16	CPU TEMP	-5 C	0 C	5 C	95 C	100 C	105 C
17	INLET TEMP	-5 C	0 C	5 C	60 C	70 C	N/A

[1] The 1.5 GHz option operates with 1.1V core voltage.

Interpreting Sensor Events

The ATCA specification includes the following definitions for the sensor event severity levels:

- IPMI non-critical / PICMG 3.0 minor / telco minor – a warning that things are somewhat out of normal range, but not really a “problem” yet. See “Non-Critical Events” below.
- IPMI critical / PICMG 3.0 major / telco major – things are still in valid operating range, but are getting close to the edge; unit still operating within vendor-specified tolerances. See “Critical Events” below.
- IPMI non-recoverable / PICMG 3.0 critical / telco critical – unit no longer operating within vendor-specified tolerances. See “Non-Recoverable Events” below.

Non-Critical Events

Non-critical events are informative only. They do not indicate that the module is outside of its operating limits. In general, no action is required. However, in certain contexts, system or shelf management software may decide that preventive action should be taken. For example, if several modules in a shelf report upper non-critical temperature events, the shelf manager might decide to increase fan speed.

Critical Events

Critical events indicate that the module is still within its operating limits, but it is close to exceeding one of those limits. Possible action in this case is to closely monitor the alarming sensor and take more aggressive action if it approaches the non-recoverable threshold.

Non-Recoverable Events

Non-recoverable events indicate that the module may no longer be functioning because it is now outside of its operating limits. It is likely that action is required or has already been taken by the local hardware/firmware. For example, a processor may have shut itself down because its maximum die temperature was exceeded, or a shelf manager may decide to deactivate the module because the processor is too hot.

Serial Interface Subsystem

The MMC firmware implements a communication protocol over the payload and/or serial debug interfaces. The communication is in the form of formatted ASCII strings.

The Serial Interface Protocol Lite (SIPL) is based on the IPMI-defined Terminal Mode of the serial/modem interface. The following sections describe the SIPL:

Terminal Mode Messages and Commands

Terminal Mode Message Format

Terminal Mode messages have the following format:

```
[<message data>]<newline>
```

The left bracket and the right bracket plus `<newline>` characters serve as START and STOP delimiters for a message. The MMC does not support multi-line IPMI messages.

Raw IPMI Messages

The SIPL supports raw IPMI messages that are entered as sequences of case-insensitive hex-ASCII pairs, each pair optionally separated from the previous one with a single `<space>` character. What follows are examples of raw IPMI request messages in Terminal Mode:

```
[18 00 22]<newline>
[180022]<newline>]
```

The MMC handles raw IPMI messages in the same way as it handles IPMI/PICMG/AMC messages coming from the IPMB-L bus and, with the exception that IPMI/PICMG/AMC replies are routed to the interfaces from which the respective requests have come (i.e. either the serial debug or payload interface of the MMC).

Terminal Mode Text Commands

The SIPL does not support Terminal Mode ASCII text commands defined by the *IPMI Specification (section 13.7.8)*.

Pigeon Point Systems (PPS) Extension Commands

The MMC firmware supports a set of PPS extension commands that are used to control and monitor the carrier Intelligent Platform Management Controller (IPMC) state over the serial debug interface. These commands are used to read the MMC status, implement graceful payload shutdown, etc.

The PPS extension commands are implemented as OEM IPMI commands with network function codes 2Eh/2Fh and message body transferred in the same manner as for raw IPMI messages (see “Raw IPMI Messages,” on page 61). Figure 5-1, “PPS Extension Command Request,” shows an example of a PPS extension command request:

Figure 5-1: PPS Extension Command Request

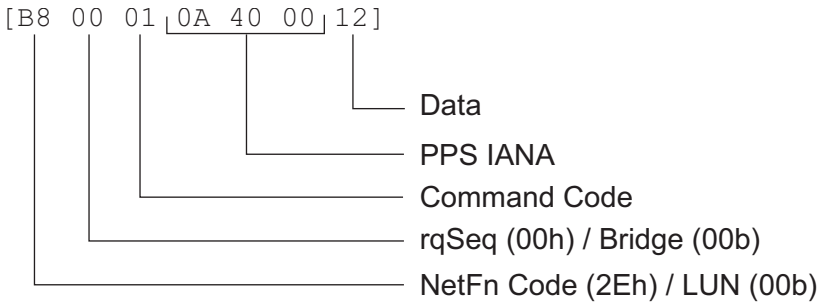
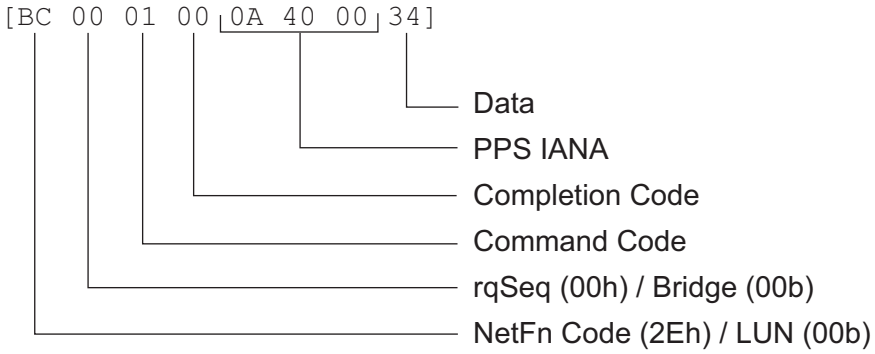


Figure 5-2, “PPS Extension Command Response,” shows an example of a PPS extension command response:

Figure 5-2: PPS Extension Command Response



Terminal Mode Line Editing

The MMC does not support input line editing functionality defined as optional in the *IPMI Specification (section 13.8)*.

Supported PPS Extension Commands

The MMC firmware supports the following PPS extension commands (see "Pigeon Point Systems (PPS) Extension Commands," on page 62):

Table 5-5: PPS Extension Commands Supported by the MMC

Command Request/Response	Code	Likely Command Source(s)	Description	See Also
Get Status	0x00	Serial debug and payload interfaces	Read the MMC status	Get Status Command
Get Serial Interface Properties	0x01	Serial debug and payload interfaces	Get the properties of a serial interface	Serial Line Properties Commands
Set Serial Interface Properties	0x02	Serial debug and payload interfaces	Set the properties of a serial interface	
Get Debug Level	0x03	Serial debug interface	Get debug/verbosity level	Debug/Verbosity Level
Set Debug Level	0x04	Serial debug interface	Set debug/verbosity level	
Get Payload Communication Timeout	0x09	Serial debug and payload interfaces	Get the timeout for payload communications	Payload Communication Timeout
Set Payload Communication Timeout	0x0A	Serial debug and payload interfaces	Set the timeout for payload communications	
Graceful Reset	0x11	Payload interface	The payload is ready to be shut down/reset	Graceful Payload Reset
Diagnostic Interrupt Results	0x12	Payload interface	Return diagnostic interrupt results	Payload Diagnostic Interrupt
Get Payload Shutdown Timeout	0x15	Serial debug and payload interfaces	Get the timeout for payload shutdown	Payload Shutdown Timeout
Set Payload Shutdown Timeout	0x16	Serial debug and payload interfaces	Set the timeout for payload shutdown	
Get Geographic Address	0x2C	Serial debug and payload interfaces	Get the geographic address	Get Geographic Address Command

The MMC accepts all PPS extension commands listed in Table 5-5 from both serial interfaces, as well as IPMB-L. This is done to achieve additional flexibility and extensibility in the MMC functionality.

The PPS extension commands listed in Table 5-5 are referred to as the SIPL commands throughout this document. The following sections discuss the SIPL commands in more detail.

Get Status Command

The IPMC status is four bytes describing the logical state of the IPMC and the payload. Table 5-6, “IPMC Status Bits,” provides a description of the IPMC status bits:

Table 5-6: IPMC Status Bits

Bit	Name	Description
Byte 1		
0 (LSB)	Control	If set to 0, the IPMC control over the payload is disabled.
1-2	NA	Reserved
3	Sensor Alert	If set to 1, indicates that at least one of the IPMC sensors detects threshold crossing.
4	Reset Alert	If set to 1, indicates that the payload is going to be reset.
5	Shutdown Alert	If set to 1, indicates that the payload is going to be shut down.
6	Diagnostic Interrupt Request	If set to 1, indicates that a payload diagnostic interrupt request has arrived.
7 (MSB)	Graceful Reboot Request	If set to 1, indicates that the payload is requested to initiate the graceful reboot sequence.
Byte 2		
0-7	NA	Reserved
Byte 3		
0-7	NA	Reserved
Byte 4		
0-3	NA	Reserved
4	Message Received	If set to 1, indicates that a message for the payload has been received.
5-7	NA	Reserved

The IPMC firmware notifies the payload about changes of all status bits except for bits 0-2 of byte 1 by sending an unprintable character (ASCII 07, BELL) over the payload interface. The payload is expected to use the **Get Status** command to identify pending events and other SIPL commands to provide a response (if necessary). The event notification character is sent in a synchronous manner, and does not appear in the contents of SIPL messages sent to the payload.

The **Get Status** command has the following synopsis:

```
[B8 xx 00 0A 40 00]
```

The IPMC responds to the **Get Status** command with the following reply:

```
[BC xx 00 00 0A 40 00 <byte1> <byte2> <byte3> <byte4>]
```

Serial Line Properties Commands

The SIPL provides commands to get/set the properties of the MMC serial interfaces (the serial debug interface and the payload interface):

- "Get Serial Interface Properties Command," on page 65
- "Set Serial Interface Properties Command," on page 65

Get Serial Interface Properties Command

The **Get Serial Interface Properties** command is used to get the properties of a particular serial interface. This command has the following synopsis:

`[B8 xx 01 0A 40 00 <interface ID>]`

The `<interface ID>` parameter can have one of the values shown in Table 5-7, "The <interface ID> Parameter Values," below.

Table 5-7: The `<interface ID>` Parameter Values

Interface ID	Description
0	Serial debug interface
1	Payload interface

The MMC responds to the **Get Serial Interface Properties** command with the following reply:

`[BC xx 01 00 0A 40 00 <interface properties>]`

The `<interface properties>` parameter has the bit fields shown in Table 5-8, "The <interface properties> Parameter Bit Fields," below.

Table 5-8: The `<interface properties>` Parameter Bit Fields

Bits	Name	Description
0-3	Baud Rate ID	The baud rate ID defines the interface baud rate as follows: 0 – 9600 bps 1 – 19200 bps 2 – 38400 bps 3 – 57600 bps 4 – 115200 bps
4-6	NA	Reserved
7 (MSB)	Echo On	If this bit is set, the MMC enables echo for the given serial interface.

Set Serial Interface Properties Command

The **Set Serial Interface Properties** command is used to change the properties of a given interface:

`[B8 xx 02 0A 40 00 <interface ID> <interface properties>]`

Debug/Verbosity Level

The SIPL provides commands to enable and disable output of error/diagnostic messages to the serial debug interface at runtime:

- "Get Debug Level Command," on page 66
- "Set Debug Level Command," on page 66

Get Debug Level Command

To get the current debug level, the **Get Debug Level** command must be used. This command has the following synopsis:

`[B8 xx 03 0A 40 00]`

The MMC responds to the **Get Debug Level** command with the following reply:

`[BC xx 03 00 0A 40 00 <debug level>]`

The `<debug level>` parameter contains the bit fields shown in Table 5-9, "MMC Debug Levels," below.

Table 5-9: MMC Debug Levels

Bit	Name	Description
0 (LSB)	Error Logging Enable	If set to 1, the MMC outputs error/diagnostic messages onto the serial debug interface.
1	Low-level Error Logging Enable	If set to 1, the MMC outputs low-level error/diagnostic messages onto the serial debug interface.
2	Alert Logging Enable	If set to 1, the MMC outputs important alert messages onto the serial debug interface.
3	Payload Logging Enable	If set to 1, the MMC provides a trace of SIPL activity on the payload interface onto the serial debug interface.
4	IPMB Dump Enable	If set to 1, the MMC provides a trace of IPMB messages that are arriving to/going from the MMC via IPMB-L.
5-7	NA	Reserved

Set Debug Level Command

To change the current debug level, the **Set Debug Level** command must be used. This command has the following synopsis:

`[B8 xx 04 0A 40 00 <debug level>]`

Payload Communication Timeout

Some of the SIPL commands are subject to payload communication timeouts. If the payload does not respond with a correct reply within a definite period of time, the MMC assumes that a payload communication timeout occurred and acts accordingly. The SIPL timeout value also limits the period of time given to the payload to prepare for a payload reset.

- "Get Payload Communication Timeout Command," on page 67
- "Set Payload Communication Timeout Command," on page 67

Get Payload Communication Timeout Command

The MMC supports reading of the payload communication timeout using the **Get Payload Communication Timeout** command. This command has the following synopsis:

`[B8 xx 09 0A 40 00]`

The MMC responds to the **Get Payload Communication Timeout** command with the following reply:

`[BC xx 09 00 0A 40 00 <payload timeout>]`

The `<payload timeout>` parameter is the payload communication timeout measured in hundreds of milliseconds. Thus, the payload communication timeout may vary from 0.1 to 25.5 seconds. The default value of the payload communication timeout is specified by the CFG_APP_SIPL_PAYLOAD_TIMEOUT Configuration Parameter.

Set Payload Communication Timeout Command

To change the payload communication timeout, the **Set Payload Communication Timeout** command is used:

`[B8 xx 0A 0A 40 00 <payload timeout>]`

Graceful Payload Reset

The MMC supports the Graceful Reboot option of the **FRU Control** command. On receiving such a command, the MMC sets the Graceful Reboot Request bit of the MMC status, sends a status update notification to the payload, and waits for the **Graceful Reset** command from the payload. If the MMC receives such a command before the payload communication timeout time, it sends the 0x00 completion code (Success) to the carrier controller. Otherwise, the 0xC3 completion code (Timeout) is sent.

The **Graceful Reset** command has the following synopsis:

`[B8 xx 11 0A 40 00]`

Note that the MMC does not reset the payload on receiving the **Graceful Reset** command or timeout. If the MMC participation is necessary, the payload must request the MMC to perform a payload reset.

The **Graceful Reset** command is also used to notify the MMC about the completion of the payload shutdown sequence (refer to "Payload Shutdown Timeout," on page 68).

Payload Diagnostic Interrupt

The MMC supports the Issue Diagnostic Interrupt feature of the **FRU Control** command. The payload is notified about a diagnostic interrupt over the SIPL as described in “Get Status Command,” on page 64. The payload is expected to return diagnostic interrupt results before the payload communication timeout using the **Diagnostic Interrupt Results** command of the SIPL. This command has the following synopsis:

`[B8 xx 12 0A 40 00 <diagnostic interrupt return code>]`

If the payload responds before the payload communication timeout, the diagnostic interrupt return code is forwarded to the carrier controller as the completion code of the **FRU Control** command response. Otherwise, the 0xC3 completion code (Timeout) is returned.

Payload Shutdown Timeout

When the carrier controller commands the MMC to shut down the payload (i.e. sends the FRU Control (Quiesce) command), the MMC notifies the payload about it by asserting appropriate alert and sending an alert notification to the payload (refer to “Get Status Command,” on page 64). Upon receiving this notification, the payload software is expected to initiate the payload shutdown sequence. After performing this sequence, the payload should send the **Graceful Reset** command (refer to “Graceful Payload Reset,” on page 67) to the MMC over the payload interface to notify the MMC that the payload shutdown is complete.

To avoid deadlocks that may occur if the payload software does not respond, the MMC provides a special timeout for the payload shutdown sequence. If the payload does not send the Graceful Reset command within a definite period of time, the MMC assumes that the payload shutdown sequence is finished, and sends a Module Quiesced hot-swap event to the carrier controller.

Get Payload Shutdown Timeout Command

The MMC supports reading of the payload shutdown timeout using the **Get Payload Shutdown Timeout** command. This command has the following synopsis:

`[B8 xx 15 0A 40 00]`

The MMC responds to the **Get Payload Shutdown Timeout** command with the following reply:

`[BC xx 15 00 0A 40 00 <LSB byte of timeout> <MSB byte of timeout>]`

The payload shutdown timeout is measured in hundreds of milliseconds and stored as a 2-byte integer. The default value of the payload shutdown timeout is specified by a dedicated Configuration Parameter.

Set Payload Shutdown Timeout Command

To change the payload shutdown timeout, the **Set Payload Shutdown Timeout** command is used:

`[B8 xx 16 0A 40 00 <LSB byte of timeout> <MSB byte of timeout>]`

Get Geographic Address Command

The MMC allows reading the geographic address of the module using the **Get Geographic Address** command, which has the following synopsis:

`[B8 xx 2C 0A 40 00]`

The MMC responds to the **Get IPMB Address** command with the following reply:

`[BC xx 2C 00 0A 40 00 <geographic address>]`

The `<geographic address>` parameter has the bit fields shown in Table 5-10, “The <geographic address> Parameter Bit Fields,” below.

Table 5-10: The <geographic address> Parameter Bit Fields

Bits	Name	Description
0-1	GA0 Signal	0 = GA0 is grounded 1 = GA0 is unconnected 3 = GA0 is pulled up
2-3	GA1 Signal	0 = GA1 is grounded 1 = GA1 is unconnected 3 = GA1 is pulled up
4-5	GA2 Signal	0 = GA2 is grounded 1 = GA2 is unconnected 3 = GA2 is pulled up
6-7	NA	Reserved

Firmware Upgrade Process

The MMC firmware supports a reliable field upgrade procedure compatible with the *HPM.1 Specification*. The key features of the firmware upgrade procedures are as follows:

- The upgrade can be performed over the serial debug/payload interface or over IPMB-L.
- The upgrade procedure is performed while the MMC firmware is online and operating normally.
- Upgrades of the firmware component are reliable. A failure in the download (error or interruption) does not disturb the MMC's ability to continue using the “old” firmware or its ability to restart the download process. Upgrades of the boot loader component are not reliable and may render the MMC non-functional in case of an incomplete upgrade.
- Upgrades of the firmware component are reversible. The MMC firmware automatically reverts back to the previous firmware if there is a problem when first running the new code and can be reverted manually using the HPM.1-defined **Manual Rollback** command. Upgrades of the boot loader component are not reversible.

HPM.1 Boot Loader

- The HPM.1 boot loader does not perform any upgrade actions
- The HPM.1 boot loader is able to boot either of two redundant copies of the MMC firmware in flash
- The HPM.1 boot loader is able to automatically rollback a failed copy of the MMC firmware and activate the backup one
- The HPM.1 boot loader can be upgraded in-field as an HPM.1-upgradeable component

HPM.1 Firmware Upgrade

The HPM.1 upgrade procedure is managed by a utility called the *upgrade agent*. The `ipmitool` utility is used as upgrade agent for upgrading the MMC firmware.

The upgrade agent communicates with the MMC firmware via serial interface or IPMB-L, and uses the ATCA commands that are described in the *HPM.1 Specification* for upgrading the firmware. Updated firmware is packed into a special image that has a format described in the *HPM.1 Specification*. That image is used by the upgrade agent to prepare and upgrade the MMC firmware. The HPM.1 upgrade procedure includes the following steps:

1. **Preparation step**. This step erases the region in the flash memory where a component will be written.
2. **Component upload step**. This step is designed to upload the component image via IPMB or a serial interface, and write it into the flash memory.
3. **Component activation step**. This step is designed to activate the previously upgraded component; for the firmware component, this step can be deferred until a later time.

The MMC firmware supports two upgradeable components: the firmware itself and the boot loader. In case of an unsuccessful firmware upgrade it is possible to roll back to the old firmware. This is not true for the boot loader.

Note: *Extreme caution should be exercised when upgrading the boot loader. There is no backup copy of the boot loader and if for any reason the boot loader upgrade procedure fails, the firmware becomes non-functional after reboot and must be reprogrammed over JTAG.*

Upgrade Utilities

The firmware upgrade procedure is performed using the upgrade agent utility, implementing the HPM.1 Upgrade Protocol and capable of programming custom firmware images into the flash memory of the MMC over a serial interface or IPMB-L. Any HPM.1-compatible Upgrade Agent can be used to upgrade the MMC firmware. It is recommended to use the `ipmitool` utility for these purposes. The `ipmitool` utility is available from PT. Contact PT Customer Support and Services for contact information.

The firmware image is supplied to the `ipmitool` utility in a single file called an HPM.1 upgrade image (for information about the format of HPM.1 upgrade images refer to the HPM.1 specification).

Detailed HPM.1 Upgrade Procedure

The following images are available from PT:

- `hpm1fw.img` - this image contains the MMC firmware
- `hpm1boot.img` - this image contains the boot loader
- `hpm1all.img` - this image contains both the firmware and the boot loader

These images can be used to upgrade corresponding components of the IPMC: the firmware, the boot loader or both.

The following snapshot samples a command performing firmware upgrade from a Linux host over LAN/IPMB:

```
ipmitool -I lan -H 192.168.0.2 -A none -T 0x82 -B 0 -t 0x7c -b 7 hpm
upgrade hpm1fw.img activate
PICMG HPM.1 Upgrade Agent 1.0:
Validating firmware image integrity...OK
Performing preparation stage...
    Services may be affected during upgrade. Do you with to continue?
y/n y...
OK
    Target Product ID : 15
    Target Manufacturer ID: 1556
Performing upgrade stage:
    Upgrading AVR-AMCm F/W
    with Version:    Major: 1
                     Minor: 70
                     Aux: 000 000 000 000
    Writing firmware: 100 % completed
Performing activation stage:
Firmware upgrade procedure successful
```

IPMI Communication Utility (ipmitool)

The `ipmitool` utility is a Linux application that can be used for a wide range of tasks involving IPMI-based communications. The following topics describe the installation process and provide information on specific applications of this utility.

Note: *Contact PT Customer Support and Services for an enhanced version of* `ipmitool`*. Besides the standard functionality, it supports the following vendor-specific enhancements, which are not available in the official release (as of version 1.8.9):*

- *Support for the serial IPMI interface (Terminal Mode)*
- *Some improvements in HPM.1 upgrade protocol implementation.*
- *Support for double bridging via LAN for accessing MMCs through the Shelf Manager and carrier IPMC.*

The enhanced version is available in binary form for Windows and in source form for Linux.

Building the ipmitool Utility

Build and install the `ipmitool` utility on a Linux host system using the following procedure:

1. Unpack the source tarball obtained from the secure Web site and change to the `ipmitool` directory:

   ```
   bash$ tar xzf <ipmitool_package_name>
   bash$ cd ipmitool
   ```

2. Run the `configure` script to prepare for the build. The `--prefix=<dir>` option can be used to specify the directory where the resulting files are installed. If not specified, `/usr/local` is used (in this case, the installation requires root privileges).

   ```
   bash$ ./configure --prefix=/home/user/ipmitool
   ```

3. Run the **`make install`** command to build and install the `ipmitool` utility.

   ```
   bash$ make install
   ```

Accessing an MMC with ipmitool

The available access methods that can be used to communicate with the MMC depend on the MMC firmware configuration and overall system setup. The most frequently used access methods are the following:

- Via an Ethernet connection to a Shelf Manager that is able to access via IPMB-0 the carrier IPMC managing the MMC. See “Accessing an MMC via a Shelf Manager,” on page 73.

  This access method can be used from any Linux or Windows host that has an Ethernet connection to the Shelf Manager of the shelf in which the MMC is installed. In this access method, the `ipmitool` utility uses an Ethernet connection to the Shelf Manager to double bridge IPMI requests to the MMC over IPMB-0 and IPMB-L.
- Via the serial debug or serial payload interface of the MMC. See “Accessing an MMC via a Serial Interface,” on page 74.

  This access method can be used from any Linux or Windows host that has a serial connection with the MMC’s serial debug or serial payload interfaces. In this access method, the `ipmitool` utility uses a serial interface to directly access the MMC.

Accessing an MMC via a Shelf Manager

To access the MMC using an Ethernet connection to a Shelf Manager, the following parameters should be specified in the command line of the `ipmitool` utility:

`-I lan`

This command line parameter instructs the `ipmitool` utility to use Ethernet for communications with the MMC.

`-H <Shelf Manager IP>`

This command line parameter specifies the IP address of the Shelf Manager.

`-T <carrier IPMC address>`

This command line parameter specifies the remote transit address (IPMB-0 address of the carrier IPMC) to which requests should be bridged by the Shelf Manager.

`-B 0`

This command line parameter specifies the remote transit channel (with 0 designating IPMB-0) to which requests should be bridged by the Shelf Manager.

`-t <MMC address>`

This command line parameter specifies the remote target address (IPMB-L address of the MMC) to which requests should be bridged by the carrier IPMC.

`-b 7`

This command line parameter specifies the remote target channel (with 7 designating IPMB-L) to which requests should be bridged by the carrier IPMC.

`-A <authtype>`

This command line parameter forces the `ipmitool` to use a specific authentication type, which must, of course, be supported by the Shelf Manager.

For example, to fetch and print Sensor Device Records of an MMC at IPMB-L address `0x72` via a Shelf Manager with the IP address `192.168.0.2`, and a carrier IPMC at IPMB-0 address `0x82`, the following command line should be used:

```
# ipmitool –I lan –H 192.168.0.2 –T 0x82 –B 0 –t 0x72 –b 7 –A none sdr
```

Accessing an MMC via a Serial Interface

The following `ipmitool` command line parameters are used for communicating with the MMC via a serial interface:

`-I serial-terminal`

This command line parameter instructs the `ipmitool` utility to use the serial interface for communications with the MMC.

`-D <dev[:baudrate]>`

This command line parameter specifies the serial device and baud rate settings to use. For Linux hosts, the serial device is the system path to the device node (e.g. **/dev/ttyS0**). For the Cygwin-flavor of the `ipmitool` utility, Windows serial device names are translated as follows: the COM1 device name is mapped to **/dev/ttyS0**, COM2 is mapped to **/dev/ttyS1** and so on.

The supported baud rates are: 2400, 9600, 19200, 38400, 57600, and 115200.

For example, to fetch and print Sensor Device Records of an MMC via a serial interface connection with a baud rate of 9600, the following command line should be used:

```
# ipmitool –I serial-terminal –D /dev/ttyS0:9600 sdr
```

Using ipmitool for HPM.1 Upgrades

The `ipmitool` utility has built-in HPM.1 upgrade functionality and can be used as an upgrade agent. To be able to send HPM.1 commands to the MMC, the proper connection options should be specified in the `ipmitool` command line.

See “Accessing an MMC with ipmitool,” on page 72 for the list of available `ipmitool` command line connection options.

HPM.1 Commands

The `ipmitool` utility supports the following HPM.1 commands, which are described on the following pages:

targetcap

Get the target upgrade capabilities. This command can be used to find out the upgrade capabilities of an MMC.

```
ipmitool hpm targetcap
```

Example:

```
ipmitool -I lan -H 192.168.0.2 -A none -T 0x82 -B 0 -t 0x74 -b 7 hpm
targetcap
PICMG HPM.1 Upgrade Agent 1.0:
TARGET UPGRADE CAPABILITIES
------------------------------
HPM.1 version.............0
Component 0 presence....[y]
Component 1 presence....[y]
Component 2 presence....[n]
Component 3 presence....[n]
Component 4 presence....[n]
Component 5 presence....[n]
Component 6 presence....[n]
Component 7 presence....[n]
Upgrade undesirable.....[n]
Aut rollback override...[n]
IPMC degraded...........[n]
Defered[1] activation......[y]
Service affected........[y]
Manual rollback.........[y]
Automatic rollback......[y]
Self test...............[n]
Upgrade timeout.........[100 sec]
Self test timeout.......[0 sec]
Rollback timeout........[5 sec]
Inaccessibility timeout.[5 sec]
```

1. “Defered” is misspelled in the `ipmitool` utility.

compprop

Get the specified component properties. This command can be used to find out component-specific properties.

```
ipmitool hpm compprop <id> <select>
```

The <id> parameter specifies the component whose properties are read; 0 corresponds to the firmware component and 1 corresponds to the boot loader component. The <select> parameter specifies the property that should be acquired. The properties are the following:

0	General properties
1	Current firmware version
2	Description string
3	Rollback firmware version
4	Deferred firmware version

Example:

```
ipmitool -I lan -H 192.168.0.2 -A none -T 0x82 -B 0 -t 0x74 -b 7 hpm
compprop 0 0
PICMG HPM.1 Upgrade Agent 1.0:
GENERAL PROPERTIES
------------------------------
Payload cold reset req....[y]
Def. activation supported.[y]
Comparison supported......[n]
Preparation supported.....[y]
Rollback supported........[y]
```

upgrade

Upgrade the firmware with the specified image. This command can be used to upgrade the firmware using a valid HPM.1 image.

```
ipmitool hpm upgrade <file> [activate]
```

The `<file>` parameter specifies the name of the HPM.1 upgrade image. If the `[activate]` parameter is specified, the upgraded firmware is activated just after the upgrade procedure. In the other case, an additional command should be issued to activate the firmware.

Example:

```
ipmitool -I lan -H 192.168.0.2 -A none -T 0x82 -B 0 -t 0x74 -b 7 hpm
upgrade hpm1fw.img
Validating firmware image integrity...OK
Performing preparation stage...
    Services may be affected during upgrade. Do you wish to continue?
y/n y
OK
    Target Product ID     : 15
    Target Manufacturer ID: 1556
Performing upgrade stage:
    Upgrading AVR-AMCm F/W
    with Version: Major: 0
                  Minor: 5
                  Aux  : 000 000 000 000
    Writing firmware: 100 % completed
```

activate

Activate the newly uploaded firmware. This command can be used for activating the newly uploaded firmware if there was no activate parameter passed to the upgrade command.

```
ipmitool hpm activate
```

Example:

```
ipmitool -I lan -H 192.168.0.2 -A none -T 0x82 -B 0 -t 0x74 -b 7 hpm
activate
PICMG HPM.1 Upgrade Agent 1.0:
```

rollback

Perform a manual rollback on the IPM controller. This command can be used to roll back from the newly uploaded firmware to the old one.

```
ipmitool hpm rollback
```

Example:

```
ipmitool -I lan -H 192.168.0.2 -A none -T 0x82 -B 0 -t 0x74 -b 7 hpm
rollback
PICMG HPM.1 Upgrade Agent 1.0:
```

rollbackstatus

Query the rollback status. This command can be used to query the firmware on the IPMC about whether a rollback event has occurred.

```
ipmitool hpm rollbackstatus
```

Example:

```
ipmitool -I lan -H 192.168.0.2 -A none -T 0x82 -B 0 -t 0x74 -b 7 hpm
rollbackstatus
PICMG HPM.1 Upgrade Agent 1.0:
Rollback occured[1] on component mask: 0x01
```

1. “occured” is misspelled in the `ipmitool` utility.

This chapter provides you with information about the various reset types and sources supported by the AMC131.

Reset Types and Sources

The AMC131 supports the following reset types that affect the entire module:

- Power reset
- Hard reset
- Soft reset

Limited resets that do not affect the entire board are also supported.

Power Reset

Power reset occurs when the power-on reset device changes from a power-off to a power-on state. The power-off to power-on state occurs when either the payload power is turned on or when the *power good* indication from the DC-to-DC converters is false and returns true.

A power reset also generates a hard reset (see “Hard Reset,” on page 80).

The AMC131 is held in reset until the PCI Express reference clock (FCLKA) source is determined. The AMC.1 R2.0 specification requires that FCLKA is e-keyed. The AMC131 is shipped from the factory configured for AMC.1 R2.0 e-keying of FCLKA. AMC.1 R1.0 carriers do not e-key FCLKA. If the AMC131 does not come out of reset when powered on, it is likely that FCLKA is not configured properly for the specific carrier. Refer to “SW2-2: FCLKA Configuration - PCIe only” on page 39 for information on configuring FCLKA.

Hard Reset

Hard resets may originate from the following sources:

- MPC8641D-initiated system reset request
- Payload reset that can be initiated by:
  - MMC
  - Carrier board
- Front panel reset button
- Setting of custom bits

Payload Reset Initiated by the MMC

The MMC can drive a payload reset that results in a hard reset. The MMC circuitry has an independent power-on reset circuit that is controlled by MMC power. A brown-out reset occurs for the MMC when the MMC power drops below 2.7V. The MMC monitors the payload reset, but does not allow this reset to directly reset the MMC circuitry.

A Watchdog timer within the MMC can reset the MMC.

The MMC ICE interface can directly reset the MMC.

Payload Reset Initiated by the Carrier Board

The carrier can reset the MMC directly through the ENABLE_N signal.

Front Panel Reset Button

When the system reset button (see Figure 2-1, “AMC131 Front Panel,” on page 23) on the front panel is pressed, the AMC131 resets itself.

Soft Reset

Soft resets may be initiated through:

- The COP In-Circuit Emulator (ICE)
- The MPC8641D (limited to the processor)

Limited Resets

Limited resets include:

- Custom logic-initiated resets
- MMC resets
- Peripheral reset (DDR2 memory, Ethernet PHY, and USB Controller)

Caution:
Do not reset the DDR2. Attempting to reset this component results in module lockup. This reset is used for initialization purposes only.

Custom Logic-Initiated Resets

Custom logic includes reset bits that control reset to payload devices such as the on-board quad PHY.

MMC Resets

The MMC has independent reset sources that do not affect the entire module such as:

- Soft reset for each core
- Watchdog Timer
- Peripheral reset

The soft reset initiated by the MMC resets each core of the MPC8641D but leaves the processor’s peripheral configuration untouched.

The MPC8641D supports a custom watchdog timer that can be configured to cause an interrupt.

Chapter 7

Programmable Registers

Several registers control and monitor a variety of functions on the AMC131. Normally, only the system boot PROM uses these registers, but they are documented here for application use as needed. Take care when modifying the contents of these registers: the system boot PROM may be relying on the state of the bits under their control.

Key topics in this chapter:

Register Definitions

The following conventions are used throughout this document.

- IPMB, SMBus, and I2C addresses as transmitted on the bus are 7-bit values. By convention, however, these addresses are listed as 8-bit values with the address in the upper 7 bits and a 0 in the least significant bit (the R/~W bit).
- All signal names use the signal name convention of using the "_N" symbol at the end of a signal name to indicate that the signal's active state occurs when it is at a low voltage. The absence of the "_N" or "n" symbol indicates that the signal is active high. Differential signals utilize _N and _P to indicate polarity.
- Binary numbers are represented by a lower case 'b' suffix. For example, 1010b.
- Hexadecimal numbers are represented by a lower case 'h' suffix. All hex letters A-F are written in uppercase. For example, 80BAh.

Custom Register Summary

The AMC131 features 24 custom registers. RO indicates that the field has read-only access. RW indicates that the field has read/write access. WO indicates that the field has write-only access. Some fields autoclear to their reset value and they are noted where applicable.

The following applies to the tables below that provide information about the custom register bits:

- Register bits reset by a soft reset are also reset by a hard reset or power reset. Writable fields reset by a soft. hard, or power reset are indicated by an S following the reset value.
- Register bits reset by a hard reset are also reset by a power reset. Writable fields reset by a hard or power reset are indicated by an H following the reset value.
- Register bits reset by a power reset are only reset by a power reset. Writable fields reset only by a power reset are indicated by a P following the reset value.
- Values in parenthesis indicate reset values.

The AMC131 includes the custom registers itemized in Table 7-1.

Table 7-1: AMC131 Registers

Register	Custom Address Offset
Register 0	0d, 0h
Register 1	1d, 1h
Register 2	2d, 2h
Register 3	3d, 3h
Register 4	4d, 4h
Register 5	5d, 5h
Register 6	6d, 6h
Register 7	7d, 7h
Register 8	8d, 8h
Register 9	9d, 9h
Register 10	10d, Ah
Register 11	11d, Bh
Register 12	12d, Ch
Register 13	13d, Dh
Register 14	14d, Eh
Register 15	15d, Fh
Register 16	16d, 10h
Register 17	17d, 11h
Register 18	RESERVED - reads all zeros
Register 19	19, 13h
Register 20-23	20-23d, 14h-17h

Register 0

This register provides the custom address offset 0d, 0h.

Table 7-2: Custom Address Offset 0d, 0h - Bits

7	6	5	4	3	2	1	0
RO (0)	RW (1) H	RW (1) H	RW (1) H	RO (0)	RO (0)	RO (0)	RO (0)
0	PHY1_ RST	DDR2_ RST_D2	DDR2_ RST_D1	0	0	0	0

Table 7-3: Custom Address Offset 0d, 0h - Description

Name	Description
DDR2_RST_D1	Writing: • 1 to this bit activates the taller SO-CDIMM DDR2 memory module reset • 0 to this bit deactivates this reset
DDR2_RST_D2	Writing: • 1 to this bit activates the shorter SO-CDIMM DDR2 memory module reset • 0 to this bit deactivates this reset
PHY1_RST	Writing: • 1 to this bit activates the on-board quad PHY reset • 0 to this bit deactivates this reset

Register 1

This register provides the custom address offset 1d, 1h.

Table 7-4: Custom Address Offset 1d, 1h - Bits

7	6	5	4	3	2	1	0
RO (0)	RO (0)	RO (0)	RO (0)	RO (0)	RO (0)	RO (0)	RO (0)
cpu_rio_id7	cpu_rio_id6	cpu_rio_id5	cpu_rio_small	SW1 4:1			

Table 7-5: Custom Address Offset 1d, 1h - Bits - Description

Name	Description
SW1 4:1	Reads current switch settings for switch pack SW1
cpu_rio_small	SRIO system addressing: 0 = Reflects the status of the SRIO port as a small system device (using 8-bit addressing). 1 = Large system device (16-bit addressing)
cpu_rio_id5	SRIO Small system device ID bit 5
cpu_rio_id6	SRIO Small system device ID bit 6
cpu_rio_id7	SRIO Small system device ID bit 7

Register 2

This register provides the custom address offset 2d, 2h.

Table 7-6: Custom Address Offset 2d, 2h - Bits

7	6	5	4	3	2	1	0
RW (0) P	RW (0) H	RO (0)	RO (0)	RO (0)	RO (0)	RO (0)	RW (0) H
KILL	FAIL	0	0	0	0	0	KILL_EN

Table 7-7: Custom Address Offset 2d, 2h - Description

Name	Description
KILL_EN	Writing: • 1 to this bit enables the KILL bit to be written with a 1 to cause a hard reset • 0 to this bit disables the KILL bit
FAIL	Writing: • 1 to this bit causes a hard reset. This bit is reset by a power or hard reset that autoclears this bit. • 0 to this bit has no affect
KILL	If KILL_EN = 1, then writing: • 1 to this bit causes a hard reset until a power reset occurs • 0 to this bit has no affect

Register 3

This register provides the custom address offset 3d, 3h.

Table 7-8: Custom Address Offset 3d, 3h - Bits

7	6	5	4	3	2	1	0
RO (0)	RO (0)	RO (0)	RO (0)	RO (0)	RO (0)	RO (0)	RO (0) H
0	0	0	0	0	0	0	RTC_SQW

Table 7-9: Custom Address Offset 3d, 3h - Description

Name	Description
RTC_SQW	M41T62 real-time clock square wave output at 32.768KHz. Used to calibrate the real-time clock. Synchronized to 25MHz to eliminate metastability.

Register 4

This register provides the custom address offset 4d, 4h.

Table 7-10: Custom Address Offset 4d, 4h - Bits

7	6	5	4	3	2	1	0
RO (0)	RO (0)	RO (0)	RO (0)	RO (0)	RO (0)	RW (0) H	RO (0)
0	0	0	0	0	0	PYLD_FLT	0

Table 7-11: Custom Address Offset 4d, 4h - Description

Name	Description
PYLD_FLT	Writing: • 1 to this bit results in the payload indicating a fault to the MMC • 0 to this bit has no affect

Register 5

This register provides the custom address offset 5d, 5h.

Table 7-12: Custom Address Offset 5d, 5h - Bits

7	6	5	4	3	2	1	0
RO (0)	RO (0)	RO (0)	RO (0)	RW (0) H	RW (0) H	RW (0) H	RW (0) H
0	CPU1_ IRQ_ OUT0	0	0	CPU1_ MCP1	CPU1_ MCP0	CPU1_ SMI1	CPU1_ SMI0

Table 7-13: Custom Address Offset 5d, 5h - Description

Name	Description
CPU1_SMI0	Writing: • 1 to this bit generates a system management interrupt to core 0. The software must clear this bit in its interrupt service routine. • 0 this bit clears this bit in the software's interrupt service routine. Software must clear this bit.
CPU1_SMI1	Writing: • 1 to this bit generates a system management interrupt to core 1. The software must clear this bit in its interrupt service routine. • 0 this bit clears this bit in the software's interrupt service routine. Software must clear this bit.
CPU1_MCP0	Writing: • 1 to this bit generates a machine check interrupt to core 0. The software must clear this bit in its interrupt service routine. • 0 this bit clears this bit in the software's interrupt service routine. Software must clear this bit.

Table 7-13: Custom Address Offset 5d, 5h - Description (Continued)

Name	Description
CPU1_MCP1	Writing: • 1 to this bit generates a machine check interrupt to core 1. The software must clear this bit in its interrupt service routine. • 0 this bit clears this bit in the software's interrupt service routine. Software must clear this bit.
CPU1_IRQ_OUT0	This read-only bit shows composite raw MPC8641D on-chip interrupt out

Register 6

This register provides the custom address offset 6d, 6h

Table 7-14: Custom Address Offset 6d, 6h - Bits

7	6	5	4	3	2	1	0
RO (0)	RO (0)	RO (0)	RO (0)	RO (0)	RO (0)	RW (0) H	RW (0) H
SD_CDSWB	0	0	0	0	0	USB_FP_ LED1_BLINK	USB_FP_ LED1

Table 7-15: Custom Address Offset 6d, 6h - Description

Name	Description
USB_FP_LED1	Writing: • 1 to this bit enables the USB LED on the front panel • 0 to this bit turns this LED off
USB_FP_LED1_BLINK	Writing: • 1 to this bit changes the USB LED every 46 ms, providing 11 on/off changes for each second • 0 to this bit ensures that the USB LED is in the solid ON state if USB_FP_LED1 = 1b
SD_CDSWB	Reading: • 1 at this bit indicates that miniSD card insertion has not been detected • 0 at this bit indicates that miniSD card insertion has been detected

Register 7

This register provides the custom address offset 7d, 7h.

Table 7-16: Custom Address Offset 7d, 7h - Bits

7	6	5	4	3	2	1	0
RO (0)	RW (0) H	RW (00b) H		RO (0)	RW (0) H	RW (00b) H	
0	CORE1_ LED_ BLINK	CORE1_LED_COLOR		0	CORE0_ LED_ BLINK	CORE0_LED_COLOR	

Table 7-17: Custom Address Offset 7d, 7h - Description

Name	Description
CORE0_LED_COLOR	Writing these bits specifies the corresponding core 0 status LED color as follows: • 00 - Yellow • 01 - Green • 10 - Red • 11 - Off
CORE0_LED_BLINK	Writing: • 1 to this bit specifies that the core 0 status LED blinks at ~11Hz. • 0 to this bit specifies that the LED does not blink
CORE1_LED_COLOR	Writing these bits specifies the corresponding core 1 status LED color. • 00 - Yellow • 01 - Green • 10 - Red • 11 - Off
CORE1_LED_BLINK	Writing: • 1 to this bit specifies that the core 1 status LED blinks at ~11Hz. • 0 to this bit specifies that the LED does not blink

Register 8

This register provides the custom address offset 8d, 8h.

Table 7-18: Custom Address Offset 8d, 8h - Bits

7	6	5	4	3	2	1	0
RO (0)	RO (0)	RO (0)	RO (0)	RO (0)	RW (0) H	RW (0) H	RW (0) H
0	0	0	0	0	SBREAK_INT_ STS _CLR	SBREAK_INT_ EN	DBREAK_ RST_EN

Table 7-19: Custom Address Offset 8d, 8h - Description

Name	Description
DBREAK_RST_EN	Writing: • 1 to this bit enables the double-break detector. Two sequential breaks from the RS232 interface cause a hard reset pulse to be generated. • 0 to this bit disables this feature.
SBREAK_INT_EN	Writing: • 1 to this bit enables the single-break detector. A payload processor interrupt (IRQ5) is generated to inform the processor that this event has occurred, so this processor can take the appropriate action. A single break from the RS232 interface indicates that the console port focus is to change to the next processor. • 0 to this bit disables this interrupt.
SBREAK_INT_STS_CLR	Writing: • 1 to this bit clears the SBREAK status bit. • 0 to this bit has no affect Reading: • 1 at this bit indicates that SBREAK_INT_EN = 1b and a single-break event (and not a double-break event) is detected • 0 at this bit indicates that the SBREAK interrupt status bit is cleared

Register 9

This register provides the custom address offset 9d, 9h.

Table 7-20: Custom Address Offset 9d, 9h - Bits

7	6	5	4	3	2	1	0
RO (0)	RO (0)	RW (0) H	RW (0) H	RW (0) H	RW (0) H	RW (0) H	RW (0) H
0	0	WDOG_ FAIL_ STS_CLR	WDOG_ INT_EN	WDOG_ KILL_EN	WDOG_ LOAD	WDOG_ KILL_ STS_CLR	WDOG_ FAIL_EN

Table 7-21: Custom Address Offset 9d, 9h - Description

Name	Description
WDOG_FAIL_EN	Writing: • 1 to this bit enables the watchdog timer to •cause a hard reset pulse •set WDOG_FAIL_STS_CLR = 1b •set the watchdog timer fail latch • 0 to this bit disables this capability Reading: • 1 at this bit indicates that watchdog timer causes a hard reset pulse when it times out • 0 at this bit indicates that watchdog timer does not cause a hard reset pulse when it times out
WDOG_KILL_STS_CLR	Writing: • 1 to this bit clears the internal watchdog kill status latch • 0 to this bit has no effect Reading: • 1 at this bit indicates that WDOG_FAIL_EN = 1b, the watchdog timer has caused a kill and the watchdog timer kill latch is set • 0 at this bit indicates that the watchdog timer latch is cleared
WDOG_LOAD	Writing: • 1 to this bit loads the 16-bit watchdog timer with HDATA and LDATA, which prevents the watchdog timer from reaching zero and barking. Watchdog timer countdown does not continue until after the load. • 0 to this bit starts the timer
WDOG_KILL_EN	Writing: • 1 to this bit enables the watchdog timer to cause a hard reset that remains active until the payload circuitry is powered down • 0 to this bit disables this capability

Table 7-21: Custom Address Offset 9d, 9h - Description (Continued)

Name	Description
WDOG_INT_EN	Writing: • 1 to this bit enables the watchdog timer to •cause an interrupt (IRQ7) to the processor •set WDOG_FAIL_STS_CLR = 1b •set the watchdog timer fail latch • 0 to this bit disables this capability Reading: • 1 at this bit indicates that the watchdog timer causes an interrupt when it times out • 0 at this bit indicates that watchdog timer does not cause an interrupt when it times out
WDOG_FAIL_STS_CLR	Writing: • 1 to this bit clears the internal watchdog interrupt (fail) status latch • 0 to this bit has no affect Reading: • 1 at this bit indicates that the WDOG_INT_EN = 1b, the watchdog timer has caused an interrupt, and the watchdog timer fail latch is set • 0 at this bit indicates that the watchdog timer interrupt status is cleared

Register 10

This register provides the custom address offset 10d, Ah.

Table 7-22: Custom Address Offset 10d, Ah - Bits

7	6	5	4	3	2	1	0
RW (00h) H							
WDOG_LDATA							

Table 7-23: Custom Address Offset 10d, Ah - Description

Name	Description
WDOG_LDATA	• The lower 8 bits of the watchdog timer are loaded with this byte when the watchdog load bit is written with 1. This byte is hexadecimal data and represents a multiple of 26 mS of time. • Reading this register returns the value last written to this address, or the default value if a power reset occurs

Register 11

This register provides the custom address offset 11d, Bh.

Table 7-24: Custom Address Offset 11d, Bh - Bits

7	6	5	4	3	2	1	0
RW (00h) H							
WDOG_HDATA							

Table 7-25: Custom Address Offset 11d, Bh - Description

Name	Description
WDOG_HDATA	• The upper 8 bits of the watchdog timer are loaded with this byte when the watchdog load bit is written with 1. This byte is hexadecimal data and represents a multiple of 256 x 26 mS of time. • Reading this register returns the value last written to this address, or the default value if a power reset has occurred

Register 12

This register provides the custom address offset 12d, Ch.

Table 7-26: Custom Address Offset 12d, Ch - Bits

7	6	5	4	3	2	1	0
RW (0h) H				RW (0h) P			
USER_REGISTER							

Table 7-27: Custom Address Offset 12d, Ch - Description

Name	Description
USER_REGISTER	This register is available for any user-defined purpose. *Note: The power reset and hard reset affect various bits differently.*

Register 13

This register provides the custom address offset 13d, Dh.

Table 7-28: Custom Address Offset 13d, Dh - Bits

7	6	5	4	3	2	1	0
RO							
See “SW2 (PCI Express/SRIO Configurations),” on page 39							

Register 14

This register provides the custom address offset 14d, Eh.

Table 7-29: Custom Address Offset 14d, Eh - Bits

7	6	5	4	3	2	1	0
RW (0) H							
SPI_PARALLEL_WRITE_DATA							

Table 7-30: Custom Address Offset 14d, Eh - Description

Name	Description
SPI_PARALLEL_WRITE_DATA	Data written to this address is serialized onto the SPI bus by the SPI master if SPI Start = 1; otherwise, this data is lost data

Register 15

This register provides the custom address offset 15d, Fh.

Table 7-31: Custom Address Offset 15d, Fh - Bits

7	6	5	4	3	2	1	0
R(0)H							
SPI_READ_DATA							

Table 7-32: Custom Address Offset 13d, Fh - Description

Name	Description
SPI_READ_DATA	Data read from this location reflects the last byte deserialized by the SPI master. Deserialized SPI data is reset to 0 by a hard reset.

Register 16

This register provides the custom address offset 16d, 10h.

Table 7-33: Custom Address Offset 16d, 10h - Bits

7	6	5	4	3	2	1	0
RO (0)	RO (0)	RO (0)	RO (0)	RO (0)	RO (0)	RO (0)	RW (0) H
0	0	0	0	0	0	0	SPI_START

Table 7-34: Custom Address Offset 16d, 10h - Description

Name	Description
SPI_START	Writing: • 1 to this bit causes the SPI master to activate the SPI slave select signal • 0 to this bit causes the SPI master to deactivate the SPI slave select signal *Note: This is a read/write bit* *Note: Slave select is a chip select/enable and must be active for data transfer to occur*

Register 17

Table 7-35: Custom Address Offset 17, 11h - Bits

7	6	5	4	3	2	1	0
RO current value				RO (0)			
CPU_ PCE_ EN_N	CPU_ RC_EN_ N	CPU_LANE[1:0]		0x0			

Table 7-36: Custom Address Offset 17, 11h - Description

Name	Description
CPU_PCE_EN_N	This bit reflects the status of the PCI Express port enable signal. Reading: • 0 means the port is enabled • 1means the port is disabled
CPU_RC_EN_N	This bit reflects the status of the PCI Express port root complex/endpoint signal. Reading: • 0 means the port is a root complex • 1 means the port is an endpoint
CPU_LANE[1:0]	These bits reflect the status of the PCI Express lane enable bit. Their value determines how many lanes are enabled on the active PCI Express port.

Register 18

This register is reserved and reads all zeros.

Register 19

This register provides the custom address offset 19, 13h

Table 7-37: Custom Address Offset 19, 13h - Bits

7	6	5	4	3	2	1	0
RO (0)						R/W (0)	RO
0x0	IRQ7_ST	0x0	IRQ5_ST	0x0		SEL_CPU_COMPORT	CONSOLE_REDIRECT

Table 7-38: Custom Address Offset 19, 13h - Description

Name	Description
CONSOLE_REDIRECT	Reading: • 1 at this bit means that CPU1 has the focus of the front panel console port • 0 at this bit means that the MMC has the focus of the front panel console port
SEL_CPU_COMPORT	Reading/Writing: • 0 at this bit means that the serial console port mux selection is controlled by the MMC. • 1 at this bit means that the board logic tries to override the MMC and force the port to be directed to the MPC8641D.
IRQ5_ST	This is the CPU1 IRQ5 line status for the single break detect interrupt. Reading: • 1 at this bit means there is a single break detect interrupt active in the PAL. Output to the CPU may be masked • 0 at this bit means there is no single break detect interrupt active in the PAL
IRQ7_ST	This is the CPU1 IRQ7 line status for the watchdog interrupt. Reading: • 1 at this bit means there is a watchdog interrupt active in the PAL. Output to the CPU may be masked • 0 at this bit means there is no watchdog interrupt active in the PAL

Register 20-23

This register provides the custom address offset 20d-23d, 14h-17h.

Table 7-39: Custom Address Offset 20d-23d, 14h-17h - Bits

7	6	5	4	3	2	1	0
RW (0) H	RW (0) H	RO (0)	RO (0)	RW (0) H	RW (0) H	RW (0) H	RW (0) H
LOCKED	READY	0	0	CORE3_ FLAG	CORE2_ FLAG	CORE1_ FLAG	CORE0_ FLAG

Table 7-40: Custom Address Offset 20d-23d, 14h-17h - Description

Name	Description	Notes
CORE0_FLAG	Successfully writing this register causes this bit to change as follows: CORE0_FLAG = XOR (CORE0_FLAG, DATA[0]	1, 2
CORE1_FLAG	Successfully writing this register causes this bit to change as follows: CORE1_FLAG = XOR (CORE1_FLAG, DATA[1]	1, 2
CORE2_FLAG	Successfully writing this register causes this bit to change as follows: CORE2_FLAG = XOR (CORE2_FLAG, DATA[2]	1, 2
CORE3_FLAG	Successfully writing this register causes this bit to change as follows: CORE3_FLAG = XOR (CORE3_FLAG, DATA[3]	1, 2
READY	Writing: • 1 to this bit unconditionally toggles the value • 0 to this bit has no effect	1, 3
LOCKED	LOCKED contributes to forming a write enable for the register (except READY bit). Writes are enabled when any of the following is true: • LOCKED is 0 and DATA[7] being written is 0 (semaphore is unlocked, resource is multi-access) • LOCKED is 0 and DATA[3:0] being written is 0000 (semaphore is unlocked, resource is unused) • LOCKED is 1 and DATA[3:0] being written is CORE[3:0]_FLAG (semaphore is locked and will be unlocked following the write)	

Notes:

1. These bits are readable and set to 0 by board reset.
2. Not all writes are successful; see LOCKED bit description.
3. All writes to the READY bit are considered valid, unlike the LOCKED bit and COREx_FLAG bits.

Chapter 8

Specifications

This chapter describes the electrical, environmental, and mechanical specifications of the AMC131.

Key topics in this chapter:

Electrical and Environmental Specifications

The subsequent topics provide tables and illustrations showing the following electrical and environmental specifications:

AMC131 Absolute Maximum Ratings

The values below are stress ratings only. Do not operate the AMC131 at these maximums. See “DC Operating Characteristics” on page 100 for operating conditions.

Supply Voltage, Vcc12 (+12V):	10-14V
Supply Voltage, Vcc3 (+3.3V):	3.0-3.6V
Storage Temperature:	-20° to +80° C (-4° to 176° F)
Non-Condensing Relative Humidity:	5 to 90%

Operating Temperature

The operating temperature range is 0 to 55° C. The AMC131 comes from the factory with an integrated heat sink for cooling the processor. The heat sink requires 700 LFM of airflow for the 1.5 GHz option and 250 LFM (linear feet per minute) for all other options.

Caution:
External airflow must be provided at all times during operation to avoid damaging the CPU module. PT strongly recommends use of a card rack fan tray to supply the external airflow.

DC Operating Characteristics

Table 8-1 shows power consumption of an AMC131 with a dual core MPC8641D processor operating without AltiVec™ and with 2GB DDR 400 SDRAM installed.

Table 8-1: Power Consumption with 3.3V VIO

Voltage (VDC)	Maximum Power (W)
12V	40 - (1.0 GHz, 1.25 GH z, 1.33 GHz options) 53 - (1.5 GHz option)
3.30 V	0.27

Battery Backup Characteristics

The battery backup circuit on the AMC131 contains an industry standard 2016 size, coin cell-battery in a surface mount battery clip. The battery life is five years.

Battery Replacement

Under normal operating conditions, it is not anticipated the battery will require replacement during the life of the product. Each customer must evaluate their operating conditions to determine if a battery maintenance program is required as part of a regular maintenance cycle for their board.

Batteries are not field serviceable items. Return the module to PT for battery replacement. See “Return Merchandise Authorization (RMA),” on page 19 for more information about returning merchandise.

Battery Voltage: 3 V

Battery Capacity: 10 mAh

Electrochemical Construction: Manganese (ML series) Lithium battery

Battery Socket Location: See Figure 4-2, “AMC131 Connector Locations,” on page 47

Caution:
The AMC131 contains a Manganese lithium battery. There is a danger of explosion if the battery is incorrectly replaced or handled. Do not disassemble or recharge the battery. Do not dispose of the battery in fire. When the battery is replaced, the same type or an equivalent type recommended by the manufacturer must be used. Used batteries must be disposed of according to the manufacturer’s instructions.

AMC131 Reliability

MTBF = 263,505 hours[1]

MTTR = 3 minutes

1. MTBF calculated using Bellcore SR-332 Issue 1.

Mechanical Specifications

The AMC131 meets the PICMG AMC.0, R 2.0 specification for all mechanical parameters.

Mechanical dimensions are shown in Figure 8-1 and are outlined below.

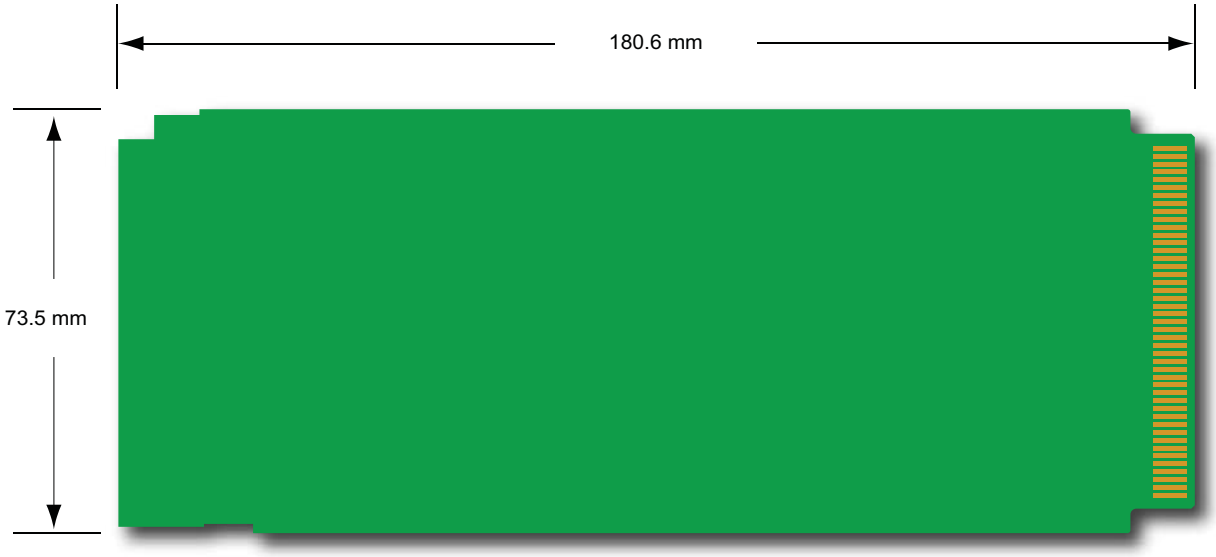
Board Length: 180.6 mm (7.1 in)

Board Width: 73.5 mm (2.9 in)

Board Height: Mid-Size Option: 18.96 mm (0.75 in)
Full-Size, 1.5 GHz Option: 26.85 mm (1.06 in))

Board Weight: 0.34 kg (.75 lbs) with mid-size front panel and 2 GB SDRAM loaded

Figure 8-1: AMC131 Board Dimensions



Chapter 9

Agency Approvals

This chapter presents pending agency approval and certification information for the AMC131 MPC8641D Processor AMC Module. Key topics in this chapter include:

Network Equipment-Building System (NEBS) and European Telecommunications Standards Institute (ETSI)

The product described in this manual is designed to meet NEBS Level 3 and ETSI Environmental Criteria:

- GR-63-CORE - *Network Equipment-Building System Requirements: Physical Protection*
- GR-1089-CORE - *Electromagnetic Compatibility and Electrical Safety - Generic Criteria for Network Telecommunications Equipment*

CE Certification

The product described in this manual meets the intent of the following European Union Directives:

- *EU 89/336/EEC Electromagnetic Compatibility Directive*, amended by 92/31/EEC, 93/68/EEC, 98/13/EEC, and 2004/108/EC
- *EU 72/23/EEC Low Voltage Directive*, amended by 93/68/EEC and 2006/95/EC

by meeting the applicable EU standards as outlined in the *Declaration of Conformance*. The *Declaration of Conformance* is available from Performance Technologies, or from your authorized distributor. Compliance will be demonstrated to the following specifications as listed in the *Official Journal of the European Communities*.

EN55022 Radiated and Conducted Emissions

EN300 386 Electromagnetic Compatibility (EMC)

EN55024 Immunity

EN61000-4-2	Electro-Static Discharge (ESD)
EN61000-4-3	Radiated Susceptibility
EN61000-4-4	Electrical Fast Transient Burst
EN61000-4-5	Surge Immunity
EN61000-4-6	Frequency Magnetic Fields
EN61000-4-11	Voltage Dips, Variations, and Short Interruptions

Safety

The product described in this manual meets the following safety regulations:

EN/IEC 60950	Safety Requirements for Information Technology Equipment
CB Scheme	CB Scheme Certificate and Report
UL60950	UL Recognized

FCC (USA) Class A Notice

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment.

This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

1. This device may not cause harmful interference, and
2. This device must accept any interference received, including interference that may cause undesired operation.

Note: *Modifications made to this device that are not approved by Performance Technologies, Inc. may void the authority granted to the user by the FCC to operate this equipment.*

Industry Canada Class A Notice

This Class A digital apparatus complies with Industry Canada's Equipment Standard for Digital Equipment (ICES-003).

Cet appareil numérique de la classe A est conforme à la norme NMB-003 du Canada.

Product Safety Information

Safety Precautions

Review the following precautions to avoid injury and prevent damage to this product, or any products to which it is connected. To avoid potential hazards, use the product only as specified.

Read all safety information provided in the component product user manuals and understand the precautions associated with safety symbols, written warnings, and cautions before accessing parts or locations within the unit. Save this document for future reference.

Caution:
To Avoid Burns: *If there is a heat sink on this module, it can get very hot during normal operation. To avoid burns, take extra care when removing the module from the chassis soon after shutdown. Wait a few minutes to allow the heat sink to cool down.*

Caution:
Handling the Module: *It is important to hold the module only by the front panel or PCB edges. Avoid touching any components unless necessary to service the product. Do not handle the heat sink, as this can adversely affect the thermal connection between the heat sink and the processor, and cause the processor to overheat under normal operating conditions.*

Caution:
To Avoid Electric Overload: *To avoid electrical hazards (heat, shock and/or fire hazard), do not make connections to terminals outside the range specified for that terminal. Refer to the product user manual for correct connections.*

Caution:
To Avoid the Risk of Electric Shock: *When supplying power to the system, always make connections to a grounded main. Always use a power cable with a grounded plug (third grounding pin). Do not operate in wet, damp, or condensing conditions.*

Caution:
System Airflow Requirements: *Platform components such as processor boards, Ethernet switches, etc., are designed to operate with external airflow. Components can be destroyed if they are operated without external airflow. Chassis fans normally provide external airflow when components are installed in compatible chassis. Filler panels must be installed over unused chassis slots so that airflow requirements are met. Please refer to the product data sheet for airflow requirements if you are installing components in custom chassis.*

Caution:
Do Not Operate Without Covers: *To avoid electric shock or fire hazard, do not operate this product with any removed enclosure covers or panels.*

Caution:
To Avoid the Risk of Electric Shock: *Do not operate in wet, damp, or condensing conditions.*

Caution:
Do Not Operate in an Explosive Atmosphere: *To avoid injury, fire hazard, or explosion, do not operate this product in an explosive atmosphere.*

Caution:
If Your System Has Multiple Power Supply Sources: *Disconnect all external power connections before servicing.*

Warning:
System power supplies must be replaced by qualified service personnel only.

Compliance with RoHS and WEEE Directives

In February 2003, the European Union issued *Directive 2002/95/EC* regarding the Restriction of the use of certain Hazardous Substances in electrical and electronic equipment (RoHS) and *Directive 2002/96/EC* on Waste Electrical and Electronic Equipment (WEEE).

This product is compliant with *Directive 2002/95/EC*. It may also fall under the *Directive 2002/96/EC*.

PT's complete position statements on the RoHS and WEEE Directives can be viewed on the Web at: http://pt.com/page/about-us/ehsms/.

Chapter 10

Data Sheet Reference

This chapter provides information on data sheets, standards, and specifications for the technology designed into the AMC131.

Table 10-1 lists the documents that you can refer to for additional details about this product.

Table 10-1: Related Documents

Author	Document Name
PICMG	AdvancedMC.0-Advanced Mezzanine Card Base Specifications
PICMG	AdvancedMC.1-PCI Express and Advanced Switching
PICMG	AdvancedMC.2-AMC Gigabit Ethernet/10Gigabit XAUI Ethernet
PICMG	AdvancedMC.4 - Serial Rapid IO
IEEE	IEEE Std. 802.3 -2002 CSMA/CD Access Method and Physical Layer Specification.
JEDEC	JEDEC Committee JC-42. PC2-4300/PC2-3200 DDR2 Unbuffered SO-DIMM Reference Design Specification
JEDEC	JEDEC Standard No. 21-C 200 Pin, PC2700/PC2100/PC1600 DDR SDRAM Unbuffered SO-DIMM Reference Design Specification
SDA	Secure Digital Memory Card Specifications Part 1 – Physical Layer Specification
Philips	The I^2C-Bus Specification
Freescale	Freescale MPC8641D PowerPC Integrated Processor Hardware Spec
Freescale	Freescale Integrated Host Processor Family Reference Manual
Broadcom	BCM5466SR Quad 1000/BASE-T and 100BASE-TX Gigabit Ethernet Transceiver
IEEE	IEEE 1149.1-2001 IEEE Standard Test Access Port and Boundary Scan Architecture

Module Management Controller

The MMC is implemented with ATMEL's 8-bit microcontroller with 128 KB in-system programmable flash. For more information, the following document can be downloaded from the ATML Web site:

http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf

See the Intel IPMI home page for information concerning the Intelligent Platform Management Interface, including the Intelligent Platform Management Interface v1.5 Specification and the Intelligent Platform Management Interface Implementer's Guide:

http://developer.intel.com/design/servers/ipmi/spec.htm

User Documentation

The latest product information and manuals are available on the PT Web site. BIOS and driver updates are also available from this site:

http://www.pt.com

Information specific to the AMC131 is available at this URL:

http://pt.com/content/amc131