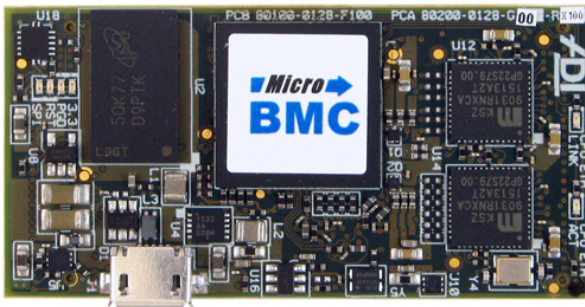


micro-BMC (μ BMC)



Open Project Specification

January 2016

Revision 0.36

Project Initiated By:



Silicom Ltd.

www.silicom-usa.com

In Collaboration with Industry Partners:



Intel Corporation

www.intel.com



Netgate

www.netgate.com



pfSense

www.pfsense.org

Date	Revision	Remarks
5/8/15	R0.10	Initial version for ADI internal distribution
6/5/15	R0.11	Changes too numerous to list.
6/11/15	R0.12	<ul style="list-style-type: none"> 3.2.4. Defined MAC address location and format in FRU EEPROM. 2.3.2. Added SSH protocol. 2.3.3. Added definition of 3 different remote administration methods (added SSH and web server) 3.3.1. Added that available features and signals are defined in a hardware config file for each product/model. 3.3.1. Removed FAN_TACH_EN signal. Available features now defined in hardware config file.
6/17/15	R0.20	<p>First preliminary external release for review.</p> <ul style="list-style-type: none"> 2.3.1, 2.3.2. Clarified RTC is local. 2.3.1, 3.2.13. Removed mention of JTAG debug interface. 3.2.4. Changed wording to note that first address in FRU EEPROM may optionally be used for uBMC MAC address storage, but the uBMC MAC address can be stored elsewhere if needed. 2.1. Changed content of what is considered minimum vs. advanced capabilities.
6/28/15	R0.3	<ul style="list-style-type: none"> Minor clarifications and enhancements throughout Added partners Netgate, pfSense 2.1. Added comments about security and clarified block diagrams. 2.2. Added discussion of security features. 2.2, 2.3, 3.2.8, 3.3, 3.4.1.4.4. Changed from RMII/NC-SI interface to switched GbE (1000base-T). 2.3.1, 3.3, 3.4. Added +5 VDC for onboard USB support. 2.3.1, 3.3.1. Changed from 8 fans to 2 fans. 3.2.7. Clarified that redundant motherboard boot flash would only allow uBMC access to the "Latest Image" flash. 3.2.8. Expanded detail of switched Ethernet hardware topology and interconnects. 2.2. Added overview of supported/unsupported feature list. 2.2. Clarified license and reuse terms. 3.3.1, 3.4.1.2,. Added +5 VDC input for USB power. 3.3.1. Added Ethernet port LED pins. 3.2.11, 3.3.1. Removed FAN_TACH_EN.

7/14/15	R0.31	<ul style="list-style-type: none"> 10. Added new section. 3.5. Added mechanical dimensions 3.3.1. Added 1k pulldown to FAN_PWM signals on uBMC module. 3.2.11. Removed 10k pullup from FAN_PWM signals on motherboard. 3.3.1. Removed RFU pins. 3.3.4. Added connector diagrams and pinouts. 3.3.3. Added connector footprint diagrams
7/14/15	R0.32	<ul style="list-style-type: none"> 3.3.1, 3.3.4. Added UBMC_PRESENT_N signal.
8/20/15	R0.33	<ul style="list-style-type: none"> 2.2. Moved Virtual Media to the supported list. 2.3.1, 2.3.2. Added Virtual Media support. 3.1. Updated block diagram with Virtual Media connection. 3.2.14. New section (Virtual Media). 3.3.1. Added USB signals for Virtual Media support. 3.3.4.2, 3.3.4.4. Added USB and GND pins for Virtual Media support. 3.4.1.4.5. New section (USB). 2.3.2. Added FreeBSD for reference design OS.
9/16/15	R0.34	<ul style="list-style-type: none"> 3.4.1.2. Increased 3.3V current rating, added 5V current rating.
1/25/16	R0.35	<ul style="list-style-type: none"> Updated company information
1/26/16	R0.36	<ul style="list-style-type: none"> Add network communication sections General marketing update and cleanup

Table of Contents

1. TERMS AND DEFINITIONS	8
2. INTRODUCTION	10
2.1. HISTORY (LEGACY BMCS)	10
2.2. μ BMC CONCEPT	13
2.3. μ BMC FEATURE SET	16
2.3.1. <i>Hardware</i>	16
2.3.2. <i>Firmware (Preliminary)</i>	17
2.3.3. <i>Remote Administrator Software Tools (Preliminary)</i>	17
3. HARDWARE	19
3.1. OVERVIEW	19
3.2. FUNCTIONAL DESCRIPTION	20
3.2.1. <i>Embedded SoC</i>	20
3.2.2. <i>Host CPU power state control</i>	20
3.2.3. <i>Host CPU status monitoring</i>	20
3.2.4. <i>FRU EEPROM</i>	21
3.2.5. <i>I2C bus</i>	21
3.2.6. <i>UART</i>	21
3.2.7. <i>SPI</i>	22
3.2.8. <i>Ethernet</i>	22
3.2.8.1. <i>Basic topology</i>	22
3.2.8.2. <i>PHYs on motherboard and μBMC module</i>	24
3.2.8.3. <i>Motherboard offboard magnetics</i>	25
3.2.8.4. <i>Motherboard onboard AC Coupler</i>	25
3.2.8.5. <i>Port LEDs</i>	27
3.2.8.6. <i>Gigabit Switch</i>	27
3.2.9. <i>GPIO</i>	27
3.2.10. <i>Real time clock (RTC)</i>	28
3.2.11. <i>Fan control</i>	28
3.2.12. <i>Voltage monitors</i>	29
3.2.13. <i>Power supply unit monitoring</i>	29
3.2.14. <i>Virtual Media (USB)</i>	30
3.3. MOTHERBOARD CONNECTORS	31
3.3.1. <i>Signal description</i>	31
3.3.2. <i>Connector type</i>	40
3.3.3. <i>Connector PCB Footprints</i>	41
3.3.3.1. <i>Module connector</i>	41
3.3.3.2. <i>Motherboard connector</i>	42
3.3.4. <i>Pin assignments by pin number</i>	43
3.3.4.1. <i>μBMC module - connector P1</i>	44
3.3.4.2. <i>μBMC module - connector P2</i>	45
3.3.4.3. <i>Motherboard - connector P1</i>	46
3.3.4.4. <i>Motherboard - connector P2</i>	47
3.4. ELECTRICAL SPECIFICATION	48
3.4.1.1. <i>Absolute maximum ratings</i>	48
3.4.1.2. <i>Normal operating conditions</i>	48
3.4.1.3. <i>DC specifications</i>	49
3.4.1.4. <i>AC specifications</i>	50

3.4.1.4.1.	SPI bus	50
3.4.1.4.2.	I2C bus	50
3.4.1.4.3.	UART interface	50
3.4.1.4.4.	Ethernet interface	51
3.4.1.4.5.	USB interface	51
3.5.	MECHANICAL SPECIFICATION	51
3.5.1.	Overview	51
3.5.2.	µBMC module dimensions	52
3.5.3.	Server motherboard dimensions	53
4.	FIRMWARE	54
4.1.	INTRODUCTION	54
4.2.	COMPONENTS	54
4.2.1.	Bootloader	54
4.2.2.	Kernel	54
4.2.3.	Development tools	54
4.2.4.	54	
5.	REMOTE ADMINISTRATOR SOFTWARE TOOLS	56
5.1.	HIGH LEVEL REQUIREMENTS	56
5.1.1.	Access control	56
5.1.2.	Database	57
5.1.3.	Display logic	58
5.1.3.1.	Login	59
5.1.3.2.	Dashboard	59
5.1.3.3.	Site administration	60
5.1.3.4.	uBMC fleet administration	60
5.1.3.5.	uBMC details	60
5.1.4.	Remote hosts	61
5.1.4.1.	Status	61
5.1.4.2.	Remote configuration	61
5.1.4.3.	Remote console access	62
5.1.4.4.	Log retrieval	62
5.1.5.	Searching	62
6.	NETWORK COMMUNICATIONS	62
6.1.	OVERVIEW	62
6.2.	REQUIREMENTS	63
6.3.	SECURITY	63
6.4.	RESPONSIVENESS	63
7.	DESIGN COLLATERAL LOCATIONS	64
8.	ABOUT ADI ENGINEERING	65
8.1.	COMPANY INFORMATION	65
8.2.	CONTACTS	65
9.	ABOUT NETGATE	66
9.1.	COMPANY INFORMATION	66
9.2.	CONTACTS	66



10. ABOUT PFSense	67
10.1. PROJECT INFORMATION	67
10.2. CONTACTS	67
11. LEGAL	68
11.1. LICENSE	68
11.2. DISCLAIMER	68

1. TERMS AND DEFINITIONS

Term	Definition
μBMC	micro BMC
ADC	Analog to Digital Converter
BIOS	Basic Input/Output System
BMC	Baseboard Management Controller
CC	Creative Commons
CLI	Command Line Interface
CPU	Central Processing Unit
DDR _x	Double Data Rate SDRAM ("x" is version 2, 3, 4) bus
EEPROM	Electrically Erasable Programmable Read Only Memory
FRU	Field-Replaceable Unit
GbE	Gigabit Ethernet
GPIO	General-Purpose Input/Output
GPL	GNU Public License
GUI	Graphical User Interface
I ² C	Inter-Integrated Circuit bus
iKVM	Keyboard/Video/Mouse over IP
IP	Internet Protocol
IPMI	Intelligent Platform Management Interface
JTAG	Joint Test Action Group
LAN	Local Area Network
LPC	Low Pin Count bus
MAC	Media Access Controller
MDI	Media Dependent Interface
MII	Media-Independent Interface

NC	No Connect
NC-SI	Network Controller Sideband Interface
NRE	Non-Recurring Engineering
NTP	Network Time Protocol
OEM	Original Equipment Manufacturer
OS	Operating System
OTG	On The Go (a USB standard)
PCB	Printed Circuit Board
PECI	Platform Environment Control Interface bus
PHY	Physical Layer (of the OSI model)
PMBus	Power Management Bus
PSU	Power Supply Unit
PWM	Pulse Width Modulation
RAM	Random Access Memory
RESTful	Representational State Transfer
RFU	Reserved for Future Use
RMII	Reduced Media-Independent Interface
RTC	Real-Time Clock
SEL	System Event Log
SMBus	System Management Bus
SMS	Short Message Service
SoC	System on Chip
SOL	Serial Over LAN
SPI	Serial Peripheral Interface (bus)
TTL	Transistor-Transistor Logic
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
VDC	Volts DC (Direct Current)
VLAN	Virtual Local Area Network

2. INTRODUCTION

2.1. History (legacy BMCs)

Networked servers typically have two internal processor subsystems:

- Large, high-performance host CPU (or CPUs in a multiprocessor configuration)
- Small, low-performance Baseboard Management Controller (BMC).

The two subsystems operate independently, with each processor running a separate operating system – often different operating systems. Each processor subsystem has independent access to the LAN. The BMC may have a dedicated Ethernet port for this purpose connecting to the “Management LAN” (Figure 1), or may have a sideband connection to one of the primary Ethernet MACs where it piggybacks traffic to get access to the main “Data LAN” (Figure 2). Systems using a dedicated Management LAN typically do so for a physical layer of security.

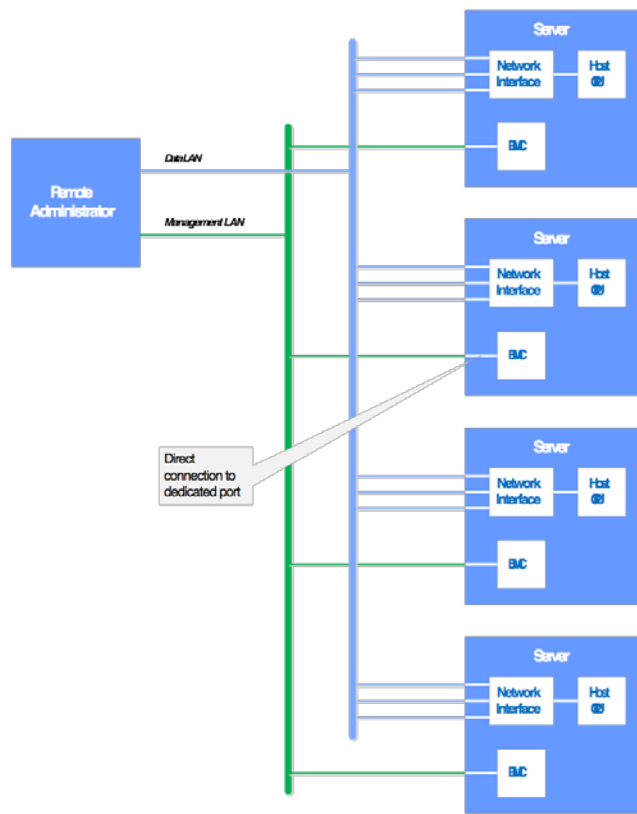


Figure 1 – μBMC with dedicated management LAN

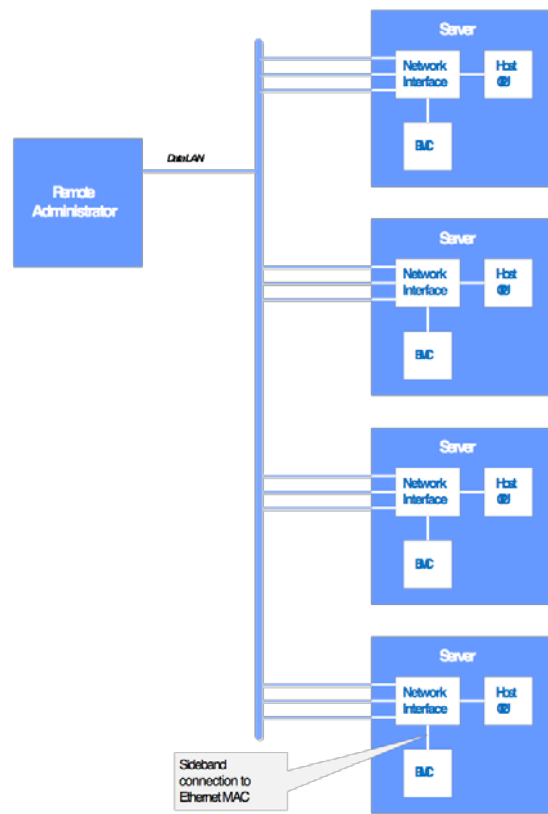


Figure 2 – μBMC in shared LAN system

The BMCs on each of many servers on the LAN are accessed by a Remote Administrator, which may or may not be onsite with the servers. The BMC typically provides the following capabilities remotely:

- Monitoring of host CPU status and health
- Power and reset control of the host CPU
- Access to the host CPU serial console UART
- Thermal sensors
- Event logging with timestamps
- Configurable event network alerts
- Fan control
- BIOS flash updates for the host CPU
- Onboard voltage monitoring



- Virtual media (USB)
- Integrated video
- iKVM support
- Intelligent bus interface to FRU power supplies
- LPC bus for communication with host BIOS and host OS

The Remote Administrator traditionally accesses the various BMCs in the system using software tools implementing Intelligent Platform Management Interface (IPMI) messaging. IPMI-based software tools may be either proprietary, or open-source (e.g. IMPItool, IPMIutil, FreeIPMI, OpenIPMI, etc.).

The BMC has traditionally been an application-specific System on Chip (SoC) processor running proprietary firmware. The firmware is royalty-based, where the OEM pays a per-unit-shipped license fee to use the firmware. There is typically significant Non-Recurring Engineering (NRE) cost associated with customizing the firmware to each specific server board design. Typically, IPMI is used. There are newer open source BMC approaches, including Redfish and OpenBMC, however these are still optimized for larger data center servers, and are not targeted for the new class of optimized, low-cost, purpose-built edge and CPE devices that MicroBMC is intended to serve.

2.2. μ BMC concept

ADI Engineering has multiple customers asking for low-to-mid-end BMC functionality on ADI-designed and manufactured server, microserver, and network appliance boards. More complex, traditional, royalty-based BMC solutions that ADI has used in the past are expensive, large, closed architectures unsuited for these applications. Furthermore, the inherent lack of security in traditional solutions leaves servers open to attack if their BMCs are not isolated to a physically-separate management LAN. Traditional BMCs' low-performance network interfaces also make implementation of higher levels of security and more advanced features challenging.

Accordingly, Silicom is developing a “micro-BMC” (μ BMC) solution based on an inexpensive System-On-Chip (SoC) running an open-source Operating System such as Linux or FreeBSD. The μ BMC will be a reusable building block on Silicom standard products or customer-specific designs. The μ BMC will be implemented as a plug-in module that attaches to connectors on the server motherboard. See Figure 3 for a conceptual rendering of the μ BMC module mounted on a server motherboard.

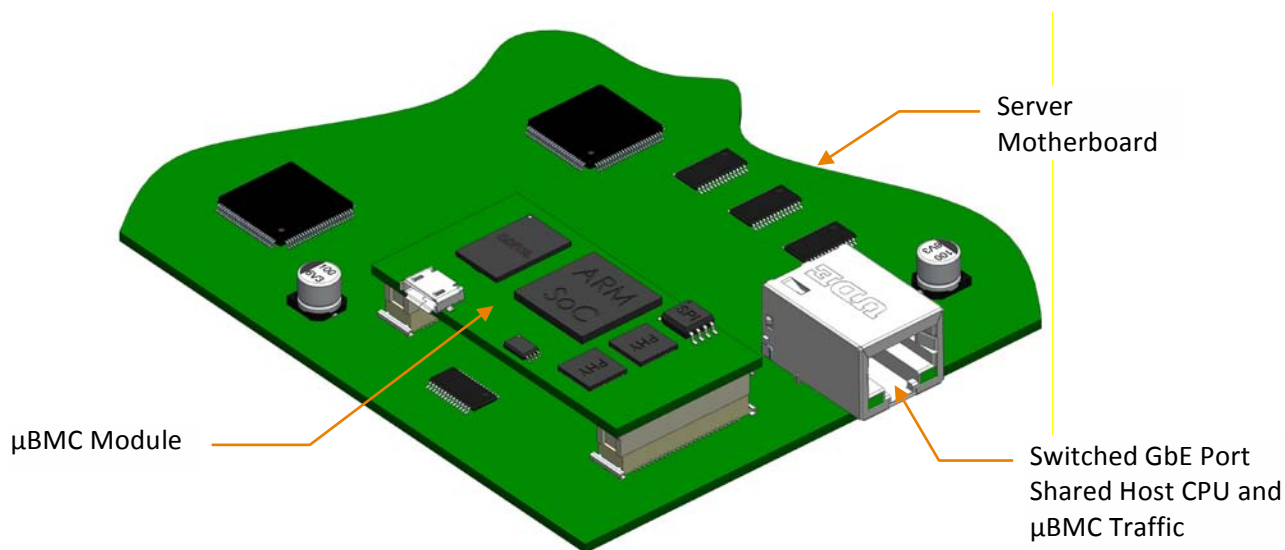


Figure 3 – Conceptual rendering of μ BMC module mounted on server motherboard

Instead of low-performance, low-security LAN sideband interfaces or physically-isolated Management LANs, the μ BMC uses onboard switched Gigabit Ethernet (GbE) to allow the motherboard host CPU and modular μ BMC SoC to share a common high-performance LAN port. Because the switch is managed, it allows the μ BMC to control packet flows and maintain VLANs. Dependency on motherboard parts with special sideband interfaces is replaced with ubiquitous 1000base-T connectivity. With the added bandwidth, modern security protocols can be utilized and more advanced features can be implemented. Figure 4 illustrates the onboard switched GbE concept.

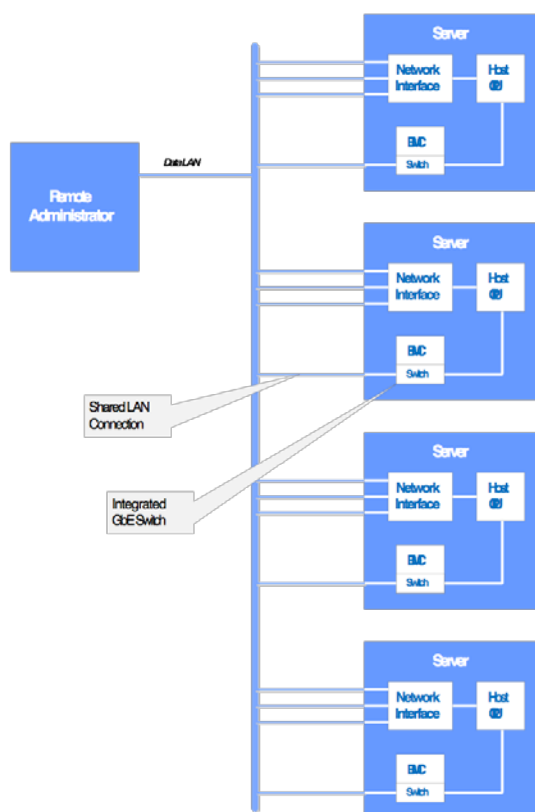


Figure 4 –μBMC module switched GbE on common LAN interface

At a high level, the μBMC provides the following features to the Remote Administrator.

- Monitoring of host CPU status and health
- Power and reset control of the host CPU
- Access to the host CPU serial console UART
- Thermal sensors
- Event logging with timestamps
- Configurable event network alerts
- Fan control
- BIOS flash updates for the host CPU
- Onboard voltage monitoring
- Virtual media (USB)
- Significantly enhanced security vs. traditional IPMI solutions



Low-end to midrange servers do not require the following more advanced and expensive features. These features have been intentionally excluded from the μ BMC definition.

- Integrated video
- iKVM support
- Intelligent bus interface to FRU power supplies
- LPC bus for communication with host BIOS and host OS

While development of the μ BMC is driven initially by a need for ADI Engineering server products, we believe that a gap between traditional “full” BMC solutions and a lightweight BMC solution currently exists throughout the industry. Existing Intelligent Platform Management Interface (IPMI) Remote Administrator software tools have become cumbersome outdated for a lean BMC solution and do not provide a RESTful interfaces that are scalable and flexible; open, modern, GUI-based Remote Administrator software tools are needed. Furthermore, there is a dangerous lack of security in IPMI which leaves servers open to attack; the new μ BMC solution completely secures all remote access.

Silicom is seeking to offer the μ BMC as a modern, open standard project. Users can purchase complete plug-and-play hardware/firmware modules from ADI Engineering, or alternately can develop their own hardware, firmware, and/or software tailored to their specific needs.

μ BMC reference design hardware design collateral, firmware design collateral, and software design collateral are all available for download and use under Creative Commons “CC” (BY – Attribution Alone) license terms, as described in section 10.

Silicom will conduct all of the initial hardware development and OS bring-up for the μ BMC module reference design. Industry partners will aid development of the μ BMC module firmware and the separate Remote Administrator software tools.

2.3. μ BMC Feature Set

As noted in other sections, the following list describes the entire set of μ BMC features that can be implemented and that are defined in this specification. Individual μ BMC module designs may support a subset of these features.

2.3.1. Hardware

- Processor/SoC.
 - Application-specific. Reference design uses the ARM Cortex-A8 TI AM3352BZCZxx, where “xx” denotes core speed grade.
 - Crypto accelerator for secure network access without affecting firmware performance.
- FRU EEPROM. I2C-based, 32 KB storage for management event logging.
- Host CPU Power state control
 - Reset button control
 - Power button control
- Host CPU status monitoring
 - Sleep S3, S4, S5 indicators
 - PROCHOT#, THERMTRIP# thermal indicators
 - CPU error indicators
 - Platform reset
- Heartbeat monitoring. Software GPIO on host CPU drives watchdog input on μ BMC.
- I2C bus
 - Bus 0 (master). Onboard FRU EEPROM, connectivity for motherboard discrete thermal sensors and other devices.
 - Bus 1 (master). Access to PECI SMBus slave inside host CPU. Used for monitoring CPU die temperature.
- UARTs
 - UART 0. Serial Over LAN (SOL) connection to host CPU
 - UART 1. μ BMC dedicated local Command Line Interface (CLI) port
- SPI. Used to access host CPU boot flash for remote BIOS updates.
- Ethernet (Remote Administrator access). 1 Gbps switched Ethernet. Uses embedded managed 3-port switch on μ BMC to allow both motherboard host CPU and μ BMC to access the LAN.
- GPIO. Undefined general purpose input and output pins.
- Voltage monitoring:

- Supports direct connection of up to 8 motherboard voltages to internal ADC.
- If more than 8 voltages are required, then a separate analog multiplexer can be added to the motherboard and controlled by μ BMC GPIO pins. Example: 16-channel TI CD74HCT4067M.
- Main Power Supply Unit (PSU) status monitoring:
 - PSU 0-3 powergood (discrete logic input)
- Fan control. Fans 0-1. PWM output and tachometer input.
- Local Real Time Clock (RTC) for timestamp of FRU EEPROM event log entries
- Virtual Media support. USB device connection to host CPU for USB boot over LAN.
- Power:
 - +3.3 VDC input from motherboard (main).
 - +5 VDC input from motherboard (for μ BMC onboard internal USB connector)
- Environmental. Application-specific. Reference design is rated for 0 to +70 C operation.

2.3.2. Firmware (Preliminary)

A reduced footprint operating system will be used based on Linux or FreeBSD as required by the particular installation. An automated build system will create each of the components required by the device, and each will be remotely upgradable.

The μ BMC firmware is detailed in section 4 of this document.

2.3.3. Remote Administrator Software Tools (Preliminary)

A suite of web based administrations tools are provided. Please see section 5 of this document for details. Three methods of remotely accessing the μ BMC are implemented:

1. SSH command line interface
2. Simplistic web server interface running on BMC.
3. Graphical “dashboard” application running on Remote Administrator computer. Features include:
 - Operating system support. Windows, Linux, Mac.
 - User interface
 - GUI-based dashboard for remote server administration.
 - Search functionality – identify and list all μ BMC server nodes on network
 - Server nodes identified by static IP address or by network host name

- Support for groups and classes of server nodes
 - Drag/drop functionality for applying configurations to nodes and groups
 - Configurable overall status display for each server node
 - List which features are supported by the server node and which features are grayed out (unavailable – not supported)
- μBMC configuration settings
 - Set/apply server node μBMC options using either graphical buttons and fields, or browse for configuration settings file
 - Save settings from a specific μBMC to a local configuration settings file
 - Apply settings to a specific μBMC, multiple μBMCs, or groups of μBMCs simultaneously
- Alerts configurable to send email or SMS text notifications
- Serial Over LAN
 - Host CPU serial console for a server node displayable in separate window or separate tab
 - Multiple serial console sessions can be open simultaneously
- μBMC firmware update applied to a specific μBMC, multiple μBMCs, or groups of μBMCs simultaneously
- Host CPU BIOS update applied to a specific node, multiple nodes, or groups of nodes simultaneously (for nodes that support remote BIOS update)
- Security. All communications with μBMC are encrypted and require login authentication.

The μ BMC module is based around an embedded SoC device capable of running an open-source embedded OS, such as Linux or FreeBSD. Figure 5 shows a high-level view of the μ BMC module hardware architecture and associated motherboard circuitry. Some of the features shown in Figure 5 are required and some features are optional; see signal description section 3.3.1 for a list of required vs. optional features.



Specific details of the SoC implementation are application-specific and not dictated by this specification. For example, the hardware architecture in Figure 5 shows external DDRx (DDR2, DDR3, DDR4) memory and eMMC flash. Specific SoC implementations may have internal memory and data storage or may use different technologies.

3.2. Functional description

3.2.1. Embedded SoC

A variety of generic SoC parts and BMC-specific SoC parts that can meet μ BMC requirements are available on the market. This document does not specify the SoC used for the μ BMC module. It is recommended that the SoC have integrated crypto accelerator hardware to handle security protocols and reduce load on firmware.

The reference design uses the TI AM3352BZCZxx SoC based on the ARM Cortex-A8 core. The “xx” denotes speed grade (TBD).

3.2.2. Host CPU power state control

The μ BMC has 2 output signals used to control host CPU operation:

- Host CPU reset button signal CPU_RSTBTN_N
- Host CPU power button signal CPU_PWRBTN_N

CPU_RSTBTN_N and CPU_PWRBTN_N allow the Remote Administrator to force the host CPU into various power states either gracefully, or immediately depending on timing of signal assertions. This is useful for restarting a hung CPU, rebooting the host CPU after a remote BIOS update, etc.

See motherboard connector signal description table in section 3.3.1 for detailed information about usage and timing of these signals.

3.2.3. Host CPU status monitoring

The host CPU typically implements a number of health/status indication output signals which are monitored by the μ BMC. These signals include the following.

Power state indicators. These signals are used to determine if the host CPU is in S0, S3, S4, or S5 power state.

- CPU_SLP_S3_N
- CPU_SLP_S4_N
- CPU_SLP_S5_N

Thermal state indicators. These signals indicate if the host CPU is operating normally, or if a thermal event has occurred.

- CPU_PROCHOT_N
- CPU_THERMTRIP_N

CPU_ERROR_N. Host CPUs may implement a number of different error outputs (MCERR, IERR, ERROR, etc.). Rather than provide pins for every type of error signal (which varies by CPU), the motherboard is responsible for logical AND-ing CPU error signals together to form this single indicator to the μ BMC.

CPU_PLTRST_N. Motherboard platform reset signal, usually generated by the host CPU. Indicates if the host CPU and other board devices are in reset.

See motherboard connector signal description table in section 3.3.1 for detailed information about usage and timing of these signals.

3.2.4. FRU EEPROM

The μ BMC module has 32 KB of non-volatile EEPROM storage for logging of timestamped system events.

The first 6 address offsets 0x00 – 0x05 in the EEPROM may optionally contain the 6 octets of the μ BMC MAC address. Some embedded SoCs have internally-programmed MAC addresses or other storage locations and would not need to use the FRU EEPROM for this purpose.

3.2.5. I2C bus

The μ BMC module is master of two separate I2C busses for different purposes.

Sensors I2C bus. This I2C bus connects to the onboard FRU EEPROM (section 3.2.4), optional motherboard LM75 type thermal sensors, and other optional motherboard devices.

PECI thermal I2C bus. This I2C bus connects to the PECT slave SMBus port on the host CPU. Previous generation CPUs used a discrete PECT pin for on-die temperature monitoring, More modern CPUs provide a dedicated slave SMBus port for access to die temperature monitor information.

3.2.6. UART

The μ BMC module has TTL-level UART connections for two separate serial ports.

Host CPU console port. This UART connects to the host CPU serial console port. It is used for SOL communications. Serial traffic is packetized and sent/received across the Ethernet LAN for Remote Administrator access to the host CPU console.

µBMC CLI port. This UART is used for local access to the µBMC command line interface and for local µBMC firmware updates.

3.2.7. SPI

Some motherboards are designed with a multiplexer providing both host CPU and µBMC access to the host CPU boot flash. This allows the µBMC to perform host CPU boot flash firmware updates at the request of the Remote Administrator.

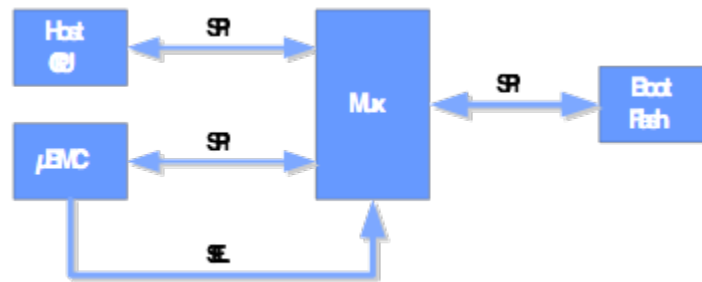


Figure 6 –Host CPU boot flash mux

The µBMC has a select pin `SPI_CPU_MUX_EN` used as a select pin for the mux. Default state of this pin is to give the host CPU access to the boot flash.

Many server motherboards implement dual boot flash in a redundant configuration such that there is a Known Good Image (KGI) flash and a Current Image (CI) flash. The host CPU will attempt to boot from the CI flash first. If that first boot fails due to a blank or corrupted flash image, then the host CPU will automatically switch over and boot from the KGI flash.

The KGI flash is loaded at the factory with a bootable image and then typically write-protected so that it can never be accidentally corrupted or overwritten. This ensures that the motherboard will always have a bootable flash image. Accordingly, it is strongly recommended that the µBMC never be given access to the KGI flash for safety reasons; boot image updates over the network should only write to the CI flash.

3.2.8. Ethernet

3.2.8.1. Basic topology

The µBMC module has two 1000base-T interfaces routed to the motherboard connectors. A 3-port GbE managed switch on the µBMC module connects one port to the µBMC SoC and connects the other two ports to 1000base-T PHYs. The PHYs may be embedded in the switch or may be separate devices. On the motherboard, the host CPU routes 1000base-T to one µBMC port and routes the other µBMC port to an external RJ-45 connector. Figure 7 illustrates the µBMC switched Ethernet topology.

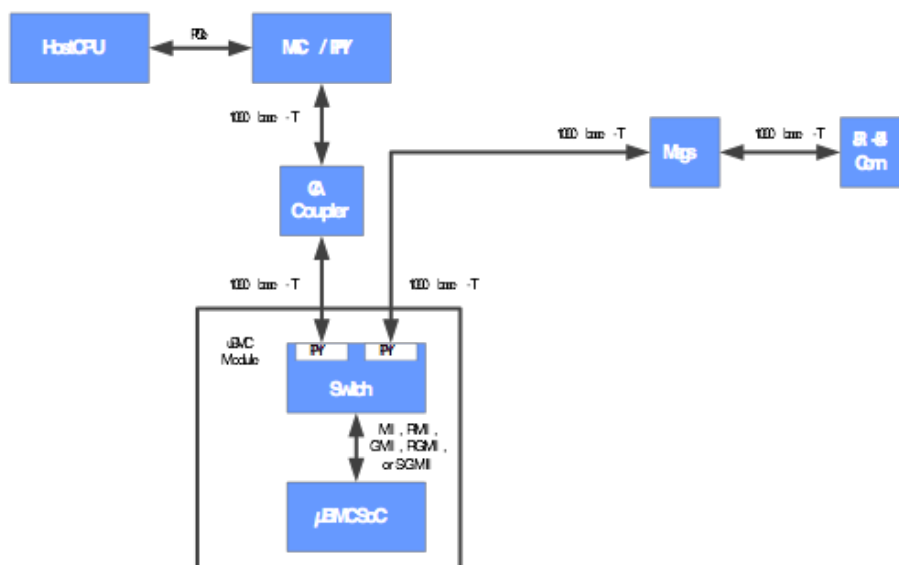


Figure 7 – μBMC switched Ethernet topology

One option shown in Figure 8 is a 1000base-T multiplexer. On motherboards that offer the μBMC module as an optional feature such that the μBMC connectors are empty in some cases, the multiplexer allows the host CPU to use the RJ-45 port directly without the μBMC module present. The μBMC module connector includes a module present pin that is used for controlling the multiplexer. On motherboards that will always have a μBMC module installed, the multiplexer may be left off and the host CPU connects 1000base-T directly to the μBMC module.

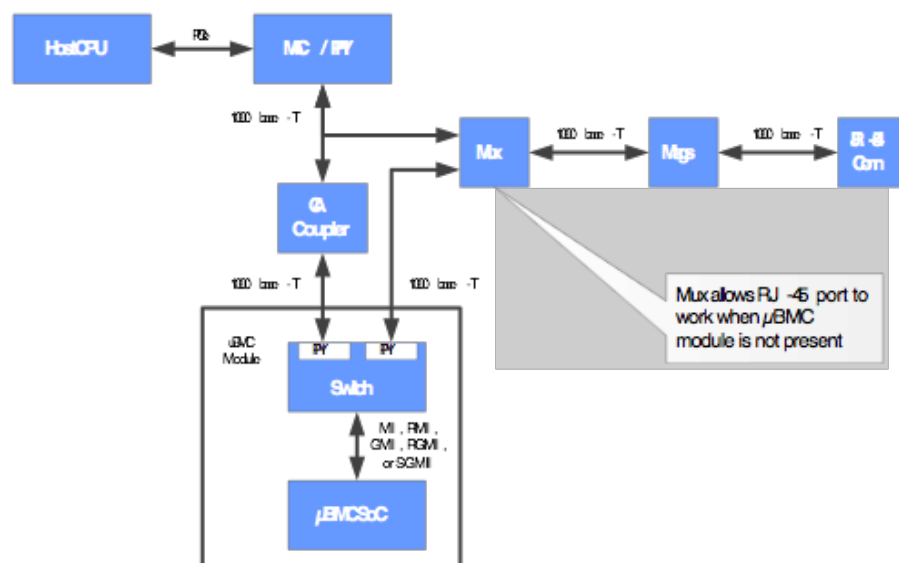


Figure 8 – μBMC switched Ethernet topology with multiplexer

When using a multiplexer and routing the PCB traces, care must be taken to keep stub lengths short to avoid signal integrity issues in the 3-way section between host CPU MAC/PHY, multiplexer, and μBMC connector.

Several example 1000base-T multiplexers that can be used are listed below. These multiplexers have the added advantage of including pins for multiplexing the RJ-45 connector LED signals as well.

- Texas Instruments TS3L500RHUR
- Maxim MAX4892EETX+

3.2.8.2. PHYs on motherboard and μ BMC module

Both motherboard and μ BMC module have 1000base-T PHY devices on them. Output line drivers of transceivers (PHYs) can be classified as either current mode, or voltage mode. It is important to distinguish the two different line driver methods used by PHYs because the method of connecting them requires slightly different design techniques, as discussed in the following sections.

Current mode PHYs generate their output waveforms by sinking a current via the center tap of the primary winding of the transformer (magnetics). The transformer center tap is also used to provide the necessary common mode voltage for the PHY. Current mode line driver technology is a legacy method still used by some PHY vendors.

Modern voltage mode PHYs do not require a center tap voltage and current. Making any kind of DC connection to the center tap would actually cause drive issues for voltage mode PHYs, which is why these PHYs typically either float (no-connect) the center tap, or connect the center tap to ground through 0.1 μ F capacitors.

The two PHYs on the μ BMC module shall both be voltage mode. Sections 3.2.8.3 and 3.2.8.4 discuss methods for connecting the 1000base-T paths on the motherboard.

3.2.8.3. Motherboard offboard magnetics

The 1000base-T path from μ BMC module connector to offboard RJ-45 connector must use 1:1 magnetics for common-mode isolation and ESD protection when connecting long CAT5e cables between systems. Most magnetics modules include integrated common-mode chokes and sometimes autotransformers on the cable side for additional common-mode noise attenuation.

Because the μ BMC module PHYs are voltage-mode devices, the PHY side of the offboard magnetics must have the center taps connected to digital ground through individual 0.1 μ F capacitors as shown in Figure 9.

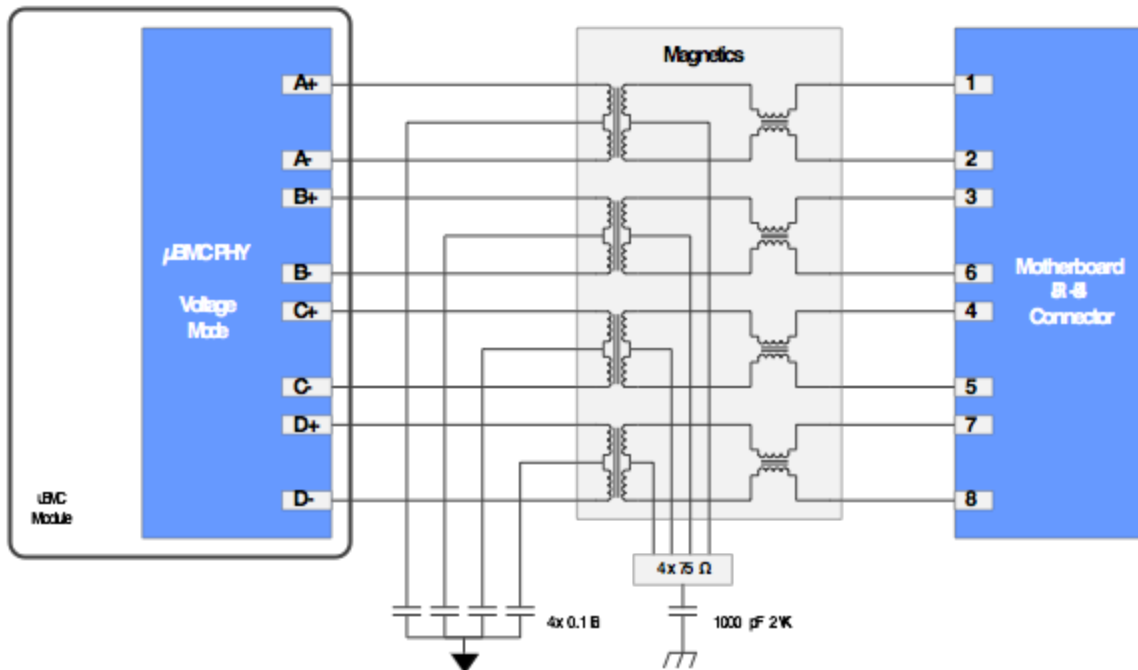


Figure 9 – Motherboard μ BMC to offboard magnetics connection

3.2.8.4. Motherboard onboard AC Coupler

The 1000base-T path from μ BMC module connector to motherboard PHY requires an “AC Coupler”, as shown in Figure 7 and Figure 8. The AC coupler eliminates the need for large and expensive magnetics in this path on the motherboard, also referred to as “transformerless” 1000base-T operation. Magnetics are not required for this path because the path consists of onboard, controlled, short PCB traces that are not subject to the isolation and ESD protection needs of long offboard cables.

The AC coupler should use 0402 or 0201 size surface-mount capacitors and resistors with short PCB traces to minimize inductance.

When a voltage mode PHY (section 3.2.8.2) is used on the motherboard, no DC biasing is required. The μ BMC module voltage mode PHY and the motherboard voltage mode PHY can be connected with an AC Coupler consisting of 0.1 μ F series capacitors, as shown in Figure 10.

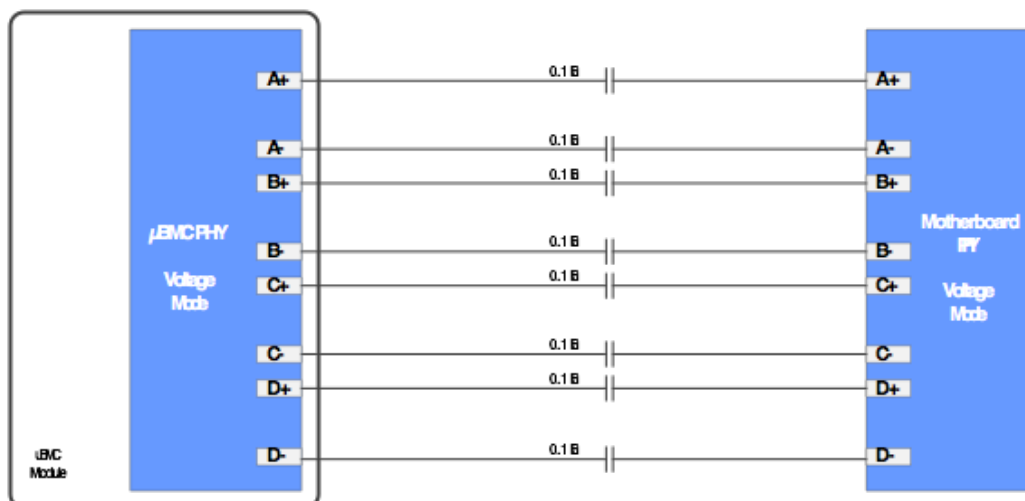


Figure 10 – Motherboard AC Coupler for voltage mode PHY

When a current mode PHY (section 3.2.8.2) is used on the motherboard, additional DC biasing is required. The μ BMC module voltage mode PHY and the motherboard current mode PHY can be connected with an AC Coupler consisting of 0.1 μ F series capacitors, as shown in Figure 11. Additional DC bias 50 ohm pullup resistors are located on the current mode PHY side of the series capacitors. The pullup voltage is typically 2.5 or 3.3 VDC; see current mode PHY datasheet to determine recommended pullup voltage for the specific PHY used.

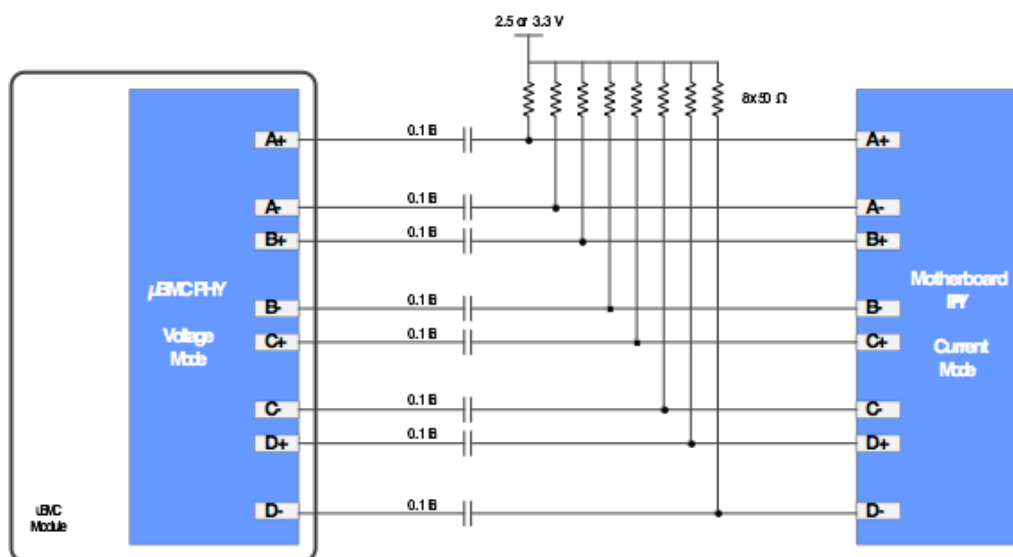


Figure 11 – Motherboard AC Coupler for current mode PHY

3.2.8.5. Port LEDs

The μ BMC module connector has 2 pins for the motherboard offboard port RJ-45 connector LEDs. The ETH2_LED_LINK_N and ETH2_LED_ACT_N signals are both active-low outputs from the μ BMC module that can be used to sink current to the cathode of each RJ-45 port LED (see section 3.4.1.3 for sink current capabilities) with appropriate series current-limiting resistors.

Only discrete link and activity indication is provided; in order to keep the LED control generic and consistent across platforms, there is no support for link speed indication or bicolor LED drive.

For motherboards that implement a multiplexer for the 1000base-T signals (Figure 8) to support a removable μ BMC module, note that the 2x LED signals must also be multiplexed between the motherboard PHY and the μ BMC module PHY so that either PHY can control the RJ-45 port LEDs. Most 1000base-T specific multiplexers include pins for this purpose; several example 1000base-T multiplexer vendor part numbers are listed in section 3.2.8.1.

3.2.8.6. Gigabit Switch

A 3-port managed Gigabit Ethernet switch device is included on the μ BMC module with the following 3 port connections:

- Switch port connected to the μ BMC SoC internal Ethernet MAC
- Switch port connected to 1000base-T PHY for motherboard PHY connection
- Switch port connected to 1000base-T PHY for motherboard offboard RJ-45 port connection

The PHYs may optionally be embedded in the switch. Alternately, the switch may be embedded in the μ BMC SoC, as is done on the μ BMC module reference design.

The Gigabit Ethernet switch supports, at a minimum, the following management features:

- Non Blocking switch fabric
- Four priority level QOS support (802.1p)
- VLAN support
- Spanning tree support
- MAC authentication (802.1x)
- Flow Control Support (802.3x)

3.2.9. GPIO

Multiple GPIO pins are available for application-specific usage. The assignment and use of these pins is not defined by this document.

3.2.12. Voltage monitors

The μ BMC module includes multiple Analog-to-Digital Converters (ADC) to allow measurement of multiple motherboard voltages. These are typically used for monitoring critical voltage rails and generating sensor events if any of the voltages are out of spec.

The ADCs have a limited input voltage range (section 3.4.1.3), so higher motherboard voltages will be scaled using resistor dividers on the motherboard. The μ BMC firmware is configured to compensate for the scaling to derive the actual motherboard voltage.

3.2.13. Power supply unit monitoring

Server chassis include one or more Power Supply Units (PSUs) which may be fixed, or hot-pluggable. Figure 13 and Figure 14 illustrate the hot-pluggable PSU concept in a server chassis with multiple PSU slots.

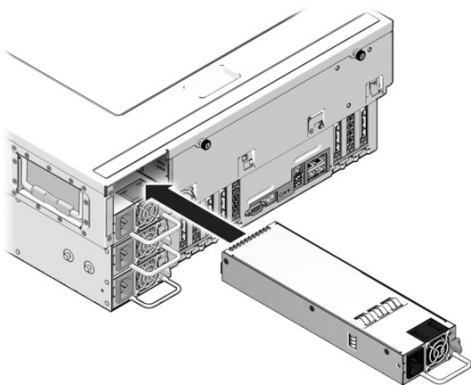


Figure 13 – PSU insertion in a multi-PSU server chassis

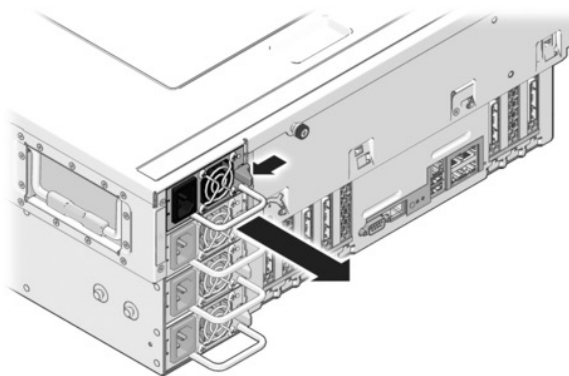


Figure 14 – PSU removal in a multi-PSU server chassis

More advanced PSUs include pins for the serial Power Management Bus (PMBus) to allow the host CPU or BMC to access advanced diagnostic registers inside each PSU.

The μ BMC module simplifies PSU monitoring by taking only discrete PSU_PWRGD power good inputs from the motherboard to indicate if the PSU is operating within specifications. This may be as simple as a voltage comparator output on the motherboard or a discrete power good output from the PSU.

3.2.14. Virtual Media (USB)

The μ BMC module provides a means for the Remote Administrator to boot the host CPU from a USB drive (thumbdrive, DVD drive, HDD, etc.) plugged into the Remote Administrator's computer. The μ BMC module does this by providing a USB device connection to the host CPU (host CPU is the USB host), identifying itself as a USB mass storage device type, and then mapping that USB device port to the LAN. Using this method, the Remote Administrator can remotely boot the host CPU from USB media for software installation, troubleshooting, etc. without requiring a local physical connection.

3.3. Motherboard connectors

3.3.1. Signal description

Pin types:

- OD Open-drain output GPIO
- TTL_{IN} 3.3V TTL level input GPIO input
- TTL_{OUT} 3.3V TTL level output GPIO output
- TTL_{INOUT} 3.3V TTL level output GPIO bidirectional
- Gbase-T Ethernet 1000base-T
- ALG_{IN} Analog ADC input
- USB Universal Serial Bus

Active-low signals are denoted by the “_N” suffix.

Signal direction (input, output) is from the perspective of the μ BMC module.

Some signals are required and some are optional. For signals marked optional, the designer may choose to implement all, none, or a subset of the signals; for example, the designer could choose to implement PSU_PWRGD [1:0] and not implement PSU_PWRGD [3:2]. In the case of grouped signals like PSU_PWRGD [3:0] (4 signals total), any non-implemented signals must be the upper signal numbers. In other words, the designer cannot implement only PSU_PWRGD [3:2] and leave PSU_PWRGD [1:0] disconnected.

A hardware configuration file loaded to the μ BMC firmware defines which signals and features a specific product supports.

Required Optional	Type	Dir	Signal	Description
Voltage Monitors				
Optional	ALG _{IN}	I	VMON_IN [7:0]	ADC voltage monitor inputs. Used for monitoring motherboard voltages. Pull low with 1k resistor on motherboard if not used.
I2C / SMBus				
Optional	TTL _{INOUT}	I/O	I2C_SENSORS_SDA	I2C master data - sensors. Used for standard LM75 thermal sensors on motherboard, μ BMC onboard FRU EEPROM, and general-purpose use. Pull high with 4.7k resistor on motherboard.
Optional	TTL _{OUT}	O	I2C_SENSORS_SCL	I2C master clock - sensors. Pull high with 4.7k resistor on motherboard.
Optional	TTL _{INOUT}	I/O	I2C_PECI_SDA	I2C master data - PECI. Used for PECI monitoring of CPU die temperature. Pull high with 4.7k resistor on motherboard.
Optional	TTL _{OUT}	O	I2C_PECI_SCL	I2C master clock - PECI. Pull high with 4.7k resistor on motherboard.
General Purpose Pins				
Optional	TTL _{INOUT}	I/O	GPIO [7:0]	General Purpose Input/Output pins. Application-specific, user-defined pins. Usage not defined by this specification.
Chassis Management				

Required Optional	Type	Dir	Signal	Description
Optional	TTL _{IN}	I	PSU_PWRGD [3:0]	Power supply unit power good indicators. 0 PSU voltage out of spec. 1 PSU operational. Pull high with 4.7k resistor on motherboard if not used.
Optional	TTL _{IN}	I	FAN_TACH [1:0]	Fan tachometer inputs. Indicates rotational speed of fan for monitoring of fan health. Pull high with 4.7k resistor on motherboard if not used.
Optional	TTL _{OUT}	O	FAN_PWM [1:0]	Fan PWM speed control outputs. PWM signal changes duty cycle from 0 to 100% to control fan speed. 0% Fan off 50% Fan half speed 100% Fan full speed Pulled low with 1k resistor on μBMC module (do not add pullup or pulldown on motherboard)
Host CPU Power State Control				
Required	OD	O	CPU_RSTBTN_N	Host CPU reset button signal. 0 Assert for 1 second or longer issues hardware reset to CPU and holds in reset until released. 1 No action
Required	OD	O	CPU_PWRBTN_N	Host CPU power button signal. 0 When in S0 or S3 states, assert 4 seconds or longer to immediately transition to S5 state. 0 When in S0 state, assert less than 4 seconds to start sequencing to S3 state. 0 When in S3 or S5 states, assert less than 4 seconds to start sequencing to S0 state.

Required Optional	Type	Dir	Signal	Description
				1 No action
UART				
Required	TTL _{IN}	I	UART_SOL_RXD	Host CPU console UART receive data. Redirected to Serial-Over-LAN (SOL) in μ BMC.
Required	TTL _{OUT}	O	UART_SOL_TXD	Host CPU console UART transmit data. Redirected to Serial-Over-LAN (SOL) in μ BMC.
Required	TTL _{IN}	I	UART_CLI_RXD	μ BMC command line interface (CLI) UART receive data. Used for local user debug/development command line communication with μ BMC.
Required	TTL _{OUT}	O	UART_CLI_TXD	μ BMC command line interface (CLI) UART transmit data. Used for local user debug/development command line communication with μ BMC.
Host CPU Status Monitors				

Required Optional	Type	Dir	Signal	Description
Optional	TTL _{IN}	I	CPU_HEARTBEAT_N	<p>Host CPU heartbeat.</p> <p>Connected to watchdog timer input on μBMC SoC. Pulse low, then high to reset timer. Pulse low at a rate of once every 900 ms to keep timer running (watchdog timeout period is 1.0 s). Watchdog is configurable in μBMC to either be disabled (default), create a sensor event for the Remote Administrator, or create a sensor event and reset the host CPU.</p> <p>Pull high with 4.7k resistor on motherboard if not used.</p> <p>Need to wait until host CPU is in S0, PLTRST_N deasserted, etc. before start counting. User application on host CPU is the one that will periodically pulse a GPIO output. How do we know when Host CPU boot flash done, OS loaded, and customer application running? Don't want to log a WDT event until the host CPU has a chance to kick it.</p>
Optional	TTL _{IN}	I	CPU_PROCHOT_N	<p>Host CPU processor hot indication.</p> <p>0 Host CPU is approaching thermal limits and has begun throttling performance to reduce temperature.</p> <p>1 CPU is operating within safe thermal limits.</p> <p>State of this signal will be ignored when PLTRST_N is asserted.</p> <p>Pull high with 4.7k resistor on motherboard if not used.</p>
Optional	TTL _{IN}	I	CPU_THERMTRIP_N	<p>Host CPU thermal trip indication.</p> <p>0 Host CPU has reached thermal limits and has hard shutdown. Asserts for a duration of approximately 200 ns.</p> <p>1 CPU is operating within safe thermal limits.</p> <p>State of this signal will be ignored when PLTRST_N is asserted.</p> <p>Pull high with 4.7k resistor on motherboard if not used.</p>

Required Optional	Type	Dir	Signal	Description
Optional	TTL _{IN}	I	CPU_ERROR_N	<p>Host CPU error indication.</p> <p>0 CPU error occurred.</p> <p>1 Normal CPU operation.</p> <p>Different host CPUs will have different error output signals (e.g. MCERR_N, IERR_N, ERROR_N, etc.). The motherboard must incorporate any logic to AND together various CPU error outputs to form the simple CPU_ERROR_N input to the µBMC.</p> <p>Pull high with 4.7k resistor on motherboard if not used.</p>
Optional	TTL _{IN}	I	CPU_SLP_S3_N	<p>Sleep S3.</p> <p>0 Host CPU is in S3 power state or below.</p> <p>1 Host CPU is in S0 state.</p> <p>Pull high with 4.7k resistor on motherboard if not used.</p>
Optional	TTL _{IN}	I	CPU_SLP_S4_N	<p>Sleep S4.</p> <p>0 Host CPU is in S4 power state or below.</p> <p>1 Host CPU is in S0 or S3 state.</p> <p>Pull high with 4.7k resistor on motherboard if not used.</p>
Optional	TTL _{IN}	I	CPU_SLP_S5_N	<p>Sleep S5.</p> <p>0 Host CPU is in S5 power state.</p> <p>1 Host CPU is on S0, S3, or S4 power state.</p> <p>Pull high with 4.7k resistor on motherboard if not used.</p>
Required	TTL _{IN}	I	CPU_PLTRST_N	<p>Platform reset.</p> <p>0 Host CPU and most board devices are in reset.</p> <p>1 Host CPU normal operation.</p>
Ethernet to Host CPU (ETH1)				

Required Optional	Type	Dir	Signal	Description
Required	Gbase-T	I/O	ETH1_MDI_A_DP	1000base-T MDI differential pair positive leg
Required	Gbase-T	I/O	ETH1_MDI_A_DN	1000base-T MDI differential pair negative leg
Required	Gbase-T	I/O	ETH1_MDI_B_DP	1000base-T MDI differential pair positive leg
Required	Gbase-T	I/O	ETH1_MDI_B_DN	1000base-T MDI differential pair negative leg
Required	Gbase-T	I/O	ETH1_MDI_C_DP	1000base-T MDI differential pair positive leg
Required	Gbase-T	I/O	ETH1_MDI_C_DN	1000base-T MDI differential pair negative leg
Required	Gbase-T	I/O	ETH1_MDI_D_DP	1000base-T MDI differential pair positive leg
Required	Gbase-T	I/O	ETH1_MDI_D_DN	1000base-T MDI differential pair negative leg
Ethernet to Offboard RJ-45 Connector (ETH2)				
Required	Gbase-T	I/O	ETH2_MDI_A_DP	1000base-T MDI differential pair positive leg
Required	Gbase-T	I/O	ETH2_MDI_A_DN	1000base-T MDI differential pair negative leg
Required	Gbase-T	I/O	ETH2_MDI_B_DP	1000base-T MDI differential pair positive leg
Required	Gbase-T	I/O	ETH2_MDI_B_DN	1000base-T MDI differential pair negative leg
Required	Gbase-T	I/O	ETH2_MDI_C_DP	1000base-T MDI differential pair positive leg
Required	Gbase-T	I/O	ETH2_MDI_C_DN	1000base-T MDI differential pair negative leg
Required	Gbase-T	I/O	ETH2_MDI_D_DP	1000base-T MDI differential pair positive leg
Required	Gbase-T	I/O	ETH2_MDI_D_DN	1000base-T MDI differential pair negative leg

Required Optional	Type	Dir	Signal	Description
Required	OD or TTL _{OUT}	O	ETH2_LED_LINK_N	<p>Link indicator LED for offboard RJ-45 Connector</p> <p>0 Ethernet port is linked.</p> <p>1 or Z No Link.</p> <p>Used on motherboard to sink current to the cathode of the RJ-45 port LED (see section 3.4.1.3 for sink current capabilities) with appropriate series current-limiting resistors.</p>
Required	OD or TTL _{OUT}	O	ETH2_LED_ACT_N	<p>Activity indicator LED for offboard RJ-45 Connector</p> <p>Toggle Ethernet port activity.</p> <p>1 or Z No activity.</p> <p>Used on motherboard to sink current to the cathode of the RJ-45 port LED (see section 3.4.1.3 for sink current capabilities) with appropriate series current-limiting resistors.</p>
SPI Host CPU Boot Flash Update				
Optional	TTL _{OUT}	O	SPI_CPU_MUX_EN	<p>SPI mux enable to Host CPU boot flash.</p> <p>For systems that support remote Host CPU boot flash updates, the motherboard must have a mux that allows either the Host CPU, or the μBMC to access the boot flash (but not both). The μBMC has control over which device can access the boot flash at any given time.</p> <p>0 BMC has access to boot flash</p> <p>1 Host CPU has access to boot flash (default)</p>
Optional	TTL _{OUT}	O	SPI_CPU_SCK	SPI clock to Host CPU boot flash.
Optional	TTL _{OUT}	O	SPI_CPU_CS_N	SPI chip select to Host CPU boot flash.
Optional	TTL _{IN}	I	SPI_CPU_MISO	<p>SPI data input from Host CPU boot flash.</p> <p>Pull low with 1k resistor on motherboard if not used.</p>
Optional	TTL _{OUT}	O	SPI_CPU_MOSI	SPI data output to Host CPU boot flash.
Virtual Media (USB)				

Required Optional	Type	Dir	Signal	Description
Optional	USB	I/O	USB_VM_DP	USB differential pair positive leg
Optional	USB	I/O	USB_VM_DN	USB differential pair negative leg
Power / Ground / Misc				
Required	Power	O	UBMC_PRESENT_N	<p>μBMC module present indicator.</p> <p>0 μBMC module present.</p> <p>Z μBMC module not present.</p> <p>Tied to GND on μBMC module. Motherboards that use a 1000base-T multiplexer to bypass the μBMC module connectors when a μBMC module is not plugged in use this signal to control the mux selection.</p> <p>Pull high with 4.7k resistor on motherboard if used. If not used on motherboard, can leave this pin floating (no connect).</p>
Required	Power	I	VDD	+3.3 VDC module power
Optional	Power	I	VDD_USB	<p>+5 VDC for onboard USB connector</p> <p>Some μBMC modules may include an onboard USB port used for booting from external USB drives and data transfer during prototype development. The reference design uses a micro-AB USB OTG connector for this purpose.</p>
Required	Power	I	GND	Module ground

3.3.2. Connector type

The μ BMC module mates to the motherboard using two pairs of 50-pin low-cost, high-density connectors. The μ BMC module side connector type and the motherboard-side connector type are illustrated in Figure 15 and Figure 16. Signal pinout for each connector is listed in section 3.3.4.

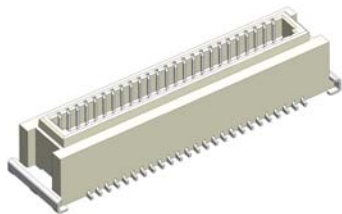


Figure 15 – μ BMC module connector ISO view

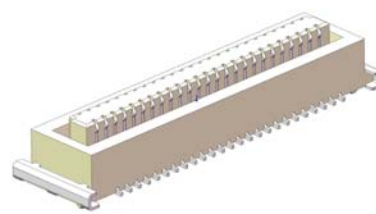


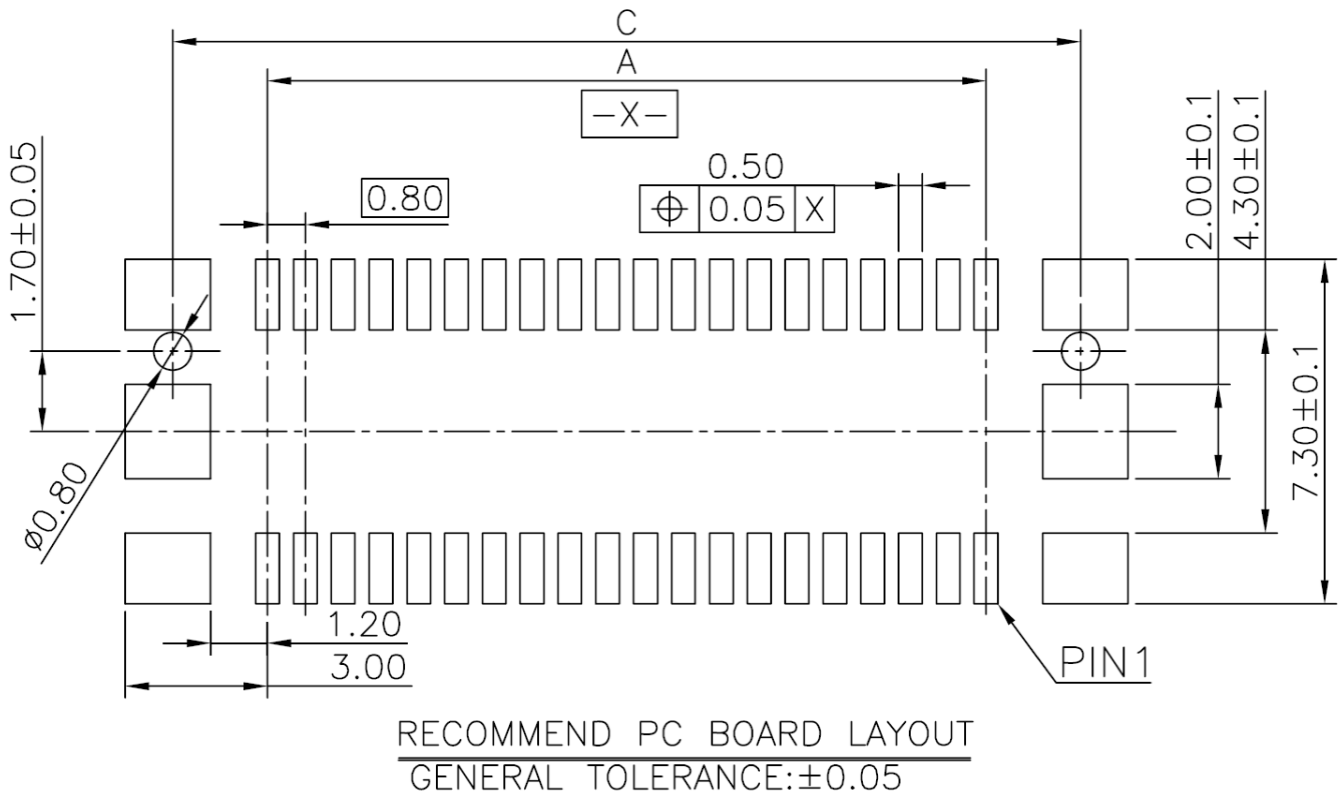
Figure 16 – Motherboard connector ISO view

Vendor	Lotes Co., Ltd.
Part	AEA-BTB-034-450-T
Contact	Crystina Xia (sa28@lotes.com.cn) + 86-15989126305
Website	www.lotes.cc

Vendor	Lotes Co., Ltd.
Part	AEA-BTB-035-350-T
Contact	Crystina Xia (sa28@lotes.com.cn) + 86-15989126305
Website	www.lotes.cc

3.3.3. Connector PCB Footprints

3.3.3.1. Module connector



Notes:

1. Dim A = 19.2 mm
2. Dim C = 23.2 mm



- micro-BMC Open Project Specification R0.35

3.3.4. Pin assignments by pin number

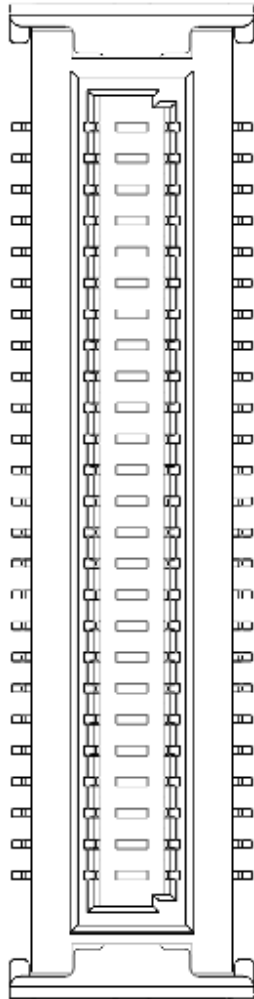


Figure 17 – μ BMC module connector pin view

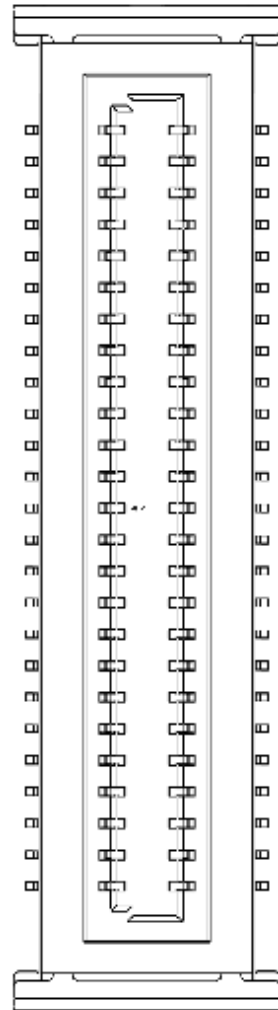


Figure 18 – Motherboard connector pin view

3.3.4.1. μ BMC module - connector P1

Signal	Pin	Pin	Signal
VMON_IN7	2	1	GND
VMON_IN6	4	3	ETH1_MDI_D_DN
VMON_IN5	6	5	ETH1_MDI_D_DP
VMON_IN4	8	7	GND
VMON_IN3	10	9	ETH1_MDI_C_DN
VMON_IN2	12	11	ETH1_MDI_C_DP
VMON_IN1	14	13	GND
VMON_IN0	16	15	ETH1_MDI_B_DN
GND	18	17	ETH1_MDI_B_DP
FAN_TACH1	20	19	GND
FAN_PWM1	22	21	ETH1_MDI_A_DN
FAN_TACH0	24	23	ETH1_MDI_A_DP
FAN_PWM0	26	25	GND
GND	28	27	ETH2_MDI_D_DN
I2C_PECI_SDA	30	29	ETH2_MDI_D_DP
I2C_PECI_SCL	32	31	GND
I2C_SENSORS_SCL	34	33	ETH2_MDI_C_DN
I2C_SENSORS_SDA	36	35	ETH2_MDI_C_DP
UART_SOL_TXD	38	37	GND
UART_SOL_RXD	40	39	ETH2_MDI_B_DN
GND	42	41	ETH2_MDI_B_DP
UART_CLI_RXD	44	43	GND
UART_CLI_TXD	46	45	ETH2_MDI_A_DN
ETH2_LED_LINK_N	48	47	ETH2_MDI_A_DP
ETH2_LED_ACT_N	50	49	GND

3.3.4.2. μ BMC module - connector P2

Signal	Pin	Pin	Signal
GND	2	1	VDD
GND	4	3	VDD
GND	6	5	VDD
GND	8	7	VDD
GND	10	9	VDD
GND	12	11	GND
GND	14	13	USB_VM_DN
GND	16	15	USB_VM_DP
GND	18	17	GND
SPI_CPU_SCK	20	19	PSU_PWRGD3
SPI_CPU_CS_N	22	21	PSU_PWRGD2
SPI_CPU_MISO	24	23	PSU_PWRGD1
SPI_CPU_MOSI	26	25	PSU_PWRGD0
UBMC_PRESENT_N	28	27	CPU_RSTBTN_N
SPI_CPU_MUX_EN	30	29	CPU_PWRBTN_N
GPIO7	32	31	CPU_PROCHOT_N
GPIO6	34	33	CPU_THERMTRIP_N
GPIO5	36	35	CPU_ERROR_N
GPIO4	38	37	CPU_SLP_S3_N
GPIO3	40	39	CPU_SLP_S4_N
GPIO2	42	41	CPU_SLP_S5_N
GPIO1	44	43	CPU_PLTRST_N
GPIO0	46	45	CPU_HEARTBEAT_N
GND	48	47	VDD_USB
GND	50	49	VDD_USB

3.3.4.3. Motherboard - connector P1

Signal	Pin	Pin	Signal
GND	1	2	VMON_IN7
ETH1_MDI_D_DN	3	4	VMON_IN6
ETH1_MDI_D_DP	5	6	VMON_IN5
GND	7	8	VMON_IN4
ETH1_MDI_C_DN	9	10	VMON_IN3
ETH1_MDI_C_DP	11	12	VMON_IN2
GND	13	14	VMON_IN1
ETH1_MDI_B_DN	15	16	VMON_IN0
ETH1_MDI_B_DP	17	18	GND
GND	19	20	FAN_TACH1
ETH1_MDI_A_DN	21	22	FAN_PWM1
ETH1_MDI_A_DP	23	24	FAN_TACH0
GND	25	26	FAN_PWM0
ETH2_MDI_D_DN	27	28	GND
ETH2_MDI_D_DP	29	30	I2C_PECI_SDA
GND	31	32	I2C_PECI_SCL
ETH2_MDI_C_DN	33	34	I2C_SENSORS_SCL
ETH2_MDI_C_DP	35	36	I2C_SENSORS_SDA
GND	37	38	UART_SOL_TXD
ETH2_MDI_B_DN	39	40	UART_SOL_RXD
ETH2_MDI_B_DP	41	42	GND
GND	43	44	UART_CLI_RXD
ETH2_MDI_A_DN	45	46	UART_CLI_TXD
ETH2_MDI_A_DP	47	48	ETH2_LED_LINK_N
GND	49	50	ETH2_LED_ACT_N

3.3.4.4. Motherboard - connector P2

Signal	Pin	Pin	Signal
VDD	1	2	GND
VDD	3	4	GND
VDD	5	6	GND
VDD	7	8	GND
VDD	9	10	GND
GND	11	12	GND
USB_VM_DN	13	14	GND
USB_VM_DP	15	16	GND
GND	17	18	GND
PSU_PWRGD3	19	20	SPI_CPU_SCK
PSU_PWRGD2	21	22	SPI_CPU_CS_N
PSU_PWRGD1	23	24	SPI_CPU_MISO
PSU_PWRGD0	25	26	SPI_CPU_MOSI
CPU_RSTBTN_N	27	28	UBMC_PRESENT_N
CPU_PWRBTN_N	29	30	SPI_CPU_MUX_EN
CPU_PROCHOT_N	31	32	GPIO7
CPU_THERMTRIP_N	33	34	GPIO6
CPU_ERROR_N	35	36	GPIO5
CPU_SLP_S3_N	37	38	GPIO4
CPU_SLP_S4_N	39	40	GPIO3
CPU_SLP_S5_N	41	42	GPIO2
CPU_PLTRST_N	43	44	GPIO1
CPU_HEARTBEAT_N	45	46	GPIO0
VDD_USB	47	48	GND
VDD_USB	49	50	GND

3.4. Electrical specification

3.4.1.1. Absolute maximum ratings

Symbol	Parameter	Min	Max	Units	Notes
V_{DD}	Power supply voltage	0	3.6	V	
V_{DD_USB}	USB power supply voltage	0	5.5	V	
V_{IN}	Voltage applied to any pin with respect to GND	0	3.6	V	2
T_A	Operating ambient temperature	--	--	°C	1

Notes:

1. Temperature range is application-specific. Different μ BMC implementations may be rated for various commercial, industrial, or extended temperature ranges. Reference design is rated for 0 to +70 C operation.
2. I/O pins are not 5V tolerant.

3.4.1.2. Normal operating conditions

Symbol	Parameter	Min	Typ	Max	Units	Notes
V_{DD}	Power supply voltage	3.0	3.3	3.6	V	
V_{DD_USB}	USB power supply voltage	4.5	5.0	5.5	V	
I_{DD}	Power supply current	0		1.82	A	
I_{DD_USB}	USB power supply current	0		0.5	A	

Symbol	Parameter	Min	Typ	Max	Units	Notes
T_A	Operating ambient temperature	--	--	--	°C	1

Notes:

1. Temperature range is application-specific. Different μ BMC implementations may be rated for various commercial, industrial, or extended temperature ranges. Reference design is rated for 0 to +70 C operation.

3.4.1.3.DC specifications

Symbol	Parameter	Min	Typ	Max	Units	Notes
V_{IL}	Input low voltage TTL _{IN} , TTL _{INOUT} pins			0.8	V	
V_{IH}	Input high voltage TTL _{IN} , TTL _{INOUT} pins	2.0			V	1
A_{VIN}	Analog input voltage range (VMON_IN pins)	0		1.8	V	
I_{IL}	Input leakage current TTL _{IN} , TTL _{INOUT} pins ALG _{IN} pins			20 20	μ A μ A	2
V_{OL}	Output low voltage TTL _{OUT} , TTL _{INOUT} pins OD pins			0.8 0.8	V V	
V_{OH}	Output high voltage TTL _{OUT} , TTL _{INOUT} pins	2.0			V	
I_{OL}	Output low current TTL _{OUT} , TTL _{INOUT} pins OD pins			5 5	mA mA	
I_{OH}	Output high current TTL _{OUT} , TTL _{INOUT} pins			5	mA	

Notes:

1. I/O pins are not 5V tolerant.

2. Nominal leakage current for input pin. Some SoC parts will implement internal weak pullup or pulldown resistors that may affect leakage current.

3.4.1.4.AC specifications

3.4.1.4.1. SPI bus

The SPI master bus speed is configurable by the Remote Administrator to match the capabilities of the motherboard components (flash, multiplexer, etc.).

3.4.1.4.2. I2C bus

The I2C clock and data signals conform to all specifications in the following documents.

- I2C-bus Specification and User Manual, Rev.03
- System Management Bus (SMBus) Specification, Version 2.0

Symbol	Parameter	Min	Typ	Max	Units	Notes
F _{I2C}	I2C clock frequency (I2C_SENSORS_SCL, I2C_PECI_SCL)	0	400	400	KHz	

3.4.1.4.3. UART interface

The µBMC implements 2 separate TTL level standard 16C550-style UART interfaces with different fixed baud rates.

- Host CPU serial console for SOL
- CLI interface for µBMC local access, debug, and firmware updates

Symbol	Parameter	Min	Typ	Max	Units	Notes
F _{BAUD_CPU}	Baud rate for host CPU serial console UART_SOL_RXD, UART_SOL_TXD	115.2	115.2	115.2	Kbps	
F _{BAUD_CLI}	Baud rate for µBMC CLI UART_CLI_RXD, UART_CLI_TXD	1.0	1.0	1.0	Mbps	1

Notes:

1. CLI baud rate set higher to reduce local firmware update times. Modern SoCs and UARTs commonly support this speed.

3.4.1.4.4. Ethernet interface

The Ethernet interface signals conform to all 1000base-T specifications in the following document.

- IEEE 802.3

3.4.1.4.5. USB interface

The USB interface signals conform to all specifications in the following document.

- Universal Serial Bus Specification, Revision 2.0.

3.5. Mechanical specification

Notes:

- Define second form factor that includes 3rd high-speed connector for supporting PCIe, VGA, LPC, etc.

3.5.1. Overview

The μ BMC module with bottom-side connectors mates with a connector site on the server motherboard, as shown in Figure 19. Dimensional details of μ BMC and server motherboard are described in following sections.

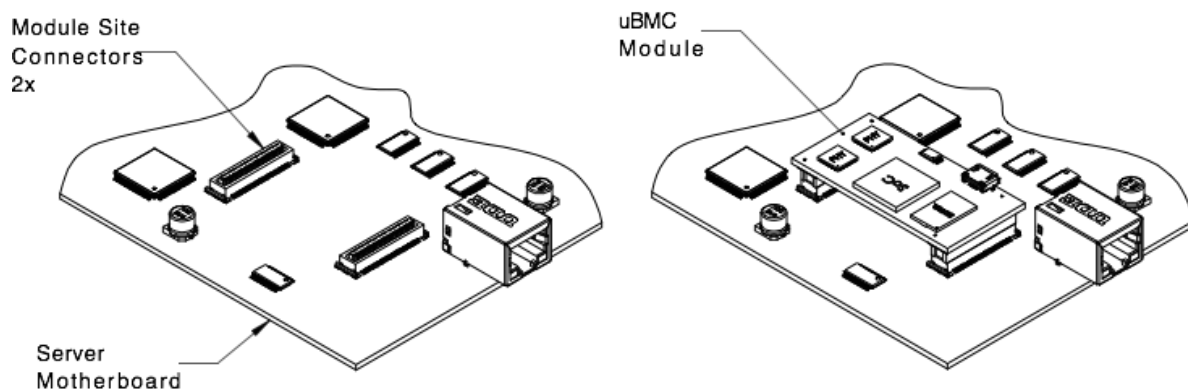
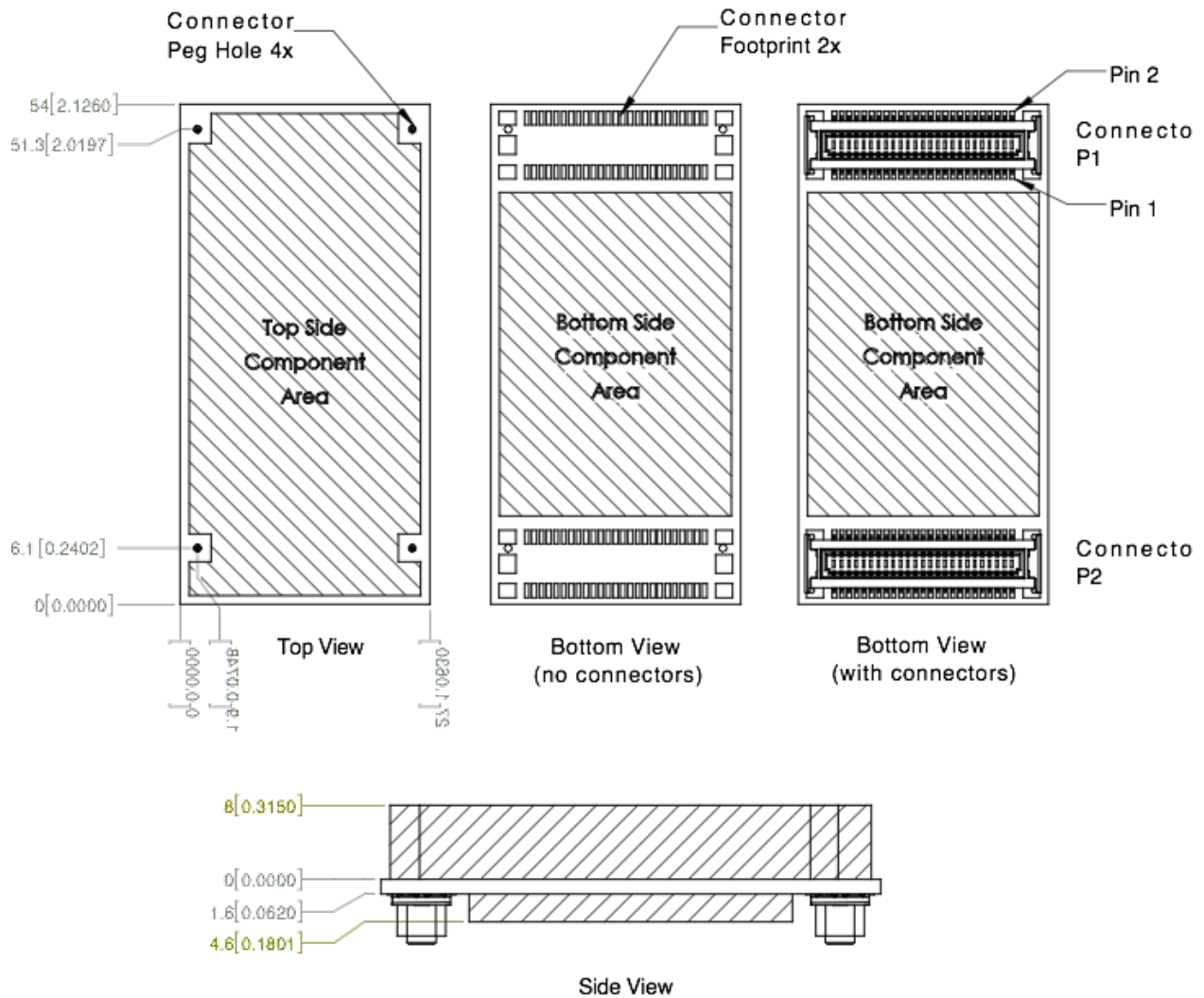
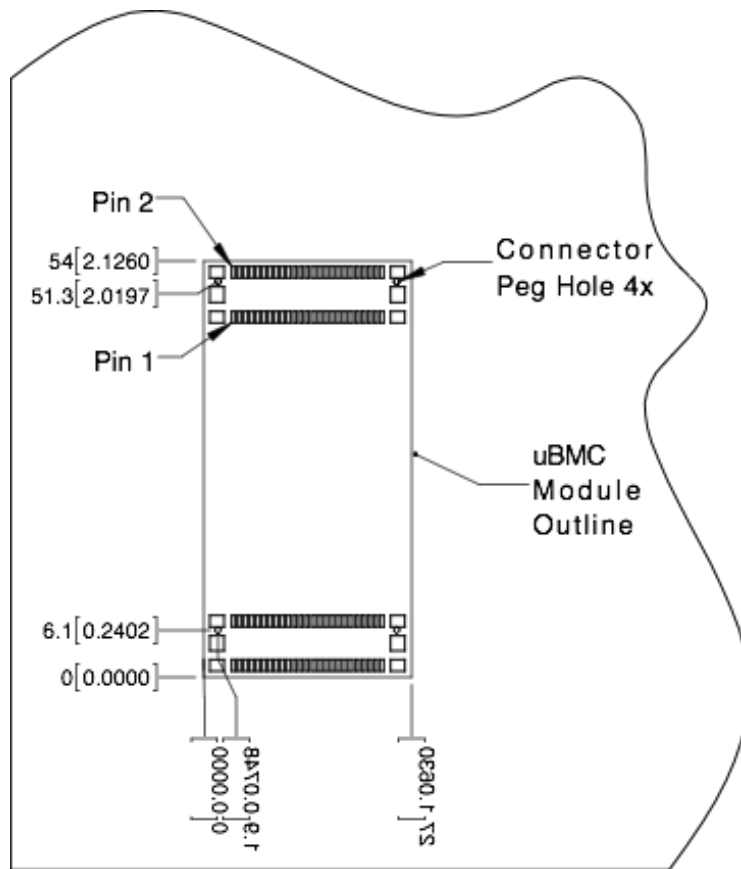


Figure 19 –Motherboard connector site for μ BMC module

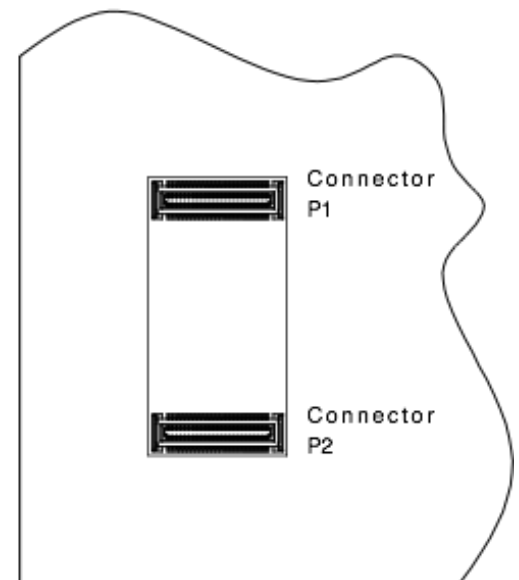
3.5.2. μ BMC module dimensions



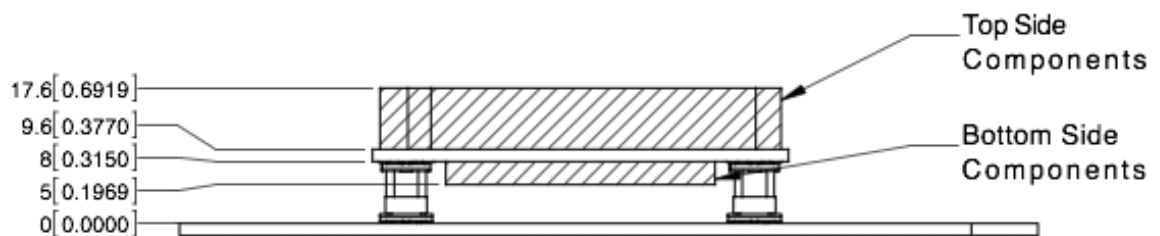
3.5.3. Server motherboard dimensions



uBMC Module Outline and Connector Locations



uBMC Module Connector Designations



uBMC Module Component Height and Keepout

4. FIRMWARE

4.1. Introduction

The uBMC requires a suite of reliable, robust yet compact firmware to allow it to monitor and supervise the host system to which it is attached.

Security, network communications and remote upgradability are also of great importance in the design of the firmware. Several components are required:

4.2. Components

4.2.1. Bootloader

Suitable bootloaders include “Das U-Boot”, Grub, and proprietary bootloaders. The final choice will depend on the SoC selected, and the support available for it.

4.2.2. Kernel

Linux or FreeBSD are considered suitable operating systems for the uBMC. They both offer extensive, secure and reliable network communications capabilities, interface with the above listed bootloaders and have exceptional reliability.

Where possible, a compact “all-in-one: suite of OS utilities will be used to minimize the required storage space and power consumption. Busybox or Toybox plus the individual GNU software components required to achieve the desired functionality will be provided.

4.2.3. Development tools

Standard GNU (or similar) development tools for the SoC will be employed. Initially the firmware images will be created by the usual command line make processes.

4.2.4.

The remote admin part of the firmware will be able to operate from behind other NAT devices (I.O.W. it does not need to be the edge device” and will use a secure polled architecture designed to both maximize security and minimize bandwidth/overhead.

The technologies used to implement this may include both polled and persistent systems based on NETCONF, Saltstack, SOAP and/or protocols.

Capabilities supported include:

- uBMC
 - Automated registration
 - Automatic firmware updates of all components
 - Remote CLI to the uBMC
 - Remote log forwarding
 - Automatic recovery from bad updates/configuration

- Remote configuration
 - Local access and updates via serial console
- Host server
 - Remote CLI access
 - Console port forwarding
 - Remote programming
 - Remote configuration
 - Fall-back to previous configuration
 - Remote restart
 - Remote power control (where supported)
 - Remote monitoring of all available host information, to include:
 - Health
 - Logs
 - Network
 - Power
 - Temperature
 - Event logs
 - Memory
 - Mass storage
 - Alerts
 - Bootflash
 - Birtual media
 - etc

5. REMOTE ADMINISTRATOR SOFTWARE TOOLS

5.1. High level requirements

The tools at the central site will necessarily be relatively complex since network communications, database, display, and security technologies will all come into play. For this reason a client/server architecture is envisaged wherein a secure webserver will provide access to the tools and web browsers will provide the distributed display capability within an organization.

By employing this model the complexity, and more importantly the security of the system can be concentrated at a single location within an organization while still allowing multiple geographically dispersed persons access to all of its facilities.

In very high security environments, there can be a one to one relationship between the server and client (browser) components, with no remote user access provided.

The main components of the system include:

1. Web server
2. Advanced access control
3. Database
4. Display logic
5. Network communications
6. Logging (Operation and access)

5.1.1. Access control

Physical access control is external and/or internal firewall functionality at the server site. The server will permit access only via HTTPS with the option to require mutual authentication, such that both the client and server must be in possession of a valid security certificate.

Within the central display application, a further level of access control is provided and that control is hierarchical in that N level of access are possible, each with different levels of access.

The highest access level is to be "System administrator" who can do anything, the lowest being "User - View Only" able to view the status of a single remotely supervised site.

A multiplicity of other access levels is to be provided controlling who may do what to which remote sites, and to the system.

5.1.2. Database

A database is required to facilitate the storage of at least the following items:

- User records (strongly encrypted)
 - Credentials
 - History
 - Privileges
 - Demographics
 - Etc.
- uBMC devices under supervision
 - Network parameters
 - Network switch configuration
 - Active interfaces
 - Local access (console) credentials
 - Demographics
 - Enrolment status
 - History
 - Operating system specification
 - Firmware components (link)
 - Host specification
 - Monitoring points
 - Alarms and thresholds
 - Etc.
- Network communication parameters (server)
 - Hostname
 - Address
 - Certificates
 - Other required network data
- Devices (servers) under supervision
 - Physical configuration (CPU, memory, storage etc.)
 - Software configuration
 - Components
 - Locations
 - Versions
 - Config files
 - Etc.
 - Bootloader image (link)
 - Console credentials
 - Network configuration
 - Notes (secure wiki link?)

- Monitoring points
- Alarms and thresholds
- Alerting (Whom to notify and how)

Firmware and software images will be stored on disk outside of the database, with the database containing links (paths) to these components.

5.1.3. Display logic

The most complex component of the central site system is the display log/application. Its purpose is allow an authorized user to interact with the database and with the network of connected uBMC agents for the purpose of monitoring and controlling both the agents and the servers/machines they supervise.

The user interface will provide the following main functional blocks:

1. Login
2. Site Administration
3. Dashboard
4. Device enrollment
5. Security (certificate) management
6. uBMC fleet administration
7. uBMC details
 - a. Status
 - b. History
 - c. Configuration
 - d. Update
 - e. CLI access
8. Remote host
 - a. Status
 - b. History
 - c. Configuration
 - d. Update
 - e. CLI access
 - f. Log retrieval
 - g. Control (Shutdown, restart, sleep etc)
9. Alarms and notification
10. Logs
11. Search

Access to the system is through a system of web pages each implementing one or more of the functions listed above. In general web pages will present a combination of graphical display elements (charts, graphs, meters and

alert icons), tables and input elements. All web pages presented by the system conform to the following design philosophy as appropriate to their content:

- A consistent, configurable color and graphic design scheme is used
- A configurable logo is displayed in the page header
- Breadcrumbs are employed in each page header to allow the user to easily see their location in the web
- A tabbed, “top menu” design is employed so that a user may navigate directly to any section without the need to traverse their way up the hierarchy
- The “Logout” link is displayed in every page header
- A configurable corporate identity and copyright notice appears in every page footer
- Where tables are presented, they may be sorted by clicking each column header
- Table rows are clickable to allow more detail to be viewed, or the tabulated entity to be edited/configured.
- Every page/section displays a “Help” link that will display information pertinent to the operation of that page.

The function of each section of the interface is as follows:

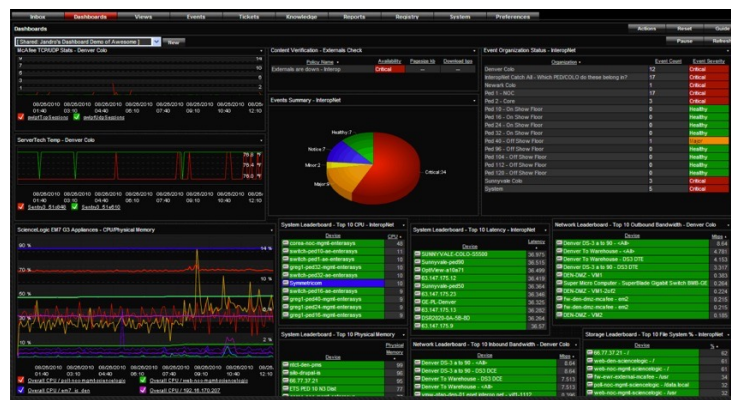
5.1.3.1.Login

A user that is not currently logged in (does not possess a valid session identity) will land on the login page, no matter which page they attempt to access. This page will provide Username and Password input elements as well as links to allow the user to recover forgotten credentials. Such links will generate a message to the administrator, but will not directly allow the user to retrieve any component of their credentials.

A configurable maximum number of login attempts will be enforced, after which the user’s account will be disabled and the administrator notified.

5.1.3.2.Dashboard

The system dashboard is one of the most important components of the system. It may be configured to show the status of remotely supervised hosts, the uBMC fleet, network operations, and other system information “at a glance”.



The dashboard may be configured and sized to facilitate its display on a large screen monitor in an operations center or public display area.

Possible display items include:

- Hosts online/offline
- Fleet update status
- Current server load
- Remote uBMCs awaiting enrollment
- Etc.

5.1.3.3.Site administration

This section facilitates the administration of user credentials, user privileges, web display settings, remote uBMC enrolment, firmware and software file provisioning, as well as a number of other administrative functions.

Access to this section is limited to those with administrator privileges, and sub-sections require more specific permission.

5.1.3.4.uBMC fleet administration

From this interface section the fleet of remote uBMC devices may be viewed, configured and maintained. The interface consists of a sortable, searchable list/table of devices which will show their current status (enrolled, on-line, up to date etc)

A device may be inspected/configured by clicking the table row in which it is displayed.

5.1.3.5.uBMC details

From the uBMC details page, a user may inspect the detailed status of the device, including its recent history. It may also be configured and updated.

Authorized users may also gain CLI (console) access to the remote uBMC through this display. An input element is provided to allow the user to type shell commands. A display area shows the response received from the system.

Shell Output - ps ax | head -n 10

PID	TT	STAT	TIME	COMMAND
0	-	DLs	0:01.56	[kernel]
1	-	ILs	0:00.06	/sbin/init --
2	-	DL	0:00.00	[crypto]
3	-	DL	0:00.00	[crypto returns]
4	-	DL	0:00.90	[cam]
5	-	DL	1:23.60	[pf purge]
6	-	DL	0:00.00	[sctp_iterator]
7	-	DL	0:05.70	[pagedaemon]
8	-	DL	0:00.00	[vmdaemon]

Execute Shell Command

<
Execute
>
Clear

Different system users may be assigned different privilege levels on the uBMC (“regular” user, super user, root user etc)

The commands entered are filtered to the extent possible to prevent accidental, irrecoverable damage to the device (e.g.: format /dev/sda1) but this is by nature imperfect. This capability must be used with extreme care.

5.1.4. Remote hosts

The remote hosts (servers) supervised by the uBMC are also listed in tabular form. Like the uBMC, table, a row may be clicked to gain access to the configuration, status and control center for each host.

5.1.4.1. Status

The status page allows the user to view any of the parameters listed earlier in this document.

It is also possible to set alarms and to configure the action to be taken should those alarms be triggered.

5.1.4.2. Remote configuration

The configuration section allows the user to view the current configuration of the server, and to make changes to that configuration remotely. Package may be monitored, updated, installed or removed, for example.

A very important capability offered by the remote configuration system is the ability to view, retrieve and copy configurations either from another remote host, or from a list of configuration templates stored in the database.

As an example, if the user wished to bring up a new host at a remote retail location in the state of Florida, they might view the list of templates and select “NX4302-Florida Branch Office”. This would retrieve a pre-stored configuration designed for a model NX4302 host, with the exact configuration required by locations in Florida.

This may then be customized for this particular site (credentials, network config etc) before being saved, and thereby deployed to the remote site.

5.1.4.3. Remote console access

A remote host may be accessed via the shell (command line) in exactly the manner described for the uBMC above. Even greater care must be taken in the use of this command. The opportunity for damage is considerable!

5.1.4.4. Log retrieval

Logs from the remote host may optionally be compressed and checksummed before regular transmission to the administrative system. Those logs may then be viewed/searched for information of interest.

5.1.5. Searching

A dedicated search interface is provided to allow a user to quickly locate uBMC devices, remote hosts, users, logs, configurations and other information. The information searched, and the results returned depend on the privileges assigned to the current users.

Administrators will be able to search almost anywhere of anything. End users may only be able to search for a list of the devices to which they have been allowed access

6. NETWORK COMMUNICATIONS

6.1. Overview

The communication mechanism used to transfer configuration settings to a remote uBMC or its host, and to retrieve status information from it is a key component of the system. It has been suggested above that technologies such as SOAP, Saltstack, NETCONF and others are suitable.

In general the network protocol employed will require no justification. If a user wishes to install a uBMC in a network appliance or server, it should not be necessary to request a change to the network configuration, to require a security audit, or convince someone that this product is safe and secure.

Where a change management process does require a review of the uBMC, its technology is such that the review should be straight forward and without concern.

6.2. Requirements

The network communications system should:

- Employ ‘standard’ network protocols whenever possible
- Operate through a NAT device (firewall, router etc.) without the need for a custom configuration in that device
- Provide a high level of security to prevent interception of the data being transferred, or unauthorized access to the uBMC
- Occupy a small bandwidth
- Provide a sufficiently responsive system (See below)
- To the extent possible, the network communications configuration will employ “Zero configuration”, allowing a uBMC to configure itself and at least “phone home”

6.3. Security

The need for extreme security is obvious. A uBMC has access to the inner most workings of its host, so access to the uBMC confers great power.

Almost all network communications will be encrypted using the most secure network protocol currently available, and when “current” is no longer “current”, it must be easily changeable. The only exception to the need for strong encryption might be a poll message in which a uBMC asks the system: “Do you have anything for me?” Polling messages contain no sensitive information, and bandwidth/processing time requirements are more important.

A second layer of security will be provided by encrypting all network message payloads before submission to the transport system. If a new configuration is to be transmitted, for example, that configuration will be blowfish encrypted before transmission, as well as being encrypted via TLS during transmission. The Blowfish key will be unique to each uBMC.

6.4. Responsiveness

The responsiveness provided by the system will depend on the communications capabilities in use by a particular device. In most cases, however, the message exchange will be essentially “real-time”. It should rarely if ever be necessary to resort to having a client uBMC poll the management system.

Two classes of responsiveness are required: When configuration, log, status and similar messages are being exchanged, the response time is relaxed. Depending on the system load factors, worst case response times measured in tens of seconds may be acceptable.

When remote command line operation is in use, the response time needs to be much shorter. Certainly no more than a very few seconds.

In practice, however, the response times for all types of operation may be equally fast unless a polling method is in use.

7. DESIGN COLLATERAL LOCATIONS

Hardware reference design repository [location](#)

Firmware reference design repository [location](#)

Software reference design repository [location](#)

8. ABOUT ADI ENGINEERING

8.1. Company information

Silicom Ltd. is an industry-leading provider of high-performance networking and data infrastructure solutions. Designed primarily to improve performance and efficiency in Cloud and Data Center environments, Silicom's solutions increase throughput, decrease latency and boost the performance of servers and networking appliances, the infrastructure backbone that enables advanced Cloud architectures and leading technologies like NFV, SD-WAN and Cyber Security. Our innovative solutions for high-density networking, high-speed fabric switching, offloading and acceleration, which utilize a range of cutting-edge silicon technologies as well as FPGA-based solutions, are ideal for scaling-up and scaling-out cloud infrastructures.

Silicom products are used by major Cloud players, service providers, telcos and OEMs as components of their infrastructure offerings, including both add-on adapters in the Data Center and stand-alone virtualized/universal CPE devices at the edge.

Silicom's long-term, trusted relationships with more than 150 customers throughout the world, its more than 400 active Design Wins and more than 300 product SKUs have made Silicom a "go-to" connectivity/performance partner of choice for technology leaders around the globe.

8.2. Contacts

Corporate Headquarters

Silicom Ltd.
14 Atir Yeda St. Kfar Sava
4464323, Israel

Phone: (972) 9-7644555

9. ABOUT NETGATE

9.1. Company information

Rubicon Communications, LLC (Netgate) is a high quality communications appliance manufacturer that designs highly configurable computers engineered for low-power, high-throughput, dependable deployments. We offer quick deployment with off-the-shelf systems, specialized custom product solutions and an end-to-end OEM service program for large customers.

Netgate directly supports Open Source software and Open Source hardware projects by providing management, engineering and support resources for the pfSense project. We also distribute Open Source hardware components such as the MinnowBoard Turbot.

We have a depth of experience in networking, computer security, embedded software and hardware engineering unmatched by larger teams. Our engineers hold multiple patents. We have industrial design, large network deployment and management, operating system, BIOS, device driver and software engineering experience starting with Unix in the 1980s. We have employees in the United States, European Union and Brazil.

9.2. Contacts

Corporate Headquarters

Rubicon Communications, LLC
7212 McNeil Road
Suite #204
Austin, TX 78729

Phone: +1-512-646-4100

Corporate sales: sales@netgate.com

General information: info@adiengineering.com

Website: www.netgate.com

10. ABOUT PFSENSE

10.1. Project information

The pfSense project is a free network firewall distribution, based on the FreeBSD operating system with a custom kernel and including third party free software packages for additional functionality. pfSense software, with the help of the package system, is able to provide the same functionality or more of common commercial firewalls, without any of the artificial limitations. It has successfully replaced every big name commercial firewall you can imagine in numerous installations around the world, including Check Point, Cisco PIX, Cisco ASA, Juniper, Sonicwall, Netgear, Watchguard, Astaro, and more.

10.2. Contacts

General information: coreteam@pfsense.org
Website: www.pfsense.org

11. LEGAL

11.1. License

µBMC reference design hardware design collateral, firmware design collateral, and software design collateral are all available for download and use under Creative Commons “CC” (BY – Attribution Alone) license terms. No per-unit-shipped royalties or fees are associated with use of the design collateral. CC (BY) means that licensees may copy, distribute, display, and perform the work and make derivative works based on it only if they give the author(s) or licensor(s) credit.

11.2. Disclaimer

This specification is provided “as is” with no warranties whatsoever, including any warranty of merchantability, non-infringement, or fitness for any particular purpose. The authors of this specification disclaim all liability, including liability for infringement of any proprietary rights, relating to use or implementation of information in this specification.