PIGEON POINT SYSTEMS
AN ACTEL COMPANY

World-Class Management Components
FOCUSED. DEPENDABLE. PROVEN.

PRODUCT BRIEF

Pigeon Point™ BMR-H8S-MCMC Reference Design

Board Management Reference Design for MicoTCA™ Carrier Hub Modules

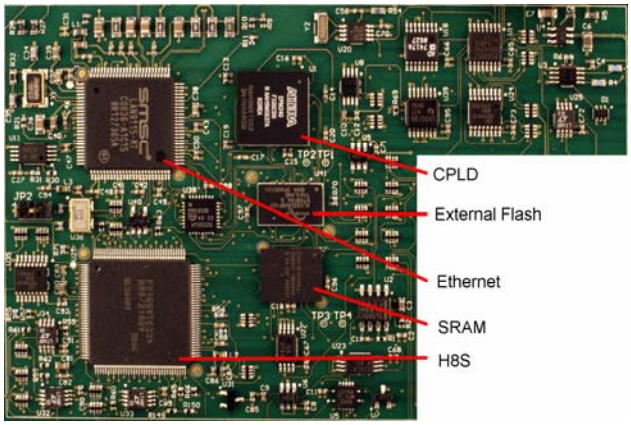
The BMR-H8S-MCMC design is one of a series of Pigeon Point Board Management Reference designs. This member of the series provides a MicroTCA Carrier Management Controller (MCMC), Carrier Manager and local Shelf Manager for µTCA MicroTCA Carrier Hub (MCH) modules and is based on the H8S/2166 controller from Renesas Technology.

The Pigeon Point µCarrier Manager™ firmware that is part of this reference design can be used with any compliant, off-MCH µTCA Shelf Manager or (more likely) combined with the included Pigeon Point µShelf Manager™ firmware to add a local µTCA Shelf Manager. In the latter case, the Carrier Manager and Shelf Manager execute together with the MCMC function on a single H8S processor.

This BMR-H8S-MCMC reference design is delivered in a Pigeon Point Board Management Starter Kit (which is also detailed in a separate Product Brief). The kit includes:

- Schematics for a complete MCMC subsystem, ready for integration into the design of your MCH, with adaptation as necessary.
- Firmware for that subsystem (including MCMC, µCarrier Manager and µShelf Manager functionality), delivered in source form and with development tools—ready for simple and quick adaptation to the specific requirements of your product.
- One-stop support for hardware, firmware and software used in developing and delivering your Pigeon Point BMR-based MCMC.

The photo below shows the core of a BMR-H8S-MCMC controller.



Full support for core hardware requirements

- Uses H8S/2166 controller, which integrates 512Kbytes of Flash and 40Kbytes of SRAM
- Supplementary external Flash and SRAM
- µTCA hot swap interfaces (handle and blue LED), implemented via CPLD, which also implements other functions listed here
- Hardware address detection from backplane
- Payload power supply controls (multiple voltage levels)
- Control of E-Keying governed fabric interfaces
- Optional persistence of above controls across MCMC resets
- Isolated IPMB-L segment for each of up to 12 AMC module sites within a MicroTCA Carrier
- Dual redundant IPMB-0 for access to other modules in a MicroTCA Carrier, such as Cooling Units and Power Modules
- Carrier FRU Info device SEEPROM access
- 10/100 Mbit Ethernet (typically connected to Ethernet switch on the MCH) for off-MCH communication
- Choice of UART- or LPC/KCS-based[1] payload interface
- UART-based serial debug interface and inter-MCH redundancy link
- Digital thermal sensor(s)
- Payload voltage monitoring
- FRU LED management

Specification compliant and interoperability tested

- MTCA.0 R1.0, the µTCA base specification
- AMC.0 R1.0, the AdvancedMC base specification relied on by MTCA.0 R1.0
- IPMI v1.5, document revision 1.1, plus relevant errata
- Successfully tested with other µTCA management components at PICMG TCA-IWs (Telecom Computing Architecture – Interoperability Workshops)

Optional support for special purpose functionality

- Telco (dry contact) alarm management
- Local System Event Log (SEL)

[1] LPC/KCS implements the IPMI-defined Keyboard Controller Style interface using the Low Pin Count version of the Peripheral Component Interconnect (PCI) bus that is included on some processors for access to low speed peripherals such as management controllers.

Comprehensive, readily adaptable firmware

- All mandatory and many optional IPMI/µTCA/AMC commands supported
- Logical subsystem for MCMC function, managing/representing local resources of the MCH
- Logical subsystem for µCarrier Manager, managing/representing the entire MicroTCA Carrier to the Shelf Manager
- Logical subsystem for an optionally installed local µShelf Manager
- Network stack subsystem for communication between:
  - µCarrier Manager and a remote Shelf Manager, or
  - local µShelf Manager and System Manager
- Numerous Pigeon Point extension commands, primarily used over the payload and serial debug interfaces
- Payload alert notifications over payload interface for sensor events and receipt of reset/shutdown commands
- Simple—but highly flexible—configuration of firmware features

Choice of serial interface protocols (SIPL variants) supported via UARTs to payload processor and serial debug interface

- SIPL-TM, based on IPMI-defined Terminal Mode of the Serial/Modem Interface,
- SIPL-BM based on IPMI-defined Basic Mode,
- Either protocol selectable individually for either serial interface
- SIPL-TM: human-oriented and ASCII-based, intended primarily for the serial debug interface
- SIPL-BM: machine-oriented and binary-based, intended primarily for the UART-based payload interface
- Both protocols use encoded forms of raw IPMI messages, which are handled by the IPM Controller essentially like IPMB messages

Simple, but powerful, firmware configuration mechanisms

- Configuration variables in a single `config.h` source file parameterize and determine inclusion/exclusion of subsystems during firmware image build
- Binary configuration files for FRU Information and Sensor Data Records (SDR) merged into firmware image
- FRU Information and SDR files produced from textual representations by special supplied compilers

Sophisticated support for firmware upgrades in the field

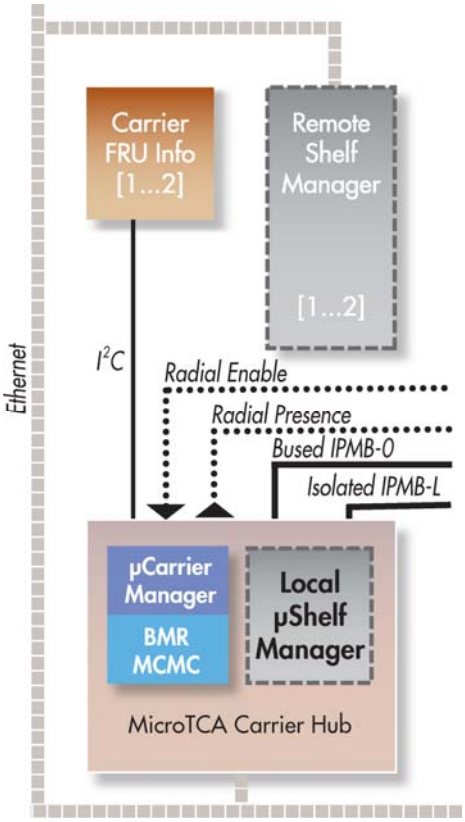
- Firmware upgrades over either payload or debug serial interfaces
- Optional enhanced firmware configuration provides redundant copy firmware, with automatic fallback to backup copy (e.g., if upgrade is interrupted or firmware corruption occurs)

µCarrier Manager manages and represents the carrier and its modules to Shelf Manager

- Communicates with modules in the carrier over IPMB-L or IPMB-0
- Manages and represents the hot swap state of modules in the carrier, including special actions when power modules are hot swapped
- Manages payload and management power for modules in the carrier (including budgeting), acting indirectly through power modules where appropriate
- Combines all local carrier module sensors in a single name space, representing them to the Shelf Manager
- Implements transparent message bridging and command/response forwarding between Shelf Manager and local carrier modules
- Handles E-Keying negotiations with all local carrier modules
- Forwards fan control commands to designated cooling unit

µCarrier Manager supports either local (on-MCH) µShelf Manager or any compliant remote Shelf Manager

- Local µShelf Manager shares single H8S with µCarrier Manager; they communicate via a private interface
- Remote Shelf Manager is off the MCH and communicates with µCarrier Manager via Remote Management Control Protocol (RMCP) over Ethernet



Optionally installed local µShelf Manager represents µTCA shelf to System Manager

- Manages a single carrier µTCA shelf
- Consistent with µTCA spec, presents an interface to the System Manager that is largely equivalent to an ATCA Shelf Manager, including for RMCP access
- Supports persistent System Event Log recording all recent events raised in the shelf
- Provides Sensor Data Record Repository for the shelf

- Implements framework for shelf cooling management
- Provides an easy to use Command Line Interface
- Represents Telco Alarm function (if implemented on the MCH) or forwards telco alarm requests within carrier if implemented elsewhere

Network stack subsystem provides Shelf Manager or System Manager external link

- Ethernet layer, including driver for LAN9115 Ethernet interface device
- Internet Protocol (IP) layer, which cooperates with ARP module to resolve IP address to MAC addresses
- IP-based protocol layer, including UDP and ICMP
- Application protocol layer, including RMCP, for communicating with remote Shelf Manager or remote System Manager

Easy-to-use Command Line Interface in µShelf Manager

- Designed for compatibility with the CLI implemented by the market leading Pigeon Point Shelf Manager for AdvancedTCA
- Accessed via H8S serial port
- Basic command line editing facilities
- Convenient status and control access to:
  - Information about IPM Controllers in the Shelf (specifically the Shelf Manager and the Carrier Manager
  - Information about Field Replaceable Units (FRUs) in the shelf, including MCHs, AMCs, Power Modules, Cooling Units, etc.
  - Activation/deactivation of a specific FRU
  - Display of FRU Info data
  - Display of sensor data for sensors in the shelf

Framework for shelf cooling management in µShelf Manager

- Application Programming Interface (API) for customer implementation of desired shelf cooling policies; key facilities include:
  - access to the aggregated temperature states of managed modules
  - access to the aggregated fan states of managed cooling units
  - access to the fan speed properties of managed cooling units
  - control of the mode and of the fan override of managed cooling units
  - access to the fan geography map
- Provided cooling policy implementation can be used as is, where appropriate, or customized if necessary

Optional use of LPC/KCS for payload interface

- Implemented in lieu of UART-based payload interface
- Based on IPMI-defined KCS variant of IPMI System Interface, implemented over LPC
- Facilitates use of existing IPMI software on payload processor, which often interfaces with management controller via KCS

Comprehensive H8S development environment included

- Cross GNU C compiler and binary utilities for H8S architecture
- JTAG-based firmware download using Renesas JTAG emulator tool (the latter purchased separately)
- Supported under both Linux and Windows on x86 hosts
- Alternate firmware download approach via H8S SCI_1 serial interface (typically assigned to serial debug interface) uses built-in functionality of H8S controller
- SCI_1-based firmware download utility h8sprg supplied in source code form, with compiled binary for Windows

Numerous extensions beyond required IPMI/µTCA commands and functionality

- Graceful Reboot and Issue Diagnostic Interrupt options in FRU Control command
- Get/Set FRU LED State commands for blue LED
- Lamp Test function of Set FRU LED State command
- Cold Reset
- Warm Reset
- Master Write-Read
- Get Device GUID
- Get/Set Sensor Hysteresis
- Get/Set Sensor Thresholds
- Get/Set Sensor Event Enable
- Re-arm Sensor Events
- Get Sensor Event Status

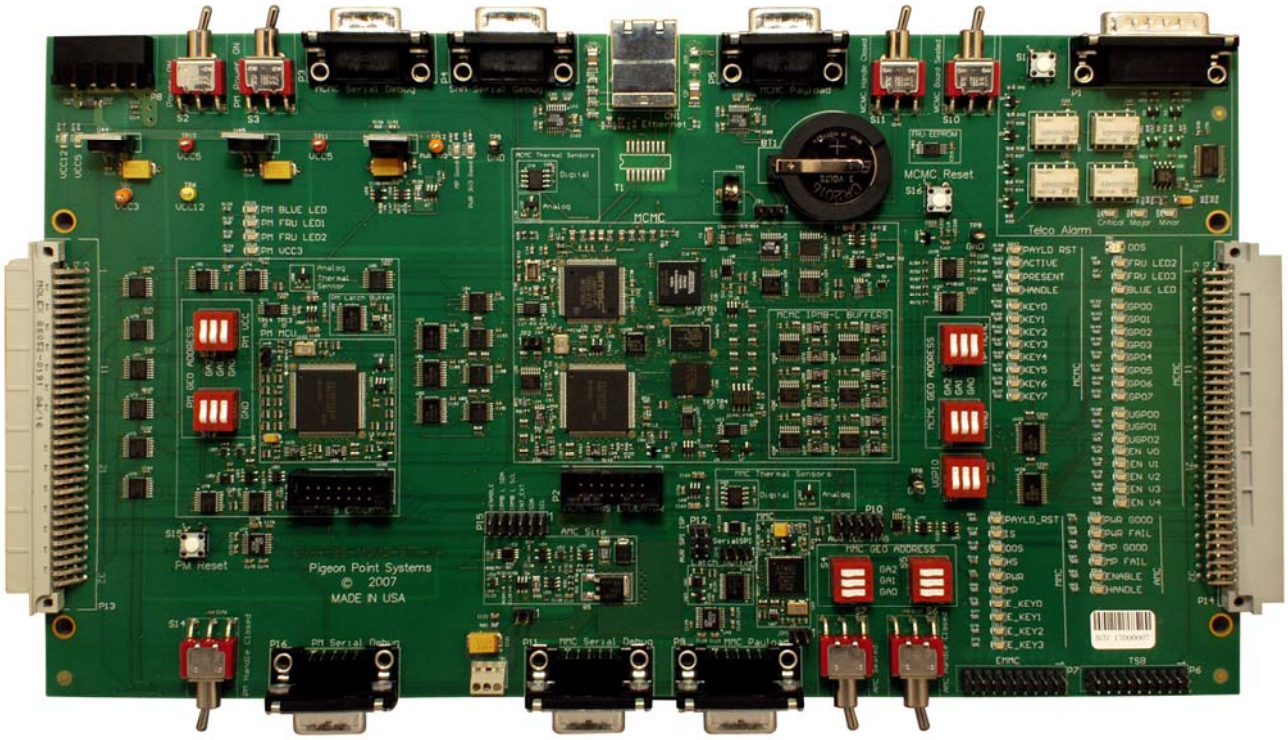
Rich set of Pigeon Point extension commands

- All extension commands implemented as IPMI-compliant OEM messages
- Get Status
- Get/Set Serial Interface Properties
- Get/Set Debug Level
- Get/Set Hardware Address
- Get/Set Handle Switch
- Get/Set Payload Communication Timeout
- Disable/Enable Payload Control
- Reset IPMC
- Hang IPMC[2]
- Bused Resource Control/Status
- Graceful Reset
- Diagnostic Interrupt Results
- Get/Set Payload Shutdown Timeout
- Get/Set Local FRU LED State
- Update Discrete Local Sensor
- Update Threshold Sensor
- Override Sensor Reading
- Get FRU State
- Get Module State
- Enable/Disable Module Site
- Wait for Payload Notify
- Get/Set Geographic Address

[2] This function is used to test the IPMC watchdog.

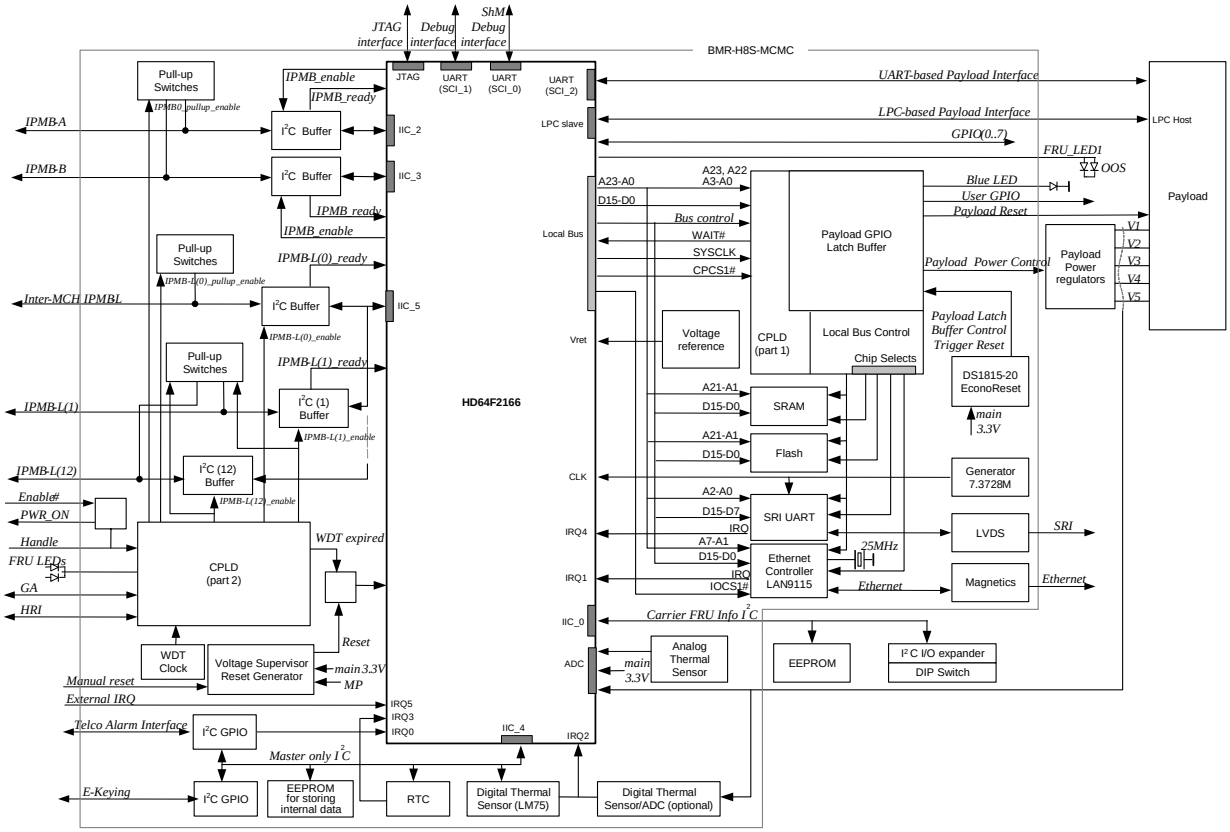
Reference Implementation

A reference implementation called the BMR-H8S-MCMC-BTR is shown below. In addition to the BMR-H8S-MCMC core, this board includes an implementation of an Enhanced Module Management Controller (EMMC) for a simplified/emulated power module that is sufficient to provide a basic power module function in a bench top context and a single emulated AMC site (emphasizing the MMC). Finally, the board includes numerous LEDs, switches and headers to allow lab experimentation with the behavior of the MCMC.



A high level block diagram of an MCMC based on the BMR-H8S-MCMC reference design is shown below. This MCMC hosts the Pigeon Point µCarrier Manager and is capable of hosting the Pigeon Point µShelf Manager.

BMR-H8S-MCMC MicroTCA Carrier Management Controller Block Diagram



For more information, visit our website at **http://www.pigeonpoint.com**

Pigeon Point Systems • P.O. Box 66989 • Scotts Valley CA 95067 • 831.438.1565