

ADVANCED PLATFORM MANAGEMENT USER'S GUIDE

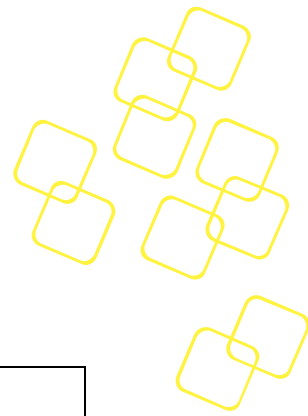
REVISION 0.1

DATE 2018/04/19

SKY-8201

**COMPACT 2U HIGH PERFORMANCE SERVER BASED
ON INTEL® XEON™ PROCESSOR SCALABLE FAMILY**





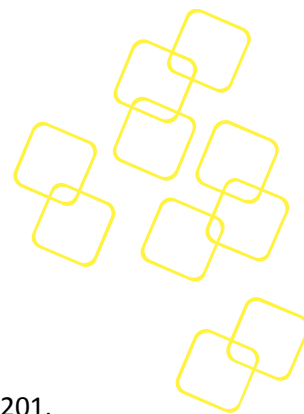
Revision History

Date [mm/dd/yyyy]	Revision	Modifications
04/19/2018	0.1	Initial version –draft based on SKY-8201
05/21/2018	0.2	Update pictures

© Copyright 2017– Advantech Co., Ltd.

All Rights Reserved

Advantech Co., Ltd. reserves the right to make improvements in the products described in this manual at any time without notice. No part of this manual may be reproduced, copied, translated or transmitted in any form or by any means without the prior written permission of Advantech Co., Ltd. Information provided in this manual is intended to be accurate and reliable. However, Advantech Co., Ltd. assumes no responsibility for its use, nor for any infringements of the rights of third parties, which may result from its use.



About this manual

The target audience of this manual includes users, developers and technicians. This document describes the features, functions and operations of the Advanced Platform Management provided by the Baseboard Management Controller (BMC) on the SKY-8201.

This manual is organized as follows:

- Section 1: **Getting Started** helps with the first steps of using the Advanced Platform Management on the SKY-8201.
- Section 2: **BMC Functionalities** provides detailed descriptions of the SKY-8201's BMC and its features.
- Section 3: **BMC Firmware and BIOS Upgrade** describes the failsafe mechanism of BMC and BIOS upgrade as well as the steps of BMC or BIOS firmware upgrade process.
- Section 4: **Essential Information for Advanced Platform Management** provides best practices and related information that may be helpful for the operation and troubleshooting of the SKY-8201 Platform.
- **Appendices** section provides supplemental information referenced in the other sections of this document and BMC firmware release policy.

This document covers:

- BMC Firmware version 1.0 and later

Revision specific features or implementations – if any – are identified using (*Rev. v010*) in the related text.

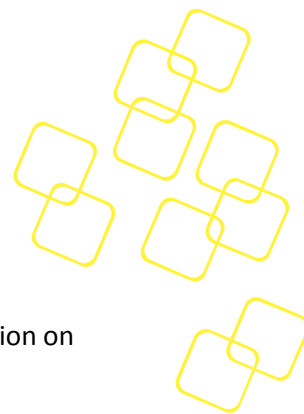
Some sections make assumptions on other 3rd party or Advantech software and the related versions of such software:

- IPMITool version 1.8.18 or greater
- SKY-8201 BIOS version 0.30 and later

Disclaimer

The information in this guide is subject to change without notice.

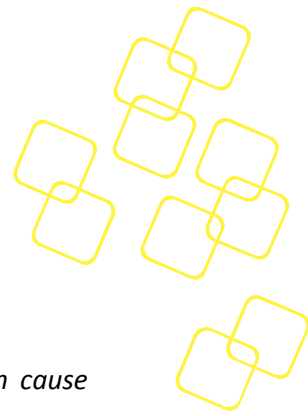
The figures in this guide can be the examples to all Advantech products with Advantech BMC solution. The figures are not 100% captured from SKY-8201; therefore, the product name and product information in the figures might be different.



Useful documents

If you cannot find the information you are looking for or need more detailed information on a specific topic, please refer to the list of additional documents and other sources of information below. Please contact your Advantech representative if you need help on obtaining these documents or still cannot find what you are looking for.

- *Intelligent Platform Management Interface Specification, Version 2.0, Revision 1.1*, October 1, 2013 – E7 April 21, 2015.
- *IPMI – Platform Management FRU Information Storage Definition, V1.0*, Document Revision 1.1, September 27, 1999.
- *IPMI - Platform Event Trap Format Specification V1.0*, Document Revision 1.0, December 7, 1998.
- *PICMG® 3.0 Revision 3.0 AdvancedTCA Base Specification*, March 24, 2008.
- *HPM.1, Hardware Platform Management IPM Controller Firmware Upgrade Specification R1.0*, PCI Industrial Computer Manufacturers Group (PICMG®) May 4, 2007.
- Information on Intel CPUs, chipsets and NIC silicon can be found at www.intel.com
- SKY-8201 User Manual
- Getting Started Guide for Advantech SKY-8201 QuickStart Linux Image
- ipmitool how-to can be found at: <http://linux.die.net/man/1/ipmitool>
- An introduction to IPMI can be found at : <http://www.intel.com/content/www/us/en/servers/ipmi/ipmi-home.html>



Warnings, Cautions and Notes



Warning! Warnings indicate conditions, which, if not observed, can cause personal injury.



Caution! Cautions are included to help you avoid damaging hardware or losing data.



Note! Notes provide additional information.

We appreciate your input

Please let us know of any aspect of this product, including the manual, which could use improvement or correction. We appreciate your valuable input in helping make our products and documentation better.

Please send all such to: ncg@advantech.com

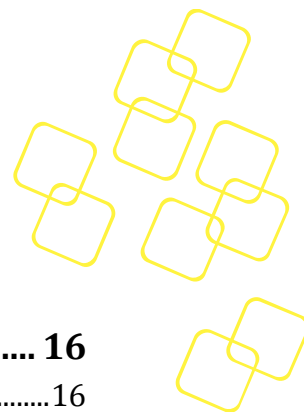
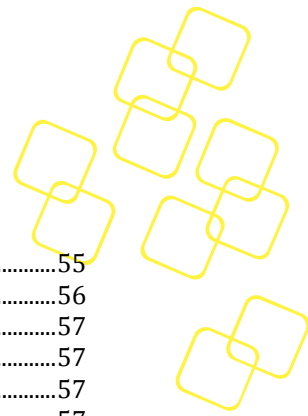



Table of Contents

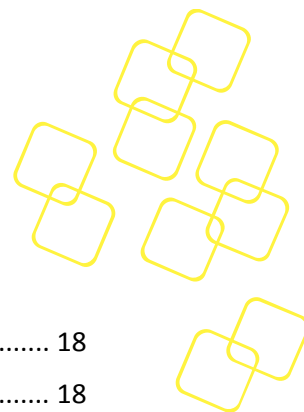
1. GETTING STARTED	16
1.1 ADVANCED PLATFORM MANAGEMENT & BMC INTRODUCTION	16
1.1.1 Integrated Platform Management Interface.....	16
1.1.2 Software Support	16
1.1.3 Advanced Features	16
1.1.4 The Advantech QuickStart Linux Image for SKY-8201.....	17
1.2 CONNECTING TO THE BMC	19
1.2.1 Internal Connection via the System Interface	19
1.2.2 External Connection via the LAN Interface	19
1.2.2.1 BMC Dual NIC Support.....	20
1.2.2.2 Configuring the BMC's LAN IP address through the KCS Interface.....	20
1.2.2.3 Enable LAN access to the BMC	21
1.3 CONNECTING TO THE X86 HOST OVER SOL	22
1.4 GETTING HELP: TECHNICAL SUPPORT AND ASSISTANCE.....	25
2. BMC FUNCTIONALITIES	27
2.1 OVERVIEW	27
2.1.1 System Interface	28
2.1.2 LAN Interface	29
2.2 OEM IPMI COMMANDS	29
2.2.1 Advantech OEM IPMI Commands	29
2.2.1.1 Command code List	29
2.2.1.2 Advantech OEM Command 'Get HW Revision'	30
2.2.1.3 Advantech OEM Command 'SEL Mode Configuration'	31
2.2.1.4 Advantech OEM Command 'Read Port 80 (BIOS POST Code)'	32
2.2.1.5 Advantech OEM Command 'Reload NVRAM Defaults'	33
2.2.1.6 Advantech OEM Command 'Reload BMC Default Configuration'	33
2.3 BMC WATCHDOG.....	34
2.4 FRU INFORMATION.....	34
2.5 SYSTEM EVENT LOG	35
2.6 SENSORS	36
2.6.1 Sensor Data Records & Handling	36
2.6.2 Sensor Types	36
2.6.3 Sensor List	38
2.6.4 Sensor Thresholds	41
2.6.4.1 Voltage Sensors	42
2.6.4.2 Temperature Sensors	44
2.6.4.3 PSU Sensor	45
2.6.5 Discrete Specific Sensors	47
2.6.5.1 BMC Health Sensor	47
2.6.5.2 Version Change Sensor	48
2.6.5.3 BMC Watchdog Sensor	49
2.6.5.4 ACPI Power Sensor	50
2.6.5.5 Processor State Sensor	50
2.6.5.6 Reset Sensor	52
2.6.5.7 FW Progress Sensor	53
2.6.5.8 Power Supply Sensor	54



2.6.5.9	Entity Presence Sensors	55
2.6.5.10	Physical Security Sensor	56
2.6.6	Advantech OEM IPMI Sensors	57
2.6.6.1	OEM BIOS POST code Sensor	57
2.6.6.2	OEM Integrity Sensor	57
2.6.6.3	Sensor Characteristics.....	57
2.6.7	Other SDR Record Types.....	61
2.6.7.1	BMC Device Locator	61
2.7	THERMAL MANAGEMENT	61
2.7.1	Cooling Management	61
2.7.1.1	Temperature Zone 1 – CPU	61
2.7.1.2	Temperature Zone 2 – CPU1	62
2.7.1.3	Thermal Protection	63
2.7.2	Fan Modules	63
2.7.3	Fan Sensors.....	64
2.7.4	Fan Failure Handling	65
2.8	BIOS SYNCHRONIZATION	66
2.8.1	System Time	66
2.8.2	FRU Info.....	66
2.9	BIOS POST WATCHDOG	66
2.10	HPM.2.....	66
2.10.1	Get HPM.x Capabilities	67
2.10.2	LAN Configuration Parameters.....	67
2.10.3	Long IPMI Messages.....	67
2.11	VLAN SUPPORT	67
2.12	BMC SECURITY.....	67
2.13	INTRUSION DETECTION.....	69
2.14	PLATFORM EVENT FILTERING & SNMP TRAPS	69
2.14.1	Simple Network Management Protocol	69
2.14.1.1	Management Information Base.....	69
2.14.1.2	SNMP Community Strings	69
2.14.1.3	SNMP Traps	70
2.14.2	Platform Event Trap (PET)	71
2.14.3	Platform Event Filtering (PEF).....	71
2.14.3.1	PEF Actions	71
2.14.3.2	Alert Policies.....	71
2.14.3.3	Event Filter Table.....	72
2.14.4	BMC PEF Alert Generation	74
2.15	BMC DEFAULT SETTINGS.....	75
2.15.1	User Account	75
2.15.2	PEF	76
2.15.3	LAN	76
2.15.4	SOL.....	78
2.15.5	Power Restore Policy	78
3.	BMC FIRMWARE AND BIOS UPGRADE.....	80
3.1	UPGRADE PLATFORM FIRMWARE.....	80
3.1.1	Upgradeable Components	80
3.1.1.1	Component 0: BMC Boot loader.....	81
3.1.1.2	Component 1: BMC firmware	81



3.1.1.3	Component 2: FPGA	82
3.1.1.4	Component 3: BIOS.....	82
3.1.1.5	Component 4: NVRAM	83
3.1.2	Check Active BMC Firmware Version.....	83
3.1.3	Upgrading BMC Firmware through KCS Interface	84
3.1.4	Upgrading BIOS through KCS Interface.....	85
3.1.5	Upgrading BMC Bootloader through IOL.....	87
4.	ESSENTIAL INFORMATION FOR ADVANCED PLATFORM MANAGEMENT	88
4.1	IDENTIFYING THE SYSTEM.....	88
4.2	LIGHTS OUT CONTROL	88
4.3	CREATING SYSTEM EVENTS FROM AN APPLICATION	89
4.4	KEEPING TIME IN SYNC.....	89
4.5	CHECK PSU PRESENCE	89
4.6	SYSTEM HEALTH STATUS	91
4.6.1	Power State LED ()	92
4.6.2	Chassis Identification LED (ID)	92
4.6.3	Critical Alarm LED (CRT)	92
4.6.4	Major Alarm LED (System Status LED) (MJR)	92
4.6.5	Minor Alarm LED (MNR).....	93
4.6.6	System Status LED	93
4.6.7	FAN Status LED	93
4.6.8	LED Panel for SKY-8201L	93
4.6.9	Audible Alarm.....	95
4.7	READING THE SEL.....	95
A.	APPENDIX: SUPPORTED IPMI COMMANDS	98
B.	APPENDIX: HOW TO INSTALL IPMITOOL.....	103
	WHY DOES IPMITOOL NOT WORK?	103
C.	APPENDIX: FIRMWARE RELEASE AND VERSIONING NUMBER	
	106	



List of Figures

Figure 1: Front Serial Console Port and USB Ports	18
Figure 2: Rear Display, Console Port and USB Ports.....	18
Figure 3: The Location of Management Ports at Rear Panel.....	18
Figure 4: Use 'ipmitool mc info' Command to get BMC Info	19
Figure 5: BMC Dual NIC	20
Figure 6: Print LAN Channel Configuration.....	21
Figure 7: Set up a Static IP for BMC.....	21
Figure 8: Configure Dynamic IP Assignment for the BMC	21
Figure 9: Set User ID, User Name, Password and Privilege Level.....	22
Figure 10: Display Current BMC SOL Setting	23
Figure 11: Activate SOL Session to Access the Advantech Standard Linux Image	24
Figure 12: Access SKY-8201 BIOS in SOL Session.....	24
Figure 13: Enter BIOS Setup Menu in SOL Session	25
Figure 14: BMC Block Diagram	28
Figure 15: Retrieve System FRU Information	35
Figure 16: Fan Module.....	64
Figure 17: Fan Module(20 inches system).....	64
Figure 18: Fan LED location	65
Figure 19: Establish an IOL Connection with Cipher Suite 2	69
Figure 20: SNMP Community Strings in Configuration File	70
Figure 21: PEF Alert Generation Flow	74
Figure 22: Check BMC Firmware Version	84
Figure 23: Command 'ipmitool hpm check'	84
Figure 24: BMC Firmware Upgrade	85
Figure 25: Check Active BMC Firmware Version	85
Figure 26: Upgrade BIOS with Ipmitool.....	86
Figure 27: Check Active BIOS Version.....	87
Figure 28: BMC Bootloader Upgrade with Ipmitool.....	87
Figure 29: Chassis Power Command Usage.....	88
Figure 30: Check the Readings of PSUx Sensors.....	90
Figure 31: Response data byte 4 and 5 of the IPMI 'Get Sensor Reading' Command	90
Figure 32: Assertion State indicates that PSU is absent.....	91

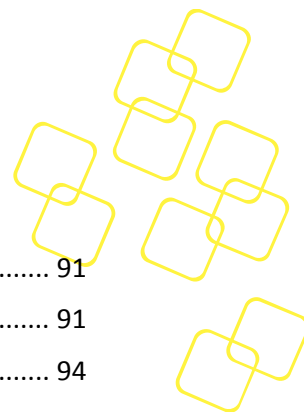
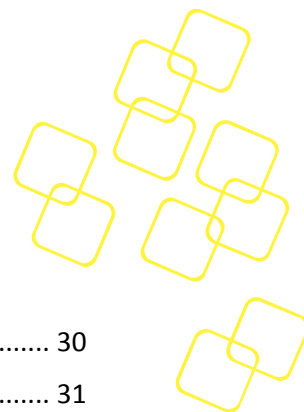


Figure 33: PSU Removal Event in SEL	91
Figure 34: The locations of LEDs.....	91
Figure 35: The locations of front LEDs.....	94
Figure 36: The locations of rear LEDs	95
Figure 37: Use 'ipmitool sel list' Command to Dump the SEL	96
Figure 38: Use 'ipmitool sel elist' Command to Dump the SEL	96
Figure 39: Use 'ipmitool sel save' Command to Store the SEL to a File	96
Figure 40: Check Saved SEL file	97
Figure 41: Use 'ipmitool sel clear' Command to Clear SEL.....	97
Figure 42: Error Message for Executing ipmitool.....	103
Figure 43: Official BMC FW Release	106



List of Tables

Table 1: OEM Command List	30
Table 2: 'Get HW Revision' Command	31
Table 3: 'SEL Mode Configuration' Command.....	31
Table 4: The BIOS Attempts to Add a New SEL Entry	32
Table 5: 'Read Port 80(BIOS POST Code)' Command	32
Table 6: 'Reload NVRAM Defaults' Command	33
Table 7: 'Reload BMC Default Configuration' Command	34
Table 8: System FRU Information.....	35
Table 9: BMC used event and reading type codes	36
Table 10: BMC used sensor type codes.....	37
Table 11: BMC used Entity IDs.....	38
Table 12: BMC Sensor list	41
Table 13: Threshold based sensor event data format.....	42
Table 14: Threshold based sensor supported events.....	42
Table 15: Voltage Sensor list	44
Table 16: Temperature Sensor list	45
Table 17: PSU temperature sensor threshold list	45
Table 18: AC PSU voltage sensor threshold list	46
Table 19: DC 800W PSU voltage sensor threshold list	46
Table 20: AC 1200W PSU current sensor threshold list	46
Table 21: DC 1400W PSU current sensor threshold list	46
Table 22: PSU speed sensor threshold list	47
Table 23: AC 800W PSU power consumption sensor threshold list.....	47
Table 24: AC 1200W PSU power consumption sensor threshold list.....	47
Table 25: AC 1400W PSU power consumption sensor threshold list.....	47
Table 26: BMC Health Sensor event data format.....	48
Table 27: BMC Health Sensor supported events.....	48
Table 28: Version Change Sensor event data format.....	48
Table 29: Version Change Sensor supported events.....	48
Table 30: Version Change Sensor event data byte 2.....	49
Table 31: BMC Watchdog Sensor event data format.....	49
Table 32: BMC Watchdog Sensor supported events.....	49

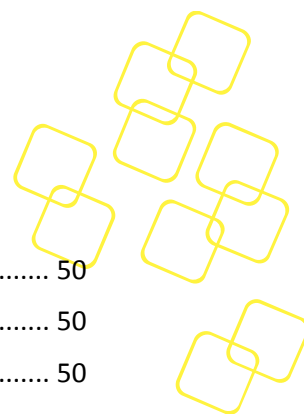


Table 33: BMC Watchdog Sensor event data byte 2	50
Table 34: System ACPI Power State Sensor event data format	50
Table 35: System ACPI Power State Sensor supported events	50
Table 36: Processor Sensor event data format	50
Table 37: Processor Event Sources	51
Table 38: Processor Sensor supported events	51
Table 39: Reset Sensor event data format	53
Table 40: Reset Sensor supported events	53
Table 41: Reset Sensor event data byte 2	53
Table 42: FW Progress Sensor event data format	54
Table 43: FW Progress Sensor supported events	54
Table 44: FW Progress Sensor event data byte 2	54
Table 45: Power Supply Sensor entities	54
Table 46: Power Supply event data format	55
Table 47: Power Supply Sensor supported events	55
Table 48: Power Supply Sensor event data byte 3	55
Table 49: Available presence sensor entities	56
Table 50: Presence Sensor event data format	56
Table 51: Entity Presence Sensor supported readings	56
Table 52: Physical Security Sensor event data format	56
Table 53: Physical Security Sensor supported events	57
Table 54: Integrity Sensor event byte definition	59
Table 55: Integrity Sensor event data table	61
Table 56: Fan Speed and CPU0 Temperature Mapping	62
Table 57: Fan Speed and CPU0 Temperature Mapping (20 inches system)	62
Table 58: Fan Speed and CPU1 Temperature Mapping	62
Table 59: Fan Speed and CPU1 Temperature Mapping(20 inches system)	63
Table 60: Fan sensor list	64
Table 61: Fan sensor list (20 inches system)	64
Table 62: Fan speed sensor threshold list	65
Table 63: Fan speed sensor threshold list (20 inches system)	65
Table 64: Supported RMCP+ Cipher Suites	68
Table 65: SNMP Community String Flag	70
Table 66: Default PEF Alert Policy Table	72

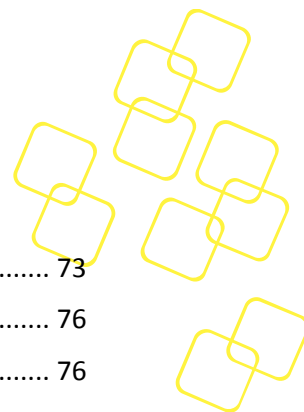
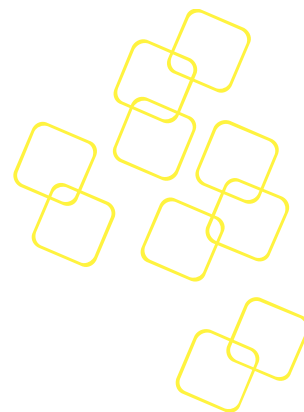
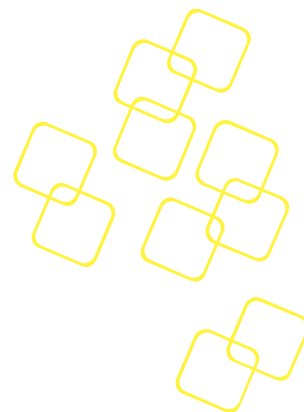


Table 67: Default PEF Event Filter Table.....	73
Table 68: User Account Default Setting	76
Table 69: Default PEF Alert Policy Table.....	76
Table 70: LAN Default Setting.....	78
Table 71: SOL Default Setting	78
Table 72: Set Power Restore Policy Command Actions	78
Table 73: Power Restore Policy	79
Table 74: HPM.1 Capability	81
Table 75: HPM.1 Component BMC Boot loader Property.....	81
Table 76: HPM.1 Component BMC application Property.....	82
Table 77: HPM.1 Component FPGA Property.....	82
Table 78: HPM.1 Component BIOS Property.....	83
Table 79: HPM.1 Component NVRAM Property	83
Table 80: System Identification	88
Table 81: Command Parameters of the 'Platform Event Message' Command.....	89
Table 82: Front LEDs Description.....	91
Table 83: Supported IPMI Commands.....	102
Table 84: The Examples of BMC FW Version.....	106

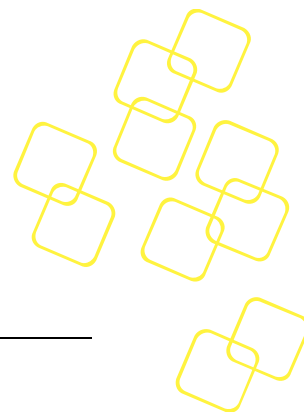


Glossary

ACPI	Advanced Configuration and Power Interface
BIOS	Basic Input Output System
BMC	Baseboard Management Controller
CGS	Carrier Grade Server
CPU	Central Processing Unit
DHCP	Dynamic Host Configuration Protocol
DIMM	Dual In-line Memory Module
DQA	Design Quality Assurance
DVT	Design Verification Test
ECC	Error Correction Check
EEPROM	Electrically Erasable Programmable Read-Only Memory
EVT	Engineering Verification Test
FPGA	Field Programmable Gate Array
FRB	Fault Resilient Booting
FRU	Field Replaceable Unit
FW	Firmware
GbE	Gigabit Ethernet
GUID	Globally Unique Identifier
HPM	Hardware Platform Management
IANA	Internal Assigned Numbers Authority
IP	Internet Protocol
IPMB	Intelligent Platform Management Bus
IPMC	Intelligent Platform Management Controller
IPMI	Intelligent Platform Management Interface
KCS	Keyboard Controller Style
KVM	Keyboard Video Mouse
LAN	Local Area Network
LED	Light Emitting Diode
LPC	Low Pin Count
MAC	Media Access Control
MIB	Module Information Block
MP	Mass Production



NC-SI	Network Controller Sideband Interface
NIC	Network Interface Controller
NVRAM	Non-Volatile Random Access Memory
OEM	Original Equipment Manufacturer
OOB	Out-Of-Band
OS	Operating System
PCH	Platform Controllers Hub
PCIe	PCI Express
PEF	Platform Event Filter
PET	Platform Event Trap
PICMG	PCI Industrial Computer Manufacturers Group
POST	Power-On Self-Test
PSU	Power Supply Unit
PVT	Pilot Run Test
RMCP	Remote Management Control Protocol
RMCP+	Advanced RMCP
RPM	Rotations per Minute
RTC	Real Time Clock
SEL	System Event Log
SDR	Sensor Data Record
SNMP	Simple Network Management Protocol
SPI	Serial Peripheral Interface Bus
SOL	Serial over LAN
SSH	Secure Shell
SW	Software
TCP	Transmission Control Protocol
UTC	Universal Time Coordinated



1. GETTING STARTED

1.1 Advanced Platform Management & BMC Introduction

Advanced Platform Management is supported via an integrated BMC running IPMI v2.0 compliant system management firmware. It provides system health monitoring to local or remote administrators, and allows them to recognize system degradation early to avoid system downtime or to shorten mean time to repair. In most cases, troubleshooting can be performed remotely alleviating physical access to the server.

1.1.1 Integrated Platform Management Interface

Platform management through the Intelligent Platform Management Interface (IPMI) is a standardized method for controlling and monitoring a device.

The IPMI specification defines a standardized interface for platform management including:

- Monitoring of system information and health, such as fans, temperatures, and power supplies
- Recovery capabilities, such as system resets and power on/off operations
- Logging capabilities, for abnormal events such as over temperature readings or fan failures
- Inventory capabilities, such as identifying failed hardware components

For additional information, see the IPMI specification.

1.1.2 Software Support

Advanced Platform Management is based on and compliant to IPMI v2.0. It is supported by most server operating systems (server OS) such as Windows Server, Linux and FreeBSD natively.

A standard software package which is most widely used for system platform management available on a number of operating systems is “ipmitool”. It is well integrated with most Linux distributions and has become an industry standard.

“ipmitool” will be used in this manual as a reference for interaction with the SKY-8201’s platform management. Other IPMI compliant software packages shall work in a similar way as well.

1.1.3 Advanced Features

Several enhancements have been made to the management code of standard white box server to enhance reliability and serviceability of the system including but not limited to:

- Improved thermal management to cover special scenarios as well as fan degradation/failure
- Chassis intrusion and FRU presence detection
- Redundant BMC and BIOS flashes for maximum reliability
- Fail safe BMC and BIOS upgrades using industry standard HPM.1 mechanisms and tools including automatic rollback on an upgrade failure
- Remote updates of firmware as long as primary power is connected to the unit
- BIOS Watchdog for reliable POST process and improved POST code sensor

- Time synchronization between the BMC and x86 host at startup for consistent event logs
- System FRU Information synchronization to the host via standard DMI tables
- Large system event log for efficient troubleshooting
- Capability to log system events from the x86 host
- Advanced Power Supply and FAN monitoring
- Intelligent and smooth FAN control supporting multiple cooling zones
- Full IPv6 support

1.1.4 The Advantech QuickStart Linux Image for SKY-8201

The operating system which runs on the SKY-8201 platform is referred to as the x86 host OS. In this document, we will take the Advantech QuickStart Linux Image (which is based on CentOS Linux distribution) as an example of the x86 host OS. Unless otherwise specified and throughout this document, command examples shown in the screenshots are executed in the Linux shell of the QuickStart Linux Image. It means that user log in to the QuickStart Linux Image via iKVM (keyboard/video/mouse) or a serial console (SOL or front console) or remote SSH connection and end up in the Linux shell to execute the/these command(s).

The easy way to access the QuickStart Linux image is through iKVM. In addition, the SKY-8201 provides a Display port and four USB ports (two at the rear of the unit and two at the front) for connection of a monitor, keyboard and mouse. See *Figure 1* and *Figure 2* for the location of these connectors.

Two additional methods are available for accessing the QuickStart Linux Image. The first one is the serial terminal connection. HW connection is made between the console port (see *Figure 1*) on the SKY-8201 and the console port on your user platform (Linux or Windows system). Putty or other serial terminal client applications can be used on the user platform to establish a connection.

SKY-8201S(20")



SKY-8201L1(27.5")



SKY-8201L2(27.5")



Figure 1: Front Serial Console Port and USB Ports

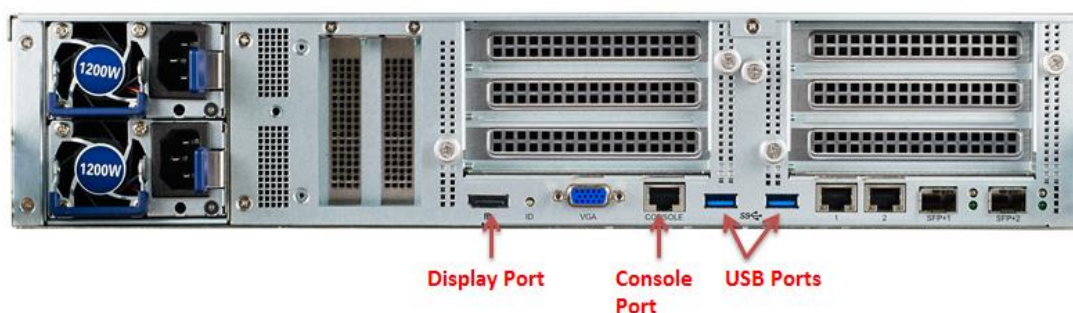


Figure 2: Rear Display, Console Port and USB Ports

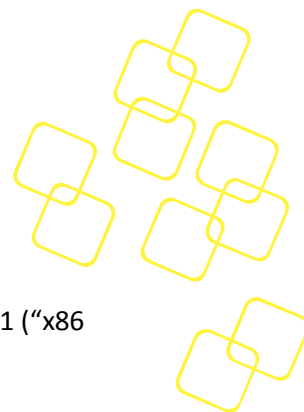
For more details regarding the HW cable connections and the SW configuration settings for the serial console connection between the SKY-8201 and the user platform, please refer to the *SKY-8201 User Manual* and *Getting Started Guide for Advantech SKY-8201 QuickStart Linux Image* documents.

The second method is a SSH connection over a LAN interface. The physical connection is made between one of the two Ethernet management ports (see MGMT1 and MGMT2 in *Figure 3*) on the SKY-8201 and the LAN port on user platform. Secure Shell (SSH) is a TCP/IP service that provides a secure mechanism for remotely logging in to the system either over local network or over Internet from another system. Putty or any other SSH client on the user platform can be used to establish a connection.



Figure 3: The Location of Management Ports at Rear Panel

For more details regarding the HW cable connections and the SW configuration settings for the SSH connection between the SKY-8201 and the user platform, please refer to *Getting Started Guide for Advantech SKY-8201 QuickStart Linux Image* documents.



1.2 Connecting to the BMC

The BMC supports an in-band (system) interface (KCS) to the Intel CPU of the SKY-8201 (“x86 host”) as well as an out-band Ethernet interface for external users.

1.2.1 Internal Connection via the System Interface

The x86 host can connect to the BMC through the KCS interface (see IPMI specification for details) using IO ports 0xCA2/0xCA3. Before connecting to the BMC, log in to the x86 host OS as described earlier.

Run `ipmitool` to access the BMC, e.g. retrieve device information:

```
# ipmitool mc info
```

```
Device ID           : 146
Device Revision     : 1
Firmware Revision   : 0.71
IPMI Version        : 2.0
Manufacturer ID     : 10297
Manufacturer Name    : Advantech
Product ID          : 33025 (0x8101)
Product Name        : Unknown (0x8101)
Device Available     : yes
Provides Device SDRs : yes
Additional Device Support :
    Sensor Device
    SEL Device
    FRU Inventory Device
    IPMB Event Generator
Aux Firmware Rev Info :
    0x00
    0x00
    0x00
    0x00
```

Figure 4: Use ‘ipmitool mc info’ Command to get BMC Info

In case of the `ipmitool` error message ‘*Could not open device at /dev/ipmi0 or /dev/ipmi/0 or /dev/ipmidev/0: No such file or directory*’ appears, please check whether the `ipmitool` drivers have been loaded well. The command for checking driver loading status and the steps of manually loading `ipmitool` drivers are provided in *Section A*.

1.2.2 External Connection via the LAN Interface

The SKY-8201 uses a shared NIC implementation to implement a management Ethernet port that allows connection to both the x86 host as well as the BMC. The required functionality to fork/aggregate traffic is built into the NIC chips used on the SKY-8201. The feature is transparent to the host OS and does not require any customized Ethernet drivers.



The two Ethernet management ports (see MGMT1 and MGMT2 in *Figure 3*) located at the rear panel are both available for providing LAN access to the BMC or the x86 host. Each port can be used in different networking domains for LAN access to BMC. See subsection below for more details.





For security reasons, LAN access to the BMC is disabled by default. This is accomplished by setting the BMC's IP address to 0.0.0.0

Some malware is probing for default IP addresses and user credentials used by many white box server OEMs. A BMC configured with some default LAN parameters that are not altered by the user presents a security risk. To avoid such security risk on Advantech hardware, Advantech disables the LAN port by default to make sure users do actively configure the LAN parameters and do not use default parameters/credentials.

1.2.2.1 BMC Dual NIC Support

On the SKY-8201 platform, dual NIC chips (two Intel i210 Ethernet controllers) are connected to the BMC via one NC-SI interface (see the figure below).

Every time the BMC starts up, it scans the LINK condition of both management ports (MGMT1 and MGMT2, see *Figure 3*).

If both ports have a link, users can reach BMC via IOL in different networking domain.

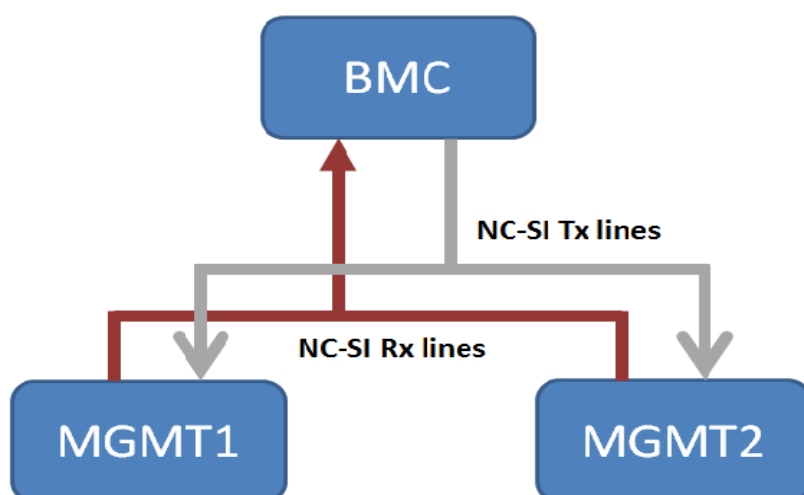


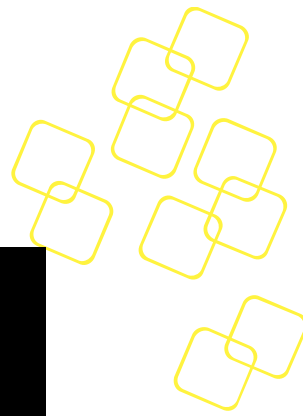
Figure 5: BMC Dual NIC

1.2.2.2 Configuring the BMC's LAN IP address through the KCS Interface

The '**Get/Set LAN Configuration Parameters**' IPMI commands can be used to configure the BMC's LAN IP address through the KCS interface. The default IP address source configured in the BMC is static IP assignment, while both static IP and DHCP IP assignments are supported. Some command examples to configure the IP address are provided here for reference, and please refer to related command usage to properly configure the BMC's IP.

For instance, the current BMC LAN configuration can be displayed through the following command:

```
#ipmitool lan print <channel number>
```

```
[root@CGS6010 ~]# ipmitool lan print 1
Set in Progress       : Set Complete
Auth Type Support     : NONE MD2 MD5 PASSWORD
Auth Type Enable     : Callback : MD2 MD5 PASSWORD
                     : User      : MD2 MD5 PASSWORD
                     : Operator  : MD2 MD5 PASSWORD
                     : Admin    : MD2 MD5 PASSWORD
                     : OEM       :
IP Address Source     : Static Address
IP Address            : 0.0.0.0
Subnet Mask           : 0.0.0.0
MAC Address           : 74:fe:48:2a:9f:c7
SNMP Community String : public
IP Header             : TTL=0x40 Flags=0x40 Precedence=0x00 TOS=0x10
BMC ARP Control       : ARP Responses Enabled, Gratuitous ARP Disabled
Gratuitous ARP Intrvl : 0.0 seconds
Default Gateway IP    : 0.0.0.0
Default Gateway MAC   : 14:dd:a9:4d:e3:40
Backup Gateway IP     : 0.0.0.0
Backup Gateway MAC    : 00:00:00:00:00:00
802.1q VLAN ID        : Disabled
802.1q VLAN Priority  : 0
RMCP+ Cipher Suites   : 0,1,2,3,6,7,8,11,12
Cipher Suite Priv Max : caaaXXaaaXXaaXX
                     : X=Cipher Suite Unused
                     : c=CALLBACK
                     : u=USER
                     : o=OPERATOR
                     : a=ADMIN
                     : O=OEM
```

Figure 6: Print LAN Channel Configuration

```
#ipmitool lan set <channel> <command> [option]
```

This command can be used to change several IPMC LAN parameters (e.g. IP address, netmask, gateway IP address,...). Below example demonstrates how to set up the static IP address for the BMC:

```
[root@CGS6010 ~]# ipmitool lan set 1 ipaddr 192.168.1.1
Setting LAN IP Address to 192.168.1.1
```

Figure 7: Set up a Static IP for BMC

Configure the BMC to get a dynamic IP address from an external DHCP server:

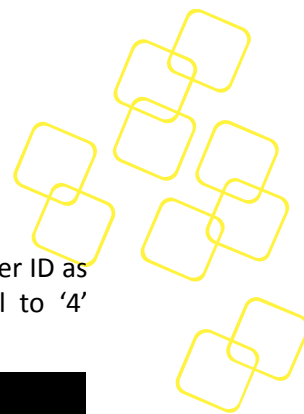
```
[root@CGS6010 ~]# ipmitool lan set 1 ipsrc dhcp
```

Figure 8: Configure Dynamic IP Assignment for the BMC

1.2.2.3 Enable LAN access to the BMC

To enable LAN access to the BMC, follow the steps below:

- Connect to the BMC via the KCS interface
- Set User ID/Password
- Set LAN parameters, especially with valid IP address (see *Section 1.2.2.2*)



Add a new BMC user (e.g. the third user, see *Section 2.15.1 User Account*). Set the user ID as '3', user name as 'SKY-8201_admin', password as 'advantech' and privilege level to '4' (Administrator), as in the example below:

```
[root@CGS6010 ~]# ipmitool user set name 3 CGS6010_admin
[root@CGS6010 ~]# ipmitool user set password 3 %cgs6010_admin%
[root@CGS6010 ~]# ipmitool channel setaccess 1 3 callin=off ipmi=on link=off privilege=4
[root@CGS6010 ~]# ipmitool user enable 3
```

Figure 9: Set User ID, User Name, Password and Privilege Level

On the user platform, issue the following command to connect to the BMC over LAN and get BMC device information:

```
#ipmitool -I lanplus <BMC IP> -U <User ID> -P <Password> mc info
```

Command Line Syntax:

- I lanplus Specifies Ethernet interface using RMCP+
- H <IP-Address> IP address assigned to the IPMC
- U<User> User account, default "administrator"
- P <Password> Password used with specified user account, default "advantech"

Default credentials used on all IPMI LAN channels:

Username: "administrator"

Password: "advantech"

1.3 Connecting to the x86 Host over SOL

Advanced Platform Management allows you to access the system's console while the host OS is absent, i.e. during BIOS execution and booting. In addition, it provides access to the OS console when no OS Ethernet interfaces are up or the OS does not provide remote login.

The underlying technology is referred to as Serial over LAN (SOL) which basically means that the BMC encapsulates host console data into an Ethernet protocol called RMCP+ and transmits this data through the BMC's IPMI-over-LAN channel.

Note that the serial console redirection is enabled by default in the SKY-8201 system BIOS.

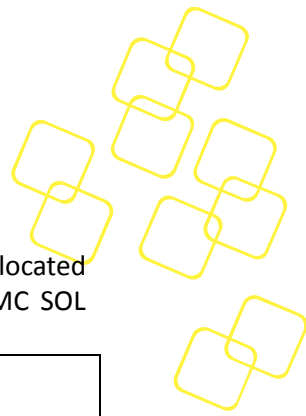
To establish a SOL session with the x86 host, do the following steps:

1. The default SOL baud rate setting on the BMC side is 115200, the same as serial console speed in BIOS (host side) is 115.2 kbps (see *Section 2.15.4 SOL*). If it has been altered, use the following command to reset the value to 115200:

```
#ipmitool sol set non-volatile-bit-rate 115.2 1
#ipmitool sol set volatile-bit-rate 115.2 1
```

2. Enable SOL for a specified user (e.g. User ID '3' we previously added in *Section 1.2.2.3*) in the BMC:

```
#ipmitool sol payload enable <channel> <User ID>
```



3. Make sure the BMC's IP address and the IP address of the user platform are located in the same subnet. On the user platform, you can display the current BMC SOL settings by using:

```
#ipmitool -I lanplus -H <BMC IP> -U <User ID> -P <Password> sol info
```

```
[root@svnsrver ~]# ipmitool -I lanplus -H 172.17.10.183 -U CGS6010_admin -P %cgs6010_admin% sol info
Set in progress      : set-complete
Enabled              : true
Force Encryption     : true
Force Authentication : false
Privilege Level      : USER
Character Accumulate Level (ms) : 150
Character Send Threshold : 220
Retry Count          : 7
Retry Interval (ms)  : 480
Volatile Bit Rate (kbps) : 115.2
Non-Volatile Bit Rate (kbps) : 115.2
Payload Channel      : 1 (0x01)
Payload Port         : 623
```

Figure 10: Display Current BMC SOL Setting

4. Use the following command to activate the SOL session and access the system console of x86 host. The examples shown here are the console outputs of the Advantech QuickStart Linux Image and BIOS, which are redirected to the activated SOL session:

```
#ipmitool -I lanplus -H <BMC IP> -U <User ID> -P <Password> sol activate
```

```
[root@svnsrvr ~]# ipmitool -I lanplus -H 172.17.10.183 -U CGS6010_admin -P %cgs6010_admin% sol activate
[SOL Session operational. Use ~? for help]

Broadcast metype=1400 audit(1441712730.805:43): avc: denied { sys_resource } for pid=5208 comm="shutdown" capa
bility=24 scontext=unconfined_u:unconfined_r:shutdown_t:s0-s0:c0.c1023 tcontext=unconfined_u:unconfined_r:shutdo
wn_t:s0-s0:c0.c1023 tclass=capability
ssage from root@CGS6010.localdomain
      (/dev/pts/0) at 19:45 ...

The system is going down Running guests on default URI: no running guests.
Stopping libvirtd daemon: [ OK ]
Stopping atd: [ OK ]
Stopping cups: [ OK ]
Stopping ksm: [ OK ]
Stopping abrt daemon: [ OK ]
Stopping sshd: [ OK ]
```

Figure 11: Activate SOL Session to Access the Advantech Standard Linux Image

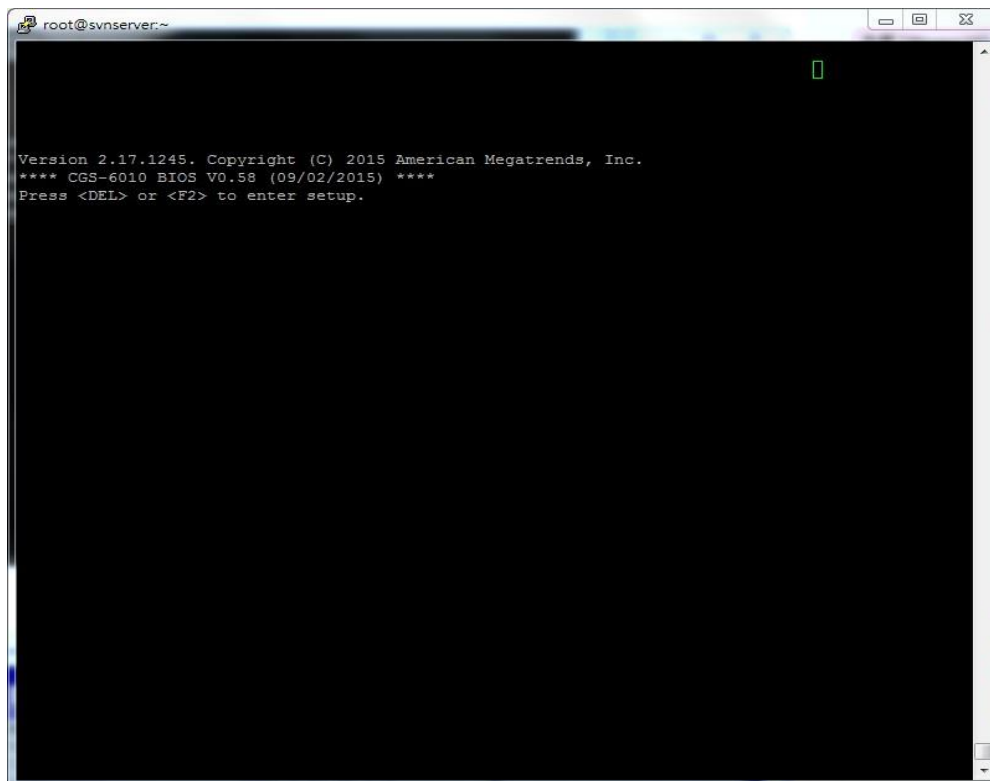


Figure 12: Access SKY-8201 BIOS in SOL Session

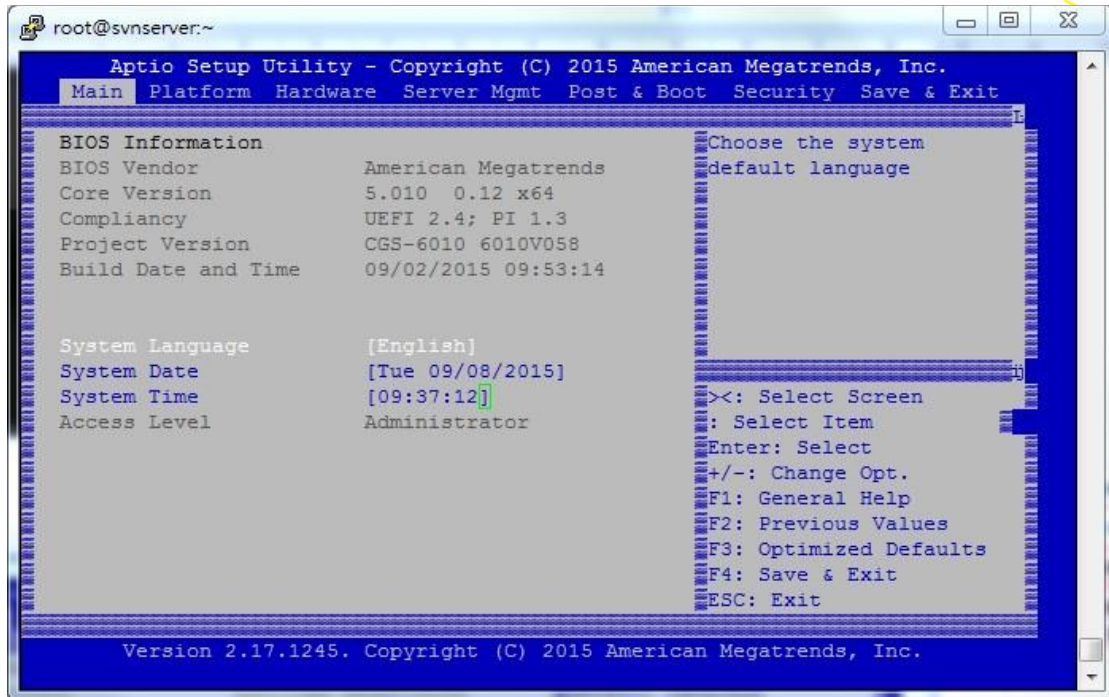


Figure 13: Enter BIOS Setup Menu in SOL Session

5. In ipmitool, '~.' is used to terminate the SOL connection. '~?' is to get other available commands during SOL connection.

```
# ipmitool -I lanplus <IP-Address> -U <User> -P <Password> sol activate
```

```
[SOL Session operational. Use ~? for help]
```

```
...
```

Support escape sequences :

```
~.    [terminate connection]
```

```
~^Z   [suspend ipmitool]
```

```
~^X   [suspend ipmitool], but don't restore tty on restart
```

```
~B    Send break
```

```
~?    This message
```

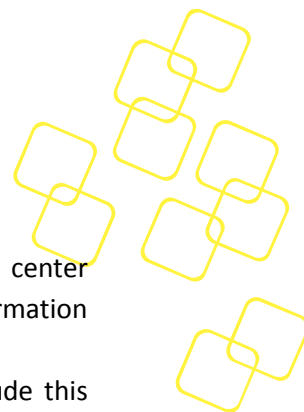
```
~~    Send the escape character by typing it twice
```

(Note that escapes are only recognized immediately after newline)

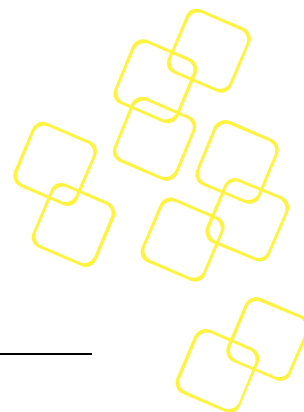
1.4 Getting Help: Technical Support and Assistance

In case the unit you received is a sample for evaluation, please contact your Advantech representative. For production units, please follow the process below:

1. Visit the Advantech web site at www.advantech.com/support to find the latest information about SKY-8201 and related products.



2. Contact your distributor, sales representative, or Advantech's customer service center for technical support if you need additional assistance. Worldwide contact information can be found on www.advantech.com.
3. Please have the following information ready before you call / be sure to include this information in your email:
 - Product name and serial number
 - Description of your peripheral attachments
 - Description of firmware and software versions installed on the product
 - A complete description of the problem
 - The exact wording of any error messages
4. In case the unit needs to be sent back for repair, please refer to the SKY-8201 User's Manual for instructions.



2. BMC FUNCTIONALITIES

2.1 Overview

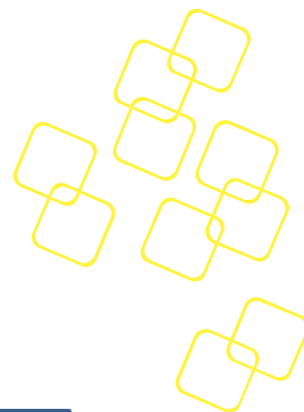
The SKY-8201 is a highly configurable carrier-grade server designed to balance the best in x86 server-class processing performance with maximum I/O and offload density in a 20" deep chassis. The system is a cost effective, highly available platform optimized to meet next-generation networking equipment needs.

With an on-board BMC and various sensors, Advanced Platform Management provides users with robust, flexible and IPMI v2.0 compliant system monitoring functionality. This section will explain more details about the basic and advanced features of the BMC. For a complete list of supported IPMI commands, refer to *Appendix A*.



The BMC provides the following functions on the SKY-8201:

- **All mandatory IPMI BMC functions** from *Table 3-1 'Required BMC Functions', Intelligent Platform Management Interface Specification, Version 2.0 (IPMIv2.0 specification)*.
- **Optional BMC functions** as follows (also refer to *Table 3-1 of IPMIv2.0 specification*):
 - Sensors
 - LAN messaging and alerting
 - Platform event filtering (PEF) and alert policies
 - External event generation
- **Firmware upgrade**
 - From host:
The BMC firmware and BIOS can be upgraded from the host system over KCS interface or OS2BMC by using the HPM.1 mechanism.
 - Over LAN:
The BMC firmware and BIOS can be upgraded over LAN management interfaces using the HPM.1 mechanism, either from an external host or product itself (OS2BMC).
 - Both BMC and BIOS upgrade use redundant flash chips for failsafe upgrades with automatic rollback
- **FRU data access** through IPMI '**FRU Read/Write**' commands
- **Environmental and health monitoring** reported through IPMI sensors, alerts, and logging
 - Temperatures
 - Voltages
 - Fan speed
 - Power supply monitoring (through PMBus)
 - Chassis intrusion
- **IPMI and OEM defined discrete sensors** enhance system health monitoring and event logs tracking
- **Chassis Power/Reset control** via IPMI and DCMI chassis commands (*IPMIv2.0 specification, Section 28 'Chassis Commands'*)



The BMC block diagram is shown below.

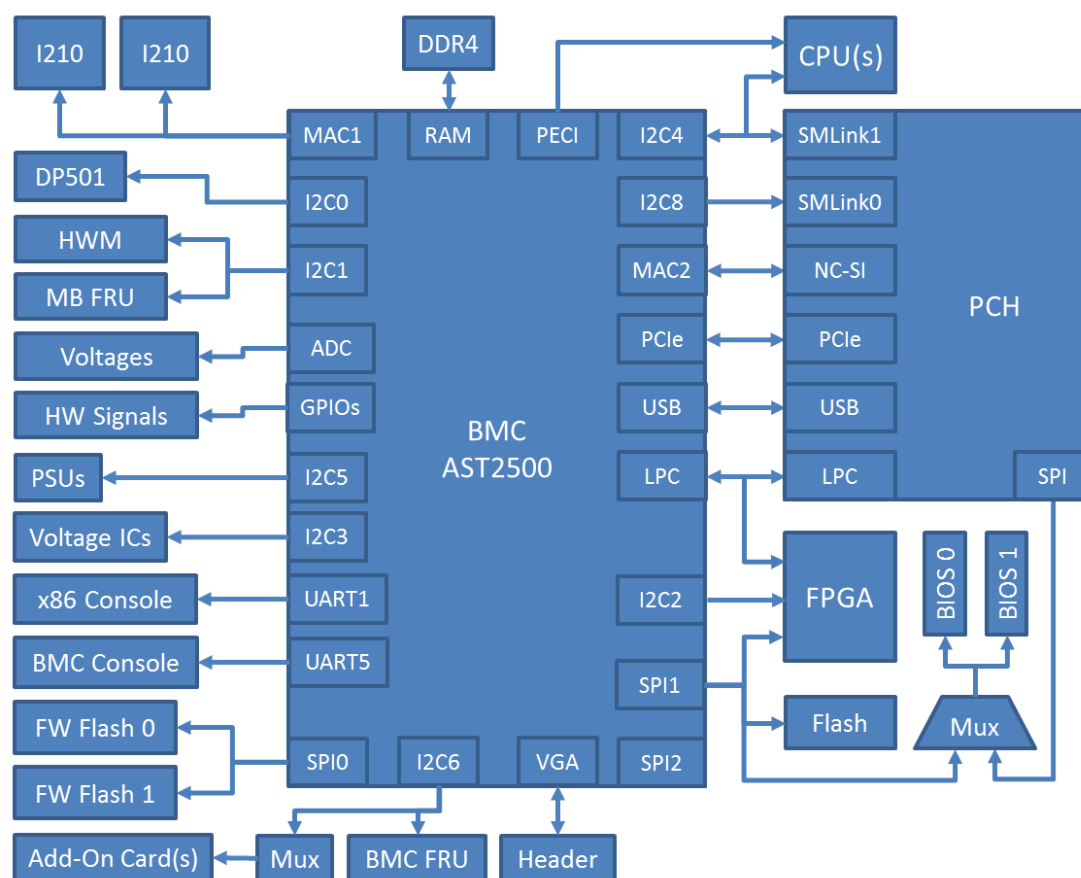


Figure 14: BMC Block Diagram

The BMC messaging interfaces comply with the *IPMIv2.0 specification*.

As system interface, a KCS interface (*IPMIv2.0 specification, Section 9 'Keyboard Controller Style (KCS) Interface'*) is supported.

In addition, direct LAN based access from x86 host to the BMC (OS2BMC) is supported as host interface.

LAN connections on MGMT1 and MGMT2 ports are supported over the RMCP/RMCP+ protocol (*IPMIv2.0 specification, Section 13 'IPMI LAN Interface'*).

In summary, there are two communication channels on the BMC:

- **KCS Channel (channel number 0Fh)**, for communication with the x86 host
- **LAN Channel (channel number 01h)**, using the MGMT1 and MGMT2 LAN ports

2.1.1 System Interface

The System Interface provides a communication path between the local x86 processor and the BMC. The physical interface is based on the LPC bus and is session-less.

The System Interface supports the IPMI defined KCS (Keyboard Controller Style) interface with the default IO addresses 0xCA2/0xCA3.



This allows IPMI drivers to auto probe for the interface and eliminates the need to specify additional parameters when loading the IPMI drivers.

- **Channel 0Fh System Interface (SMS)**
 - Access Mode -> Always available
 - Protocol -> KCS
 - Medium -> System Interface
 - Channel Privilege -> Administrator
 - User & Password Support -> N/A
 - Session Type -> Session-less, so no user and authentication support
 - Session Quantity -> 0

2.1.2 LAN Interface

Out-of-band (OOB) management over LAN (IPMI over LAN) is implemented via a shared NIC, which allows the BMC to be accessed through the system's LAN management ports MGMT1 and MGMT2.

The IP settings and other privileges of this channel can be configured through '**Get/Set LAN Configuration Parameters**' commands (*Section 23 'IPMI LAN Commands' in the IPMIv2.0 specification*).

- **Channel 01h for LAN Interface**
 - Access Mode -> Always available
 - Protocol -> IPMB
 - Medium -> LAN 802.3
 - Channel Privilege -> As set by '**Set Channel Access**' command
 - User & Password Support -> Total 7 users are supported
 - Session Type -> Multi-session with authentication
 - Session Quantity -> Total 4 simultaneous sessions are supported

2.2 OEM IPMI Commands

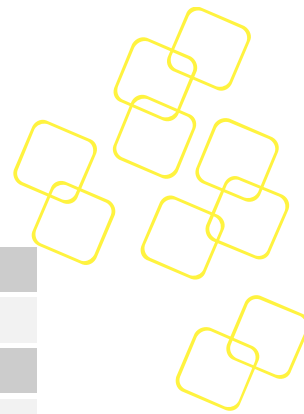
In addition on the standard IPMI commands listed in *Appendix A*, the SKY-8201 supports a number of OEM specific IPMI Commands.

2.2.1 Advantech OEM IPMI Commands

2.2.1.1 Command code List

The BMC supported Advantech IPMI OEM commands are listed in below table.

Command Name	NetFn Code	Command Code
Get HW Revision	2Eh	05h
Get Payload CPU ID	2Eh	06h
Write I2C Device	2Eh	20h
Read I2C Device	2Eh	21h
Store Configuration Settings	2Eh	40h
Read Configuration Settings	2Eh	41h
SEL Mode Configuration	2Eh	62h



Set SW Bank Selection	2Eh	70h
Get SW Bank Selection	2Eh	71h
Set Community String Flag	2Eh	74h
Get Community String Flag	2Eh	75h
Read Port 80 (BIOS POST Code)	2Eh	80h
Reload NVRAM Defaults	2Eh	81h
BIOS Data Exchange	2Eh	82h
BIOS Rollback Options	2Eh	85h
Trigger Payload OS Interrupt	2Eh	90h
Get BIOS Boot Bank ID	2Eh	93h
Set BIOS Boot Bank ID	2Eh	94h
Set Factory Mode	2Eh	E0h
Write EEPROM Test Byte	2Eh	E6h
Read EEPROM Test Byte	2Eh	E7h
Reload BMC Default Configuration	2Eh	F2h
Get SW Component Information	2Eh	F3h
Reload Factory Defaults	2Eh	F4h

Table 1: OEM Command List

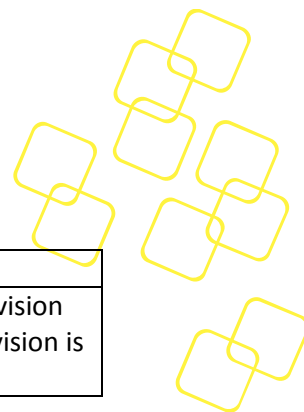
2.2.1.2 Advantech OEM Command ‘Get HW Revision’

This command is intended to distinguish about the boards HW revision.

Request	Byte	Data Field
	1	NetFn: 0x2e
	2	Cmd: 0x05
	3	IANA Byte 1: 0x39
	4	IANA Byte 2: 0x28
	5	IANA Byte 3: 0x00

Response	Byte	Data Field
	1	Completion Code 00h: command completed normally
	2	IANA Byte 1: 0x39
	3	IANA Byte 2: 0x28
	4	IANA Byte 3: 0x00
	5	Major HW revision (e.g. “A1” or “B1”)
	6	Minor HW revision (e.g. 01h, 02h, ...)

Command example:



	Command Sequence	Description
ipmitool raw	>ipmitool raw 0x2e 0x05 0x39 0x28 0x00 39 28 00 a1 02	Get HW Revision The HW revision is 0xa1 0x02

Table 2: 'Get HW Revision' Command

2.2.1.3 Advantech OEM Command 'SEL Mode Configuration'

This command is used to change System Event Log (SEL) behaviour when the SEL runs full. Two options are available: stop logging when the SEL is full, or roll over (overwrite oldest event in case of new event). This rule is applied to subsequent entries being written to the SEL.

Request	Byte	Data Field
	1	NetFn: 0x2e
	2	Cmd: 0x62
	3	IANA Byte 1: 0x39
	4	IANA Byte 2: 0x28
	5	IANA Byte 3: 0x00
	6	SEL Mode [7:0]: 00h: Stop on full. BMC will not log any new event to SEL. 01h: Wrap around when full. BMC will overwrite the oldest event with a new incoming event.

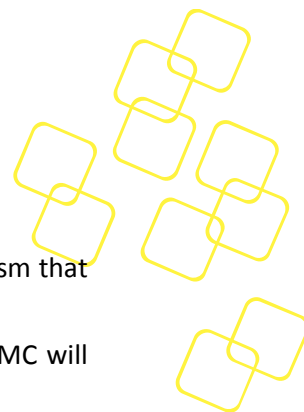
Response	Byte	Data Field
	1	Completion Code 00h: command completed normally
	2	IANA Byte 1: 0x39
	3	IANA Byte 2: 0x28
	4	IANA Byte 3: 0x00
	5	SEL Mode [7:0]: 00h: stop on full 01h: wrap around when full

Command example:

	Command Sequence	Description
ipmitool raw	>ipmitool raw 0x2e 0x62 0x39 0x28 0x00 0x01 39 28 00	Set SEL full action as "Stop on full"

Table 3: 'SEL Mode Configuration' Command

Please note that the setting of the SEL full action (**Do Nothing** or **Erase Immediately** options) in the BIOS setup menu (under the submenu: **Platform** -> **Platform**



Management -> System Event Log -> When SEL is Full) is a parallel mechanism that only applies to BIOS writing events to the SEL.

When the BIOS attempts to write a new event log but the SEL is full, the BMC will behave like this based on the SEL full settings of BMC and BIOS:

BMC Setting	BIOS Setting	Result
Stop on full	Do Nothing	Will not log new event to SEL
Stop on full	Erase Immediately	Erase all contents of the SEL then add new event to SEL
Wrap around when full	Do Nothing	Overwrite the oldest event on SEL
Wrap around when full	Erase Immediately	Erase all contents of the SEL then add new event to SEL

Table 4: The BIOS Attempts to Add a New SEL Entry

2.2.1.4 Advantech OEM Command 'Read Port 80 (BIOS POST Code)'

This command is used to retrieve the latest BIOS POST Code.

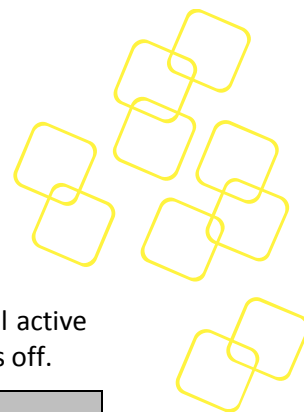
Request	Byte	Data Field
	1	NetFn: 0x2e
	2	Cmd: 0x80
	3	IANA Byte 1: 0x39
	4	IANA Byte 2: 0x28
	5	IANA Byte 3: 0x00

Response	Byte	Data Field
	1	Completion Code 00h: command completed normally
	2	IANA Byte 1: 0x39
	3	IANA Byte 2: 0x28
	4	IANA Byte 3: 0x00
	5	latest BIOS POST Code

Command example:

	Command Sequence	Description
ipmitool raw	>ipmitool raw 0x2e 0x80 0x39 0x28 0x00 39 28 00 b2	Read Port 80 (BIOS POST Code) The latest POST Code is 0xb2

Table 5: 'Read Port 80(BIOS POST Code)' Command



2.2.1.5 Advantech OEM Command 'Reload NVRAM Defaults'

This command is used to reload the UEFI BIOS NVRAM defaults of the actual active BIOS at the next reboot. The command is only allowed when payload power is off.

Request	Byte	Data Field
	1	NetFn: 0x2e
	2	Cmd: 0x81
	3	IANA Byte 1: 0x39
	4	IANA Byte 2: 0x28
	5	IANA Byte 3: 0x00

Response	Byte	Data Field
	1	Completion Code D5h = not supported in present state
	2	IANA Byte 1: 0x39
	3	IANA Byte 2: 0x28
	4	IANA Byte 3: 0x00

Command example:

	Command Sequence	Description
ipmitool raw	>ipmitool raw 0x2e 0x81 0x39 0x28 0x00 39 28 00	Reload NVRAM Default Command

Table 6: 'Reload NVRAM Defaults' Command

2.2.1.6 Advantech OEM Command 'Reload BMC Default Configuration'

This command reloads BMC related, product specific settings.

Request	Byte	Data Field
	1	NetFn: 0x2e
	2	Cmd: 0xF2
	3	IANA Byte 1: 0x39
	4	IANA Byte 2: 0x28
	5	IANA Byte 3: 0x00

Response	Byte	Data Field
	1	Completion Code
	2	IANA Byte 1: 0x39
	3	IANA Byte 2: 0x28
	4	IANA Byte 3: 0x00

Command example:

	Command Sequence	Description
ipmitool raw	>ipmitool raw 0x2e 0xF2 0x39 0x28 0x00 39 28 00	Reload BMC default configuration

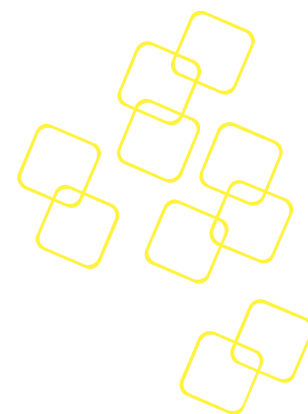


Table 7: ‘Reload BMC Default Configuration’ Command

2.3 BMC Watchdog

The IPMI specification includes support for an IPMI BMC watchdog. This watchdog can be configured by BIOS, OS, external management software or the user. The watchdog owner and action can be configured by the IPMI **‘Set Watchdog Timer’** command.

This IPMI watchdog is also used for fail safe BIOS execution, especially after a BIOS firmware update on the SKY-8201.

The BIOS POST watchdog is started on platform reset, watchdog owner is set as “BIOS FRB2” and action is set as “Hard Reset”.

When an IPMI watchdog timeout occurs, the BMC logs the current timestamp, action and watchdog owner to the System Event Log (SEL) so this information can be used for debugging purposes.

More sophisticated operation modes such as pre-timeout interrupts are also supported. Standard utilities such as ipmitool can be used for configuring and strobing this watchdog. Refer to the *IPMIv2.0 specification, Section 27 ‘BMC Watchdog Timer Commands’* for more details about the BMC watchdog features and operation.

2.4 FRU Information

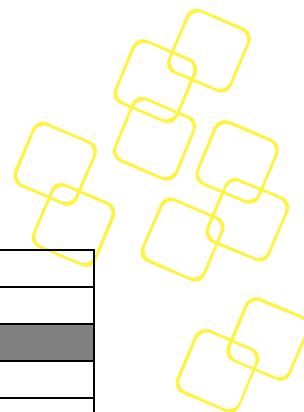
The IPMI specification includes support for storing and accessing multiple sets of non-volatile Field Replaceable Unit (FRU) information for different modules in the system. The FRU data includes information such as product name, HW version, serial number, part number, modules, and asset tag.

The system FRU information is stored in an EEPROM on the mainboard (FRU Device 0).

FRU information is accessed using IPMI commands (*IPMIv2.0 specification, Section 34 ‘FRU Inventory Device Commands’*).

BMC FRU Information (Device ID: 0)

Common Header
Version = 1
Chassis Info
Version = 1
Type = Rack Mount Chassis
Part Number = SKY-8201
Serial Number = TPE0000000 (programmed during manufacturing)
Board Info
Version = 1
Language Code = en
Manufacturing Date/Time = WW MM DD TIME YYYY (programmed during manufacturing)
Manufacturer = Advantech
Product Name = GSMB-8201MB



Serial Number = AA00000000 (programmed during manufacturing)
Part Number = 00000000A0A (programmed during manufacturing)
Product Info
Version = 1
Language Code = en
Manufacturer = Advantech
Product Name = SKY-8201
Part/Model Number = 0000000000A (programmed during manufacturing)
Product Version = A1-02 (programmed during manufacturing)
Serial Number = AAA0000000 (programmed during manufacturing)

Table 8: System FRU Information

You can use the 'ipmitool fru' command to retrieve system FRU information:

```
#ipmitool fru
```

```
FRU Device Description : Builtin FRU Device (ID 0)
Chassis Type          : Rack Mount Chassis
Chassis Part Number   : SKY-8101
Chassis Serial        : AKA1234567
Board Mfg Date        : Mon Jan  1 08:00:00 1996
Board Mfg             : Advantech
Board Product         : GSMB-8101MB
Board Serial          : AKA1234567
Board Part Number     : SKY-8101
Product Manufacturer  : Advantech
Product Name          : SKY-8101
Product Part Number   : SKY-8101
Product Version       : A1 02
Product Serial        : AKA1234567
```

Figure 15: Retrieve System FRU Information

2.5 System Event Log

The System event log (SEL) is stored in a flash memory device. Note that each SEL entry is 16 bytes in length. The SEL consists of 64 kB of memory, resulting in approximately 4095 entries.

OEM SEL events can be used to log events from the host OS and/or application software into the SEL. This is supported through the IPMI commands '**Platform Event Message**' and '**Add SEL Entry**'.



When the SEL is full, the BMC will stop logging by default. However, the OEM command '**SEL Mode Configuration**' can be used to change the behaviour on SEL full condition. See *Section 2.2.1.3 Advantech OEM Command 'SEL Mode Configuration'* for more details.



2.6 Sensors

The BMC monitors system health and represents related data through sensors. In case any sensor thresholds are exceeded, the BMC will send a sensor event to the default event receiver using IPMI messages, or to a higher level monitoring entity via a SNMP trap.

The important voltages and temperatures are connected to the BMC management system in different ways (see hardware chapters for more details).

Moreover, the BMC also registers several logical, discrete sensors (e.g. IPMI BMC Watchdog sensor, IPMI Version change sensor, etc).

All sensors are defined according to the IPMI specification and as such available through standard IPMI sensor commands.

2.6.1 Sensor Data Records & Handling

The BMC uses Sensor Data Records (SDRs) to describe the sensors and their capabilities, e.g. the sensor type, how to interpret the readings, the supported events and the sensor thresholds.

The set of sensor data records is stored in a device sensor data repository. Caution needs to be taken when using the term SDR, as it is commonly used for sensor data records as well as sensor data repositories.

For a fixed hardware configuration, the set of sensors as such is typically static, so the sensor data records are typically a fixed part of the BMC firmware image. While there are IPMI commands to manipulate sensor data records at runtime, this might be a security concern.

The BMC on the platform supports volatile manipulation of sensor data parameters for testing/debugging purposes but the default values will be reloaded from the FW image at each BMC reset.

The device sensor data repository is implemented in a semi-static way, which means that static sensor data records for the main FRU are stored in BMC firmware image, but additional sensors of optional FRUs are dynamically added to either non-volatile storage or in RAM.

2.6.2 Sensor Types

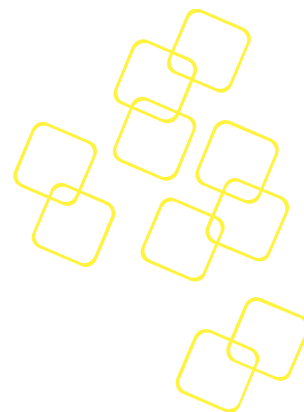
This section summarizes the IPMI Sensor Type, Event/Reading Type and Entity ID codes used and listed in following subsections.

The board overall sensor list includes below IPMI Event/Reading Type codes:

Event/Reading Type	Description
01h	Threshold-based
03h	Generic 'digital' discrete (state asserted / state deasserted)
70h	OEM discrete

Table 9: BMC used event and reading type codes

Below table gives an overview about the used IPMI Sensor Type codes:

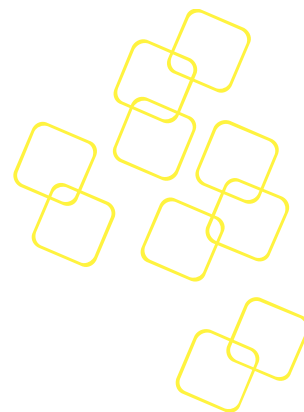


Sensor Type Code	Event Description
01h	IPMI Temperature
02h	IPMI Voltage
03h	IPMI Current
04h	IPMI Fan
05h	IPMI Physical Security (Chassis Intrusion)
07h	IPMI Processor
08h	IPMI Power Supply
09h	IPMI Power Unit
0Bh	Other Units-based Sensor
0Dh	Drive Slot (Bay)
0Fh	IPMI System Firmware Progress
12h	IPMI System Event
1Dh	IPMI System Boot / Restart Initiated
22h	IPMI System ACPI Power State
23h	IPMI Watchdog 2
25h	IPMI Entity Presence
28h	IPMI Management Subsystem Health
2Bh	IPMI Version Change
C0h	OEM sensor

Table 10: BMC used sensor type codes

The last table shows the used IPMI Entity ID codes for this product:

Entity ID Code	Entity
03h	IPMI Processor
04h	IPMI Disk or disk bay
06h	IPMI System management module
07h	IPMI System board
0Ah	IPMI Power supply
0Bh	IPMI Add-in Card
0Fh	IPMI Drive backplane
11h	IPMI Other system board (part of board set)



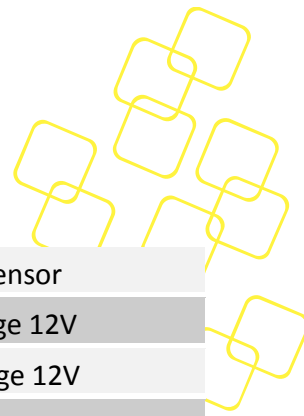
13h	IPMI Power unit / power domain
14h	IPMI Power module
15h	IPMI Power management / distribution
17h	IPMI System chassis
1Ah	IPMI Disk Drive Bay
1Dh	IPMI Fan / cooling device
1Eh	IPMI Cooling unit / domain
20h	IPMI Memory device
31h	IPMI PCI Express Bus
37h	IPMI Air Inlet
A0h	PICMG Front Board
C0h	PICMG RTM
C1h	PICMG AMC
F0h	PICMG ShMC

Table 11: BMC used Entity IDs

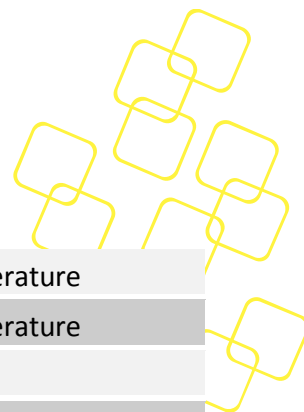
2.6.3 Sensor List

The following table is the list of sensors provided by the BMC:

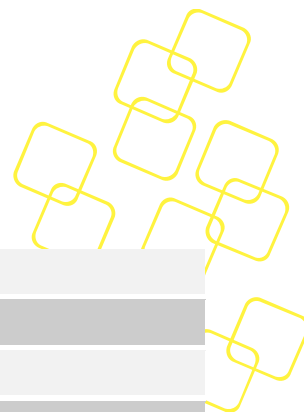
No	Sensor ID	Sensor Type	Event Reading Type	Entity ID	Description
0	SKY-8201	-	-		IPMI FRU Device Locator
1	BMC_HEALTH	28h	6Fh	07h	IPMI Management Subsystem Health
2	VERSION_CHANGE	2Bh	6Fh	07h	IPMI Version Change sensor
3	BMC_WATCHDOG	23h	6Fh	07h	IPMI BMC Watchdog sensor
4	ACPI_STATE	22h	6Fh	06h	IPMI System ACPI Power State sensor
5	PROC_STATE	07h	6Fh	03h	IPMI Processor sensor
6	SYSTEM_RESET	1Dh	6Fh	07h	Payload system reset indication
7	FW_PROGRESS	0Fh	6Fh	07h	IPMI System FW Progress sensor
8	CASE_INTRUSION	05h	6Fh	17h	IPMI Physical Security sensor
9	PCIE_CR_INT	13h	6Fh	31h	IPMI Critical Interrupt sensor
10	INTEGRITY	C0h	6Fh	06h	Advantech Integrity OEM sensor
11	PSU1	08h	6Fh	0Ah	IPMI Power Supply sensor



12	PSU2	08h	6Fh	0Ah	IPMI Power Supply sensor
13	P12V-VOL	02h	01h	14h	Payload Power voltage 12V
14	P12V_STBY-VOL	02h	01h	14h	Standby Power voltage 12V
15	P5V-VOL	02h	01h	14h	Payload Power voltage 5V
16	P5V_AUX-VOL	02h	01h	14h	Standby Power voltage 5V
17	P3V3-VOL	02h	01h	14h	Payload Power voltage 3.3V
18	P3V3_AUX-VOL	02h	01h	14h	Standby Power voltage 3.3V
19	BAT_3_0-VOL	02h	01h	14h	Battery Backup voltage 3.0V
20	PVPP_ABC-VOL	02h	01h	14h	DDR Power voltage 2.6V
21	PVPP_DEF-VOL	02h	01h	14h	DDR Power voltage 2.6V
22	PVPP_GHJ-VOL	02h	01h	14h	DDR Power voltage 2.6V
23	PVPP_KLM-VOL	02h	01h	14h	DDR Power voltage 2.6V
24	PVCCIN_CPU0-VOL	02h	01h	14h	CPU0 Power voltage 1.8V
25	PVCCIN_CPU1-VOL	02h	01h	14h	CPU1 Power voltage 1.8V
26	P1V8_PCH_AUX-VOL	02h	01h	14h	PCH Power voltage 1.8V
27	PVDDQ_ABC-VOL	02h	01h	14h	DDR Power voltage 1.2V
28	PVDDQ_DEF-VOL	02h	01h	14h	DDR Power voltage 1.2V
29	PVDDQ_GHJ-VOL	02h	01h	14h	DDR Power voltage 1.2V
30	PVDDQ_KLM-VOL	02h	01h	14h	DDR Power voltage 1.2V
31	P1V2_DP501-VOL	02h	01h	14h	DP501 Power voltage 1.2V
32	P1V05_PCH-VOL	02h	01h	14h	PCH Power voltage 1.05V
33	PVCCIO_CPU0-VOL	02h	01h	14h	CPU0 Power voltage 1V
34	PVCCIO_CPU1-VOL	02h	01h	14h	CPU1 Power voltage 1V
35	PVNN_PCH_AUX-VOL	02h	01h	14h	PCH Power voltage 0.9V
36	PVCCSA_CPU0-VOL	02h	01h	14h	CPU0 Power voltage 0.85V
37	PVCCSA_CPU1-VOL	02h	01h	14h	CPU1 Power voltage 0.85V
38	PVTT_ABC-VOL	02h	01h	14h	DDR Power voltage 0.6V
39	PVTT_DEF-VOL	02h	01h	14h	DDR Power voltage 0.6V
40	PVTT_GHJ-VOL	02h	01h	14h	DDR Power voltage 0.6V
41	PVTT_KLM-VOL	02h	01h	14h	DDR Power voltage 0.6V
42	INLET-TMP	01h	01h	17h	Inlet Temperature
43	OUTLET-TMP	01h	01h	17h	Outlet Temperature



44	CPU0-TMP	01h	01h	03h	CPU0 Internal Temperature
45	CPU1-TMP	01h	01h	03h	CPU1 Internal Temperature
46	HWM-TMP	01h	01h	07h	
47	PCH-TMP	01h	01h	07h	PCH Internal Temperature
48	CPU0_DIMM_A1-TMP	01h	01h	20h	DIMM Module A1 Temperature
49	CPU0_DIMM_A2-TMP	01h	01h	20h	DIMM Module A2 Temperature
50	CPU0_DIMM_B1-TMP	01h	01h	20h	DIMM Module B1 Temperature
51	CPU0_DIMM_C1-TMP	01h	01h	20h	DIMM Module C1 Temperature
52	CPU0_DIMM_D1-TMP	01h	01h	20h	DIMM Module D1 Temperature
53	CPU0_DIMM_D2-TMP	01h	01h	20h	DIMM Module D2 Temperature
54	CPU0_DIMM_E1-TMP	01h	01h	20h	DIMM Module E1 Temperature
55	CPU0_DIMM_F1-TMP	01h	01h	20h	DIMM Module F1 Temperature
56	CPU1_DIMM_G1-TMP	01h	01h	20h	DIMM Module G1 Temperature
57	CPU1_DIMM_G2-TMP	01h	01h	20h	DIMM Module G2 Temperature
58	CPU1_DIMM_H1-TMP	01h	01h	20h	DIMM Module H1 Temperature
59	CPU1_DIMM_J1-TMP	01h	01h	20h	DIMM Module J1 Temperature
60	CPU1_DIMM_K1-TMP	01h	01h	20h	DIMM Module K1 Temperature
61	CPU1_DIMM_K2-TMP	01h	01h	20h	DIMM Module K2 Temperature
62	CPU1_DIMM_L1-TMP	01h	01h	20h	DIMM Module L1 Temperature
63	CPU1_DIMM_M1-TMP	01h	01h	20h	DIMM Module M1 Temperature



64	FAN1-SPEED	04h	01h	1Dh	FAN1 RPM
65	FAN2-SPEED	04h	01h	1Dh	FAN2 RPM
66	FAN3-SPEED	04h	01h	1Dh	FAN3 RPM
67	FAN4-SPEED	04h	01h	1Dh	FAN4 RPM
68	PSU1-FRU	-	-	-	PSU FRU Device Locator 1
69	PSU1_IN-POWER	0Bh	01h	0Ah	PSU input power sensor
70	PSU1_OUT-POWER	0Bh	01h	0Ah	PSU output power sensor
71	PSU1_IN-CUR	03h	01h	0Ah	PSU input current sensor
72	PSU1_OUT-CUR	03h	01h	0Ah	PSU output voltage sensor
73	PSU1_IN-VOL	02h	01h	0Ah	PSU input voltage sensor
74	PSU1_OUT-VOL	02h	01h	0Ah	PSU output voltage sensor
75	PSU1_INTAKE-TMP	01h	01h	0Ah	PSU intake temperature
76	PSU1_HOTSPOT-TMP	01h	01h	0Ah	PSU hot spot temperature
77	PSU1_FAN-SPEED	04h	01h	0Ah	PSU FAN Speed sensor
78	PSU2-FRU	-	-	-	PSU FRU Device Locator 2
79	PSU2_INTAKE-TMP	01h	01h	0Ah	PSU input power sensor
80	PSU2_HOTSPOT-TMP	01h	01h	0Ah	PSU output power sensor
81	PSU2_IN-VOL	02h	01h	0Ah	PSU input current sensor
82	PSU2_OUT-VOL	02h	01h	0Ah	PSU output voltage sensor
83	PSU2_IN-CUR	03h	01h	0Ah	PSU input voltage sensor
84	PSU2_OUT-CUR	03h	01h	0Ah	PSU output voltage sensor
85	PSU2_IN-POWER	0Bh	01h	0Ah	PSU intake temperature
86	PSU2_OUT-POWER	0Bh	01h	0Ah	PSU hot spot temperature
87	PSU2_FAN-SPEED	04h	01h	0Ah	PSU FAN Speed sensor

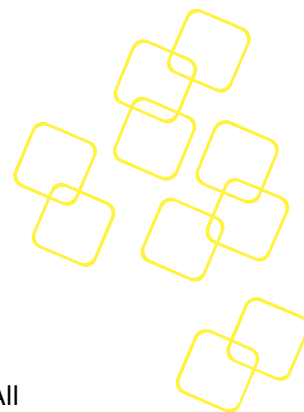
Table 12: BMC Sensor list

2.6.4 Sensor Thresholds

Sensor definitions and related properties are stored in Sensor Data Records (SDRs) located in the BMC's flash.

The IPMI specification defines the following thresholds as part of the sensor properties:

- **UNC** Upper Non-Critical
- **UC** Upper Critical
- **UNR** Upper Non-Recoverable



- **LNC** Lower Non-Critical
- **LC** Lower Critical
- **LNR** Lower Non-Recoverable

The threshold based sensor event data format is according to the IPMI specification. All defined sensors in this sub-section will use below described event data bytes 1 – 3:

Event Data 1	Event Data 2	Event Data 3
[7:6] = 01b (Trigger reading in byte 2), [5:4] = 01b (Trigger threshold in byte 3), [3:0] = Offset (threshold event)	Reading that triggered the event generation	Threshold value that was crossed

Table 13: Threshold based sensor event data format

All thresholds specified subsequent will generate events (assertion and deassertion event direction). Unused thresholds have the event generation disabled (event mask bits in SDR data not set).

Sensor Types	Type Codes	Generic Offset	Event
Temperature, Voltage, Current, Fan Speed, Power Supply	01h,	00h	Lower Non-critical - going low
	02h,	01h	Lower Non-critical - going high
	03h,	02h	Lower Critical - going low
	08h	03h	Lower Critical - going high
		04h	Lower Non-recoverable - going low
		05h	Lower Non-recoverable - going high
		06h	Upper Non-critical - going low
		07h	Upper Non-critical - going high
		08h	Upper Critical - going low
		09h	Upper Critical - going high
		0Ah	Upper Non-recoverable - going low
		0Bh	Upper Non-recoverable - going high

Table 14: Threshold based sensor supported events

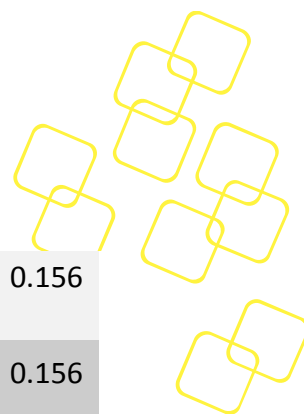
Following subchapters just use the usual abbreviations for the different thresholds (LNR, LCR, LNC and UNC, UCR, UNR).

2.6.4.1 Voltage Sensors

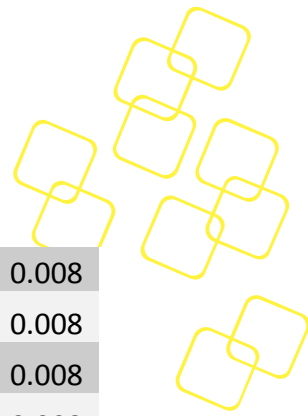
The input, standby and most payload power voltages are monitored by the BMC.

All sensors use a hysteresis for the event generation. For all sensors, the hysteresis is 1.5% of the nominal sensor voltage plus ripple from the voltage regulator.

Sensor Name	Nominal	LNR	LCR	LNC	UNC	UCR	UNR	HYST
-------------	---------	-----	-----	-----	-----	-----	-----	------



P12V-VOL	12		11.388			12.636		0.156
P12V_STBY-VOL	12		11.388			12.636		0.156
P5V-VOL	5		4.748			5.279		0.066
P5V_AUX-VOL	5		4.748			5.279		0.066
P3V3-VOL	3.3		3.133			3.47		0.048
P3V3_AUX-VOL	3.3		3.133			3.47		0.048
BAT_3_0-VOL	3		2.699			3.302		0.048
PVPP_ABC-VOL	2.6		2.488			2.714		0.035
PVPP_DEF-VOL	2.6		2.488			2.714		0.035
PVPP_GHJ-VOL	2.6		2.488			2.714		0.035
PVPP_KLM-VOL	2.6		2.488			2.714		0.035
PVCCIN_CPU0-VOL	1.8		1.694			1.912		0.024
PVCCIN_CPU1-VOL	1.8		1.694			1.912		0.024
P1V8_PCH_AUX-VOL	1.8		1.694			1.912		0.024
PVDDQ_ABC-VOL	1.2		1.093			1.306		0.021
PVDDQ_DEF-VOL	1.2		1.093			1.306		0.021
PVDDQ_GHJ-VOL	1.2		1.093			1.306		0.021
PVDDQ_KLM-VOL	1.2		1.093			1.306		0.021
P1V2_DP501-VOL	1.2		1.093			1.306		0.021
P1V05_PCH-VOL	1.05		1			1.104		0.016
PVCCIO_CPU0-VOL	1		0.928			1.056		0.016
PVCCIO_CPU1-VOL	1		0.928			1.056		0.016
PVNN_PCH_AUX-VOL	0.9		0.8			1		0.016
PVCCSA_CPU0-VOL	0.85		0.736			0.952		0.016
PVCCSA_CPU1-VOL	0.85		0.736			0.952		0.016



PVTT_ABC-VOL	0.6		0.496			0.704		0.008
PVTT_DEF-VOL	0.6		0.496			0.704		0.008
PVTT_GHJ-VOL	0.6		0.496			0.704		0.008
PVTT_KLM-VOL	0.6		0.496			0.704		0.008

Table 15: Voltage Sensor list

2.6.4.2 Temperature Sensors

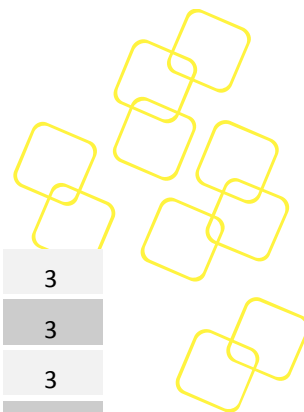
Several temperature sensors are supported, either via board populated IC's (e.g. TMP75) or Intel PECI readings from CPU.

All temperature sensors use a hysteresis of about 3°C for the event generation.

To prevent false temperature sensor events caused by a single incorrect reading, a mechanism is implemented to ignore those readings. Each new reading is compared to the previous reading and if the difference is more than 10°C, the reading is ignored.

This only applies for one reading. If there are more false readings in a row, the reading will be treated as correct reading and a temperature sensor event might be generated. This avoids stuck temperature sensors. The same mechanism is implemented for "0xFF" and "0x00" temperature value readings.

Sensor Name	Nominal	LNR	LCR	LNC	UNC	UCR	UNR	HYST
INLET-TMP					55	65	75	3
OUTLET-TMP					65	75	85	3
CPU0-TMP					94	97		3
CPU1-TMP					94	97		3
HWM-TMP					60	70	80	3
PCH-TMP					90	100		3
CPU0_DIMM_A1-TMP					85	95		3
CPU0_DIMM_A2-TMP					85	95		3
CPU0_DIMM_B1-TMP					85	95		3
CPU0_DIMM_C1-TMP					85	95		3
CPU0_DIMM_D1-TMP					85	95		3
CPU0_DIMM_D2-TMP					85	95		3
CPU0_DIMM_E1-TMP					85	95		3
CPU0_DIMM_F1-TMP					85	95		3
CPU1_DIMM_G1-TMP					85	95		3
CPU1_DIMM_G2-TMP					85	95		3
CPU1_DIMM_H1-TMP					85	95		3
CPU1_DIMM_J1-TMP					85	95		3



CPU1_DIMM_K1-TMP					85	95		3
CPU1_DIMM_K2-TMP					85	95		3
CPU1_DIMM_L1-TMP					85	95		3
CPU1_DIMM_M1-TMP					85	95		3

Table 16: Temperature Sensor list

2.6.4.3 PSU Sensor

PSU Current Sensor

The used system power supply (PSU) provides an input and output current draw reading. Both power values are available to read and are supported by BMC as PSU current sensors.

The BMC PSU current sensors use the Sensor Type code 03h and the Event/Reading Type code of 01h (threshold). See below tables for supported thresholds (events).

PSU Power Sensor

The actual used PSU power is calculated and IPMI readable via two PSU power sensors (Sensor Type code 0Bh). These sensors indicate the instantaneous, power supply reported, input and output power consumption.

The power consumption sensors use Event/Reading Type code 01h (threshold). Please see below tables for the supported power thresholds (events).

Each PSU does provide two temperature sensors (unit degrees C) with following threshold values:

Sensor Name	Nominal	LNR	LCR	LNC	UNC	UCR	UNR	HYST
PSU1_INTAKE-TMP	40	-	-	-	50	60	70	3
PSU1_HOTSPOT-TMP	40	-	-	-	50	60	70	3
PSU2_INTAKE-TMP	40	-	-	-	50	60	70	3
PSU2_HOTSPOT-TMP	40	-	-	-	50	60	70	3

Table 17: PSU temperature sensor threshold list

The main PSU sensors are the two voltage sensors (unit Volts) available. Below defined threshold values are defined.

Sensor Name	Nominal	LNR	LCR	LNC	UNC	UCR	UNR	HYST
PSU1_IN-VOL	220	-	90	-	-	264	-	2
PSU1_OUT-VOL	12	-	11.5	-	-	12.8	-	0.1
PSU2_IN-VOL	220	-	90	-	-	264	-	2
PSU2_OUT-VOL	12	-	11.5	-	-	12.8	-	0.1

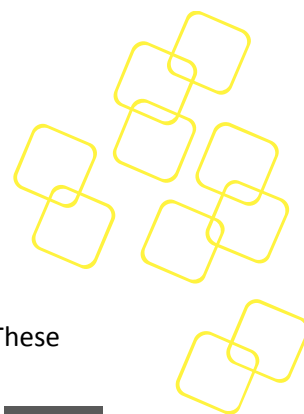


Table 18: AC PSU voltage sensor threshold list

The used PSU models do provide current readings for the input and output voltages. These readings are available via current sensors (unit Amps).

Sensor Name	Nominal	LNR	LCR	LNC	UNC	UCR	UNR	HYST
PSU1_IN-CUR	0.75	-	-	-	-	11	-	0.25
PSU1_OUT-CUR	5	-	-	-	-	65	-	1
PSU2_IN-CUR	0.75	-	-	-	-	11	-	0.25
PSU2_OUT-CUR	5	-	-	-	-	65	-	1

Table 19: DC 800W PSU voltage sensor threshold list

Sensor Name	Nominal	LNR	LCR	LNC	UNC	UCR	UNR	HYST
PSU1_IN-CUR	0.75	-	-	-	-	14	-	0.25
PSU1_OUT-CUR	5	-	-	-	-	97	-	1
PSU2_IN-CUR	0.75	-	-	-	-	14	-	0.25
PSU2_OUT-CUR	5	-	-	-	-	97	-	1

Table 20: AC 1200W PSU current sensor threshold list

Sensor Name	Nominal	LNR	LCR	LNC	UNC	UCR	UNR	HYST
PSU1_IN-CUR	2	-	-	-	-	12	-	0.25
PSU1_OUT-CUR	13	-	-	-	-	114	-	1
PSU2_IN-CUR	2	-	-	-	-	12	-	0.25
PSU2_OUT-CUR	13	-	-	-	-	114	-	1

Table 21: DC 1400W PSU current sensor threshold list

As each PSU does include one fan, a fan speed sensor (unit RPM, revolutions per minute) is provided per power supply with below specified thresholds.

Sensor Name	Nominal	LNR	LCR	LNC	UNC	UCR	UNR	HYST
PSU1_FAN-SPEED	4000	-	1200	-	-	30000	-	150
PSU2_FAN-SPEED	4000	-	1200	-	-	30000	-	150

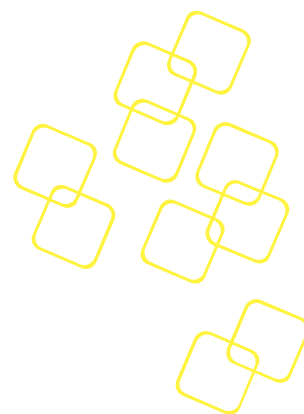


Table 22: PSU speed sensor threshold list

Finally PSU power consumption values for input and output direction are provided via following sensors (unit Watt).

Sensor Name	Nominal	LNR	LCR	LNC	UNC	UCR	UNR	HYST
PSU1_IN-POWER	80	-	-	-	-	800	-	10
PSU1_OUT-POWER	80	-	-	-	-	800	-	10
PSU2_IN-POWER	80	-	-	-	-	800	-	10
PSU1_OUT-POWER	80	-	-	-	-	800	-	10

Table 23: AC 800W PSU power consumption sensor threshold list

Sensor Name	Nominal	LNR	LCR	LNC	UNC	UCR	UNR	HYST
PSU1_IN-POWER	80	-	-	-	-	1200	-	10
PSU1_OUT-POWER	80	-	-	-	-	1200	-	10
PSU2_IN-POWER	80	-	-	-	-	1200	-	10
PSU1_OUT-POWER	80	-	-	-	-	1200	-	10

Table 24: AC 1200W PSU power consumption sensor threshold list

Sensor Name	Nominal	LNR	LCR	LNC	UNC	UCR	UNR	HYST
PSU1_IN-POWER	80	-	-	-	-	1400	-	10
PSU1_OUT-POWER	80	-	-	-	-	1400	-	10
PSU2_IN-POWER	80	-	-	-	-	1400	-	10
PSU1_OUT-POWER	80	-	-	-	-	1400	-	10

Table 25: AC 1400W PSU power consumption sensor threshold list

2.6.5 Discrete Specific Sensors

The sections sub-chapters describe all BMC implemented discrete IPMI sensors (event reading type code 6Fh) in detail.

2.6.5.1 BMC Health Sensor

The IPMI defined Management Subsystem Health sensor (type code 28h) is part of the designs sensor repository with below specified event data format.



Event Direction	Event Data 1	Event Data 2	Event Data 3
[7] = 0b (Assertion) 1b (Deassertion)	[7:4] = Ch (sensor-specific event extension code in byte 2, unspecified byte 3), [3:0]: Specific Offset	[7:0] = Sensor number	FFh (unspecified)

Table 26: BMC Health Sensor event data format

Following BMC health events can be generated by this sensor:

Sensor Type	Type Code	Specific Offset	Event
Management	28h	01h	Controller access degraded or unavailable
Subsystem Health		04h	Sensor failure (number in Event Data 2)

Table 27: BMC Health Sensor supported events

2.6.5.2 Version Change Sensor

A Version Change sensor with IPMI sensor type code 2Bh is supported according to the IPMI specification.

Event Direction	Event Data 1	Event Data 2	Event Data 3
[7] = 0b (Assertion)	[7:4] = Ch (sensor-specific event extension code in byte 2, unspecified byte 3), [3:0]: Specific Offset	[7:0] = Version change type (see table below)	FFh (unspecified)

Table 28: Version Change Sensor event data format

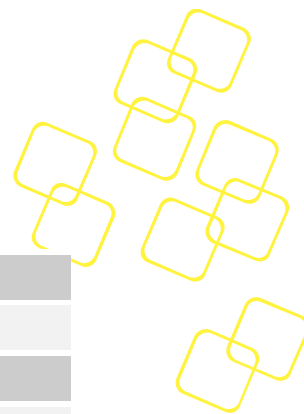
The Version Change sensor is used to generate events in case of specific Software changes.

Sensor Type	Type Code	Specific Offset	Event
Version Change	2Bh	01h	Firmware or software change detected with associated Entity. Informational. Success or failure not implied.

Table 29: Version Change Sensor supported events

Some SW components available in the board design are defined by the IPMI specification for this sensor type and indicated by the event data byte 2.

Event Data 2	Version Change Type
00h	Unspecified (BMC FRU data)



02h	Management controller firmware revision (BMC)
07h	Management controller firmware boot block (BMC Uboot)
09h	System firmware (EFI / BIOS) change (BIOS)
11h	Programmable hardware change (e.g. FPGA)

Table 30: Version Change Sensor event data byte 2

2.6.5.3 BMC Watchdog Sensor

The BMC Watchdog sensor is supported according to the Watchdog 2 sensor type listed in the IPMI specification.

Event Direction	Event Data 1	Event Data 2	Event Data 3
[7] = 0b (Assertion)	[7:4] = Ch (sensor-specific event extension code in byte 2, unspecified byte 3), [3:0]: Specific Offset	[7:4] = Interrupt type [3:0] = Timer use at expiration (see table below)	FFh (unspecified)

Table 31: BMC Watchdog Sensor event data format

The IPMI defined BMC Watchdog supports following events for this BMC:

Sensor Type	Type Code	Specific Offset	Event
Watchdog 2	23h	00h	Timer expired, status only (no action)
		01h	Hard Reset
		02h	Power Down
		03h	Power Cycle

Table 32: BMC Watchdog Sensor supported events

The event data 2 field for the BMC Watchdog sensor provides an event extension code:

Value	Interrupt type [7:4]	Timer use at expiration [3:0]
0h	None	reserved
1h	SMI	BIOS FRB2
2h	NMI	BIOS/POST
3h	Messaging Interrupt	OS Load
4h	reserved	SMS/OS
5h	reserved	OEM
Fh	unspecified	unspecified

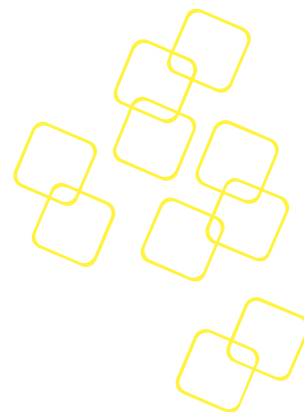


Table 33: BMC Watchdog Sensor event data byte 2

2.6.5.4 ACPI Power Sensor

The design provides an IPMI System ACPI Power State sensor to reflect the payload x86 system supported power states defined by ACPI.

Event Direction	Event Data 1	Event Data 2	Event Data 3
[7] = 0b (Assertion)	[7:4] = 0h (unspecified Event Data bytes 2 and 3), [3:0]: Specific Offset	FFh (unspecified)	FFh (unspecified)

Table 34: System ACPI Power State Sensor event data format

Below table summarizes the supported events of this sensor.

Sensor Type	Type Code	Specific Offset	Event
System ACPI Power State	22h	00h	S0 / G0 "working"
		05h	S5 / G2 "soft-off"
		07h	G3 / Mechanical Off

Table 35: System ACPI Power State Sensor supported events

2.6.5.5 Processor State Sensor

A processor sensor according to the IPMI specification is implemented with support for several CPU related events.

Event Direction	Event Data 1	Event Data 2	Event Data 3
[7] = 0b (Assertion)	[7:6] = 10b (OEM code in Event Data 2), [5:4] = 10b or 00b (event specific, see below) [3:0]: Specific Offset	(see table below)	BIOS POST code, if specific offset = 03h (Hang in POST failure), otherwise FFh (unspecified).

Table 36: Processor Sensor event data format

The sensor event data byte 2 holds the CPU source of the above defined events (if distinguishable) as specified below:

Value	Processor Event Source
00h	CPU 0
01h	CPU 1

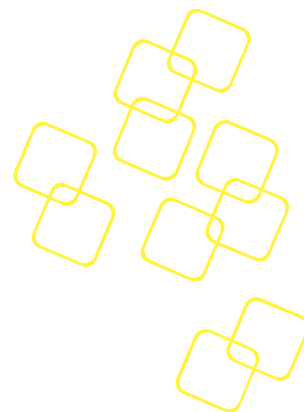


Table 37: Processor Event Sources

The available events are specified in below table.

Sensor Type	Type Code	Specific Offset	Event
Processor	07h	00h	IERR
		01h	Thermal Trip
		03h	FRB2/Hang in POST failure
		0Ah	Processor Automatically Throttled (PROCHOT)
		0Bh	Machine Check Exception (Uncorrectable)
		0Ch	Correctable Machine Check Error

Table 38: Processor Sensor supported events

A FRB2/Hang in POST failure (offset 03h) event will be generated if the BMC Watchdog bits with timer use BIOS FRB2. The current BIOS POST code will be logged in event data 3 for this event in addition (event data byte 1, [7:4] = Ah, OEM code in Event Data 2 and 3).

For all other supported sensor event offsets, the event data 3 will be unused (FFh, unspecified). Thus event data byte 1 [7:4] will be filled with 8h (OEM code in Event Data 2 and unspecified byte 3) for all events other than 03h.

2.6.5.5.1 IERR

IERR is a critical processor internal error, which may indicate a

- processor unrecoverable error
- non-CPU event, such as a system BUS interruption
- memory unrecoverable error

The error details will be printed out on the payload console and the system is not operational anymore and stops responding.

2.6.5.5.2 Thermal Trip

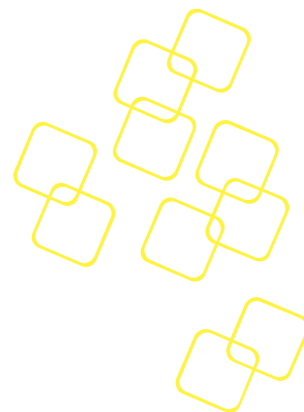
THERMTRIP will be activated, when the CPU internal temperature monitoring sensor detects that the CPU reached critical junction temperature, where permanent damage of the silicon may happen.

Upon assertion of THERMTRIP, the CPU will shut off its internal clocks trying to reduce the processor junction temperature.

The error details will be printed out on the payload console and the system is not operational anymore and stops responding.

2.6.5.5.3 FRB2/Hang In POST Fail

A FRB2/Hang in POST failure event will be generated if the BMC Watchdog bits, with timer use set to BIOS FRB2.



This may happen when:

- BIOS image is corrupted / empty
- Hardware failure preventing BIOS from normal startup
- Watchdog handling not activated in BIOS

2.6.5.5.4 **PROCHOT**

PROCHOT will be activated, when the CPU internal temperature monitoring sensor detects that the CPU reached the maximum tested operating temperature.

This indicates that the CPU activated its thermal control circuit, throttling to lower CPU frequency, trying to reduce its power dissipation and thus reducing its temperature.

2.6.5.5.5 **Uncorrectable Machine Check Exceptions**

Processors are designed to be able to handle and correct certain errors that may occur. If this is not possible, an Uncorrectable Machine Check Exception will be indicated by the CPU, on:

- Memory errors or Error Correction Code (ECC) problems
- Inadequate cooling / processor over-heating
- System bus errors
- Cache errors in the processor or hardware

The error details will be printed out on the payload console and the system is not operational anymore and stops responding.

2.6.5.5.6 **Correctable Machine Check Exceptions**

Processors are designed to be able to handle and correct certain errors that may occur. These types of errors are called Correctable Errors. A Correctable Machine Check Error is a warning error, indicated by the CPU, on:

- Memory errors or Error Correction Code (ECC) problems
- Inadequate cooling / processor over-heating
- System bus errors
- Cache errors in the processor or hardware

The system is still operational. The warning will be logged in the payload system logs.

2.6.5.6 **Reset Sensor**

The IPMI defined “System Boot / Restart Initiated” sensor is available in the BMC SDR. This sensor is intended to acknowledge about payload resets and is asserted with board resets.

The event message for this sensor is filled with following content:

Event Direction	Event Data 1	Event Data 2	Event Data 3
[7] = 0b (Assertion)	[7:4] = Fh (sensor-specific event extension code in byte 2 and byte 3), [3:0]: Specific Offset	FFh or reset cause if specific offset is 07h (see table below)	FFh or channel number if specific offset is 07h

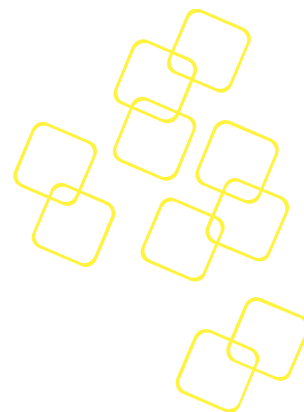


Table 39: Reset Sensor event data format

Following event offsets are supported for this System Boot / Restart Initiated sensor:

Sensor Type	Type Code	Specific Offset	Event
System Boot / Restart Initiated	1Dh	00h	Initiated by power up
		01h	Initiated by hard reset
		02h	Initiated by warm reset
		07h	System Restart (with use of Event Data bytes2 and 3)

Table 40: Reset Sensor supported events

Details to the occurred system restart are available in the event data byte 2, bits [3:0] (bits [7:4] are reserved), if event data 1 specific offset is 07h (System Restart). The reset cause is similar as returned by the Get System Restart Cause command.

Event Data 2 [3:0]	Restart Cause
0h	Unknown (system start/restart detected, but cause unknown)
1h	Chassis Control command [required]
3h	Power-up via power pushbutton (front panel handle) [optional]
4h	Watchdog expiration (see watchdog flags) [required]

Table 41: Reset Sensor event data byte 2

If the reset cause is watchdog (4h), additional details are located in the BMC Watchdog sensor.

If event data 1 specific offset is 07h (System Restart), the event data byte 3 holds the channel number used to deliver command that generated restart (per Get System Restart Cause command).

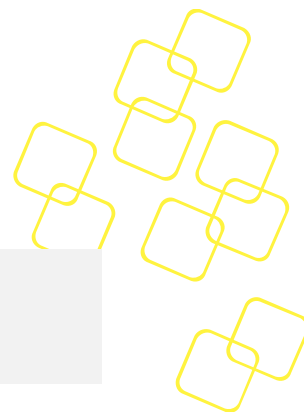
2.6.5.7 FW Progress Sensor

The BMC SDR contains an IPMI defined FW Progress sensor (formerly POST Error sensor) in order to support logging of the OS boot process. The BMC supports adding and forwarding of SEL entries from the BIOS/OS system firmware progress events by sending 'Add SEL Entry' commands with the matching sensor type to the BMC through the KCS interface.

AMI BIOS supports the System Firmware Progress Sensor to report the current status to the SEL.

The sensors event messages have below specified event data format:

Event Direction	Event Data 1	Event Data 2	Event Data 3
[7] =	[7:4] = Ch (sensor-specific	(see table below)	FFh (unspecified)



0b (Assertion)	event extension code in byte 2, unspecified byte 3), [3:0]: Specific Offset		
----------------	--	--	--

Table 42: FW Progress Sensor event data format

The following events are currently supported by BIOS and thus for the System Firmware Progress Sensor:

Sensor Type	Type Code	Specific Offset	Event
System Firmware Progress	0Fh	00h	System Firmware Error (POST Error)
		01h	System Firmware Hang
		02h	System Firmware Progress

Table 43: FW Progress Sensor supported events

Extension codes for the System Firmware Progress events are provided in Event Data byte 2. Complete lists can be found in the IPMI 2.0 specification. Following sub events are supported by the sensor on this product for the Event Data 1 offset 02h:

Value	Event Extension Code
01h	Memory initialized
02h	Hard-Disk Initialization
03h	Secondary processor(s) initialization
07h	PCI Resource Configuration
09h	Video Initialization
0Ch	Keyboard Controller Initialization
13h	Starting operating system boot process

Table 44: FW Progress Sensor event data byte 2

2.6.5.8 Power Supply Sensor

The IPMI Power Supply sensor (type code 08h) is defined for the BMC to provide information about the systems power supply units (PSUs).

No.	Power Supply
1	PSU 1
2	PSU 2

Table 45: Power Supply Sensor entities



The power supply IPMI sensor event data details are described below:

Event Direction	Event Data 1	Event Data 2	Event Data 3
[7] = 0b (Assertion) 1b (De-assertion)	[7:4] = 0h (unspecified byte 2 and byte 3) or [7:4] = 3h (unspecified byte 2, sensor-specific event extension code in byte 3), [3:0]: Specific Offset	FFh	[7:4] = 0h and [3:0] = Error Type for specific offset 06h, otherwise FFh

Table 46: Power Supply event data format

Following sensor event specific offsets are available for the IPMI Power Supply sensor:

Sensor Type	Type Code	Specific Offset	Event
Power Supply	08h	00h	Presence detected
		01h	Power Supply Failure detected
		06h	Configuration error (type in byte 3)

Table 47: Power Supply Sensor supported events

For configuration error cases (specific offset 06h), the IPMI specification defines error types for this sensor (provided via event data byte 3). This BMC does only use the voltage mismatch type.

Event Data 3	Error Type
4h	Voltage rating mismatch. The voltage rating of the supply does not match the system's requirements.

Table 48: Power Supply Sensor event data byte 3

2.6.5.9 Entity Presence Sensors

Several IPMI Entity Presence sensors are used by the BMC to indicate the presence of pluggable system entities inside the SKY-8201 system.

No.	Entity
1	Fan Module 1
2	Fan Module 2
3	Fan Module 3
4	Fan Module 4
5	Fan Module 5
6	Fan Module 6

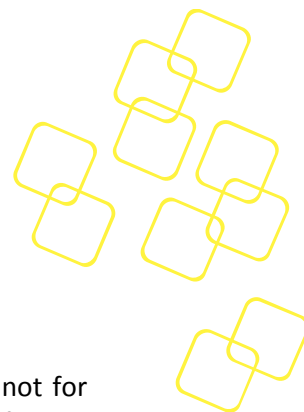


Table 49: Available presence sensor entities

The Entity Presence sensor type is typically used to provide a present reading and not for event generation. But for SKY-8201 fan modules, the BMC do generate sensor events for this IPMI sensor type.

Event Direction	Event Data 1	Event Data 2	Event Data 3
[7] = 0b (Assertion)	[7:4] = 0h (unspecified byte 2 and byte 3), [3:0]: Specific Offset	FFh	FFh

Table 50: Presence Sensor event data format

Following sensor readings and event specific offsets are available to judge about the above specified entities presence:

Sensor Type	Type Code	Specific Offset	Event
Entity Presence	25h	00h	Entity Present
		01h	Entity Absent

Table 51: Entity Presence Sensor supported readings

2.6.5.10 Physical Security Sensor

An IPMI defined Physical Security (Chassis Intrusion) sensor with IPMI sensor type code 05h is supported according to the IPMI specification.

Event Direction	Event Data 1	Event Data 2	Event Data 3
[7] = 0b (Assertion) 1b (De-assertion)	[7:4] = 0h (unspecified Event Data bytes 2 and 3), [3:0]: Specific Offset	FFh (unspecified)	FFh (unspecified)

Table 52: Physical Security Sensor event data format

The Physical Security sensor is used to generate events in case of intrusion to specific HW areas in the system. The BMC does support the chassis intrusion event if a user opens the chassis physically (via HWM case open pin and connected intrusion HW switch).

Sensor Type	Type Code	Specific Offset	Event
Physical Security	05h	00h	General Chassis Intrusion

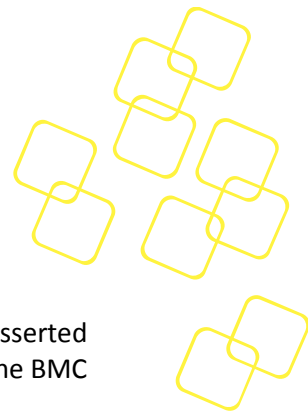


Table 53: Physical Security Sensor supported events

An assertion event is created if the chassis top cover is opened (and sensor was not asserted before). Chassis intrusion is latched in HWM and BMC will log the event to SEL once the BMC is alive, no matter if standby power or AC on.

De-assertion event is only triggered if the cover is closed (open before) and the sensor is re-armed (e.g. via sensor rearm IPMI command). Means without the rearm, no event is created even the chassis is closed.

The chassis intrusion sensor will only trigger again if rearmed by IPMI command. Otherwise it will stay disabled after having triggered once.

2.6.6 Advantech OEM IPMI Sensors

All following sub-chapters do describe Advantech defined, discrete OEM IPMI sensors (OEM event reading type code 70h) in detail.

2.6.6.1 OEM BIOS POST code Sensor

The Advantech BIOS POST code sensor is a discrete OEM IPMI sensor (OEM event reading type code 70h with sensor type code 0Fh, System Firmware Progress) to allow users to read the actual BIOS POST code (Port 80h code) similar to the Advantech Read Port 80 OEM IPMI command.

The BIOS POST code sensor is intended to provide a reading (only) and thus don't support sensor event generation. Please verify the Processor State IPMI sensor and the special offset "FRB2/Hang in POST failure" (3h), because this sensor does log the POST code in SEL (in event data 3 byte) if BIOS does hang.

2.6.6.2 OEM Integrity Sensor

The Advantech Integrity Sensor is an OEM sensor according to the SDR (Sensor Data Record) definitions in the IPMI specification. Its main purpose is to monitor internal firmware states and report events to the operator that would otherwise go unnoticed (hence "integrity sensor").

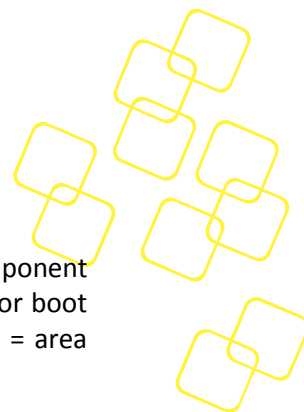
Examples for those events are checksum errors, firmware update success/failure, and firmware rollbacks.

2.6.6.3 Sensor Characteristics

The Integrity sensor does not support sensor reading, but generates event messages only. These events are stored in the local System Event Log (SEL) and sent to the default event receiver.

The event message contains three bytes of event data. The first byte defines how the event is supposed to be treated: the value of 0xA0 defines that event data 2 and 3 contain OEM data (please verify the IPMI specification for details on OEM sensors).

Event data 2 is used to identify which component the event relates to. This can either be a HPM.1 component, a logical component/feature on the board (for example FRU, RTC) or simply a board specific event.



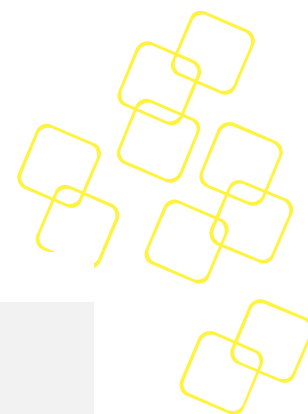
Event data 3 [7..3] identifies the action or a subcomponent. For example: If the component in byte 2 was a HPM.1 component, it might report if this was an update, a rollback, or boot failure. If the component in byte 2 was “FRU”, it might indicate the subcomponent = area within the FRU that the event relates to.

Event data 3 [2..0] holds the result code. For the HPM.1 example above, it might report that an update or rollback either succeeded or failed. For the FRU example, it might indicate a checksum error.

2.6.6.3.1 Event Data Byte Definition

The following list provides the exact Integrity sensor event bytes definition.

Data Byte	[Bit]	Description	Value	Event Data
1	[7:0]	IPMI Header	0xA0	Event data 2 & 3 used as OEM data
2	[7:0]	Component	0x00 – 0x07 0x08 – 0xFE 0xFF	HPM.1 component (FW, FPGA, BIOS...) Logical component (FRU, RTC...) Board specific event
3	[7:3]	Action / Subcomponent	b00000 b00001 b00010 b00011 b00100 b00101 b00110 b00111 b01000 b01001 b01010 b01011 b01100 b01101 b01110 b01111 b10000 b10001 b10010 b10011 b10100 b10101 b10110 b10111...	Update Recovery/Rollback Manual Rollback Automatic Rollback Activation Flash 0 Boot Flash 1 Boot Common Header Internal Area Chassis Info Area Board Info Area Product Info Area Multi Record Area Time synchronization Graceful Shutdown IP Address VBAT low SW WDT bite Power Boot POST Unexpected Version Change Manufacturing Mode Not defined yet...



			...b11111	Not defined yet
3	[2:0]	Result	b000	Successful
			b001	Failed
			b010	Aborted
			b011	Checksum Error
			b100	Timeout
			b101	Initiated
			b110	Finished
			b111	Unspecified Error

Table 54: Integrity Sensor event byte definition

2.6.6.3.2 Event Data Translation

The structured definition allows simple translation of each Integrity Sensor event message. Below is an example Integrity Sensor SEL event (0x0A0100). The three event data bytes could be translated in following manner:

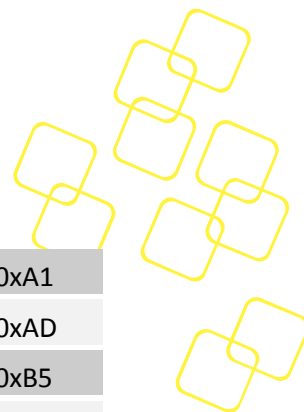
Data 1: 0x0A: Header
Data 2: 0x01: logical Component (BMC FW)
Data 3: 0x00: b 0 0 0 0 0 0 0 0
Update Successful

The example Integrity Sensor event reports a successful BMC Firmware update.

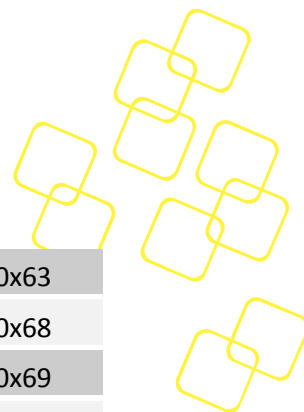
2.6.6.3.3 Event Data Table

All event data combinations supported by the BMC Integrity Sensor can be found in the following list.

Component	Action / Subcomponent	Result	Byte 1	Byte2
BMC FW	Update	Successful	0x01	0x00
	Update	Timeout	0x01	0x04
	Update	Aborted	0x01	0x02
	Activation	Failed	0x01	0x21
	Manual Rollback	Initiated	0x01	0x15
	Manual Rollback	Successful	0x01	0x10
	Manual Rollback	Failed	0x01	0x11
	Automatic Rollback	Initiated	0x01	0x1D
	Automatic Rollback	Successful	0x01	0x18
	Automatic Rollback	Failed	0x01	0x19
	Graceful Shutdown	Timeout	0x01	0x74



	POST	Failed	0x01	0xA1
	Unexpected Version Change	Initiated	0x01	0xAD
	Manufacturing Mode	Initiated	0x01	0xB5
	Power (On Boot)	Successful	0x01	0x90
	Boot (Reboot)	Successful	0x01	0x98
	Watchdog (Reset Boot)	Successful	0x01	0x88
FPGA	Update	Successful	0x02	0x00
	Update	Timeout	0x02	0x04
	Update	Aborted	0x02	0x02
	Activation	Failed	0x02	0x21
	Manual Rollback	Initiated	0x02	0x15
	Manual Rollback	Successful	0x02	0x10
	Manual Rollback	Failed	0x02	0x11
	Automatic Rollback	Initiated	0x02	0x1D
	Automatic Rollback	Successful	0x02	0x18
	Automatic Rollback	Failed	0x02	0x19
BIOS	Update	Successful	0x03	0x00
	Update	Timeout	0x03	0x04
	Update	Aborted	0x03	0x02
	Activation	Failed	0x03	0x21
	Flash 0 Boot	Failed	0x03	0x29
	Flash 1 Boot	Failed	0x03	0x31
	Boot	Failed	0x03	0x99
	Unexpected Version Change	Initiated	0x03	0xAD
NVRAM	Update	Successful	0x04	0x00
	Update	Timeout	0x04	0x04
	Update	Aborted	0x04	0x02
	Activation	Failed	0x04	0x21
	Boot	Failed	0x04	0x99
	Unexpected Version Change	Initiated	0x04	0xAD
FRU	Common Header	Checksum Error	0x08	0x3B
	Internal Area	Checksum Error	0x08	0x43
	Board Info Area	Checksum Error	0x08	0x53
	Product Info Area	Checksum Error	0x08	0x5B



	Multi Record Area	Checksum Error	0x08	0x63
RTC	Time sync	Successful	0x09	0x68
	Time sync	Failed	0x09	0x69
Payload	Power	Failed	0x0B	0x91
EEPROM	Automatic Rollback	Initiated	0x0C	0x1D
Power Input Module	Power	Failed	0x0D	0x91

Table 55: Integrity Sensor event data table

2.6.7 Other SDR Record Types

The BMC SDR data includes some non-sensor records in addition to Full or Compact Sensor Records (Type 01h / 02h) described in previous chapters.

2.6.7.1 BMC Device Locator

The BMC maintains a Device Locator Record for its own management controller identification (FRU Device ID 0). This Management Controller Device Locator Record (Type 12h) is used to hold location and type information of the BMC.

2.7 Thermal Management

The SKY-8201 system fans are controlled by BMC based on temperature sensor events. To ensure system components against damage caused by overheating, the BMC is able to change system fan speed(s) if needed. Means proper thermal conditions inside the chassis are monitored and controlled by BMCs cooling management.

2.7.1 Cooling Management

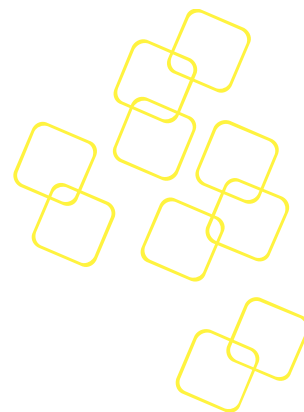
The SKY-8201 uses a Hardware Monitor, located on the main board, for fan module control. The BMC is connected via I2C channel 1 to the HWM slave I2C interface. This HWM is responsible for all 12 system fans (on the six fan modules).

The BMC supports two temperature zones, because SKY-8201 system is divided into two thermal zones (one for CPU and other for PCIe cards).

2.7.1.1 Temperature Zone 1 – CPU

The speed of system fans 1 and 2 will be adjusted according to the one processor temperature sensor “CPU0-TMP” reading to control the processor 1 area cooling. Furthermore, a default fan table will ensure correct system temperature(s).

Temperature (°C)		Fan Duty (%)
T1	45	20
T2	55	35
T3	65	50



T4	80	75
TR Critical	90 (default)	100

Table 56: Fan Speed and CPU0 Temperature Mapping

Temperature (°C)		Fan Duty (%)
T1	45	10
T2	55	25
T3	65	40
T4	80	65
TR Critical	90 (default)	100

Table 57: Fan Speed and CPU0 Temperature Mapping (20 inches system)

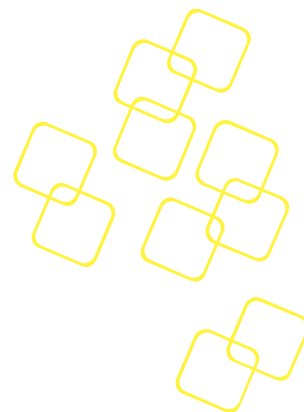
The NCT7904D HWM will use the highest CPU temperature as temperature source to control the affected temperature zone fans. The CPUs maximum junction temperature (Tjmax) will automatically retrieved by BMC through the PECL interface as soon as the payload (x86) is powered on. The BMC uses the read Tjmax value to configure the base temperature (Tbase) in the HWM.

2.7.1.2 Temperature Zone 2 – CPU1

The speed of system fans 3 and 4 will be adjusted according to the temperature sensor “CPU1-TMP” reading. The second thermal zone covers the processor 2 area. Furthermore, a default fan table will ensure correct system temperature(s).

Temperature (°C)		Fan Duty (%)
T1	45	20
T2	55	35
T3	65	50
T4	80	75
TR Critical	90 (default)	100

Table 58: Fan Speed and CPU1 Temperature Mapping



Temperature (°C)		Fan Duty (%)
T1	45	10
T2	55	25
T3	65	40
T4	80	65
TR Critical	90 (default)	100

Table 59: Fan Speed and CPU1 Temperature Mapping(20 inches system)

2.7.1.3 Thermal Protection

In order to deal with exceptional situations, the BMC fan control implements an override mechanism where the BMC will set the fans to full speed if any temperature sensor crosses any upper limit. BMC will keep the fans at full speed until all sensors are back in normal operating ranges. The BMC will return to normal fan speeds only in this case, otherwise maximum possible fan speeds are kept.

Furthermore, if any of the system fans do fail and not proper work, all remaining fans will be set to full speed.

2.7.2 Fan Modules

The SKY-8201 system includes 4 single fans, distributed on 4 fan modules (one fan per fan module). And 20 inches system includes 6 single fans, distributed on 3 fan modules (two fans per fan module). The fan modules are connected to a separate fan board via box headers. The fan board itself is connected to the SKY-8201 main board (with the BMC and fan control HWM) via a pin header.



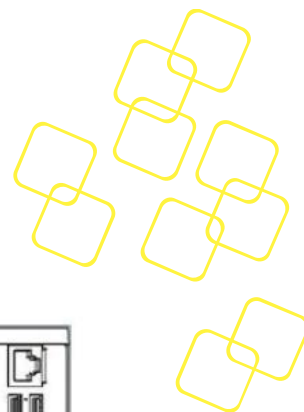


Figure 16: Fan Module

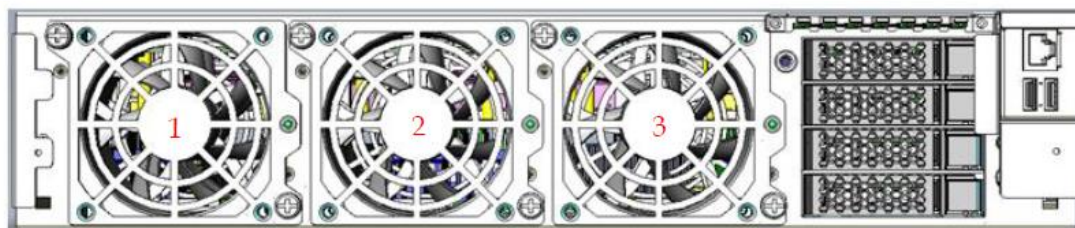


Figure 17: Fan Module(20 inches system)

2.7.3 Fan Sensors

In addition to the BMC build-in sensors, following chassis fan module sensors are provided by the BMC FW.

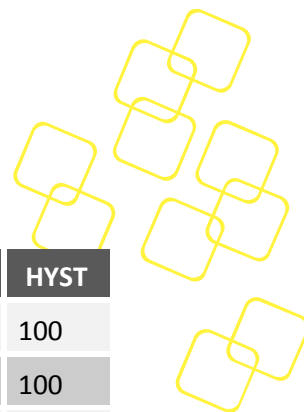
No	Sensor ID	Sensor Type	Event Reading Type	Entity ID	Description
1	FAN1-SPEED	04h	01h	17h	RPM of fan 1
2	FAN2-SPEED	04h	01h	17h	RPM of fan 2
3	FAN3-SPEED	04h	01h	17h	RPM of fan 3
4	FAN4-SPEED	04h	01h	17h	RPM of fan 4

Table 60: Fan sensor list

N o	Sensor ID	Sensor Type	Event Reading Type	Entity ID	Description
1	EXT_FAN1-SPEED	04h	01h	17h	RPM of fan 1
2	EXT_FAN2-SPEED	04h	01h	17h	RPM of fan 2
3	EXT_FAN3-SPEED	04h	01h	17h	RPM of fan 3
4	EXT_FAN4-SPEED	04h	01h	17h	RPM of fan 4
5	EXT_FAN5-SPEED	04h	01h	17h	RPM of fan 4
6	EXT_FAN6-SPEED	04h	01h	17h	RPM of fan 4

Table 61: Fan sensor list (20 inches system)

The fan speed sensors do use revolutions per minute (RPM) as unit and below thresholds are specified.



Sensor Name	Nominal	LNR	LCR	LNC	UNC	UCR	UNR	HYST
FAN1-SPEED	7000	-	1200	-	-	-	-	100
FAN2-SPEED	7000	-	1200	-	-	-	-	100
FAN3-SPEED	7000	-	1200					100
FAN4-SPEED	7000	-	1200					100

Table 62: Fan speed sensor threshold list

Sensor Name	Nominal	LNR	LCR	LNC	UNC	UCR	UNR	HYST
EXT_FAN1-SPEED	7000	-	1200	-	-	-	-	100
EXT_FAN2-SPEED	7000	-	1200	-	-	-	-	100
EXT_FAN3-SPEED	7000	-	1200					100
EXT_FAN4-SPEED	7000	-	1200					100
EXT_FAN5-SPEED	7000	-	1200					100
EXT_FAN6-SPEED	7000	-	1200					100

Table 63: Fan speed sensor threshold list (20 inches system)

2.7.4 Fan Failure Handling

One LED per fan module is designed to indicate the state of the fan module.

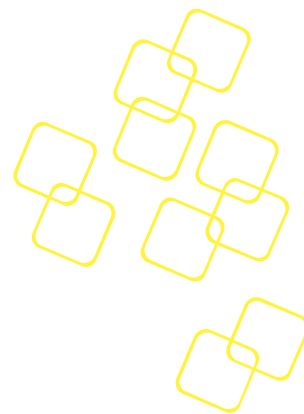


Figure 18: Fan LED location

Basic failure handling is defined to indicate single fan issues via the visible green fan module LED on the rear side of the chassis. In addition, events are logged in the system event log (SEL) via each fan speed sensor.

The fan modules green LED will be in “on” state when the two fans of each fan module are operating inside the defined ranges (, speed value higher than lower threshold.)

If a condition of low fan revolution speed is detected, the green LED is “blinking” (500ms on and 500ms off). Furthermore, a sensor event is generated in SEL, including the value of low fan speed (in RPM).



2.8 BIOS Synchronization

The SKY-8201's Advanced Platform Management synchronizes important information between the BMC and the system BIOS for platform wide consistency.

2.8.1 System Time

The BMC contains a real-time-clock to create time stamps for system events. However, the BMC's RTC does not implement a backup battery.

To make sure that the BMC timestamps are valid and are in sync with the x86 host's system time, the BIOS sets the BMC's time every time the system starts up. That yields consistent BMC and host OS log time stamps which helps to correlate events for troubleshooting purposes.

Events logged by the BMC before the time sync with BIOS will be flagged with "Pre-Init" as a timestamp.

2.8.2 FRU Info

The BMC store the System GUID and other FRU information (such as Serial Number, etc.) in the FRU EEPROM so BIOS can read the information by using standard IPMI command. The BIOS uses this information to dynamically populate the related DMI Info tables that can be queried by the host OS using DMI parsing tools like **dmidecode**.

2.9 BIOS POST Watchdog

The IPMI 2.0 compliant BMC watchdog is used to monitor OS during runtime or observe BIOS boot progress and initiate a rollback when the BIOS is found to be corrupted.

The BIOS watchdog timeout is predefined as 180 seconds (configurable by BIOS setting) and automatically starts when the payload power for the x86 subsystem is being turned on or when an x86 reset is detected. The time out action is set to "Hard Reset", with the timer use indicating "BIOS FRB2" use.

If the watchdog timer times out with this configuration two times, it triggers a BIOS chip failover followed by a system reset and a restart of the watchdog timer.

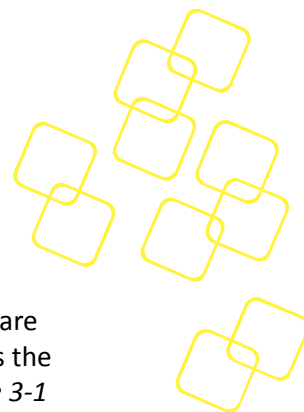
The BIOS does not touch the watchdog timer except for the following situations:

1. It disables the watchdog right before jumping into the x86 OS boot loader so it does not trigger after BIOS execution. It could alternatively reconfigure the watchdog to act as boot watchdog (i.e. change timeout action), based on BIOS configuration.
2. It temporarily disables the watchdog once the BIOS setup menu is entered, so the watchdog won't be triggered while the user is in the BIOS menu.

2.10 HPM.2

Advanced Platform Management implements the most important HPM.2 features applicable for the design to improve and extend the management part feature set.

The supported HPM.2 features and functionalities are described shortly in the following subsections. Please refer to the *HPM.2 specification, Table 3-1 Get HPM.x Capabilities command* for more information.



2.10.1 Get HPM.x Capabilities

The HPM.2 command '**Get HPM.x Capabilities**' indicates which HPM.2 functionalities are supported by the BMC. This command can be sent via any session-less interface and is the minimum requirement for HPM.2 compliance. Refer to the *HPM.2 specification, Table 3-1 'Get HPM.x Capabilities' command* for the details.

2.10.2 LAN Configuration Parameters

To support most of the new HPM.2 features and existing IPMI feature extensions by HPM.2, the standard IPMI LAN parameter commands '**Get/Set LAN Configuration Parameters**' are extended.

The new parameters are specified as OEM parameters (OEM range C0h - FFh) and reported via the previously mentioned '**Get HPM.x Capabilities**' HPM.2 IPMI command.

2.10.3 Long IPMI Messages

The maximum IPMI message size for LAN channels with HPM.2 extension is increased; compared to the default IPMI defined message sizes. The long message feature saves time for transferring a high amount of payload data significantly, for example during HPM.1 updates. Long IPMI Messages using HPM.2 over LAN are only available if RMCP+ (lanplus) is used.

2.11 VLAN Support

The BMC supports VLAN tagged traffic according to 802.1Q for all available network communication protocols and interfaces, including IOL, SOL, DHCP and ARP.

The VLAN configuration of all channels can be applied per channel independently. Having a VLAN configured for one channel does not require a VLAN configured on the other channel.

When a VLAN ID is configured into the LAN parameters, the BMC will only accept packets with that VLAN tag. All BMC-generated packets will include the given VLAN tag.

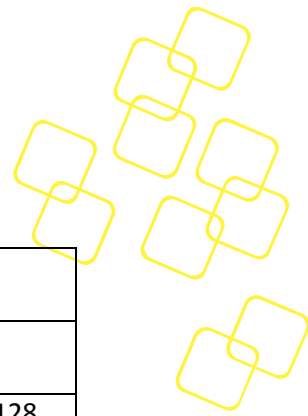
The VLAN ID (and VLAN Priority) for one IPMI channel can be configured along with other LAN settings via the available IPMI '**Set LAN Configuration Parameter**' command. See the *IPMIv2.0 specification, Section 13.7 and 23.1* for more details.

2.12 BMC Security

The IPMI '**Get Channel Cipher Suites**' command can be executed prior to establish a session with the BMC. This command is used to check what authentication, integrity, and confidentiality algorithms are supported. The algorithms are used in combination as **Cipher Suites**. This command only applies to the implementations that support IPMI v2.0/RMCP+ sessions.

Advanced Platform Management supports the following IPMIv2.0/RMCP+ cipher suites:

ID	Characteristics	Cipher Suite	Authentication Algorithm(s)	Integrity Algorithm(s)	Confidentiality Algorithm(s)
0*	"no password"	00h, 00h, 00h	RAKP-none	None	None



1	S	01h, 00h, 00h	RAKP-HMAC-SHA1	None	None
2	S, A	01h, 01h, 00h		HMAC-SHA1-96	None
3	S, A, E	01h, 01h, 01h			AES-CBC-128
6	S	02h, 00h, 00h	RAKP-HMAC-MD5	None	None
7	S, A	02h, 02h, 00h		HMAC-MD5-128	None
8	S, A, E	02h, 02h, 01h			AES-CBC-128
11	S, A	02h, 03h, 00h		MD5-128	None
12	S, A, E	02h, 03h, 01h			AES-CBC-128
15	S	03h, 00h, 00h	RAKP-HMAC-SHA256	None	None
16	S, A	03h, 04h, 00h		HMAC-SHA256-128	None
17	S, A, E	03h, 04h, 00h			AES-CBC-128

Table 64: Supported RMCP+ Cipher Suites

For security concern, Cipher Suite 0 is disabled. Per default Cipher Suite 17 is used.

The example below selects Cipher Suite 2 to establish an IOL connection:

```
#ipmitool -I lanplus <BMC IP> -U <User ID> -P <Password> mc info -C <Cipher Suite ID>
```

```
[root@CGS6010 ~]# ipmitool -I lanplus -U administrator -P advantech -H 172.17.7.222 mc info -C 2
Error sending request
Device ID           : 121
Device Revision     : 1
Firmware Revision   : 0.26
IPMI Version        : 2.0
Manufacturer ID     : 10297
Manufacturer Name    : Unknown (0x2839)
Product ID          : 12896 (0x3260)
Product Name        : Unknown (0x3260)
Device Available    : yes
Provides Device SDRs : yes
Additional Device Support :
  Sensor Device
  SEL Device
  FRU Inventory Device
  IPMB Event Generator
Aux Firmware Rev Info :
  0x00
  0x00
  0x00
  0x00
```

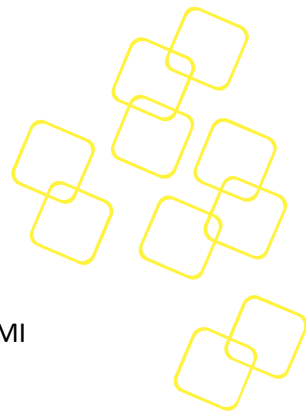


Figure 19: Establish an IOL Connection with Cipher Suite 2

Please note that establishing an IOL session with RMCP+ cipher suites also requires IPMI management tool's support (i.e. **ipmitool**).

2.13 Intrusion Detection

The SKY-8201 supports chassis intrusion detection by default. If the top cover of the chassis is removed, this gets detected even when the box is unpowered or unplugged, and the corresponding sensor (see *Section 2.6.5.10*) will report the event.

2.14 Platform Event Filtering & SNMP Traps

To monitor the systems health condition, asynchronous alert notifications through SNMP traps, triggered via IPMI Platform Event Filtering (PEF), are supported by BMC. This chapter does describe basic mechanism and implementation details.

2.14.1 Simple Network Management Protocol

The Internet standard Simple Network Management Protocol (SNMP) is used for collecting and organizing data about network connected devices. This data and information's can be modified to change a devices behaviour.

The BMC can be connected to networks and supports SNMP.

2.14.1.1 Management Information Base

A management information base (MIB) describes the system status and configuration. It's organized in the form of variables (in hierarchies) on the managed system using the SNMP obtained data. These variables can be queried and manipulated (in some circumstances) remotely by managing applications. MIB is usually an ASCII text file that describes SNMP network elements as a list of data objects.

The BMC does provide the MIB as part of the Advantech SAL SW implementation (see following chapter). The used MIB file is called "NCG_SAL-MIB.txt" [14] and stored within SAL. The MIB file content is an overall Advantech NCG SNMP definition and not project specific. Please refer to NCG_SAL-MIB.txt file [14] for details about defined data content.

2.14.1.1.1 SNMP versions

So far three main versions of SNMP are defined. The BMC implementation does support version SNMPv1, SNMPv2c and SNMPv3.

2.14.1.2 SNMP Community Strings

The SNMP private and public community strings are a security feature for SNMP GET/SET commands and for SNMP traps.

In a standard Linux environment the community strings can be changed by changing the SNMP configuration file. This option is not available for the BMC implementation as the user might not be able to directly access the BMC Linux shell.

For IPMI based PETs an IPMI mechanism is already defined to change the (public) community string. This mechanism is extended to use an IPMI OEM command to switch between public



and private community string to give the user the possibility to finally modify the private and public community strings using standard IPMI functions.

2.14.1.2.1 Change Community String

The IPMI specification defines standard functions to change the PET public community string. Please refer to IPMI specification, Table 23-4 “LAN Configuration Parameters” and “Community String” parameter 16.

This standard IPMI command is extended to change the community string for SNMP GET and SNMP SET. A flag will be used to indicate if the new string will be written as public or private community string, possible values are 0 and 1, the default value is 0.

Flag Value	Description
0	Change the community string for PET and SNMP GET
1	Change the community string for SNMP SET

Table 65: SNMP Community String Flag

The flag value can be read and changed using Advantech OEM IPMI commands.

After the community string is changed, a BMC reset is required in order to reload the new SNMP configuration.

2.14.1.2.2 Configure Community String

The community strings are saved in the sal.conf file. The keyword for the public community string is “rocommunity” and “rwcommunity” for the private community string. The default strings are “public” and “private”.

Example sal.conf with community string definition:

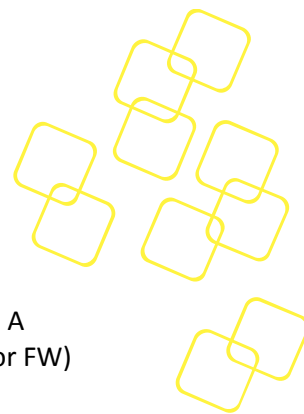
```
#####
# Access Control
#####

# sec.name source community
rocommunity public
rwcommunity private
```

Figure 20: SNMP Community Strings in Configuration File

2.14.1.3 SNMP Traps

SNMP traps are notifications send from an SNMP agent to a SNMP manager asynchronous. This allows notifying the management station about significant events happened via a SNMP message.



2.14.2 Platform Event Trap (PET)

The IPMI Platform Event Trap (PET) Format is defined in a separate IPMI specification. A platform event is defined as an event that is generated directly from a platform (HW or FW) independent of any other system parts (e.g. OS).

This BMC does support sending platform event traps compliant to the IPMI Platform Event Trap (PET) Format specification.

The Platform Event Trap format is used for sending a platform event in an SNMP Trap (see previous chapter).

The PET trap data fields are filled from the event message that generated the alert and from the PET LAN configuration parameters.

2.14.3 Platform Event Filtering (PEF)

Users can configure the BMC via IPMI defined Platform Event Filtering (PEF) to take selected actions (e.g. system power-off, reset or alert generation) on event messages that BMC receives or has internally generated. The given IPMI event is compared against BMC maintained event filter table entries. If the event does match a table entry, the BMC can perform specific action(s).

Event logging is a basic BMC feature and independent from event filtering, means event filtering can be configured without any impact on the event logging capabilities of the BMC.

2.14.3.1 PEF Actions

The BMC does implement a reduced PEF mechanism, focus is on mandatory alert generation action ("Send Alert", see IPMI Specification, Table 17-1 PEF Action Priorities). Furthermore, OEM actions are supported ("OEM"). Other optional PEF actions (power down, power cycle, reset etc.) are not supported. Please see following sub-chapter for details about the OEM defined PEF action.

2.14.3.1.1 OEM Action (System Status / Alert LED)

BMC provides a PEF OEM action (turning on/off critical LED and/or audible alarm) for onside alert notifications. With proper configuration of sensors in PEF event filter table, the user can manage critical LED to light as desired and beep alarm, too.

Each PEF entry will be processed after an event occurs with specific means on assertion or de-assertion. The OEM action is programmed BMC internally and turns on LED / beep if an assertion event occurs and turns off LED / beep if a de-assertion event occurs. Utilizing the critical LED and audible alarm should help users to create proper PEF entry configurations.

Note: Since the SKY-8201 system does only have one critical alert LED on front panel, the alert stage (LED on) and audible alarm (beep on) will remain until all OEM action PEF entries are de-asserted.

2.14.3.2 Alert Policies

An Alert Policy defines where a triggered alert (after a matched event) is directed (send) to. One or more alert destinations (with different media types or channels) can be specified. If more used, the destinations are processed in sequence.



The Alert Policy Table does include all possible alert media and destination sets. For more details, please see IPMI specification, chapters 17.1 and 17.11.

The BMC does specify a single, 3-byte long, Alert Policy, means the Alert Policy Table (as part of the PEF configuration parameters) does only have the Alert Policy entry number “1” (minimum requirement).

Below table defines the default BMC content for the Alert Policy including the alert channel and destination parameter details (in IPMITool notation: “pef policy -v”).

Description	Setting
Alert policy table entry	1
Policy set	1
Policy entry rule	Match-always
Channel number	1
Channel medium	802.3 LAN
Alert destination type	PET
PET Community	public
ACK timeout/retry (secs)	0
Retries	0
IP address	0.0.0.0
MAC address	00:00:00:00:00:00

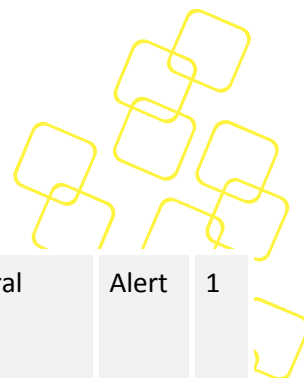
Table 66: Default PEF Alert Policy Table

2.14.3.3 Event Filter Table

The PEF Event filter table defines which kind of event does trigger a specific PEF action. It’s defined by IPMI specification to consist of a set of rows (entries). The single fields used by any row are defined in IPMI specification 2.0, Table 17-2.

One event table entry can match either a single or also multiple events received by BMC. 16 entries are provided by BMC as recommended by IPMI specification. See default configuration below (in IPMITool notation: “pef list -v”).

Entry	Status	Version	Sensor Type	Sensor Number	Event Severity	Event Class	Event Trigger(s)	Action	Policy set
1	active	0x11	Fan (0x04)	Any (0xFF)	Critical	Thres hold	(0x01/0x0004) < LC	Alert, OEM	1
2	active	0x11	Temp. (0x01)	Any (0xFF)	Critical	Thres hold	(0x01/0x0280) >UNC, >UC	Alert, OEM	1



3	inactive	0x11	Chassis Intrusion (0x05)	8	Critical	Discrete	(0x6F/0x0001) General Chassis Intrusion	Alert	1
4	inactive	0x11	Power Supply (0x08)	Any (0xFF)	Critical	Discrete	(0x6F/0x000B) Presence detected Power Supply Failure detected Power Supply input lost (AC/DC)	Alert, OEM	1
5	active	0x11	Voltage (0x02)	Any (0xFF)	Critical	Threshold	(0x01/0x0204), <LC, <UC Action	Alert, OEM	1
6	inactive	-	-	-	-	-	-	-	-
7	inactive	-	-	-	-	-	-	-	-
8	inactive	-	-	-	-	-	-	-	-
9	inactive	-	-	-	-	-	-	-	-
10	inactive	-	-	-	-	-	-	-	-
11	inactive	-	-	-	-	-	-	-	-
12	inactive	-	-	-	-	-	-	-	-
13	inactive	-	-	-	-	-	-	-	-
14	inactive	-	-	-	-	-	-	-	-
15	inactive	-	-	-	-	-	-	-	-
16	inactive	-	-	-	-	-	-	-	-

Table 67: Default PEF Event Filter Table

Some Platform Event Filter table entries are pre-configured by BMC to ensure SNMP alerts are created for common system failure events. Each BMC default PEF table entry is shortly described in next sub-chapters.

2.14.3.3.1 Fan Failure Alert Notification

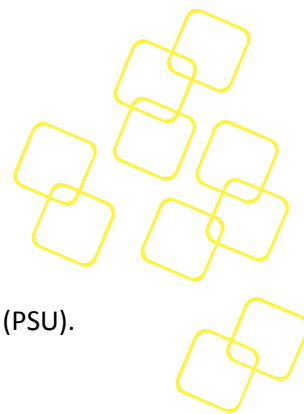
The BMC will generate an alert when any of the fan speeds fall below the lower critical threshold defined in the BMC sensor table / SDR.

2.14.3.3.2 CPU Temperature Alert Notification

If the CPU temperature either exceeds the upper non-critical or upper critical thresholds defined in the BMC sensor table / SDR, the BMC will generate an alert.

2.14.3.3.3 Chassis Intrusion Alert Notification

The BMC supports an alert notification while sensor "CASE_INTRUSION" detects intrusion occurred.



2.14.3.3.4 PSU Status Alert Notification

BMC supports alert notifications for status updates related to the Power Supply Units (PSU).

2.14.3.3.5 Voltage Alert Notification

If any BMC monitored voltage exceeds either the lower critical or upper critical thresholds defined in the BMC sensor table / SDR, the BMC will generate an alert.

2.14.4 BMC PEF Alert Generation

The BMC does use following mechanism to send out platform alerts, filtered via PEF Event Filter and Alert Policy Tables, to configured channel / destination:

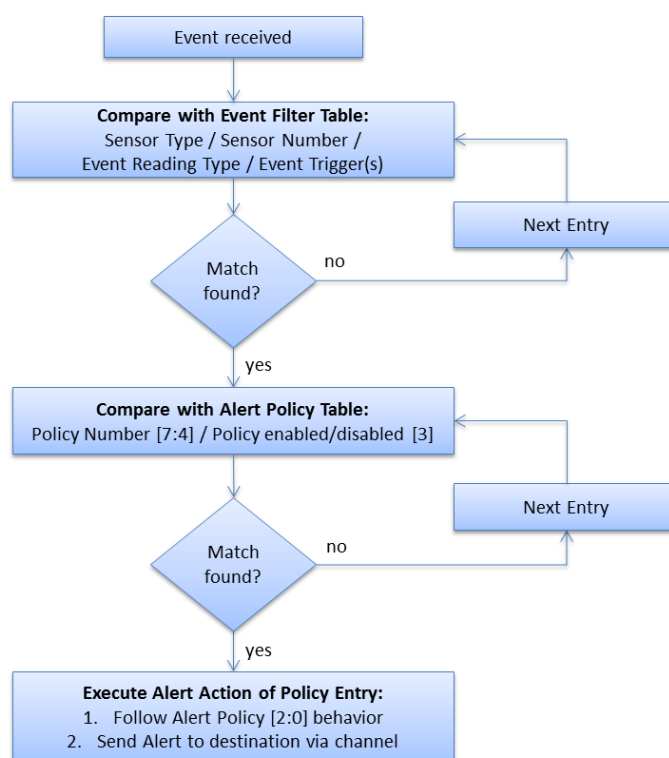
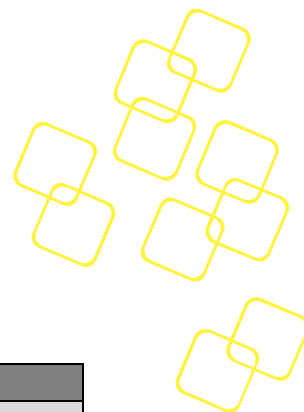


Figure 21: PEF Alert Generation Flow

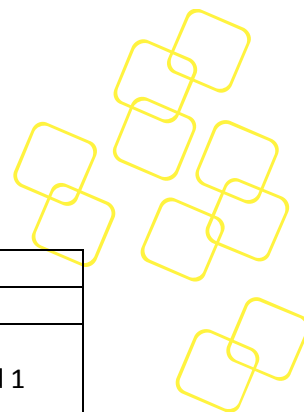
A received platform event is stored into SEL. This new event is processed through the PEF Event Filter Table entries and compared via sensor number, sensor type, event/reading type and the event trigger fields. If the event does match any filter entry, the filters Alert Policy Number is used to search for a matching (and enabled) policy entry in the Alert Policy table. If any Alert Policy entry does match, BMC knows the channel and destination to be used for the alert. Afterwards BMC can send out the alert to this destination via the found channel. The channel determines which set of destination addresses are used and other details like the alert type (these parameters are set via the LAN configuration parameters commands).



2.15 BMC Default Settings

2.15.1 User Account

Parameter Name	Default Value
User 1	
User ID (16 bytes)	NULL
User Password (16 bytes)	NULL
User Privilege Limit	2h = ADMINISTRATOR
Link Authentication	1h = Enable
IPMI Messaging	1h = Enable
User 2	
User ID (16 bytes)	callback
User Password (16 bytes)	advantech
User Privilege Limit	1h = CALLBACK
Link Authentication	1h = Enable
IPMI Messaging	1h = Enable
User Payload Access	Byte 1 : Standard Payload enables 1 2h = enable standard payload 1 (SOL) Byte 2 : OEM Payload enables 1 0h = disable OEM Payload 0-7
User 3	
User ID (16 bytes)	user
User Password (16 bytes)	advantech
User Privilege Limit	2h = USER
Link Authentication	1h = Enable
IPMI Messaging	1h = Enable
User Payload Access	Byte 1 : Standard Payload enables 1 2h = enable standard payload 1 (SOL) Byte 2 : OEM Payload enables 1 0h = disable OEM Payload 0-7
User 4	
User ID (16 bytes)	operator
User Password (16 bytes)	advantech
User Privilege Limit	3h = OPERATOR
Link Authentication	1h = Enable
IPMI Messaging	1h = Enable
User Payload Access	Byte 1 : Standard Payload enables 1: 2h = enable standard payload 1 (SOL) Byte 2 : OEM Payload enables 1: 0h = disable OEM Payload 0-7
User 5	
User ID (16 bytes)	administrator
User Password (16 bytes)	advantech
User Privilege Limit	4h = ADMINISTRATOR



Link Authentication	1h = Enable
IPMI Messaging	1h = Enable
User Payload Access	Byte 1 : Standard Payload enables 1: 2h = enable standard payload 1 (SOL) Byte 2 : OEM Payload enables 1: 0h = disable OEM Payload 0-7

Table 68: User Account Default Setting

2.15.2 PEF

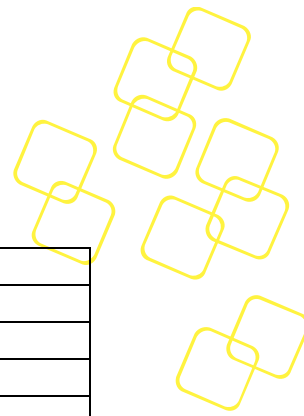
See chapter 2.14.3 for more detail.

Description	Setting
Alert policy table entry	1
Policy set	1
Policy entry rule	Match-always
Channel number	1
Channel medium	802.3 LAN
Alert destination type	PET
PET Community	public
ACK timeout/retry (secs)	0
Retries	0
IP address	0.0.0.0
MAC address	00:00:00:00:00:00

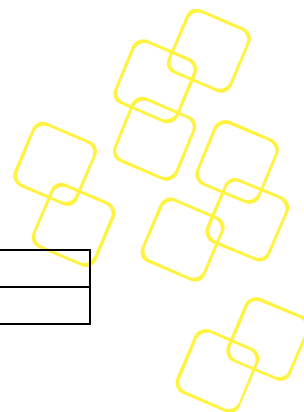
Table 69: Default PEF Alert Policy Table

2.15.3 LAN

Parameter Name	Default Value
Channel Access	
PEF Alerting	Enabled
Per-message Authentication	Disabled
User level Authentication	Disabled
Authentication Type Supported	MD5/PASSWORD (*Note: type None is disabled for security reasons)
Access Mode	Always available
Privilege Level Limit	ADMINISTRATOR
Authentication type enables	
Callback level	14h = MD5 + PASSWORD
User level	14h = MD5 + PASSWORD



Operator level	14h = MD5 + PASSWORD
Administrator level	14h = MD5 + PASSWORD
OEM level	00h
IP Address	
IP address	0.0.0.0
IP address source	1h – static address (manually configured)
MAC address	Programmed during manufacturing
Subnet mask	0.0.0.0
Default gateway	0.0.0.0
Default gateway MAC	0:0:0:0:0:0
Backup gateway	0.0.0.0
Backup gateway MAC	0:0:0:0:0:0
Community string	Public
Destination 1	
Alert acknowledge	0h – unacknowledged
Destination type	0h – PET Trap destination
Alert acknowledge timeout/retry interval (s)	0
Number of retries	0h – none
Address format	IPv4 IP followed by 802.3 MAC
Gateway selector	0h – use default gateway
Alerting IP address	0.0.0.0
Alerting MAC address	0:0:0:0:0:0
Destination 2	
Alert acknowledge	0h – unacknowledged
Destination type	0h – PET Trap destination
Alert acknowledge timeout/retry interval (s)	0
Number of retries	0h – none
Address format	IPv4 IP followed by 802.3 MAC
Gateway selector	0h – use default gateway
Alerting IP address	0.0.0.0
Alerting MAC address	0:0:0:0:0:0
Destination 3	
Alert acknowledge	0h – unacknowledged
Destination type	0h – PET Trap destination
Alert acknowledge timeout/retry interval (s)	0
Number of retries	0h – none
Address format	IPv4 IP followed by 802.3 MAC
Gateway selector	0h – use default gateway
Alerting IP address	0.0.0.0
Alerting MAC address	0:0:0:0:0:0
Destination 4	
Alert acknowledge	0h – unacknowledged
Destination type	0h – PET Trap destination
Alert acknowledge timeout/retry interval (s)	0
Number of retries	0h – none
Address format	IPv4 IP followed by 802.3 MAC
Gateway selector	0h – use default gateway



Alerting IP address	0.0.0.0
Alerting MAC address	0:0:0:0:0:0

Table 70: LAN Default Setting

2.15.4 SOL

Parameter Name	Default Value
SOL Enable	1h – Enable
SOL Authentication	82h – Force encryption + USER level
Character Accumulate Interval & Character Send Threshold	Byte 1 : Character Accumulate Interval in 5 ms increments: 04h = 20 ms 1Eh – 150 ms Byte 2 : Character Send Threshold: 20h = 32 characters DCh – 220 characters
SOL Retry	Byte 1 – Retry Count: 7h – 7 times Byte 2 – Retry Interval in 10 ms increments: 30h – 480 ms
SOL non-volatile bit rate	Ah – 115200 bps
SOL volatile bit rate	Ah – 115200 bps

Table 71: SOL Default Setting

2.15.5 Power Restore Policy

The Power Restore Policy determines how the system behaves when system power returns after a power loss.

The policy can be set as one of those options:

ID	Action	Meaning
0	Always-off	Always stay off
1	Previous	Restore to state that was in effect before power loss
2	Always-on	Always power on

Table 72: Set Power Restore Policy Command Actions

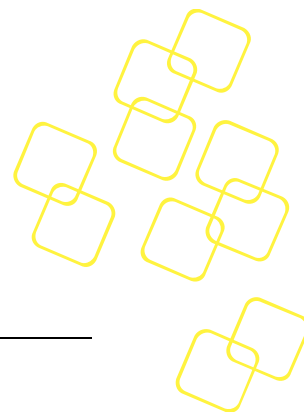
See Section 28.8 'Set Power Restore Policy command' in the IPMIv2.0 specification for more details in usage.

Parameter Name	Default Value
Power Restore Policy	10b = chassis always powers up after AC/mains returns



Table 73: Power Restore Policy

Please note that the power restore policy setting (options: **Power On**, **Power Off** and **Last State**) in the BIOS setup menu (under **Hardware** -> **PCH State after G3**) is in sync with the policy setting in the BMC. On each x86 system start up, the BIOS will update its setting by inquiring the BMC for current policy setting in the BMC. On the other hand, if the policy setting had been altered in the BIOS setup menu, BIOS will also notify BMC to update its policy setting in the BMC.



3. BMC FIRMWARE AND BIOS UPGRADE

3.1 Upgrade Platform Firmware

Advanced platform Management allows users to update the SKY-8201's firmware via the KCS or LAN interfaces using the HPM.1 protocol and related definitions.

Supported components include the BMC firmware itself as well as the system BIOS. For improved reliability, most updateable components support a backup image stored in a dedicated, redundant flash chip. The BMC will perform an automatic rollback in case of an upgrade failure to recover the unit to its previous known good state.

Please note that the functionality of the BMC will be degraded while upgrading the BMC firmware. Some functions including sensor listing, BMC information, etc., will not be available during that timeframe.

The BMC firmware of SKY-8201 uses physically redundant images. In case of a firmware malfunction or a corrupted update, the BMC bootloader will switch over to a backup image. The same rule applies to BIOS image.

3.1.1 Upgradeable Components

There are three components subject to firmware upgrades in the SKY-8201:

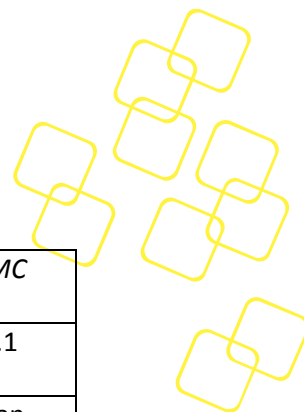
- Component 0: BMC boot loader
- Component 1: BMC firmware
- Component 2: FPGA
- Component 3: System BIOS
- Component 4: NVRAM

Please note that the BMC boot loader is a hardened and minimized firmware component which is not subject to field updates as it does not support fail safe updates via rollback capabilities. The upgrade capability of the boot loader is only intended for factory use. A failing update in the field may leave the unit non-functional and may result in the requirement for on-premises service access and/or RMA.



HPM.1 defines a mechanism and data structure that allow an update tool (**upgrade agent**) to identify the upgradeable components and related properties:

Property	Value	Description
Component 0 presence	Y	Component 0 is present
Component 1 presence	Y	Component 1 is present
Component 2 presence	Y	Component 2 is present
Component 3 presence	Y	Component 3 is present
Component 4 presence	Y	Component 4 is present
Upgrade undesirable	N	Upgrades are supported
Auto rollback override	N	No manual override of rollback performed by BMC
IPMC degraded	Y	BMC functionality will be degraded while performing an update
Deferred activation	Y	Upgraded image can be activated later
Services affected	Y	Service is affected during an upgrade



		<i>Note: This only applies to BIOS update. BMC updates should not affect any services.</i>
Manual rollback	Y	BMC supports a manual rollback via HPM.1 command.
Automatic rollback	Y	BMC supports automatic rollback in case an upgrade fails.
Self test	Y	BMC supports a built in self-test.
Upgrade timeout	360 sec	Timeout for performing an upgrade action. After timeout expires the upgrade agent will report an upgrade failure.
Self test timeout	25 sec	Timeout for a built in self-test after an update. After timeout expires the upgrade agent will report an upgrade failure.
Rollback timeout	120 sec	Timeout when BMC performs a rollback. After timeout expires the upgrade agent will report an upgrade failure.
Inaccessibility timeout	50 sec	Timeout for BMC not accessible. After timeout expires the upgrade agent will report an upgrade failure.

Table 74: HPM.1 Capability

3.1.1.1 Component 0: BMC Boot loader

This component represents the BMC bootloader. It does not support rollback and shall only be updated in factory or during system integration. Field upgrade is strongly discouraged.

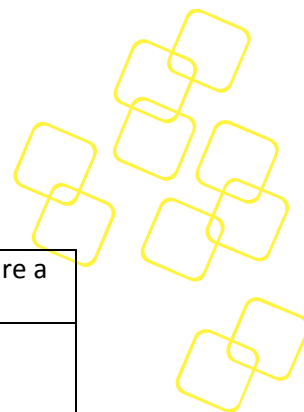
Property	Value	Description
Payload cold reset req.	N	Upgrading this component does not require a reset of the x86 host
Def. activation supported	Y	Upgrade will be activated automatically when image has been transferred
Comparison supported	N	Comparing the actual component versus the new one is not supported
Preparation supported	Y	Prepare Upgrade Command is supported
Rollback supported	N	No rollback is supported

Table 75: HPM.1 Component BMC Boot loader Property

3.1.1.2 Component 1: BMC firmware

This component represents the BMC application. Two redundant images are stored in physically distinct flash chips for a maximum of reliability and availability. Which image is the active image depends on the previous upgrade. For instance, if the current active image is the first SPI flash, then backup image is the second one. After upgrading and activating new BMC firmware, the active image will change to the second SPI flash.

Property	Value	Description
----------	-------	-------------



Payload cold reset req.	N	Upgrading this component does not require a reset of the x86 host
Def. activation supported	Y	Upgrade can be activated later using a deferred activation command
Comparison supported	N	Comparing the actual component versus the new one is not supported
Preparation supported	Y	Prepare Upgrade Command is supported
Rollback supported	Y	Rollback is supported

Table 76: HPM.1 Component BMC application Property

3.1.1.3 Component 2: FPGA

This component represents FPGA. The FPGA image will be uploaded (stored non-volatile) to a file (called deferred image) located on the external BMC configuration flash (SPI flash 2). After the upload, Afterwards, the new FPGA configuration will be written from the deferred image file to the FPGA internal flash.

Property	Value	Description
Payload cold reset req.	Y	Upgrading this component require a reset of the x86 host
Def. activation supported	Y	Upgrade can be activated later using a deferred activation command
Comparison supported	N	Comparing the actual component versus the new one is not supported
Preparation supported	Y	Prepare Upgrade Command is supported
Rollback supported	Y	Rollback is supported

Table 77: HPM.1 Component FPGA Property

3.1.1.4 Component 3: BIOS

This component represents the system BIOS. The active BIOS SPI flash is connected to PCH and backup BIOS SPI flash is connected to the BMC.

Property	Value	Description
Payload cold reset req.	Y	Upgrading this component requires a reset of the x86 host to take effect
Def. activation supported	Y	Upgrade can be activated later using a deferred activation command
Comparison supported	N	Comparing the actual component versus the new one is not supported
Preparation supported	Y	Prepare Upgrade Command is supported
Rollback supported	Y	Rollback is supported

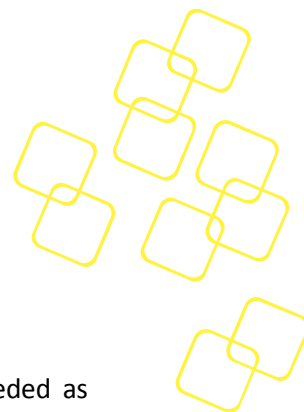


Table 78: HPM.1 Component BIOS Property

3.1.1.5 Component 4: NVRAM

This component represents the system NVRAM. NVRAM HPM.1 component is needed as part of x86 BIOS and the upgrade do have the same properties of BIOS (payload reboot or power cycle required). There is no rollback supported for the NVRAM component.

Property	Value	Description
Payload cold reset req.	Y	Upgrading this component requires a reset of the x86 host to take effect
Def. activation supported	Y	Upgrade can be activated later using a deferred activation command
Comparison supported	N	Comparing the actual component versus the new one is not supported
Preparation supported	Y	Prepare Upgrade Command is supported
Rollback supported	Y	Rollback is supported

Table 79: HPM.1 Component NVRAM Property

3.1.2 Check Active BMC Firmware Version

Either the 'ipmitool mc info' command or 'ipmitool hpm check' command can be used to retrieve the active BMC firmware version:

```
#ipmitool mc info
```

```
[root@CGS6010 ~]# ipmitool mc info
Device ID           : 110
Device Revision     : 1
Firmware Revision   : 0.06
IPMI version        : 2.0
Manufacturer ID     : 10297
Manufacturer Name   : Unknown (0x2839)
Product ID          : 24592 (0x6010)
Product Name        : Unknown (0x6010)
Device Available    : yes
Provides Device SDRs : no
Additional Device Support :
  Sensor Device
  SDR Repository Device
  SEL Device
  FRU Inventory Device
  IPMB Event Receiver
  IPMB Event Generator
  Chassis Device
Aux Firmware Rev Info :
  0x00
  0x00
  0x00
  0x00
```

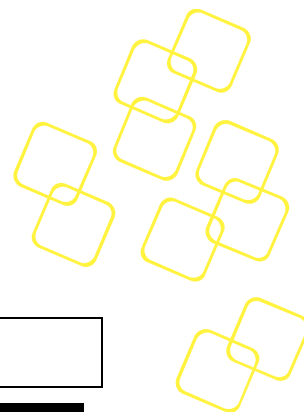


Figure 22: Check BMC Firmware Version

```
#ipmitool hpm check
```

```
[root@CGS6010 ~]# ipmitool hpm check
PICMG HPM.1 Upgrade Agent 1.0.9:
-----Target Information-----
Device Id       : 0x6e
Device Revision  : 0x1
Product Id      : 0x6010
Manufacturer Id  : 0x2839 (Unknown (0x2839))

-----
|ID | Name      | Active      | Versions      | Deferred      |
|   |           |             | Backup        |               |
|---|---|---|---|---|
| 0 | BOOT      | 1.16 00000000 | ---.-- --- | ---.-- --- |
| 1 | APP       | 0.06 00000000 | 0.04 00000000 | 0.00 00000000 |
|* 2 | BIOS      | 0.56 00000000 | 0.54 00000000 | 0.00 00000000 |
|---|---|---|---|---|
(*) Component requires Payload Cold Reset
```

Figure 23: Command 'ipmitool hpm check'

3.1.3 Upgrading BMC Firmware through KCS Interface

Please make sure that the ipmitool utility works normally (refer to *Section 1.2.1*) and the version of ipmitool is 1.8.17 or higher. Use the following command to perform a BMC firmware upgrade:

```
#ipmitool hpm upgrade <BMC image file> -z 255 activate
```

```
[root@CGS6010 cgs6010_bmc_00_08]# ipmitool hpm upgrade cgs6010_bmc_00_08.hpm -z 255 activate
Setting large buffer to 255
PICMG HPM.1 Upgrade Agent 1.0.9:
Validating firmware image integrity...OK
Performing preparation stage...
Services may be affected during upgrade. Do you wish to continue? (y/n): y
OK
Performing upgrade stage:
-----
|ID | Name      | Active      | Versions      | File          | % |
|   |           |             | Backup        |               |   |
|---|---|---|---|---|
| 0 | BOOT      | 1.16 00000000 | ---.-- --- | 1.16 00000000 | Skip |
| 1 | APP       | 0.06 00000000 | 0.04 00000000 | 0.08 00000000 | 100% |
|   | Upload Time: 04:42 | Image Size: 16515072 bytes |
|---|---|---|---|---|
Active version Upgrade version
(*) Component requires Payload Cold Reset
Performing activation stage:
Waiting firmware activation...OK
Firmware upgrade procedure successful
```

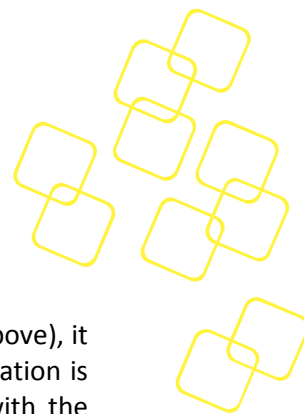


Figure 24: BMC Firmware Upgrade

When the message *'Firmware upgrade procedure successful'* shows (see the figure above), it will take about 40 to 60 seconds to activate the new BMC firmware. After the activation is done, you may check the active firmware version to make sure it is consistent with the upgraded BMC version.

```
#ipmitool hpm check
```

```
[root@CGS6010 cgs6010_bmc_00_08]# ipmitool hpm check

PICMG HPM.1 Upgrade Agent 1.0.9:

-----Target Information-----
Device Id       : 0x6e
Device Revision : 0x1
Product Id      : 0x6010
Manufacturer Id : 0x2839 (Unknown (0x2839))

-----
|ID| Name | Active | Versions | Deferred |
|  |      |        | Backup   |           |
-----
| 0|BOOT | 1.16 00000000 | --- -- | --- -- |
| 1|APP  | 0.08 00000000 | 0.06 00000000 | 0.00 00000000 |
|* 2|BIOS | 0.56 00000000 | 0.54 00000000 | 0.00 00000000 |
-----
(*) Component requires Payload Cold Reset
```

Figure 25: Check Active BMC Firmware Version

Please note that the functionality of the BMC will be degraded while upgrading the BMC firmware. Some functionality including sensor listing, BMC information, etc., will not be available at that time.



3.1.4 Upgrading BIOS through KCS Interface

Please make sure that the ipmitool utility works normally (refer to *Section 1.2.1*) and the version of ipmitool is 1.8.17 or higher. Use the following command to perform a BIOS firmware upgrade:

```
#ipmitool hpm upgrade <BIOS image file> -z 255 activate
```



```
[root@CGS6010 V058]# ipmitool hpm upgrade CGS6010_bios_standard_00.58.img -z 255 activate
Setting large buffer to 255

PICMG HPM.1 Upgrade Agent 1.0.9:

Validating firmware image integrity...OK
Performing preparation stage...
Services may be affected during upgrade. Do you wish to continue? (y/n): y
OK

Performing upgrade stage:

-----
|ID | Name      | Active      | Versions      | File      | % |
|----|-----|-----|-----|-----|-----|
|* 2|BIOS      | 0.56 00000000 | 0.54 00000000 | 0.58 00000000 |100%|
|   |Upload Time: 05:46 | Image Size: 16777216 bytes |
-----
Active version Upgrade version

(*) Component requires Payload Cold Reset
Performing activation stage:
Waiting firmware activation...OK

Firmware upgrade procedure successful
```

Figure 26: Upgrade BIOS with Ipmitool

BIOS upgrade requires a reset of the x86 host to take effect. Use the 'ipmitool chassis power cycle' command to reboot the system:

```
#ipmitool chassis power cycle
```

After the system reboot is completed, you may check if the active BIOS version is consistent with the upgraded BIOS version.

```
#ipmitool hpm check
```

```
[root@CGS6010 ~]# ipmitool hpm check

PICMG HPM.1 Upgrade Agent 1.0.9:

-----Target Information-----
Device Id       : 0x6e
Device Revision : 0x1
Product Id      : 0x6010
Manufacturer Id : 0x2839 (Unknown (0x2839))

-----
|ID | Name      | Active      | Versions      | Deferred  |
|----|-----|-----|-----|-----|
| 0|BOOT      | 1.16 00000000 | ---.-- --- | ---.-- --- |
| 1|APP       | 0.08 00000000 | 0.06 00000000 | 0.00 00000000 |
|* 2|BIOS      | 0.58 00000000 | 0.56 00000000 | 0.00 00000000 |
-----
(*) Component requires Payload Cold Reset
```

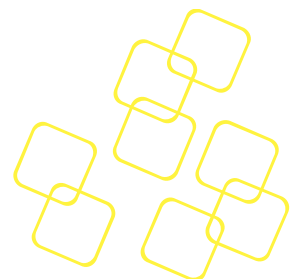



Figure 27: Check Active BIOS Version

Please note that the functionality of the BMC will be degraded while upgrading the BIOS firmware. Some functionality including sensor listing, BMC information, etc., will not be available at that time.



3.1.5 Upgrading BMC Bootloader through IOL

For most of new BMC firmware releases, the BMC bootloader shall remain at the same version and thus does not require bootloader updates. However, in some cases (e.g. bug fixing for the bootloader) bootloader updates is unavoidable. If users find an individual bootloader image file within a new BMC release package, please upgrade the BMC bootloader with the command provided below.

Please make sure that the ipmitool utility works normally (refer to *Section 1.2.1*) and the version of ipmitool is 1.8.17 or higher. Use the following command to perform a BMC bootloader upgrade:

```
#ipmitool -I lan -H <BMC IP> -U <User ID> -P <Password> hpm upgrade <Bootloader image file> -z 1024 force activate
```

```
[root@CentOS7 tmp]# ipmitool -I lan -H 172.17.10.89 -U administrator -P advantech hpm upgrade ./namb3260_bootloader_standard_0.26.img -z 1024 force activate
Error sending request
Setting large buffer to 1024

PICMG HPM.1 Upgrade Agent 1.0.9:

Validating firmware image integrity...OK
Performing preparation stage...
Services may be affected during upgrade. Do you wish to continue? (y/n): y
OK

Performing upgrade stage:

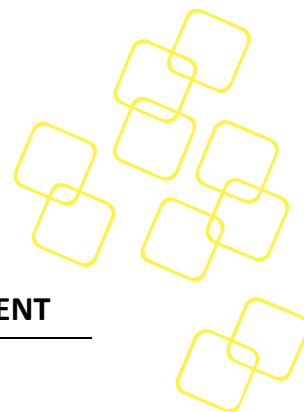
-----
| ID | Name | Active | Versions | File | % |
|----|-----|-----|-----|-----|---|
| 03260 BL | 1.00 | 0.12 | 0.02 | 100% |
| Upload Time: 02:41 | Active Version | Image Size: 154620 bytes | Upgrade Version |
-----

(*) Component requires Payload Cold Reset
Performing activation stage:
Waiting firmware activation...Error: Unable to establish LAN session
OK

Firmware upgrade procedure successful
```

Figure 28: BMC Bootloader Upgrade with Ipmitool

When the message ‘Performing activation stage:’ shows (see the figure above), it will take a while to activate the new BMC bootloader. After the activation is done, you may check the active BMC bootloader version to make sure it is consistent with the upgraded BMC bootloader version.



4. ESSENTIAL INFORMATION FOR ADVANCED PLATFORM MANAGEMENT

4.1 Identifying the System

The system identity can be discovered using the IPMI 'Get Device ID' and 'Get System GUID' commands.

The command 'Get Device ID' also returns a manufacturer ID indicating Advantech and a product ID indicating SKY-8201.

	Entry
Device ID	0x92
Manufacturer ID	0x2839
Product ID	0x8201
System GUID	bc86d520-b725-4813-9c97-e6c5800a4bc9

Table 80: System Identification

4.2 Lights Out Control

One of the basic lights out management functions is performing a system reset or power cycle from remote.

These functions are available through the IPMI chassis commands described in the *IPMIv2.0 specification, Section 28.3 'Chassis Control Commands'*.

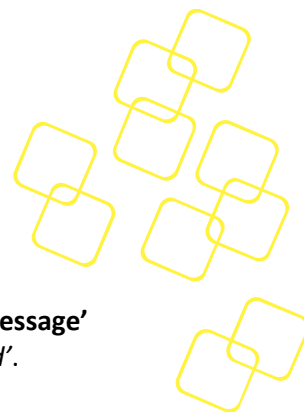
The BMC has full control over system power and reset which allows it to perform:

- **Power on**
Through direct power control & simulated power button press if needed
- **Power off**
Through direct power control
- **Power cycle**
Through direct power control
- **Reset**
Through direct reset signal control
- **Graceful shutdown**
Users can initiate graceful shutdown by pressing the **Power** button if an ACPI-compliant OS is used. If the system does not shut down within 60 seconds, the BMC will perform a hard power off.

You can use the 'ipmitool chassis power' command to control the system:

```
[root@svnserver ~]# ipmitool -I lanplus -H 172.17.10.183 -U CGS6010_admin -P %cgs6010_admin% chassis power
chassis power Commands: status, on, off, cycle, reset, diag, soft
[root@svnserver ~]# ipmitool -I lanplus -H 172.17.10.183 -U CGS6010_admin -P %cgs6010_admin% chassis power off
Chassis Power Control: Down/Off
[root@svnserver ~]# ipmitool -I lanplus -H 172.17.10.183 -U CGS6010_admin -P %cgs6010_admin% chassis power on
Chassis Power Control: Up/On
```

Figure 29: Chassis Power Command Usage



4.3 Creating System Events from an Application

System software can log events into the BMC's SEL using the '**Platform Event Message**' command per *IPMIv2.0 specification, Section 29.3 'Platform Event Message Command'*.

The command parameters are shown below:

Command Parameter	Description
Generator ID	7bit ID available for software use (ID)
EvMRev	Event Message Revision shall be 04h for Event Messages that comply with the format given in the <i>IPMI v2.0 specification</i>
Sensor Type	Sensor Type per <i>IPMIv2.0 specification, Table 42-3</i> . OEM Reserved Types starting at 0xD0 to 0xFF are available for customer use
Sensor #	Sensor Number representing the 'sensor' within the BMC that generated the Event Message
Event Dir	1-bit. Indicates the event transition direction. (0 = Assertion Event, 1 = Deassertion Event)
Event Type	7-bits. This field indicates the type of threshold crossing or state transition (trigger) that produced the event. This is encoded using the Event/Reading Type Code. See <i>Section 42, 'Sensor and Event Code Tables'</i> per the <i>IPMIv2.0 specification</i> .
Event Data(1:3)	Event Data

Table 81: Command Parameters of the 'Platform Event Message' Command

4.4 Keeping Time in Sync

As described in *Section 2.8.1 System Time*, the BIOS synchronizes the BMC's RTC with the main system time on each start up. However, as the SKY-8201 is barely shut down when operated in the field, the x86 host's system time and the BMC's time will drift apart due to tolerance of the respective RTCs. Over a long period of time that may lead to significant differences in the timestamps created by logging software on the x86 host and the BMC.

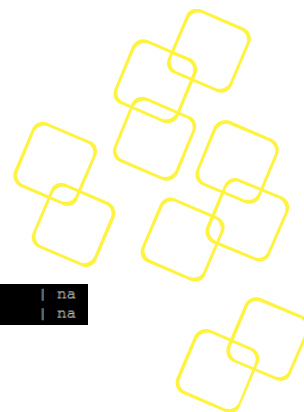
Keeping time stamps consistent for correlating x86 host and BMC logs for advanced troubleshooting is desirable. Advantech recommends to sync the BMC's time with the x86's host system time by sending the IPMI '**Set SEL Time**' command from the OS regularly.

Using this mechanism, it is also possible to sync the OS time to a NTP time server on regular intervals, and to pass this accurate time to the BMC using the '**Set SEL Time**' command.

4.5 Check PSU Presence

The 'ipmitool sensor' command can be used to check the presence of redundant PSU modules (**PSU1** and **PSU2**):

```
#ipmitool sensor
```



PSU1	0x0	discrete	0x0100	na	na	na	na	na	na
PSU2	0x0	discrete	0x0100	na	na	na	na	na	na

Figure 30: Check the Readings of PSUx Sensors

The ipmitool will use the IPMI 'Get Sensor Reading' command to read back each sensor's reading (if the readings are applicable), and provide the response data bytes 4 and 5 of the reading command (refer to *IPMIv2.0 specification, Section 35.14 'Get Sensor Reading Command'*). As shown in the figure above, the readings 00h and 00h are response data bytes 4 and 5 of the **PSUx** sensors under the circumstances two PSU modules are both present.

This can be interpreted as (refer to *Figure 31*):

- Byte 4 : 00h = 0000 0000
No any sensor state (sensor offset, 0 to 7) is asserted
- Byte 5 : 80h = 1000 0000
No any sensor state (sensor offset, 8 to 14) is asserted

(4)	<p><u>For threshold-based sensors</u></p> <p>Present threshold comparison status</p> <p>[7:6] - reserved. Returned as 1b. Ignore on read.</p> <p>[5] - 1b = at or above (\geq) upper non-recoverable threshold</p> <p>[4] - 1b = at or above (\geq) upper critical threshold</p> <p>[3] - 1b = at or above (\geq) upper non-critical threshold</p> <p>[2] - 1b = at or below (\leq) lower non-recoverable threshold</p> <p>[1] - 1b = at or below (\leq) lower critical threshold</p> <p>[0] - 1b = at or below (\leq) lower non-critical threshold</p> <p><u>For discrete reading sensors</u></p> <p>[7] - 1b = state 7 asserted</p> <p>[6] - 1b = state 6 asserted</p> <p>[5] - 1b = state 5 asserted</p> <p>[4] - 1b = state 4 asserted</p> <p>[3] - 1b = state 3 asserted</p> <p>[2] - 1b = state 2 asserted</p> <p>[1] - 1b = state 1 asserted</p> <p>[0] - 1b = state 0 asserted</p>
(5)	<p><u>For discrete reading sensors only. (Optional)</u></p> <p>(00h Otherwise)</p> <p>[7] - reserved. Returned as 1b. Ignore on read.</p> <p>[6] - 1b = state 14 asserted</p> <p>[5] - 1b = state 13 asserted</p> <p>[4] - 1b = state 12 asserted</p> <p>[3] - 1b = state 11 asserted</p> <p>[2] - 1b = state 10 asserted</p> <p>[1] - 1b = state 9 asserted</p> <p>[0] - 1b = state 8 asserted</p>

Figure 31: Response data byte 4 and 5 of the IPMI 'Get Sensor Reading' Command

However, if the PSU2 module is removed, the response bytes 4 and 5 of the PSU2 sensor **PSU2** will be changed from '01h' to '00h' (see the figure below). This time bit 0 of byte 4 had been altered and it shall be interpreted as:

- Byte 4 : 01h = 0000 0001
Sensor state (sensor offset) 0 is asserted.

PSU1	0x0	discrete	0x0100	na	na	na	na	na	na
PSU2	0x0	discrete	0x0000	na	na	na	na	na	na

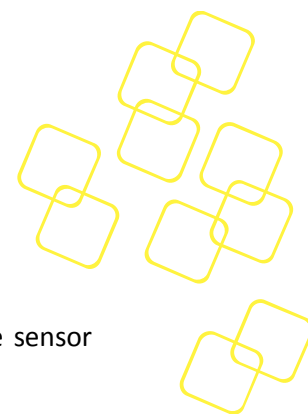


Figure 32: Assertion State indicates that PSU is absent

Refer to the description of **PSUx** sensor in *Section 2.6.5.8*, the reading '1b' for the sensor offset 0 ("**Presence Detected**" bit) means that the PSU is present.

Power Restore Policy

Meanwhile an event log will be added to the SEL to indicate the PSU was failure detected. Use the 'ipmitool sel elist' command to check the SEL:

```
#ipmitool sel elist
```

```
1c | 02/08/2018 | 10:16:23 | Power Supply PSU2 | Presence detected | Deasserted
1d | 02/08/2018 | 10:18:09 | Power Supply PSU2 | Presence detected | Asserted
```

Figure 33: PSU Removal Event in SEL

4.6 System Health Status

The SKY-8201 supports an array of status and alarm LEDs at the front panel. The locations and the functions of front LEDs are described in *Figure 35* and *Table*.



Figure 34: The locations of LEDs


Item	Label	Colour	Description
F15	PWR	Green	Power Alarm
F16	MNR	Amber	Minor Alarm
F17	MJR	Red	Major Alarm
F18	CRT	Red	Critical Alarm
F19	ID	Blue	Chassis Identification LED
F20	Status	Red	System Status LED
F21		Green	Power State LED

Table 82: Front LEDs Description

The SKY-8201 20" sku has three hot swappable fan modules at the front. Each of the fan modules carries two high performance fans for optimized air flow and there are up to six



fans supported on the SKY-8201. Each fan module has an integrated fan status LED for indicating its health status.

4.6.1 Power State LED (⏻)

The SKY-8201 system chassis does provide one green LED to indicate the system power state. The LED is visible on the front side of the SKY-8201 chassis and will be “on” once the system is powered on successful.

4.6.2 Chassis Identification LED (ID)

The SKY-8201 system chassis supports one blue LED (FRU LED ID 0) to allow users to identify the chassis. The LED is visible on the rear side of the chassis and can be controlled by the standard IPMI Chassis Identify command. Furthermore, BMC will turn on Chassis Identification LED if user does press the CHASSIS ID button on the IO board (detected via CPLD register).

The LEDs are ON or OFF with the following conditions:

1. The LEDs will be ON if the IPMI ‘**Chassis Identify**’ command (*IPMIv2.0 specification, Section 28.5*) is received, and they will be automatically OFF after 15 seconds (by default).
2. The LEDs will be ON if ‘Chassis ID’ button on the front panel is pressed. Press the button again to turn off the LEDs.

The LED ID of the Chassis Identify LED for the ‘**Get/Set FRU LED State**’ commands is 0x00.

4.6.3 Critical Alarm LED (CRT)

The Critical Alarm LED is ON when the BMC detects any critical system failure event. A critical system failure has a significant impact to the system, it means that the system can either not continue to operate or it is operating under a non-redundant power or cooling condition, resulting in increased risk for system failure.

The SKY-8201 system red critical LED (FRU LED ID 0x01) and the audible alarm will turn on while a critical fault has been detected by the BMC. The fault condition is defined following:

- If any BMC sensor does pass a critical threshold
- Or a critical discrete sensors is asserted (e.g. processor thermal trip, PSU failure)

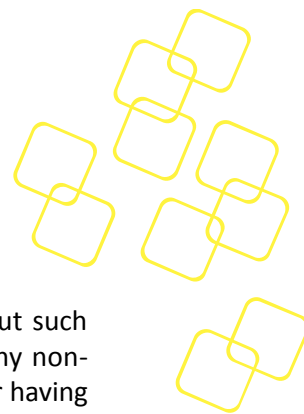
The critical LED can be changed to overwrite mode and turned on or off by the PICMG Set FRU LED State IPMI command. The LED state is not changed by BMC in any case if set to overwrite mode.

The LED ID of the Critical Alarm LED for the ‘**Get/Set FRU LED State**’ commands is 0x01

4.6.4 Major Alarm LED (System Status LED) (MJR)

One system status LED in red colour (FRU LED ID 0x04) is provided for the SKY-8201 system chassis. The system status LED behaviour is based on Platform Event Filtering (PEF) alert rules, which can be configured by user. The default configuration is to light up if any critical sensor threshold or critical discrete events (fan failure, power supply failure) occurs. These events will trigger the LED to light up. And the LED will turn off automatically when all critical events are absent.

The LED ID of the Major Alarm LED for the ‘**Get/Set FRU LED State**’ commands is 0x04



4.6.5 Minor Alarm LED (MNR)

The Minor Alarm LED is ON when the BMC detects a minor system failure event but such event does not affect the system's normal operation. Such events are defined as any non-critical BMC sensor event (such as SEL_FULL) being asserted or **Case_Intrusion** sensor having triggered.

The MNR LED (FRU LED ID 0x05) can be changed to overwrite mode and turned on or off by the PICMG Set FRU LED State command. In this mode, BMC does not change the LED in any condition.

The LED ID of the Minor Alarm LED for the '**Get/Set FRU LED State**' commands is 0x05

4.6.6 System Status LED

The SKY-8201 20 inches system chassis does provide one red LED (FRU LED ID 0x06) to indicate system status events. System status LED will be turned on based on Platform Event Filtering rules, which can be configured by user. The default configuration is that fan/voltage/temperature critical sensor threshold will trigger this LED. It will be cleared automatically when all critical events are absent.

System status LED can be changed to overwrite mode and turned on or off by the PICMG [3] Set FRU LED State command. In this mode, BMC does not change the LED in any condition.

4.6.7 FAN Status LED

There are totally up to 6 fans which are grouped into 3 fan modules inserted into the front side of the system chassis of SKY-8201 20" sku. Each fan module has an integrated FAN status LED for indicating its health status. (For SKY-8201L 27.5" sku, there are total up to 4 fans which are grouped into 4 fan modules be installed internally, without FAN status LED indicators.)

Basic failure handling is defined to indicate single fan issues via the visible green fan module LED on the rear side of the chassis. In addition, events are logged in the system event log (SEL) via each fan speed sensor.

The fan modules green LED will be in "on" state when the two fans of each fan module are operating inside the defined ranges (speed value higher than lower threshold.)

If a condition of low fan revolution speed is detected, the green LED is "blinking" (500ms on and 500ms off). Furthermore, a sensor event is generated in SEL, including the value of low fan speed (in RPM).

4.6.8 LED Panel for SKY-8201L

The SKY-8201L supports an array of status and alarm LEDs at the front & rear panel. The locations and the functions of front LEDs are described in *Figure 35* and *Table*.

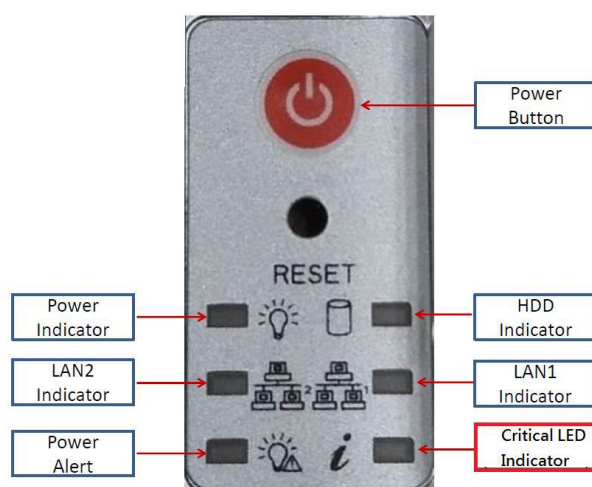
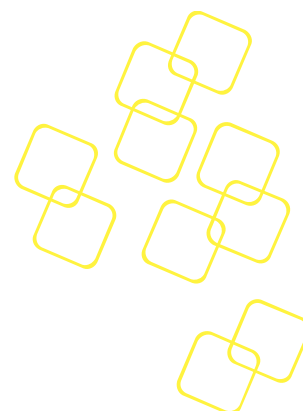


Figure 35: The locations of front LEDs





Label	Color	Description
<i>i</i>	Blue/Red	Critical LED
	Green	NIC#2 (MGMT2 RJ45) Link/Act LED
	Green	NIC#1 (MGMT1 RJ45) Link/Act LED
	Amber	HDD Act LED
	Green	Power State LED

Table 79: Front LEDs Description

4.6.8.1 Critical LED

The SYK-8201 system red critical LED (FRU LED ID 1) will turn on while a critical fault has been detected by the BMC. The fault condition is defined following:

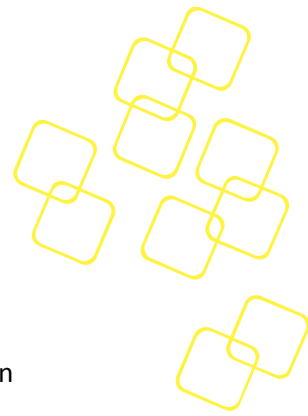
- If any BMC sensor does pass a critical threshold

Or a critical discrete sensor is asserted (e.g. processor thermal trip, PSU failure)

The critical LED can be changed to overwrite mode and turned on or off by the PICMG [3] Set FRU LED State IPMI command. The LED state is not changed by BMC in any case if set to overwrite mode.

4.6.8.2 Power Alert LED

Power alert LED in red color (FRU LED ID 2) is provided for the SKY-8201 system chassis. The power alert LED behavior is based on PSU sensors related alert event. The default configuration is to light up if any PSU sensor threshold events (e.g. voltage, current, watt, fan etc.) occurs. These error events will trigger the LED to light up. And the LED will turn off automatically when all error events are absent.



Two modes are available for the alert LED configuration:

1. Local control mode:

System critical event(s) occurred. Configurable options are provided in PEF OEM action entries.

2. Override mode:

User can control the LED manually by PICMG [3] defined Set FRU LED State IPMI command.

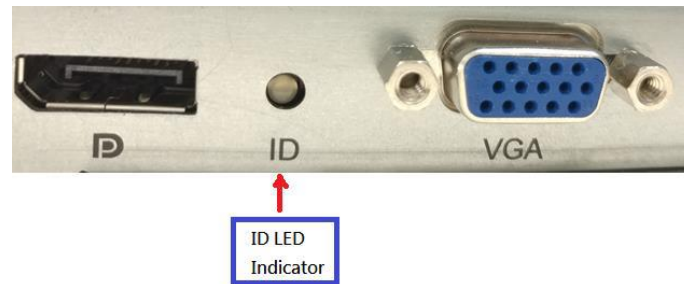


Figure 36: The locations of rear LEDs

Label	Color	Description
ID	Blue	Chassis Identification LED

Table 80: Rear LEDs Description

4.6.8.3 Chassis Identification LED

The SKY-8201L system chassis supports one blue LED (FRU LED ID 0) to allow users to identify the chassis. The LED is visible on the rear side of the chassis and can be controlled by the standard IPMI Chassis Identify command

4.6.9 Audible Alarm

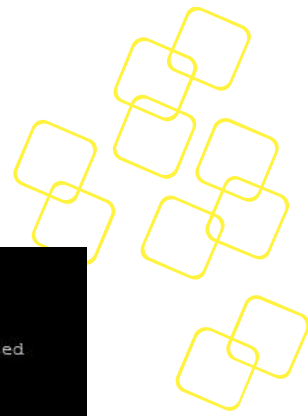
The BMC does support an audible alarm via HWM and a populated speaker. The audible alarm (beep) is triggered together with a specified SKY-8201 system LED and occurring critical event(s).

The SKY-8201 does trigger the audible alarm together with the critical LED (see Chapter 4.6.3).

4.7 Reading the SEL

The 'ipmitool sel list' command or 'ipmitool sel elist' command can be used to dump the SEL:

```
#ipmitool sel list
```

```
[root@CGS6010 ~]# ipmitool sel list
1 | Pre-Init | 0000000044 | Power Unit #0x36 | Power off/down | Asserted
2 | Pre-Init | 0000000045 | Power Supply #0x38 | Presence detected | Asserted
3 | Pre-Init | 0000000049 | Power Unit #0x36 | Power off/down | Deasserted
4 | Pre-Init | 0000000058 | Physical Security #0x03 | General Chassis intrusion | Asserted
5 | Pre-Init | 0000000061 | System Event | Timestamp Clock Sync | Asserted
6 | Pre-Init | 0000000061 | System Event #0x39 | Timestamp Clock Sync | Asserted
7 | 01/01/2009 | 00:07:59 | System Event #0x39 | Timestamp Clock Sync | Asserted
8 | 01/01/2009 | 00:08:01 | System Event | Timestamp Clock Sync | Asserted
9 | 01/01/2009 | 00:08:01 | Voltage #0x34 | Lower Critical going low | Asserted
a | 01/01/2009 | 00:12:31 | Power Unit #0x36 | Power off/down | Asserted
b | 01/01/2009 | 00:12:31 | Power Supply #0x37 | Presence detected | Asserted
c | Pre-Init | 0000000021 | Power Supply #0x38 | Presence detected | Asserted
```

Figure 37: Use 'ipmitool sel list' Command to Dump the SEL

```
#ipmitool sel elist
```

```
[root@CGS6010 cgs6010_bmc_00_08]# ipmitool sel elist
1 | Pre-Init | 0000000044 | Power Unit #0x36 | Power off/down | Asserted
2 | Pre-Init | 0000000045 | Power Supply PSU_1-Status | Presence detected | Asserted
3 | Pre-Init | 0000000049 | Power Unit #0x36 | Power off/down | Deasserted
4 | Pre-Init | 0000000058 | Physical Security Case Intrusion | General Chassis intrusion | Asserted
5 | Pre-Init | 0000000061 | System Event | Timestamp Clock Sync | Asserted
6 | Pre-Init | 0000000061 | System Event #0x39 | Timestamp Clock Sync | Asserted
7 | 01/01/2009 | 00:07:59 | System Event #0x39 | Timestamp Clock Sync | Asserted
8 | 01/01/2009 | 00:08:01 | System Event | Timestamp Clock Sync | Asserted
9 | 01/01/2009 | 00:08:01 | Voltage PSU_12V-VOL | Lower Critical going low | Asserted | Reading 0 < Threshold 11.40 Volts
a | 01/01/2009 | 00:12:31 | Power Unit #0x36 | Power off/down | Asserted
b | 01/01/2009 | 00:12:31 | Power Supply #0x37 | Presence detected | Asserted
c | Pre-Init | 0000000021 | Power Supply PSU_1-Status | Presence detected | Asserted
d | Pre-Init | 0000000033 | System Event | Timestamp Clock Sync | Asserted
e | Pre-Init | 0000000033 | System Event #0x39 | Timestamp Clock Sync | Asserted
f | 01/01/2009 | 00:13:28 | System Event #0x39 | Timestamp Clock Sync | Asserted
```

Figure 38: Use 'ipmitool sel elist' Command to Dump the SEL

The 'ipmitool sel save <file name>' command can be used to store the SEL to a file:

```
#ipmitool sel save <file name>
```

```
[root@CGS6010 ~]# ipmitool sel save sel_log
1 | Pre-Init | 0000000044 | Power Unit #0x36 | Power off/down | Asserted
2 | Pre-Init | 0000000045 | Power Supply #0x38 | Presence detected | Asserted
3 | Pre-Init | 0000000049 | Power Unit #0x36 | Power off/down | Deasserted
4 | Pre-Init | 0000000058 | Physical Security #0x03 | General Chassis intrusion | Asserted
5 | Pre-Init | 0000000061 | System Event | Timestamp Clock Sync | Asserted
6 | Pre-Init | 0000000061 | System Event #0x39 | Timestamp Clock Sync | Asserted
7 | 01/01/2009 | 00:07:59 | System Event #0x39 | Timestamp Clock Sync | Asserted
8 | 01/01/2009 | 00:08:01 | System Event | Timestamp Clock Sync | Asserted
9 | 01/01/2009 | 00:08:01 | Voltage #0x34 | Lower Critical going low | Asserted
a | 01/01/2009 | 00:12:31 | Power Unit #0x36 | Power off/down | Asserted
b | 01/01/2009 | 00:12:31 | Power Supply #0x37 | Presence detected | Asserted
```

Figure 39: Use 'ipmitool sel save' Command to Store the SEL to a File

After that, the file contents can be dumped to view the SEL entries: `#cat <file name>`


```
[root@CGS6010 ~]# cat sel_log
0x04 0x09 0x36 0x6f 0x00 0xff 0xff # Power Unit #0x36 Power off/down
0x04 0x08 0x38 0x6f 0x00 0xff 0xff # Power Supply #0x38 Presence detected
0x04 0x09 0x36 0x6f 0x00 0xff 0xff # Power Unit #0x36 Power off/down
0x04 0x05 0x03 0x6f 0x00 0xff 0xff # Physical Security #0x03 General Chassis intrusion
0x04 0x12 0x00 0x6f 0x05 0x00 0xff # System Event #0x00 Timestamp Clock Sync
0x04 0x12 0x39 0x6f 0x05 0x00 0xff # System Event #0x39 Timestamp Clock Sync
0x04 0x12 0x39 0x6f 0x05 0x80 0xff # System Event #0x39 Timestamp Clock Sync
0x04 0x12 0x00 0x6f 0x05 0x80 0xff # System Event #0x00 Timestamp Clock Sync
0x04 0x02 0x34 0x01 0x52 0x00 0xbe # Voltage #0x34 Lower Critical going low
0x04 0x09 0x36 0x6f 0x00 0xff 0xff # Power Unit #0x36 Power off/down
0x04 0x08 0x37 0x6f 0x00 0xff 0xff # Power Supply #0x37 Presence detected
0x04 0x08 0x38 0x6f 0x00 0xff 0xff # Power Supply #0x38 Presence detected
0x04 0x12 0x00 0x6f 0x05 0x00 0xff # System Event #0x00 Timestamp Clock Sync
0x04 0x12 0x39 0x6f 0x05 0x00 0xff # System Event #0x39 Timestamp Clock Sync
0x04 0x12 0x39 0x6f 0x05 0x80 0xff # System Event #0x39 Timestamp Clock Sync
```

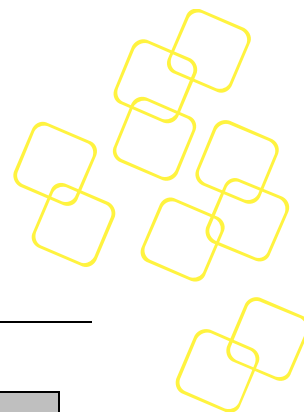
Figure 40: Check Saved SEL file

The 'ipmitool sel clear' command can be used to clear the SEL in the BMC:

```
#ipmitool sel clear
```

```
[root@CGS6010 ~]# ipmitool sel clear
Clearing SEL. Please allow a few seconds to erase.
```

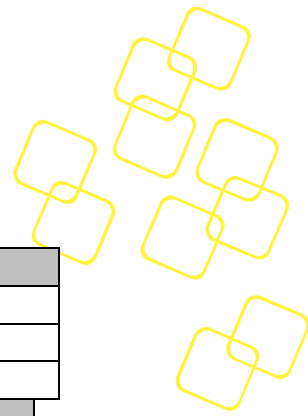
Figure 41: Use 'ipmitool sel clear' Command to Clear SEL



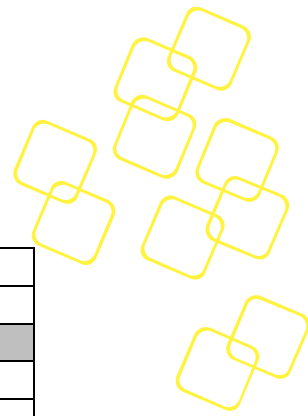
A. APPENDIX: SUPPORTED IPMI COMMANDS

The following standard IPMI commands are supported:

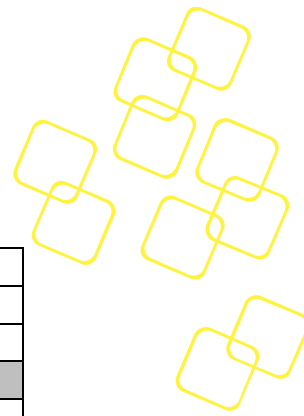
IPMI Device Global Commands	NetFn	Cmd	IPMI Section
Get Device Id	App	0x01	20.1
Cold Reset	App	0x02	20.2
Warm Reset	App	0x03	20.3
Get Self Test Results	App	0x04	20.4
Set ACPI Power State	App	0x06	20.6
Get Device GUID	App	0x08	20.8
BMC Device and Messaging Commands	NetFn	Cmd	IPMI Section
Set BMC Global Enables	App	0x2E	22.1
Get BMC Global Enables	App	0x2F	22.2
Clear Message Flags	App	0x30	22.3
Get Message Flags	App	0x31	22.4
Enable Message Channel Receive	App	0x32	22.5
Get Message	App	0x33	22.6
Send Message	App	0x34	22.7
Read Event Message Buffer	App	0x35	22.8
Get System GUID	App	0x37	22.14
Get Channel Authentication Capabilities	App	0x38	22.13
Get Session Challenge	App	0x39	22.15
Activate Session	App	0x3A	22.17
Set Session Privilege Level	App	0x3B	22.18
Close Session	App	0x3C	22.19
Get Session Information	App	0x3D	22.20
Get AuthCode	App	0x3F	22.21
Set Channel Access	App	0x40	22.22
Get Channel Access	App	0x41	22.23
Get Channel Info	App	0x42	22.24
Set User Access	App	0x43	22.25
Get User Access	App	0x44	22.27
Set User Name	App	0x45	22.28
Get User Name	App	0x46	22.29
Set User Password	App	0x47	22.30
Activate Payload	App	0x48	24.1
Deactivate Payload	App	0x49	24.2
Set User Payload Access	App	0x4C	24.6
Get User Payload Access	App	0x4D	24.7
Get Channel Payload Support	App	0x4E	24.8
Get Channel Payload Version	App	0x4F	24.9
Master Write-Read	App	0x52	22.11
Get Channel Cipher Suites	App	0x54	22.15
Set Channel Security Keys	App	0x56	22.25



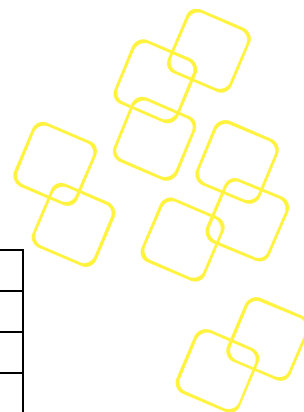
BMC Watchdog Timer Commands	NetFn	Cmd	IPMI Section
Reset Watchdog Timer	App	0x22	27.5
Set Watchdog Timer	App	0x24	27.6
Get Watchdog Timer	App	0x25	27.7
Chassis Device Commands	NetFn	Cmd	IPMI Section
Get Chassis Capabilities	Chassis	0x00	28.1
Get Chassis Status	Chassis	0x01	28.2
Chassis Control	Chassis	0x02	28.3
Chassis Identify	Chassis	0x04	28.5
Set Chassis Capabilities	Chassis	0x05	28.7
Set Power Restore Policy	Chassis	0x06	28.8
Get System Restart Cause	Chassis	0x07	28.11
Set System Boot Options	Chassis	0x08	28.12
Get System Boot Options	Chassis	0x09	28.13
Set Front Panel Button Enables	Chassis	0x0A	28.6
Set Power Cycle Interval	Chassis	0x0B	28.9
Event Commands	NetFn	Cmd	IPMI Section
Set Event Receiver	S/E	0x00	29.1
Get Event Receiver	S/E	0x01	29.2
Platform Event	S/E	0x02	29.3
PEF and Alerting Commands	NetFn	Cmd	IPMI Section
Get PEF Capabilities	S/E	0x10	30.1
Arm PEF Postpone Timer	S/E	0x11	30.2
Set PEF Configuration Parameters	S/E	0x12	30.3
Get PEF Configuration Parameters	S/E	0x13	30.4
Set Last Processed Event ID	S/E	0x14	30.5
Get Last Processed Event ID	S/E	0x15	30.6
Alert Immediate	S/E	0x16	30.7
PET acknowledge	S/E	0x17	30.8
SEL Device Commands	NetFn	Cmd	IPMI Section
Get SEL Info	Storage	0x40	31.2
Get SEL Allocation Info	Storage	0x41	31.3
Reserve SEL	Storage	0x42	31.4
Get SEL Entry	Storage	0x43	31.5
Add SEL Entry	Storage	0x44	31.6
Clear SEL	Storage	0x47	31.9
Get SEL Time	Storage	0x48	31.10
Set SEL Time	Storage	0x49	31.11
Get SEL Time UTC Offset	Storage	0x5C	31.11a
SDR Device Commands	NetFn	Cmd	IPMI Section
Get SDR Repository Info	Storage	0x20	33.9
Get SDR Repository Allocation Info	Storage	0x21	33.10
Reserve SDR Repository	Storage	0x22	33.11
Get SDR	Storage	0x23	33.12
Clear SDR Repository	Storage	0x27	33.16
Get SDR Repository Time	Storage	0x28	33.17



Set SDR Repository Time	Storage	0x29	33.18
Run Initialization Agent	Storage	0x2C	33.21
FRU Device Commands	NetFn	Cmd	IPMI Section
Get FRU Inventory Area Info	Storage	0x10	34.1
Read FRU Data	Storage	0x11	34.2
Write FRU Data	Storage	0x12	34.3
Sensor Device Commands	NetFn	Cmd	IPMI Section
Get Device SDR Info	S/E	0x20	35.2
Get Device SDR	S/E	0x21	35.3
Reserve Device SDR Repository	S/E	0x22	35.4
Get Sensor Reading Factors	S/E	0x23	35.5
Set Sensor Hysteresis	S/E	0x24	35.6
Get Sensor Hysteresis	S/E	0x25	35.7
Set Sensor Threshold	S/E	0x26	35.8
Get Sensor Threshold	S/E	0x27	35.9
Set Sensor Event Enable	S/E	0x28	35.10
Get Sensor Event Enable	S/E	0x29	35.11
Get Sensor Event Status	S/E	0x2B	35.13
Get Sensor Reading	S/E	0x2D	35.14
Get Sensor Type	S/E	0x2F	35.16
FRU Device Commands	NetFn	Cmd	IPMI Section
Get FRU Inventory Area Info	Storage	0x10	34.1
Read FRU Inventory Data	Storage	0x11	34.2
Write FRU Inventory Data	Storage	0x12	34.3
Sensor Device Commands	NetFn	Cmd	IPMI Section
Get Device SDR Info	S/E	0x20	35.2
Get Device SDR	S/E	0x21	35.3
Reserve Device SDR Repository	S/E	0x22	35.4
Set Sensor Hysteresis	S/E	0x24	35.6
Get Sensor Hysteresis	S/E	0x25	35.7
Set Sensor Threshold	S/E	0x26	35.8
Get Sensor Threshold	S/E	0x27	35.9
Set Sensor Event Enable	S/E	0x28	35.10
Get Sensor Event Enable	S/E	0x29	35.11
Re-arm Sensor Events	S/E	0x2A	35.12
Get Sensor Event Status	S/E	0x2B	35.13
Get Sensor Reading	S/E	0x2D	35.14
LAN Device Commands	NetFn	Cmd	IPMI Section
Set LAN Configuration Parameters	Transport	0x01	23.1
Get LAN Configuration Parameters	Transport	0x02	23.2
Serial/Modem Device Commands	NetFn	Cmd	IPMI Section
Set Serial/Modem Configuration Parameters	Transport	0x10	25.1
Get Serial/Modem Configuration Parameters	Transport	0x11	25.2
Set Serial/Modem Mux	Transport	0x12	25.3

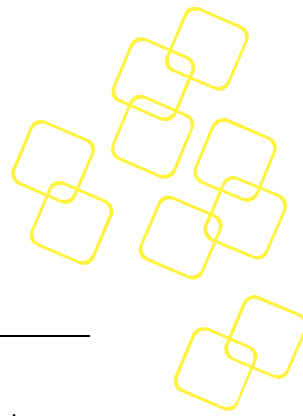


SOL Activating	Transport	0x20	26.1
Set SOL Configuration Parameters	Transport	0x21	26.2
Get SOL Configuration Parameters	Transport	0x22	26.3
PICMG Commands	NetFn	Cmd	PICMG 3.0 Table
Get PICMG Properties	PICMG	0x00	3-11
FRU Control	PICMG	0x04	3-27
Get FRU LED Properties	PICMG	0x05	3-29
Get LED Color Capabilities	PICMG	0x06	3-30
Set FRU LED State	PICMG	0x07	3-31
Get FRU LED State	PICMG	0x08	3-32
Get Fan Speed Properties	PICMG	0x14	3-86
Set Fan Level	PICMG	0x15	3-88
Get Fan Level	PICMG	0x16	3-87
Set Fan Policy	PICMG	0x1C	3-89
Get Fan Policy	PICMG	0x1D	3-90
Firmware Upgrade Commands	NetFn	Cmd	HPM.1 Table
Get target upgrade capabilities	PICMG	0x2E	3-3
Get component properties	PICMG	0x2F	3-5
Abort Firmware Upgrade	PICMG	0x30	3-15
Initiate upgrade action	PICMG	0x31	3-8
Upload firmware block	PICMG	0x32	3-9
Finish firmware upload	PICMG	0x33	3-10
Get upgrade status	PICMG	0x34	3-2
Activate firmware	PICMG	0x35	3-11
Query self-test results	PICMG	0x36	3-12
Query Rollback status	PICMG	0x37	3-13
Initiate Manual Rollback	PICMG	0x38	3-14
HPM.2 Commands	NetFn	Cmd	HPM.2 Table
Get HPM.X Capability	PICMG	0x3E	3-1
Get Dynamic Credentials	PICMG	0x3F	3-12
Advantech OEM Commands	NetFn	Cmd	Section in this document
Get HW Revision	2Eh	05h	2.2.1.2
Get Payload CPU ID	2Eh	06h	
Write I2C Device	2Eh	20h	
Read I2C Device	2Eh	21h	
SEL Mode Configuration	2Eh	62h	2.2.1.3
Set SW Bank Selection	2Eh	70h	
Get SW Bank Selection	2Eh	71h	
Set Community String Flag	2Eh	74h	
Get Community String Flag	2Eh	75h	
Read Port 80 (BIOS POST Code)	2Eh	80h	
Reload NVRAM Defaults	2Eh	81h	2.2.1.5
BIOS Data Exchange	2Eh	82h	
BIOS Rollback Options	2Eh	85h	



Trigger Payload OS Interrupt	2Eh	90h	
Get BIOS Boot Bank ID	2Eh	93h	
Set BIOS Boot Bank ID	2Eh	94h	
Set Factory Mode	2Eh	E0h	
Write EEPROM Test Byte	2Eh	E6h	
Read EEPROM Test Byte	2Eh	E7h	
Reload BMC Default Configuration	2Eh	F2h	2.2.1.6
Get SW Component Information	2Eh	F3h	
Reload Factory Defaults	2Eh	F4h	

Table 83: Supported IPMI Commands



B. APPENDIX: HOW TO INSTALL IPMITOOL

Follow the instructions to download, build, and install the latest version of the IPMITool utility, although the IPMITool utility is available with most recent Linux distributions.

1. Users need to install IPMITool (example installation follows) to be able to run HPM.1 upgrades:
2. Download the latest version of IPMITool from the official website:

```
http://ipmitool.sourceforge.net/  
http://sourceforge.net/projects/ipmitool/files/
```

3. Get the patch tarball “ipmitool-1.8.18.tar.gz”:
4. Unzip the file, configure and build source:

```
# tar zxvf ipmitool-1.8.18.tar.gz  
# cd ipmitool-1.8.18  
# ./configure --enable-intf-lanplus  
# make  
# make install
```

5. Load IPMITool driver:

```
# modprobe ipmi_msghandler  
# modprobe ipmi_devintf  
# modprobe ipmi_si
```

6. Verify the IPMITool functionality:
7. Execute first IPMITool command, to read the current HPM.1 version configuration.

```
#ipmitool mc info
```

Why does IPMITool not Work?

For most Linux distributions, ipmitool shall work normally as mentioned in *Section 1.1.2 Software Support*. However, you may see the following error message while executing ipmitool:

```
root@ubuntu:~# ipmitool  
Could not open device at /dev/ipmi0 or /dev/ipmi/0 or /dev/ipmidev/0: No such file or  
directory
```

Figure 42: Error Message for Executing ipmitool



Once it happens, please check driver loading status of the ipmitool drivers (*ipmi_devintf*, *ipmi_si* and *ipmi_msghandler*) with the '*lsmod | grep ipmi*' command. If they are loaded well, Linux shall response with the message shown as the picture below:

```
# lsmod | grep ipmi
```

```
root@ubuntusky:~# lsmod | grep ipmi
ipmi_devintf      20480 0
ipmi_ssif         24576 0
ipmi_si           57344 0
ipmi_msghandler   49152 3 ipmi_ssif,ipmi_devintf,ipmi_si
```

If the driver was not loaded, follow the steps below to load the driver manually. The IPMI system interface driver (*ipmi_si*) can successfully auto-probe the address (IO ports 0xCA2/0xCA3).

Load the *ipmi_si* driver first:

```
# modprobe ipmi_si
```

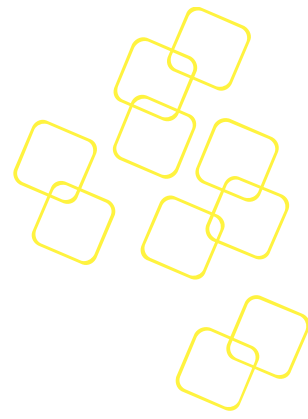
```
[root@CentOS7 ~]# modprobe ipmi_si
7443.000584] ipmi message handler version 39.2
7443.007222] IPMI System Interface driver.
7443.011341] ipmi_si: probing via ACPI
7443.015097] ipmi_si 00:06: [io 0x0ca2] regsize 1 spacing 1 irg 0
7443.021227] ipmi_si: Adding ACPI-specified kcs state machine
7443.026959] ipmi_si: probing via SMBIOS
7443.030833] ipmi_si: SMBIOS: io 0x0ca2 regsize 1 spacing 1 irg 0
7443.036789] ipmi_si: Adding SMBIOS-specified kcs state machine duplicate interface
7443.044427] ipmi_si: probing via SPMI
```

When the kernel is loading the *ipmi_si* driver, the related message handler driver *ipmi_msghandler* shall be automatically loaded as well. One more step is required to allow the user space applications to access the BMC, which is loading the related device interface driver *ipmi_devintf*:

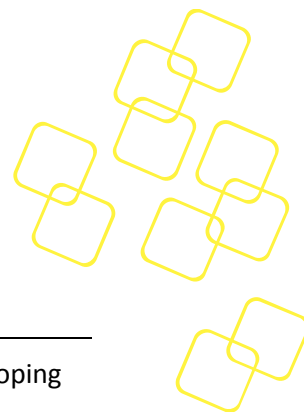
```
# modprobe ipmi_devintf
```

```
root@CentOS7 ~] modprobe ipmi_devintf
7451.259524] ipmi device interface
```

You may check driver loading status again then execute ipmitool to see whether it works.



Trouble



C. APPENDIX: FIRMWARE RELEASE AND VERSIONING NUMBER

The official BMC firmware releases are aligned with product HW and/or system developing stages (EVT/DVT/PVT/MP), see the picture below.



Figure 43: Official BMC FW Release

The BMC firmware version number is represented as M.NN, which is separated into two parts as major number and minor number. Engineering versions pre mass production will use the major number 0 and the major number 1 (or higher) is used for mass production release.

The minor numbers are aligned with milestone releases so that the first milestone 1 release is version 0.10, the first milestone 2 release is version 0.20 and so on.

The even minor numbers are used for official release which means the release is passed DQA verification. The odd minor numbers are used for test image which means the release is only for evaluation, debugging or pre-verifying bug without Advantech DQA qualification. The test image may release to customer if customer agrees in written form to exclude any warranty/liability claims and absolutely do not use the test image in production.

Here are some examples of firmware version release:

Version Number	Description
0.10	The first milestone 1 official release
0.12	The second milestone 1 official release
0.21	Test image of milestone 2 release
1.00	MP official release

Table 84: The Examples of BMC FW Version