

Pigeon Point™ BMR-H8S-EMMC Reference Design Board Management Reference Design for μ TCA™ Modules

The BMR-H8S-EMMC design is one of a series of Pigeon Point Board Management Reference designs. This member of the series provides an Enhanced Module Management Controller (EMMC) for μ TCA and is based on the H8S/2168 family of controllers from Renesas Technology.

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

- Schematics for a complete EMMC subsystem, ready for integration into the μ TCA module, especially a Cooling Unit or OEM module, with adaptation as necessary.
- Firmware for that subsystem, 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 EMMC.

The photo shows the core of a BMR-H8S-EMMC Enhanced Module Management Controller. The active components are:

- The H8S/2168 highly integrated micro controller.
- Dual IPMB buffers to isolate the EMMC from the dual redundant IPMB-0 implemented on the backplane.
- An external oscillator to provide the operating frequency.

Oscillator H8S Processor



I²C Buffers

29mm

Specification compliant and interoperability tested

- PICMG MTCA.0 R1.0
- IPMI v1.5, document revision 1.1, plus relevant errata
- Successfully tested with other management components at PICMG TCA-IW (Telecom Computing Architecture – Interoperability Workshops)

Full support for core hardware requirements

- Supports H8S/2168, H8S/2167 and H8S/2166 variants, with 256K, 384K and 512Kbytes of Flash, respectively
- μ TCA hot swap interfaces (handle and blue LED)
- 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 EMMC resets
- Dual redundant IPMB-0
- Choice of UART- or LPC/KCS-based¹ payload interface
- UART-based serial debug interface
- Thermal sensors (LM60 analog and/or DS75 digital)
- Payload voltage monitoring
- FRU LED management

Optional support for special purpose functionality

- Fan management for Cooling Unit application
- Telco alarm management

Small footprint and low power

- Core EMMC can fit in 29mm x 29mm footprint
- Active components consume only 79mA of max power

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 commands supported over IPMB-0
- 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
- Sophisticated support for firmware upgrades in the field
- 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

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

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)

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

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

- Reset/Get/Set BMC Watchdog Timer
- 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
- Get Device GUID
- Get Message
- Get/Set Sensor Hysteresis
- Get/Set Sensor Thresholds
- Get/Set Sensor Event Enable
- Re-arm Sensor Events
- Get Sensor Event Status
- Get/Set Fan Level
- Get IPMB Link Info
- Get Address Info
- Get Telco Alarm Capability
- Get/Set Telco Alarm State

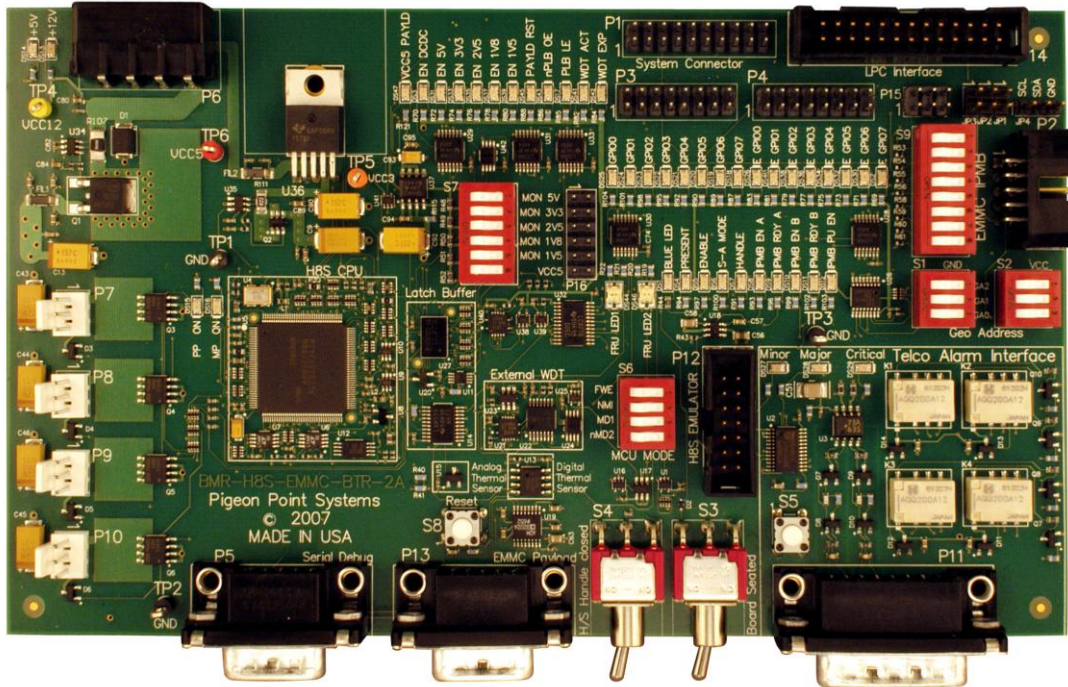
Rich set of Pigeon Point extension commands

- 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 EMMC
- Hang EMMC²
- Graceful Reset
- Diagnostic Interrupt Results
- Get/Set Payload Shutdown Timer
- Set Test Flags
- Get/Set Geographic Address

² This function is used to test the EMMC watchdog.

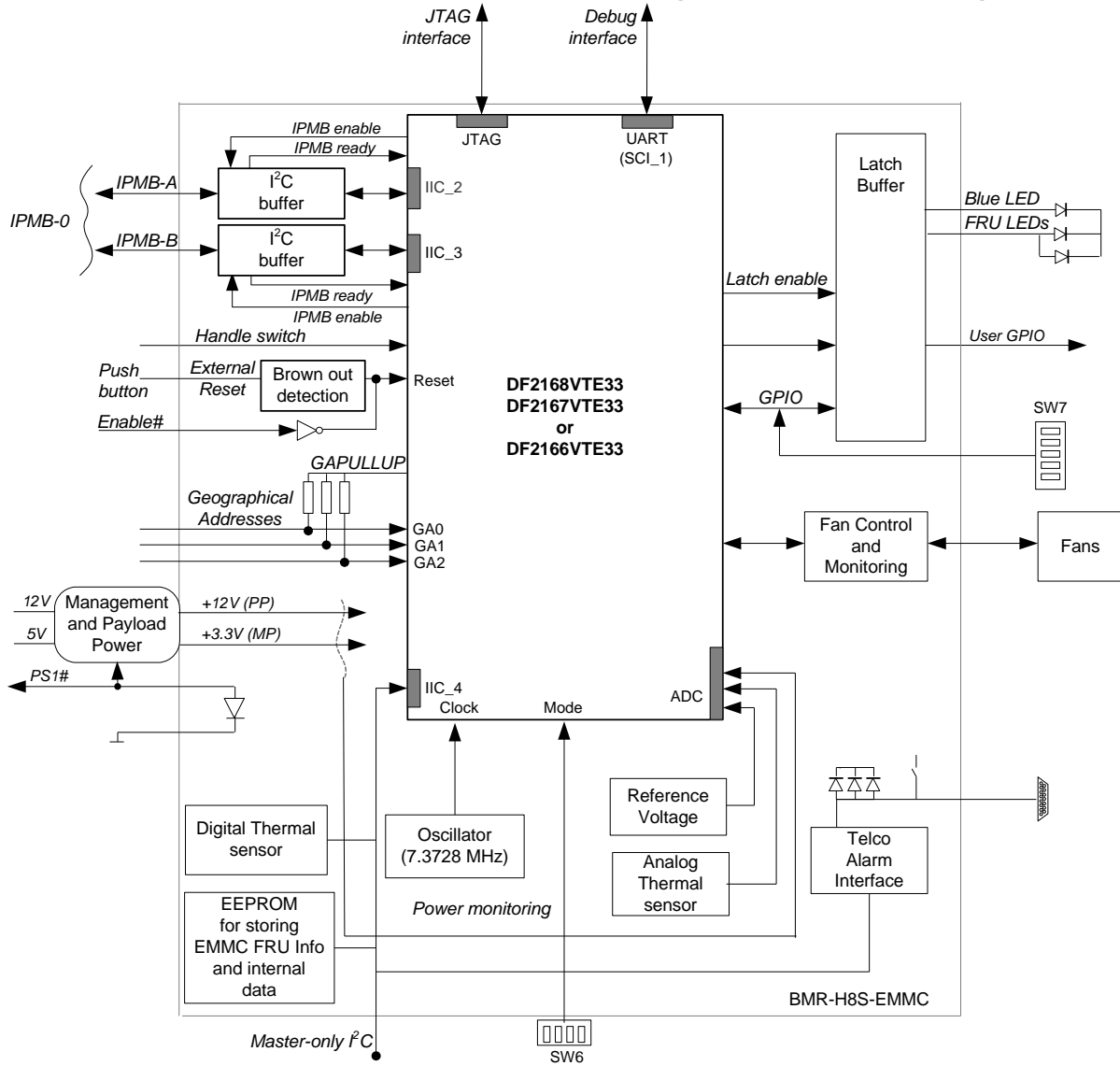
Reference Implementation

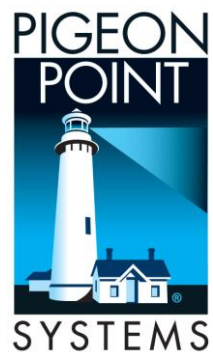
A bench top implementation called the BMR-H8S-EMMC-BTR is shown below. In addition to the BMR-H8S-EMMC core, the board includes implementations of the optional EMMC features and numerous LEDs, switches and headers to allow lab experimentation with the behavior of the EMMC.



A high level block diagram of an EMMC based on the BMR-H8S-EMMC reference design is shown below.

BMR-H8S-EMMC Enhanced Module Management Controller Block Diagram





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

Pigeon Point Systems • 2191 S. El Camino Real, Suite 209 • Oceanside CA 92054 • 760.757.2304