**COM-HPC Server Carrier BMC Software Specification**

***Revision: 5.0***

Date: 03. 16, 2022

|  |  |
| --- | --- |
| ADLINK Technology, Inc.  Authorized Representative Approval | |
| Prepared by  Ken Pan  Senior Engineer, System Platform Department Section IV, TPDC | **Approved by**  **Ken Pan**  **Senior Engineer, System Platform Department Section IV, TPDC** |

Revision History

|  |  |  |
| --- | --- | --- |
| Release | Date | Summary |
| R1 | March 16, 2021 | Initial version |
| R2 | September 1, 2021 | Add a supported command table |
| R3 | February 16, 2022 | Remove SMTP information |
| R4 | March 3, 2022 | Remove FAN 2 sensor and Add ADC2 5V sensor |
| R5 | March 16, 2022 | Modify sensor threshold values |
| R6 | October 16, 2023 | Add SSIF supported |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Table of Contents

[Table of Contents 1](#_Toc148363862)

[1. General Information 4](#_Toc148363863)

[1.1 References 4](#_Toc148363864)

[1.2 Glossary and Abbreviations 4](#_Toc148363865)

[2. COM-HPC Server Carrier Firmware 5](#_Toc148363866)

[2.1 General System Overview 5](#_Toc148363867)

[2.2 Switch Design of BIOS 6](#_Toc148363868)

[2.3 Physical layout of the COM-HPC Server Carrier Server Carrier 7](#_Toc148363869)

[2.4 Software Architecture 8](#_Toc148363870)

[2.5 Baseboard Management Controller (BMC) 9](#_Toc148363871)

[2.6 Software Interfaces 9](#_Toc148363872)

[2.6.1 IPMI 10](#_Toc148363873)

[2.6.2 SSH Console 10](#_Toc148363874)

[2.6.3 Debug Console 10](#_Toc148363875)

[2.6.4 WebUI 11](#_Toc148363876)

[2.6.5 HTML5 KVM 12](#_Toc148363877)

[2.6.6 Local connection 13](#_Toc148363878)

[2.6.7 SSIF 13](#_Toc148363879)

[2.7 Basic Network Settings 14](#_Toc148363880)

[3. Hardware Control 14](#_Toc148363881)

[3.1 Reading Voltage Sensors 14](#_Toc148363882)

[3.2 Controlling the Fans and Reading the Fan Sensors 16](#_Toc148363883)

[3.3 Switching BIOS controller and flashes 17](#_Toc148363884)

[3.4 General Purpose I/O (GPIO) Table 18](#_Toc148363885)

[4. BMC Operation Features 18](#_Toc148363886)

[4.1 Sensor Data Record (SDR) 18](#_Toc148363887)

[4.1.1 Reading sensor data from Web Interface 23](#_Toc148363888)

[4.1.2 Reading Sensor data by IPMI command 25](#_Toc148363889)

[4.2 FRU Data Information 27](#_Toc148363890)

[4.2.1 FRU Data and Manager Control 27](#_Toc148363891)

[4.2.2 Access FRU data 28](#_Toc148363892)

[4.2.2.1 Access FRU via IPMI 28](#_Toc148363893)

[4.2.2.2 Access FRU via WebUI 29](#_Toc148363894)

[4.3 System Event Log 29](#_Toc148363895)

[4.4 Firmware Upgrade 32](#_Toc148363896)

[4.4.1 Supported Tools 32](#_Toc148363897)

[4.4.1.1 Yafu(Yet Another Firmware Update) Flash 33](#_Toc148363898)

[4.4.1.2 WebUI 35](#_Toc148363899)

[4.5 Audit Logs 36](#_Toc148363900)

[4.6 Reset to factory default 37](#_Toc148363901)

[4.6.1 Reset BMC to default 37](#_Toc148363902)

[4.7 BMC Health Monitoring 37](#_Toc148363903)

[5. Appendix A Error Log Collection 38](#_Toc148363904)

[5.1 Enable the Remote Logs 38](#_Toc148363905)

[5.2 Retrieve logs on the BMC 39](#_Toc148363906)

[5.3 Supported command table 39](#_Toc148363907)

# General Information

## References

[1] [Redis](https://redis.io/): An open source database

[2] [RFC7235](https://tools.ietf.org/html/rfc7235) Hypertext Transfer Protocol (HTTP/1.1): Authentication

[3] [Intelligent Platform Management Interface Spec, v.2, Rev. 1.1](https://www.intel.com/content/www/us/en/servers/ipmi/ipmi-second-gen-interface-spec-v2-rev1-1.html)

[4] IPMI- Platform Management FRU Information Storage Definition V1.0 document Revision 1.2

[5] [Intel Rack Scale Design Architecture Specification Software version 2.1](https://www.intel.com/content/dam/www/public/us/en/documents/guides/platform-hardware-design-guide-v2-1.pdf)

[6] AMI Generic MegaRAC SP-X Firmware RESTful Web service for SPX

[7] AMI Generic MegaRAC SP-X Firmware User’s Guide

[8] [DMTF](http://www.dmtf.org): Distributed Management Task Force, INC.

[9] [Loggly](https://www.loggly.com/): A commercial log service

## Glossary and Abbreviations

|  |  |
| --- | --- |
| Term | Description |
| BMC | Baseboard Management Controller |
| PCH | Platform Controller Hub |
| BIOS | Basic Input/Output System |
| HTTPS | Hypertext Transfer Protocol Secure |
| HTML | Hypertext Markup Language |
| DHCP | Dynamic Host Configuration Protocol |
| SDR | Sensor Data Record |
| SEL | System Event Log |
| IPMI | Intelligent Platform Management Interface |
| KVM | Keyboard Video Mouse |
| COM-HPC | Computer-On-Module – High Performance Compute |
| NTP | Network Time Protocol |
| LPC | Low Pin Count |
| SPI | Serial Peripheral Interface |
| eSPI | Enhanced Serial Peripheral Interface |
| RMII | Reduced Media Independent Interface |
| FRU | Field Replaceable Unit |
| SSH | Secure Shell |
| SSL | Secure Socket Layer |
| RAM | Random Access Memory |
| MAC | Media Access Control |
| ADC | Analog-to-Digital Converter |
| PWM | Pulse Width Modulation |
| EEPROM | Electrically-Erasable Programmable Read-Only Memory |
| GPIO | General Purpose Input/Output |
| ACPI | Advanced Configuration and Power Interface |

# COM-HPC Server Carrier Firmware

The COM-HPC Server Carrier Firmware is composed of 2 major components.

* BMC
* BIOS

**BMC**

An ASPEED® AST2500 BMC is applied to COM-HPC Server Carrier for system management. It provides several features for system management locally and remotely.

## General System Overview

The BMC system is established based on the Debian Jessie with a customized kernel based on version 3.14.17, following table shows the general feature of the BMC system.

|  |  |
| --- | --- |
| AMI Feature | Supported on AST2500 |
| SMBUS Read/Write | O |
| Linux Kernel 3.14.17 | O |
| IPMI specification v2.0, Rev. 1.1 | O |
| DHCP Client | O |
| IPv4 | O |
| IPv6 | O |
| Debian Jessie support | O |
| BMC upgrade | O |
| Serial over LAN | O |
| KVM | O |

To maintain the BMC, there are 1 general background tasks defined in the system crontab:

|  |  |  |
| --- | --- | --- |
| Binary | Description | Frequency |
| /etc/init.d/setntpdate | NTP time synchronization | Hourly |

The default shell of the firmware is the /bin/bash which is linked to the BusyBox v2.9.0, it provides following tools when using the console:

adjtimex, arp, arping, ash, basename, bash, brctl, cat, catv, chgrp, chmod, chown, chroot, clear, cp, cpio, cut, date, dd, depmod, devmem, df, diff, dirname, dmesg, dnsdomainname, du, echo, egrep, expr, false, fgrep, find, free, freeramdisk, ftpget, getty, grep, gunzip, gzip, head, hostname, id, ifconfig, ifdown, ifenslave, ifplugd, ifup, inetd, insmod, kill, killall, killall5, last, ln, logger, login, losetup, ls, lsmod, mkdir, mkfifo, mknod, mktemp, modinfo, modprobe, more, mount, mountpoint, mv, netstat, nslookup, passwd, ping, pivot\_root, pwd, readlink, realpath, rm, rmdir, rmmod, route, run-parts, runlevel, sed, sh, sleep, sort, stty, sulogin, swapoff, swapon, switch\_root, sync, sysctl, tail, test, tftp, tftpd, time, top, touch, traceroute, traceroute6, true, tty, udhcpc, udhcpc6, udpsvd, umount, uname, uniq, uptime, vconfig, vi, watchdog, wc, wget, which, who, whoami, xargs, zcat, zcip.

Except for described above, there are utilities located in:

/bin

/sbin

/usr/bin

/usr/sbin

/usr/local/bin

Most of them were redirected to the BusyBox except for those functions require standard support such as top, zip-upzip and wget.

Customized tools are located in the /usr/local/bin includes the flasher and IPMIMain, the former is the utility to write the firmware, the latter is the IPMI stack.

## Switch Design of BIOS

To make sure the host system is provided its function during running and upgrading time. The BMC could switch the controller of BIOS to the PCH or to the BMC itself by a GPIO pin as described in Figure 1. For normal situation, the BIOS is connected to the PCH. One may ask that in the normal situation how the BMC can communicate with the BIOS. The answer is the LPC connection between the PCH and the BMC. With this LPC connection, the BMC and BIOS could communicate via the registers in the PCH.

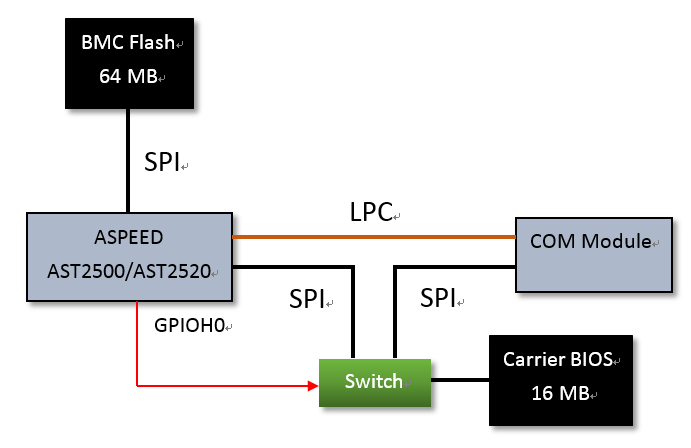


Figure 1: BIOS switch layout

## Physical layout of the COM-HPC Server Carrier Server Carrier

The physical layout of the COM-HPC Server Carrier is as below.



## Software Architecture

The COM-HPC Server Carrier firmware is compliant with following Specifications:

* IPMI package fully compliant with IPMI [3].

The BMC software stack is established on AMI MegaRAC® SP-X RR11-7, it is a powerful management stack enabling fast, realistic and high-quality remote management of server systems from anywhere in the world. Administrators can complete Out-of-Band, OS-independent server control. The firmware not only follows the IPMI 2.0 Spec. but also provides several user interfaces. For example, the WebUI provides a graphical HTML interface with KVM supported.

## Baseboard Management Controller (BMC)

The core of COM-HPC Server Carrier hardware management system is an AST2500 chipset manufactured by ASPEED. The software interface is described as below:

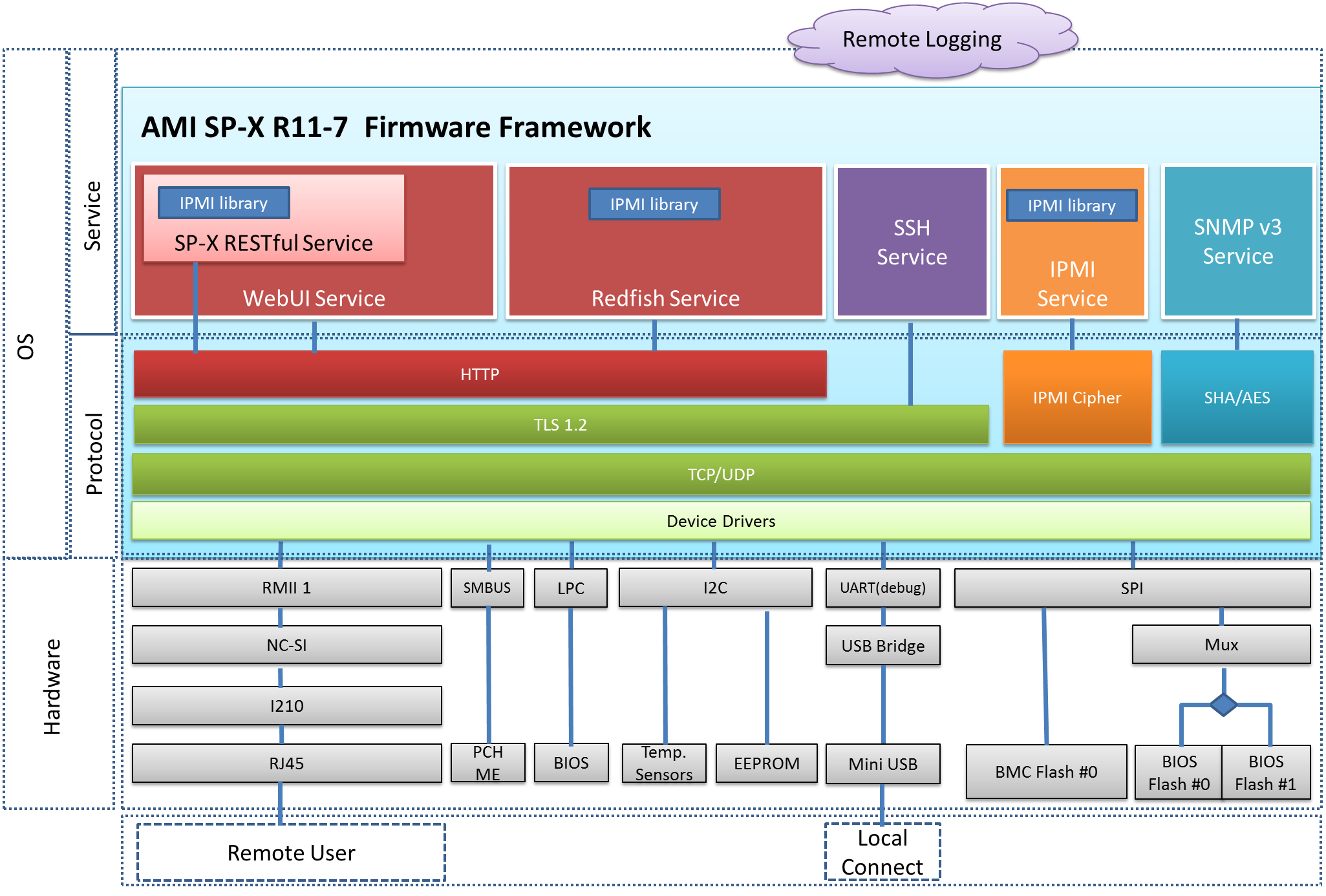


Figure 4: The software/hardware interface of the firmware, from the user’s point of view.

Since the firmware is established based on the Linux Kernel, as described in the above figure, the hardware is controlled by the device drivers and protected by the OS. Applications could only access the hardware via the device drivers.

The BMC firmware is highly related to the payload hardware, the hardware components are accessed via following interfaces:

* RMII network interface (NIC)
* LPC interface
* eSPI interface
* UART-based serial debug interface, in order to support console debug and Serial Over LAN function (SOL)
* I2C/SMBus interface

## Software Interfaces

As described in the previous section, the firmware provides several user interfaces for BMC.

### IPMI

The firmware follows the IPMI standard 2.0 and provides IPMI commands for users to access the BMC to show FRU data, SEL logs, etc.

The default username with administrator privileges is admin/admin, users have to modify it before deploying them into the enterprise environment.

### SSH Console

The firmware provides a Linux built-in SSHv2 rev. 7.7p1-2 service. Only 5 concurrent sessions are allowed for SSH users. This is done by checking the number of SSH connections before establishing it. If there are already 5 SSH connections are established, the shell would deny the login attempt.

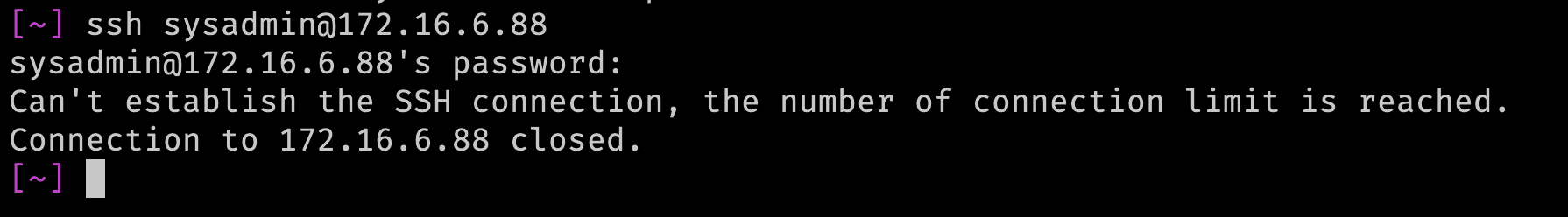


Figure 5: The SSH connection would be denied if it reaches the limitation.

The SSH console is using the authentication of the OS installed on the BMC. The default username and password of the SSH connection is sysadmin/superuser.

The SSH console and debug console are using the same system account, a built-in account which has administrator privileges is sysadmin/superuser.

### Debug Console

The debug console could be connected via the COM port on the front panel of the server. The username and password are also the same as the SSH interface since they are all directly connected to the Linux OS on the BMC.

The most difference between the SSH console and the debug console is: The debug messages and the system standard error would be displayed on the debug console only.

### WebUI

The WEBUI is a graphical interface that controls the BMC and Payload via the web service. Users are able to access the WEBUI with a browser that supports HTML5. The WEBUI provides several functionalities as follows:

* Dashboard
  + BMC Up Time
  + Pending De-assertions
  + Access Logs
  + Today events
  + 30 days events
  + Sensor Monitoring
* Sensor
  + Critical Sensors
  + Discrete Sensor States
  + Normal Sensors
  + Disabled Sensors
* FRU Information
  + Available FRU Devices
  + Board Information
  + Product Information
* Logs & Reports
  + IPMI Event Log
  + System Log
  + Audit Log
* Settings
  + Date & Time
  + Log Settings
    - SEL Log Settings Policy
    - Advanced Log Settings
  + Network Settings
    - Network IP Settings
  + Platform Event Filters
    - Event Filters
    - Alert Policies
    - LAN Destinations
  + Services
    - web
    - kvm
    - ssh
    - solssh
  + SSL Settings
    - View SSL certificate
    - Generate SSL certificate
    - Upload SSL certificate
  + User Management
* Remote Control
  + Launch KVM
* Power Control
  + Power off
  + Power on
  + Power cycle
  + Hard reset
  + ACPI shutdown
* Maintenance
  + Firmware Information
  + Firmware Update
  + System Administrator

The username and password of WebUI is synchronized with IPMI users.

### HTML5 KVM

The term KVM is the abbreviation of the Keyboard, Video, and Mouse. It is able to re-direct all keyboard, video, and mouse activities to the local machine. Through the help of BMC hardware, it is also able to redirect local devices such as CDROM and disk image to the remote server as it is physically mounted on the server. The KVM could only be launched by the WEBUI since it is created by HTML5.

Once a user gets in the KVM, he/she is able to access the remote server just like a nearby server. He/she can access the BIOS and access the host system.

The KVM is tested and supported by AMI.

|  |  |
| --- | --- |
| Brower | Version |
| Windows | |
| IE | 11+ |
| Edge |  |
| Firefox | 43+ |
| Chrome | 47+ |
| Mac | |
| Safari |  |
| Linux | |
| Firefox | 38+ |
| Chrome | 47+ |
| Opera | 26.0 |

The KVM supports following 3 types of keyboard layout:

* English (U.S)
* German
* Japanese

To use the video part of KVM without any video latency, it is recommended to have more than 8GM RAM on the client machine.

### Local connection

The local connection equals to the SSH connection and is provided by the com-port located on the server board.

### SSIF

The SMBus System Interface (SSIF) acts as a specialized bridge for communication between the CPU and the BMC in a server environment. Instead of depending on a COM port, SSIF utilizes the SMBus to achieve its objectives.

The primary role of SSIF is to facilitate the CPU in sending IPMI commands to the BMC.

Although both the debug console and SSH interface have a direct link to the Linux OS on the BMC and share the same username and password, SSIF operates on a distinct layer. Its focus is solely on streamlining CPU-to-BMC communication via IPMI commands. Unlike the debug console, which is designed to display system errors and debug messages, SSIF is tailored for executing system management commands. Notably, using SSIF to execute IPMI commands does not require a username or password. However, the user issuing the commands must possess sufficient administrative privileges, such as being a sudoer or administrator.

## Basic Network Settings

The firmware supports both IPv4 and IPv6 connection. It could be set to static IP manually or dynamic IP by setting a DHCP server. The DHCP would be automatically enabled if the BMC is not set to a static IP.

Since the MAC Address is located in the /conf section and would not be over written by any upgrade process. All network setting would remain the same even if the user upgrades the BMC without preserving configuration.

When a new image is written by DediProg tool, the network setting would be erased. The only way to make it available is to use the debug console. If a user has to re-configure the network via the debug console, below procedure should be followed:

1. Reset the BMC and click “B/b” key to enter Uboot environment.
2. Type following command:

*set ethaddr <MAC Address>*

*saveenv*

*bootfmh*

The BMC would now have a valid mac address and users are able to modify the network via WebUI. Please notice that **\*\*you should only modify the MAC Address according to the label on the board\*\***.

# Hardware Control

The chapter describes the mechanism of BMC of controlling the hardware, since they are not exposed to the end-users, that is, the end-users are not able to access the hardware directly via any wired connection, it is still worthy to describe the controlling since it is about the detail of the firmware design. The software interface for end-users to access the hardware is described in Section 2.6.

## Reading Voltage Sensors

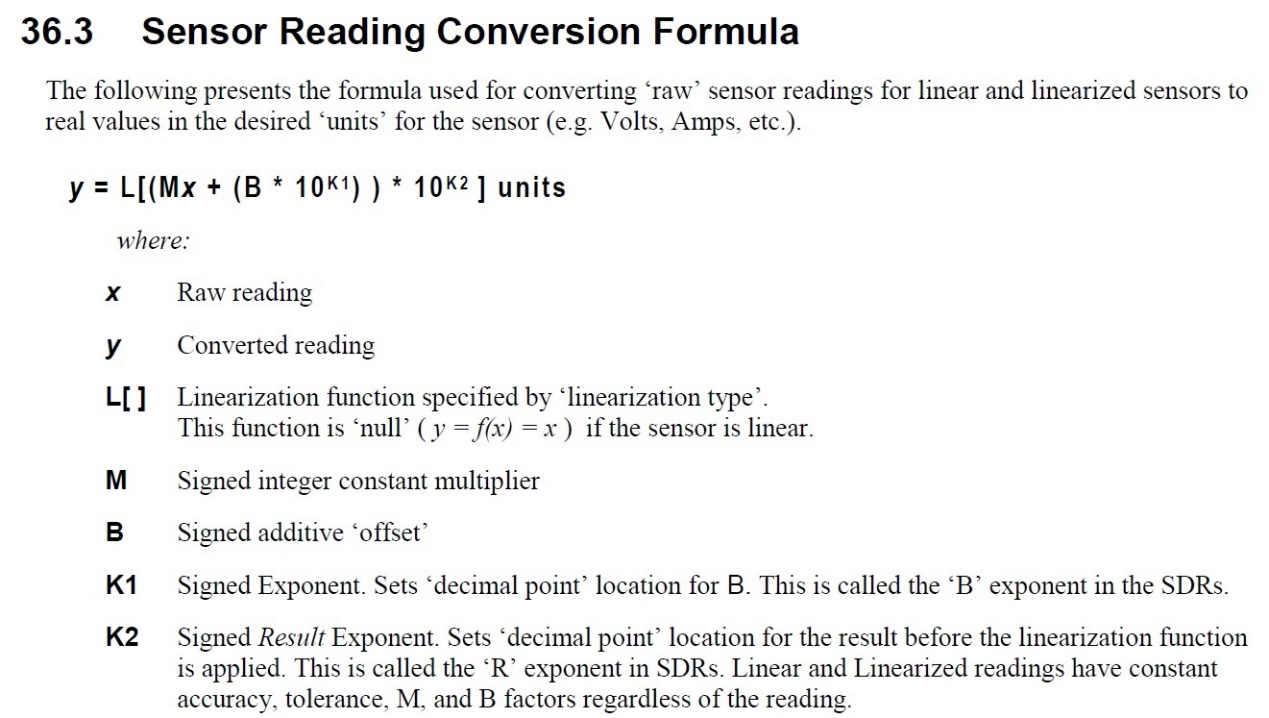
The voltage sensor values are provided by the ADC controller of AST2500 chipset. It provides 16 channels (pins) to monitor current voltage readings on the server. Those pins have been defined in ADC unit via the chip to monitor sensor values on the board.

The ADC controller has sixteen channels and each ADC channel could be assigned to a GPIO pin to detect voltage sensor on the payload. The ADC sensor table (Table 4) is defined how many voltage sensors and description in the system.

|  |  |  |  |
| --- | --- | --- | --- |
| PIN No | I/O | PIN NAME | Voltage Sensors Description |
| F4 | INPUT | P\_+2V5\_BMC\_AUX | Sensor reading from P\_+2V5\_BMC\_AUX |
| F5 | INPUT | P12V\_MOD | Sensor reading from P12V\_MOD |
| E2 | INPUT | VCC\_5V\_SBY\_IN | Sensor reading from VCC\_5V\_SBY\_IN |
| E1 | INPUT | P\_+1V15\_BMC\_AUX | Sensor reading from P\_+1V15\_BMC\_AUX |
| E3 | INPUT | P\_+5VSB | Sensor reading from P\_+5VSB |
| G5 | INPUT | P\_+5V | Sensor reading from P\_+5V |
| G4 | INPUT | P\_+3V3 | Sensor reading from P\_+3V3 |
| F2 | INPUT | P\_+3V3SB | Sensor reading from P\_+3V3SB |
| G2 | INPUT | P\_+1V8 | Sensor reading from P\_+1V8 |
| H5 | INPUT | P\_+1V8SB | Sensor reading from P\_+1V8SB |
| G1 | INPUT | VBAT\_HM | Sensor reading from VBAT\_HM |
| F3 | INPUT | GND | GND |
| G3 | INPUT | GND | GND |
| F1 | INPUT | GND | GND |
| H3 | INPUT | GND | GND |
| H4 | INPUT | GND | GND |

Table 4: ADC Sensor Table. The pin number is defined by AST2500/AST2520. The pin name is defined by the HW which is used to describe the sensor.

The raw data read from the ADC is not the current reading of the sensor, it has to be transformed to the correct value via some predefined constraints. According to IPMI v2.0 specifications, the following presents the formula used for converting raw sensor readings for linear and non-linear sensors to real values in the desired units for the sensors.



## Controlling the Fans and Reading the Fan Sensors

The firmware provides a Smart Fan Control mechanism to control the fan speed according to the system temperatures of the chassis since the fan speed is highly related to the temperature of the system temperature reading. The algorithm of the fan control is basically using the system temperatures as input values.

The fan related PWM controllers and sensors are listed below:

|  |  |  |  |
| --- | --- | --- | --- |
| PIN No | I/O | PIN NAME | Voltage Sensors Description |
| U5 | INPUT | BMC\_FAN\_TACH0 | RPM value of the fan.0 |
| U4 | INPUT | BMC\_FAN\_TACH1 | RPM value of the fan 1 |
| AB4 | INPUT | BMC\_FAN\_TACH2 | RPM value of the fan 2 |
| AB3 | INPUT | BMC\_FAN\_TACH3 | RPM value of the fan 3 |
| V2 | OUTPUT | BMC\_PWM\_ FAN0 | This pin controls the duty cycle time of fan 0 |
| W2 | OUTPUT | BMC\_PWM\_ FAN1 | This pin controls the duty cycle time of fan 1 |
| U3 | OUTPUT | BMC\_PWM\_ FAN2 | This pin controls the duty cycle time of fan 2 |
| W3 | OUTPUT | BMC\_PWM\_ FAN3 | This pin controls the duty cycle time of fan 3 |

Table 5: FAN GPIO Table

## Switching BIOS controller and flashes

As described in chapter 2, a single GPIO pin, GPIOH0, could be used to switch the BIOS’ control among the module and BMC.

|  |  |
| --- | --- |
| GPIOH0 | SPI Connection Target |
| High (Default) | module |
| Low | BMC |

Also, when the control is switched back to the BMC, jumper JP27, JP28, and JP29, are used for selecting the BIOS flashes.

|  |  |  |  |
| --- | --- | --- | --- |
| JP29 | JP28 | JP27 | Flash Selection |
| Pin 1-2 | **Pin 1-2** | **Pin 2-3** | 1st Flash |
| Pin 1-2 | **Pin 2-3** | **Pin 1-2** | 2nd Flash |

## General Purpose I/O (GPIO) Table

On the COM-HPC Server Carrier product system, The Generic purpose IO pin on an integrated circuit whose behavior, including whether it is an input or output pin, can be controlled or detected at run time in general GPIO table (Table 6).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Function | I / O | GPIO | PIN NAME | Description |
| Power Button | OUTPUT | GPIOZ2 | FM\_BMC\_PWRBTN\_OUT\_N | BMC needs to output FM\_BMC\_PWRBTN\_OUT\_N signal (active low pulse) when BMC received power bottom command or message from module. |
| Reset | OUTPUT | GPIOE1 | RST\_BMC\_SYSRST\_BTN\_OUT\_N | When receiving Reset button event from module, used as GPIO for sending Reset button event to PCH |
| SPI Switch | OUTPUT | GPIOH0 | FM\_PCH\_SPI\_BMC\_CTRL\_N | Use as GPIO output for module SPI & BMC SPI1 bus switching  SPI source select for EEPROM flash of BIOS.  High level: module SPI bus to BIOS EEPROM  Low level: BMC SPI1 bus to BIOS EEPROM |
| Power Good | INPUT | GPIOE4 | PWRGD\_P3V3 | Payload 3V3 voltage power is ready. |
| INPUT | GPIOP7 | BMC\_RDY\_N | To be set high when BMC boot completed. |

Table 6: GPIO Table

# BMC Operation Features

## Sensor Data Record (SDR)

As defined in IPMI Specification, firmware defines several sensors as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sensor Number | Sensor Name | Sensor Class | | Short Description |
| #0x14 | P\_2V5\_BMC\_AUX | threshold | 2.5V BMC auxiliary | |
| #0x1 | P12V\_MOD | threshold | Module 12V | |
| #0x2 | VCC\_5V\_SBY\_IN | threshold | VCC 5V standby | |
| #0x3 | P\_1V15\_BMC\_AUX | threshold | 1.15V BMC auxiliary | |
| #0x4 | P\_5VSB | threshold | 5V standby | |
| #0x5 | P\_5V | threshold | 5V power | |
| #0x6 | P\_3V3 | threshold | 3.3V power | |
| #0x7 | P\_3V3SB | threshold | 3.3V standby | |
| #0x8 | P\_1V8 | threshold | 1.8V power | |
| #0x9 | P\_1V8SB | threshold | 1.8V standby | |
| #0xA | VBAT\_HM | threshold | 3V battery | |
| #0xB | BMC\_FAN\_0 | threshold | Fan 0 tachometer | |
| #0xC | BMC\_FAN\_1 | threshold | Fan 1 tachometer | |
| #0xE | BMC\_FAN\_2 | threshold | Fan 2 tachometer | |
| #0xF | BMC\_FAN\_3 | threshold | Fan 3 tachometer | |
| #0x10 | REAR\_TEMP | threshold | Thermal sensor (Rear side) | |
| #0x11 | FRONT\_TEMP | threshold | Thermal sensor (Front side) | |
| #0x12 | WATCHDOG | discrete | Watchdog sensor | |
| #0x13 | ACPIPowerStatus | discrete | Power status sensor | |

Table 7: The type and the sensor number of sensors.

There are two kinds of sensors in the firmware, threshold sensor and discrete sensor. The threshold sensor is used for presenting numerical value of HW sensors, each of the threshold type sensor has the following thresholds:

* upper non-recoverable (UNR)
* upper critical (UC)
* upper non-critical (UNC)
* lower non-critical (LNC)
* lower critical (LC)
* lower non-recoverable (LNR)

The following sensor table is defined by electrical engineering.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sensor Name | Reading Type (unit) | Nominal Reading | LNR | | LC | | LNC | | | UNC | UC | UNR | |
| P\_2V5\_BMC\_AUX | VOLT | 2.5 | | 2.14 | | 2.26 | | 2.38 | | 2.64 | 2.76 | | 2.88 |
| P12V\_MOD | VOLT | 12 | | 10.15 | | 10.78 | | 11.34 | | 12.60 | 13.16 | | 13.79 |
| VCC\_5V\_SBY\_IN | VOLT | 5 | | 4.26 | | 4.50 | | 4.74 | | 5.25 | 5.49 | | 5.76 |
| P\_1V15\_BMC\_AUX | VOLT | 1.15 | | 0.98 | | 1.04 | | 1.09 | | 1.21 | 1.27 | | 1.32 |
| P\_5VSB | VOLT | 5 | | 4.26 | | 4.50 | | 4.74 | | 5.25 | 5.49 | | 5.76 |
| P\_5V | VOLT | 5 | | 4.26 | | 4.50 | | 4.74 | | 5.25 | 5.49 | | 5.76 |
| P\_3V3 | VOLT | 3.3 | | 2.82 | | 2.98 | | 3.14 | | 3.48 | 3.62 | | 3.80 |
| P\_3V3SB | VOLT | 3.3 | | 2.82 | | 2.98 | | 3.14 | | 3.48 | 3.62 | | 3.80 |
| P\_1V8 | VOLT | 1.8 | | 1.53 | | 1.62 | | 1.71 | | 1.89 | 1.98 | | 2.07 |
| P\_1V8SB | VOLT | 1.8 | | 1.53 | | 1.62 | | 1.71 | | 1.89 | 1.98 | | 2.07 |
| VBAT\_HM | VOLT | 3 | | 1.82 | | 1.96 | | 2.1 | | 3.3 | 3.46 | | 3.6 |
| BMC\_FAN\_0 | RPM | n/a | | 0 | | 0 | | 0 | | 25500 | 25500 | | 25500 |
| BMC\_FAN\_1 | RPM | n/a | | 0 | | 0 | | 0 | | 25500 | 25500 | | 25500 |
| BMC\_FAN\_2 | RPM | n/a | | 0 | | 0 | | 0 | | 25500 | 25500 | | 25500 |
| BMC\_FAN\_3 | RPM | n/a | | 0 | | 0 | | 0 | | 25500 | 25500 | | 25500 |
| REAR\_TEMP | ℃ | n/a | | -128 | | -128 | | -26 | | 76 | 127 | | 127 |
| FRONT\_TEMP | ℃ | n/a | | -128 | | -128 | | -26 | | 76 | 127 | | 127 |
| WATCHDOG | discrete | 0x0080 | | n/a | | n/a | | | n/a | n/a | n/a | | n/a |
| ACPIPowerStatus | discrete | 0x0180 | | n/a | | n/a | | | n/a | n/a | n/a | | n/a |

Table 8: The threshold values of sensors.

An SEL (System Event Log) would be generated when the reading is higher or lower than one of the thresholds listed above and would be recorded to SEL logs. Each SEL has 2 different subtypes, assertion and de-assertion.

* Assertion: Across the threshold from previous better status to worse status
* De-assertion: Across the threshold from previous worse status to better status

An example for 5V sensor as below:

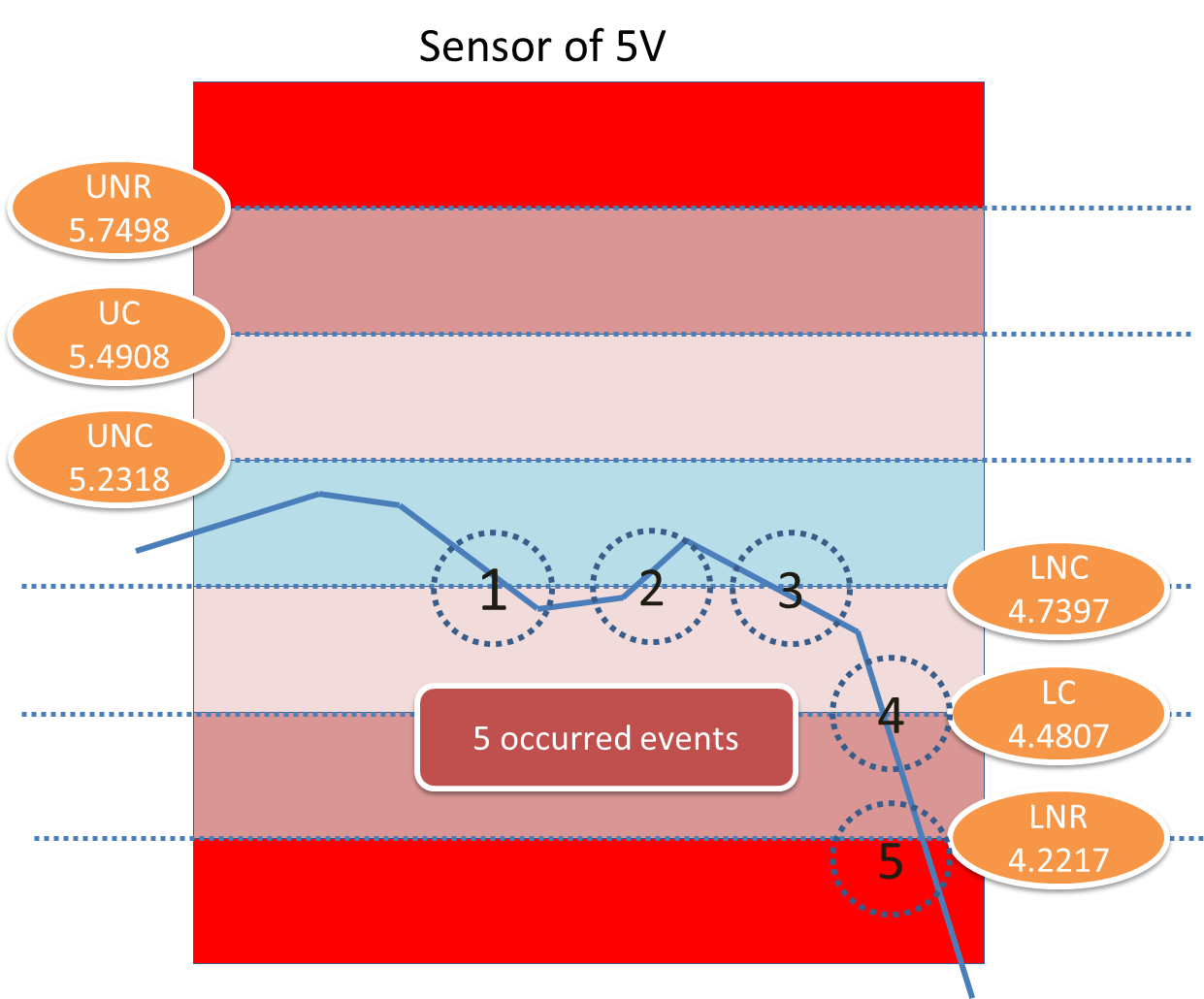


Figure 15: Thresholds of 5V sensor and 5 events for example.

There are 5 events generated by the sensor:

|  |  |  |
| --- | --- | --- |
| Example No. | Sensor Type |  |
| 1 | LNC | Assertion |
| 2 | LNC | De-assertion |
| 3 | LNC | Assertion |
| 4 | LC | Assertion |
| 5 | LNR | Assertion |

It is easy to imagine that once an event occurs, there should be a pair of assertion and de-assertion of the same type. These pairs show that the HW is unstable but is recovered to a normal status. If the assertion and the de-assertion are not occurring in pair, the hardware is not recovered from the status. If there’s any UNR or LNR event occurs, the HW should have a physical failure and needs to be repaired.

To prevent from the situation that a vibration sensor creates too many events, a value Hysteresis is also defined. For example, when an upper assertion is triggered, the corresponding de-assertion is only triggered when the reading value is lower than the threshold minus the hysteresis. Please see below figure for details:

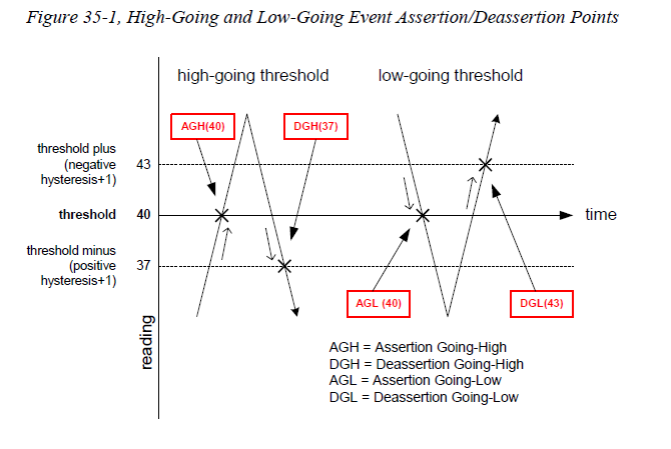


Figure 16: Hysteresis of sensors

In COM-HPC Server Carrier, we can use the following IPMI command to get SEL of events like power state, system event, fan status, etc.

*ipmitool -I lanplus -H <BMC\_IP> -U admin -P admin sel list*

The entry of SEL via the above IPMI command can be divided into 6 fields. The first field is record ID in hexadecimal format. This ID is used for SEL Record access. The second field and the third field are date and time, respectively when event was logged. The fourth field is the sensor type and sensor number of the sensor that generated the event. There are System ACPI Power state, Temperature, Fan, Voltage, Current, System Event, and the Microcontroller/Coprocessor sensor types. The fifth field is the event type of trigger for the event. The last field indicates the event transition direction, assertion and de-assertion. The detailed explanation of these fields can be found in IPMI v2.0 specification [3] Table 32-1. Figure 17 is the SEL example via IPMI command.

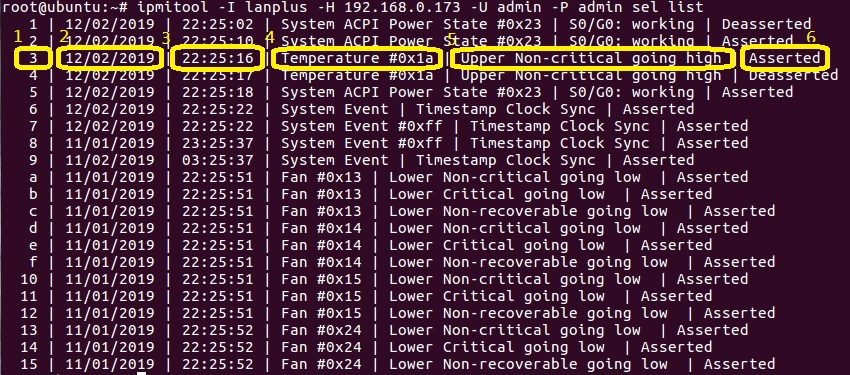


Figure 17: SEL record via IPMI command

|  |  |
| --- | --- |
| Sensor Type | Event Type |
| Temperature | Upper Non-critical going high… |
| Fan | Lower Non-critical going low, Lower Critical going low, Lower Non-recoverable going low… |
| Voltage | Lower Non-critical going low, Lower Critical going low, Lower Non-recoverable going low… |
| System Event | Timestamp Clock Sync |
| Microcontroller/Coprocessor | Transition to Running |

Table 9: The fourth field (Sensor Type) and the fifth field (Event Type) of SEL via IPMI command

### Reading sensor data from Web Interface

The Web Service provides a graphical sensor data reading in the main page. The critical sensor data would be alerted with red color for a user who is monitoring on a single machine.

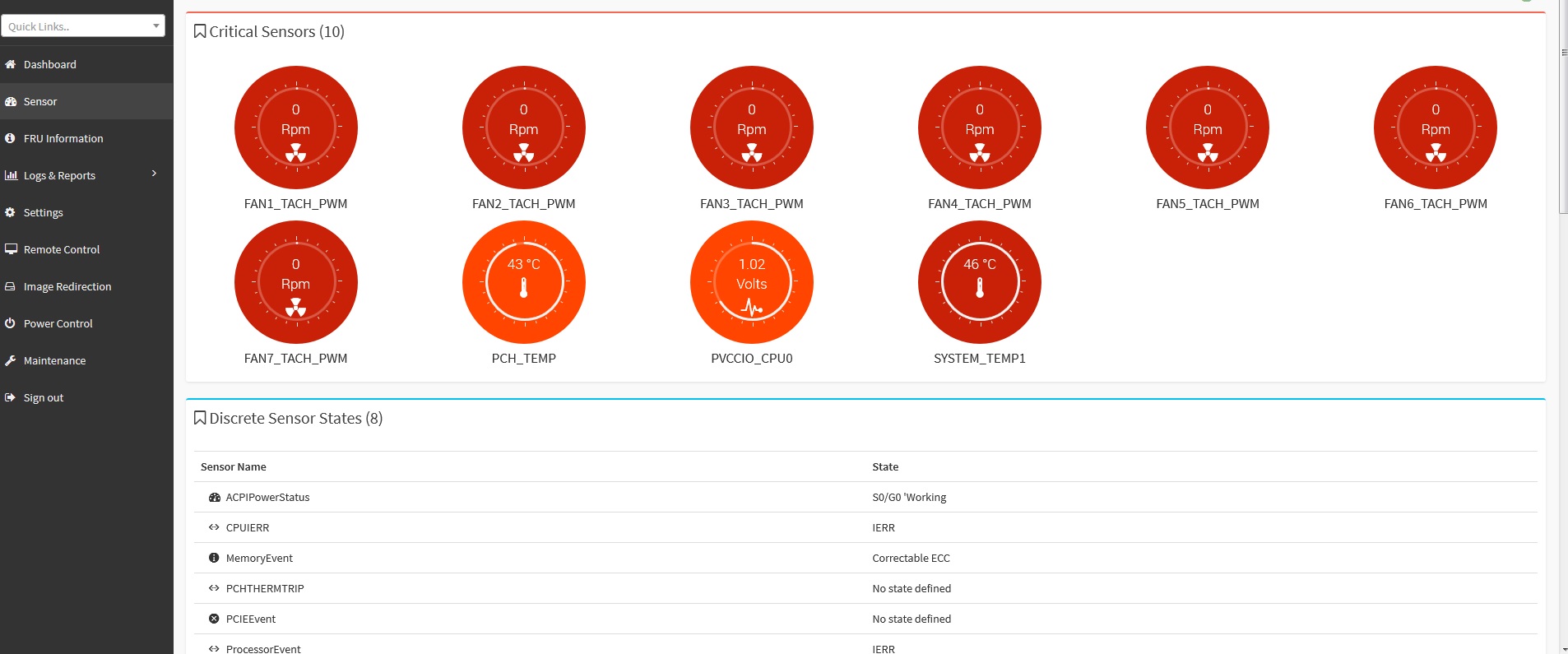


Figure 18: The Critical sensor data

In the bottom of the page, the normal sensors are also shown with reading values and the history data.

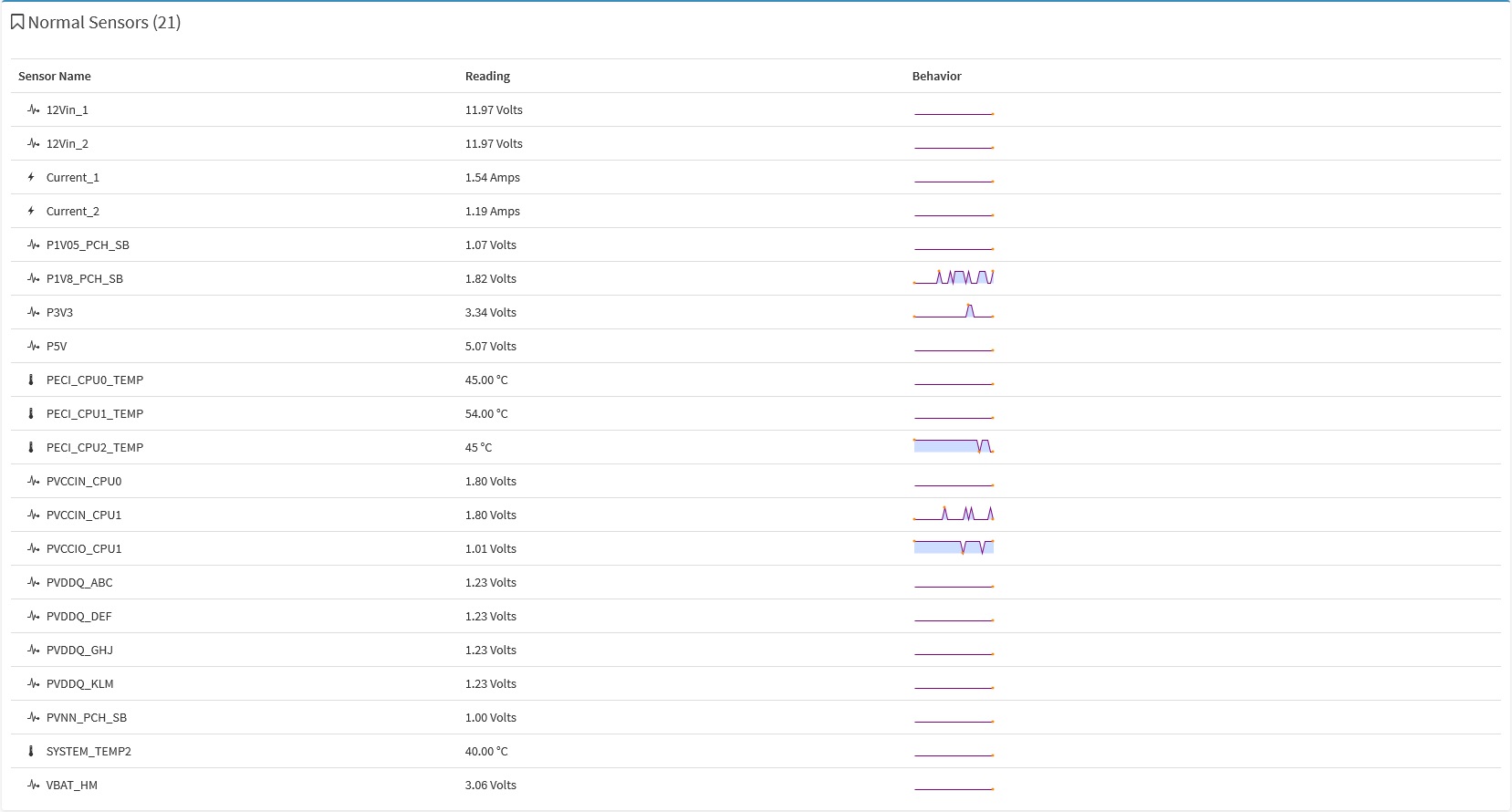


Figure 19: Sensor

Users are able to see real time values by clicking each of the sensors. The threshold values are also changeable in the detail page of a particular sensor.

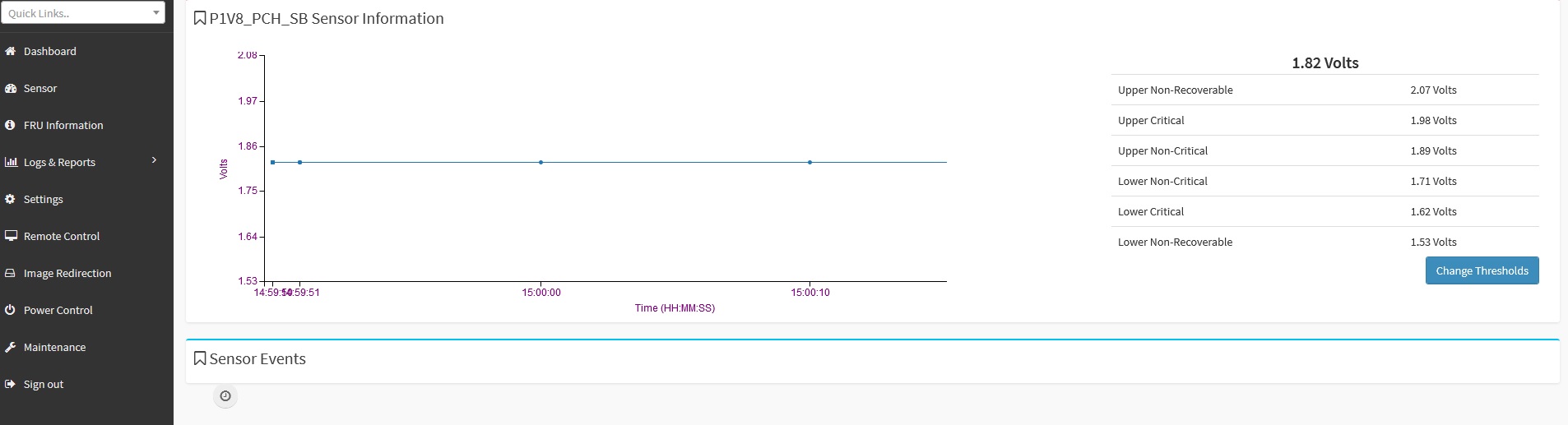


Figure 20: Real time sensor reading

### Reading Sensor data by IPMI command

For the remote users, following 2 commands are useful for monitoring sensor data.

*ipmitool -I lanplus -H <BMC\_IP> -U admin -P admin sdr*

*ipmitool -I lanplus -H <BMC\_IP> -U admin -P admin sensor*

The former one shows the brief status of all sensors, the latter one displays the detail of all the sensors.

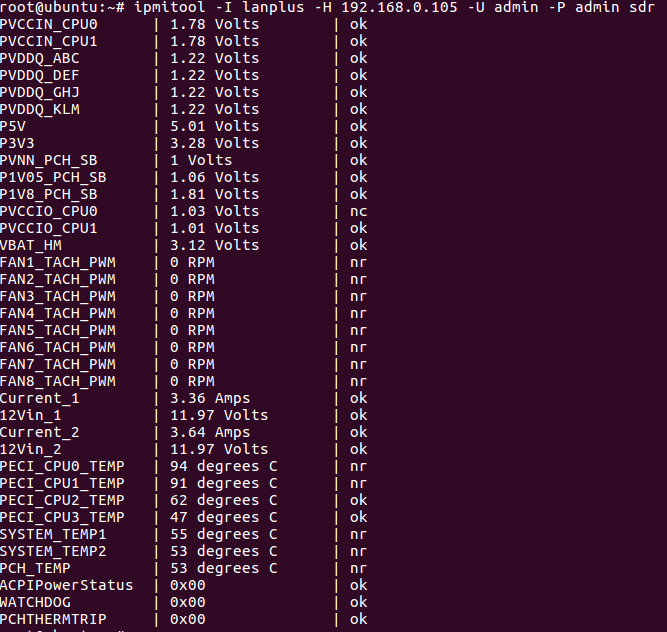


Figure 21: The result of sensor data record via IPMI command.

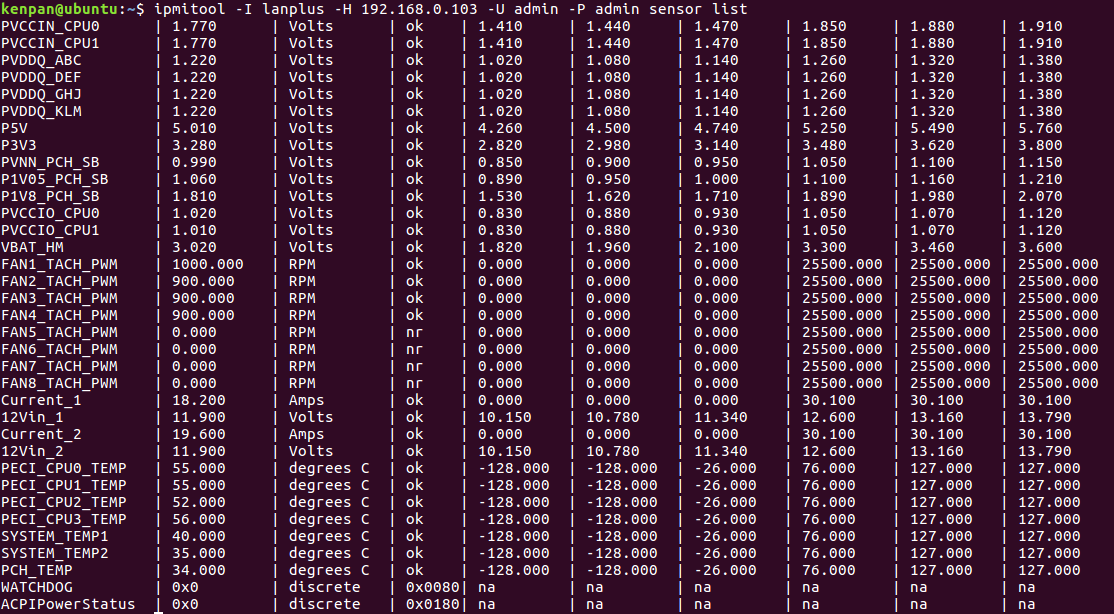


Figure 22: The detail result of sensor data record via IPMI command.

## FRU Data Information

### FRU Data and Manager Control

The FRU information is created based the IPMI FRU specification [4]. For COM-HPC Server Carrier, the FRU information is customized according to SRS. Following terms are defined to be required in the FRU information.

* Board
* Product Name
* Product Version
* Part Number
* Serial Number
* Manufacturer

These data should be kept and are not able to be erased during the upgrade process. The format of FRU information is as below which contains above 6 required fields:

|  |  |
| --- | --- |
| FRU Data Field | Nokia value |
| Product Manufacturer | COM-HPC Server Carrier |
| Product Name | COM-HPC Server Carrier |
| Product Serial Number | 123456789ABCDE |
| Product Version | 0001 |
| Product Asset Tag | 123456 |
| Board Manufacturer | COM-HPC Server Carrier |
| Board Product Name | COM-HPC Server Carrier |
| Board Manufacturer Date | 2021/02/18 |
| Board Serial Number | 123456789ABCDE |
| Board Part Number | 123456789ABCDE |

Table 10: FRU data format

The firmware offers a tool to write above information into the FRU.

The firmware provides a Linux utility, AdFruWrite\_F, for COM-HPC Server Carrier to write required FRU information, it reads a configuration file with particular format and then verifies it. If the configuration is verified as a valid format, the utility and then writes the data into the EEPROM. Please see User Manual for details.

Traditionally, it is easy to use utilities such as IPMITOOL to write FRU information into the EEPROM. To prevent the FRU information from being over-written accidentally and to guarantee the factory default FRU information is un-modified, COM-HPC Server Carrier disable IPMI’s writing privilege to the EEPROM in the firmware level. Only the utility AdFruWrite\_F is authorized to write the FRU.

Since the FRU information is only allowed to be written by the AdFruWrite\_F, using IPMITOOL to write FRU would result in an error.

### Access FRU data

#### Access FRU via IPMI

To read FRU information from IPMI, following command could be issued.

*ipmitool -I lanplus -H <BMC\_IP> -U admin -P admin fru*

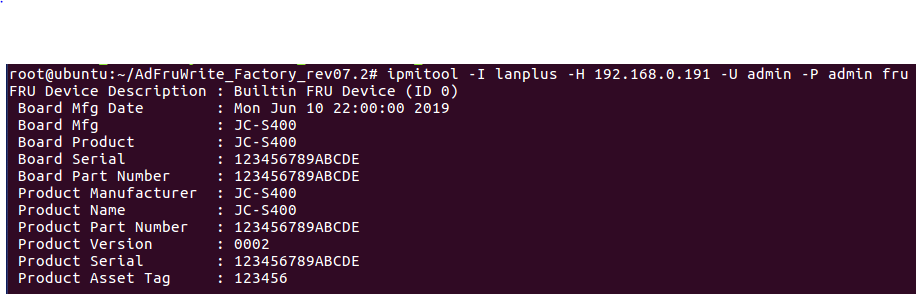


Figure 23: FRU information via IPMI command

#### Access FRU via WebUI

The FRU information can also be displayed via accessing the WebUI.

*“Control & Maintenance” -> “FRU Information”*

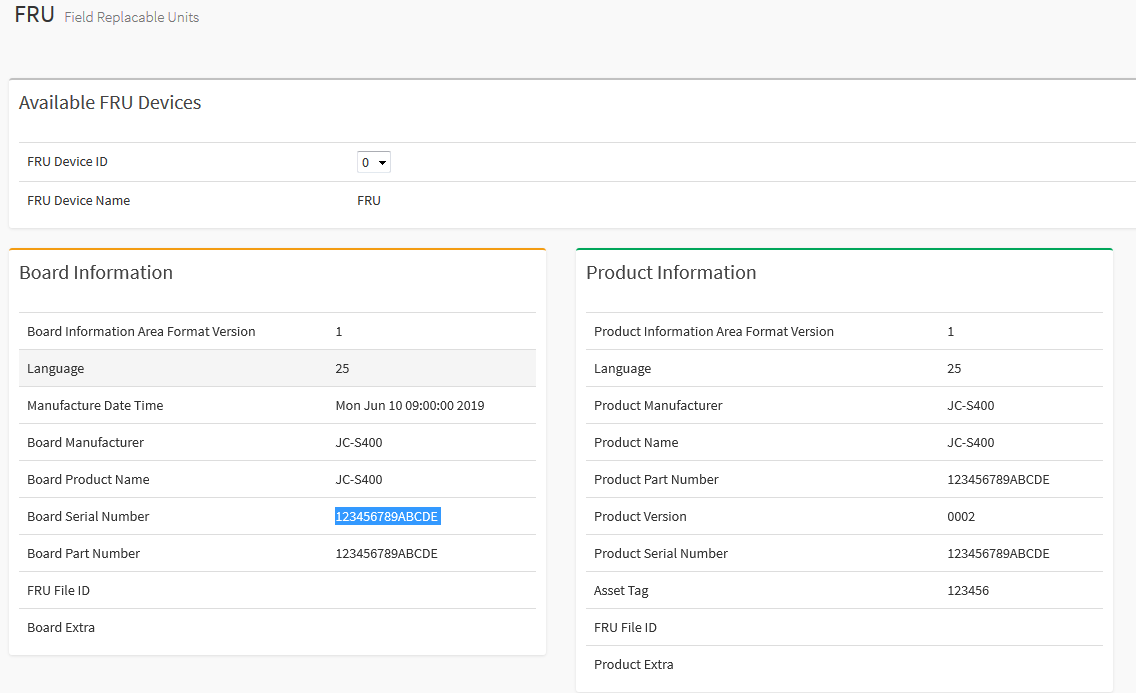


Figure 24: FRU information via WebUI

## System Event Log

The most important mission of BMC is 1. Monitor the sensors and 2. Monitor the events. The former is described in the previous chapter and the latter is achieved by providing the SEL. The SEL is defined by the IPMI spec. and the firmware follows the specification to provide below event logs.

The SEL is allocated in a 64KB non-volatile EEPROM which allows more than several thousands of log entries. BMC could access the EEPROM via I2C interface.

There are two supported storage policies for SEL logs, the linear and the circular, the firmware supports the latter by default. The circular policy is also called “SEL rotation”, once the storage reaches the limitation, the latest one would overwrite the oldest one as described below:

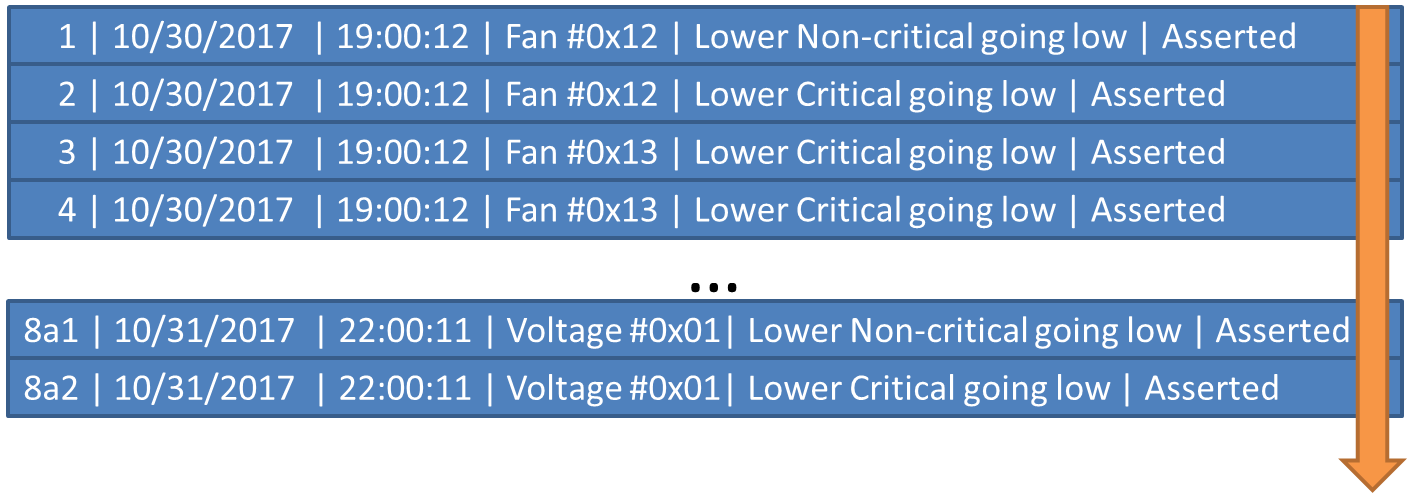


Figure 25: The order of writing SEL logs when the size of SEL area is not full

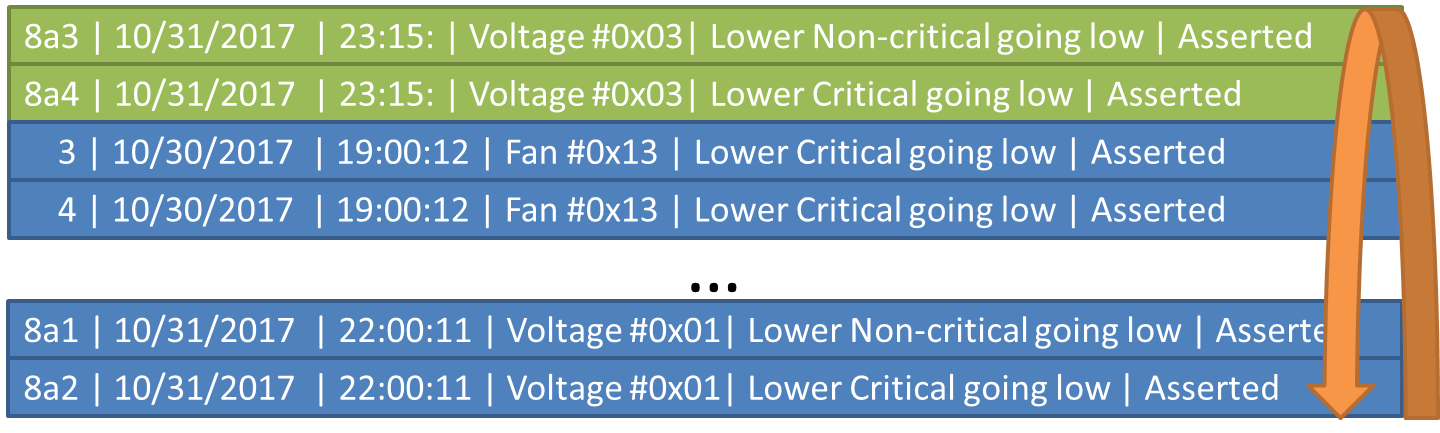


Figure 26: Log rotation when SEL area is full, the latest one would overwrite the oldest one.

Since it is a non-volatile storage, it would not be erased after a power cycle. To filter the unnecessary logs, it is able to enable/disable selected SEL type. For monitoring, the SEL log could be accessed remotely including reading and deleting.

The format of the SEL log is as below:

|  |  |  |
| --- | --- | --- |
| Bytes | Field | Description |
| 1-2 | Record ID | ID used for SEL Record access. |
| 3 | Record Type | [7:0] - Records |
| 4-7 | Timestamp | Time when event was logged. |
| 8-9 | Generator ID | RqSA & LUN if event was generated from IPMB. Software ID if event was generated from system software. |
| 10 | EvM Rev | Event Message from version. |
| 11 | Sensor Type | Sensor Type Code for sensor that generated the event. |
| 12 | Sensor # | Number of Sensor that generated the event |
| 13 | Event Dir | Event Type | Event Dir/Event Type. |
| 14 | Event Data 1 | Refer Event Request Message Event Data Field Contents. IPMI2.0 specification. |
| 15 | Event Data 2 | Refer Event Request Message Event Data Field Contents. IPMI2.0 specification. |
| 16 | Event Data 3 | Refer Event Request Message Event Data Field Contents. IPMI2.0 specification. |

Table 11: The System Event Record

The format listed above is provided by the IPMI Spec., please check IPMI Spec for details. There are three ways for checking the SEL logs described below, please noticed that all SEL is synchronized since they are all from the same source, once the SEL is cleared by one of the interface, it is no longer accessible via other interfaces.

IPMI:

The following IPMITOOL can be used to check the SEL.

*ipmitool -I lanplus -H <BMC\_IP> -U admin -P admin sel list*

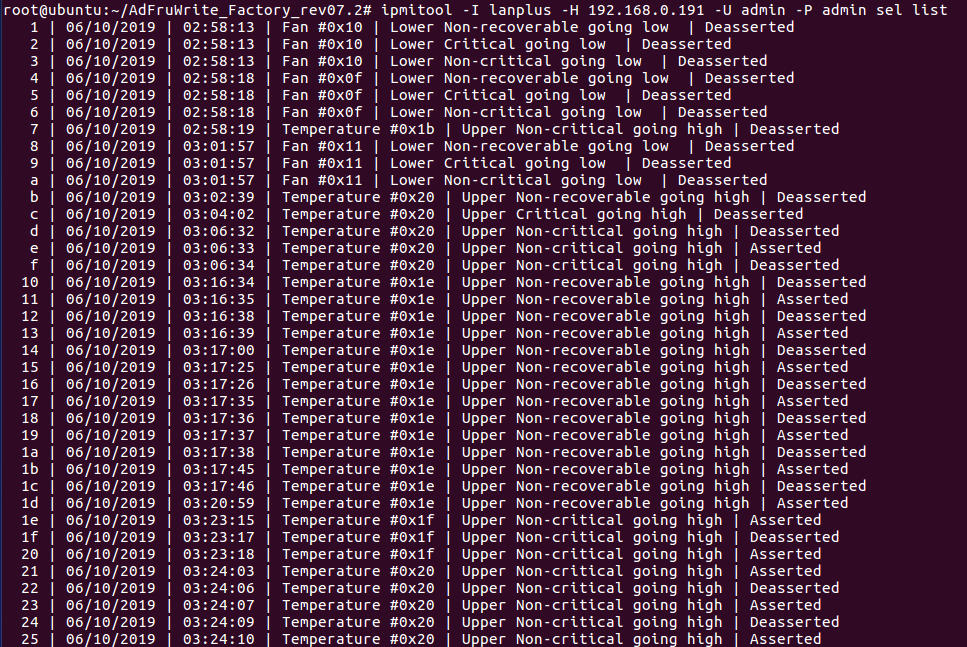


Figure 27: SEL via IPMI command

Use following command to clear the SEL

*ipmitool -I lanplus -H 172.16.6.118 -U admin -P admin sel clear*

WebUI:

Following page provides the ability to read, export, and clear SEL.

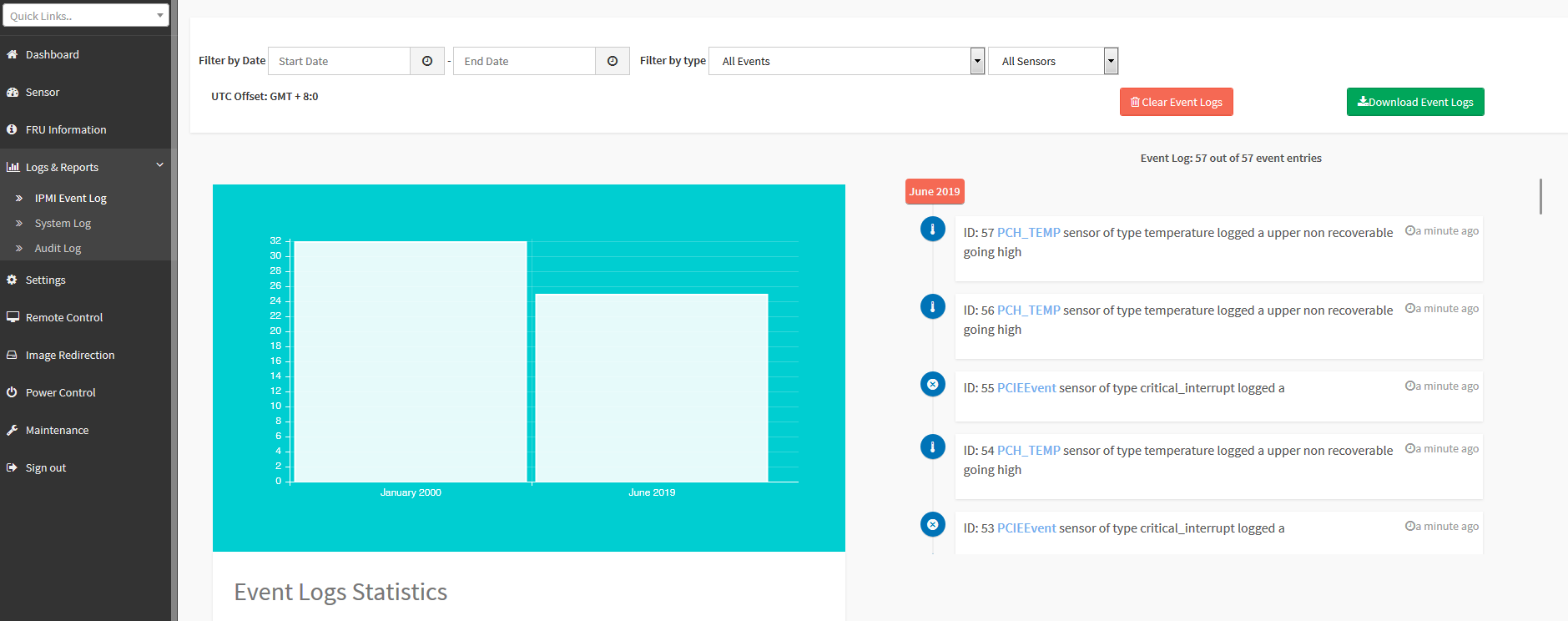


Figure 28: SEL via WebUI

## Firmware Upgrade

### Supported Tools

Following table shows the supported tools which can perform firmware upgrade.

|  |  |  |
| --- | --- | --- |
| Target/Tools | YafuFlash | WebUI |
| BMC | ✓ | ✓ |

Table 12: Supported tools for firmware update

To upgrade the firmware remotely, firmware provides two interfaces for the purpose. The WebUI provides the graphical interface for users to upgrade firmware via several mouse clicks, the YafuFlash provides console command for users to upgrade firmware via Windows command prompt and Linux terminal. BMC could be updated on-the-fly when it is running.

#### Yafu(Yet Another Firmware Update) Flash

YafuFlash is a command line tool which runs on Linux and Windows to perform the firmware upgrade remotely.

A simple YafuFlash command could be done as below:

*Yafuflash -nw -ip <BMC\_IP> -u admin -p admin -d 1 <image path>*

|  |  |
| --- | --- |
| Parameters | Suported |
| -nw | Upgrade from network medium |
| -ip | Option to enter IP when using network medium |
| -host | Option to enter host name when using network medium |
| -u | Option to enter Usernamem when using network medium |
| -p | Option to enter Passwrod when using network medium |
| -port | Option to enter Port Number |

Table 13: Parameters for using YafuFlash

The detail procedure of upgrading firmware with YafuFlash is described in the following flow chart:

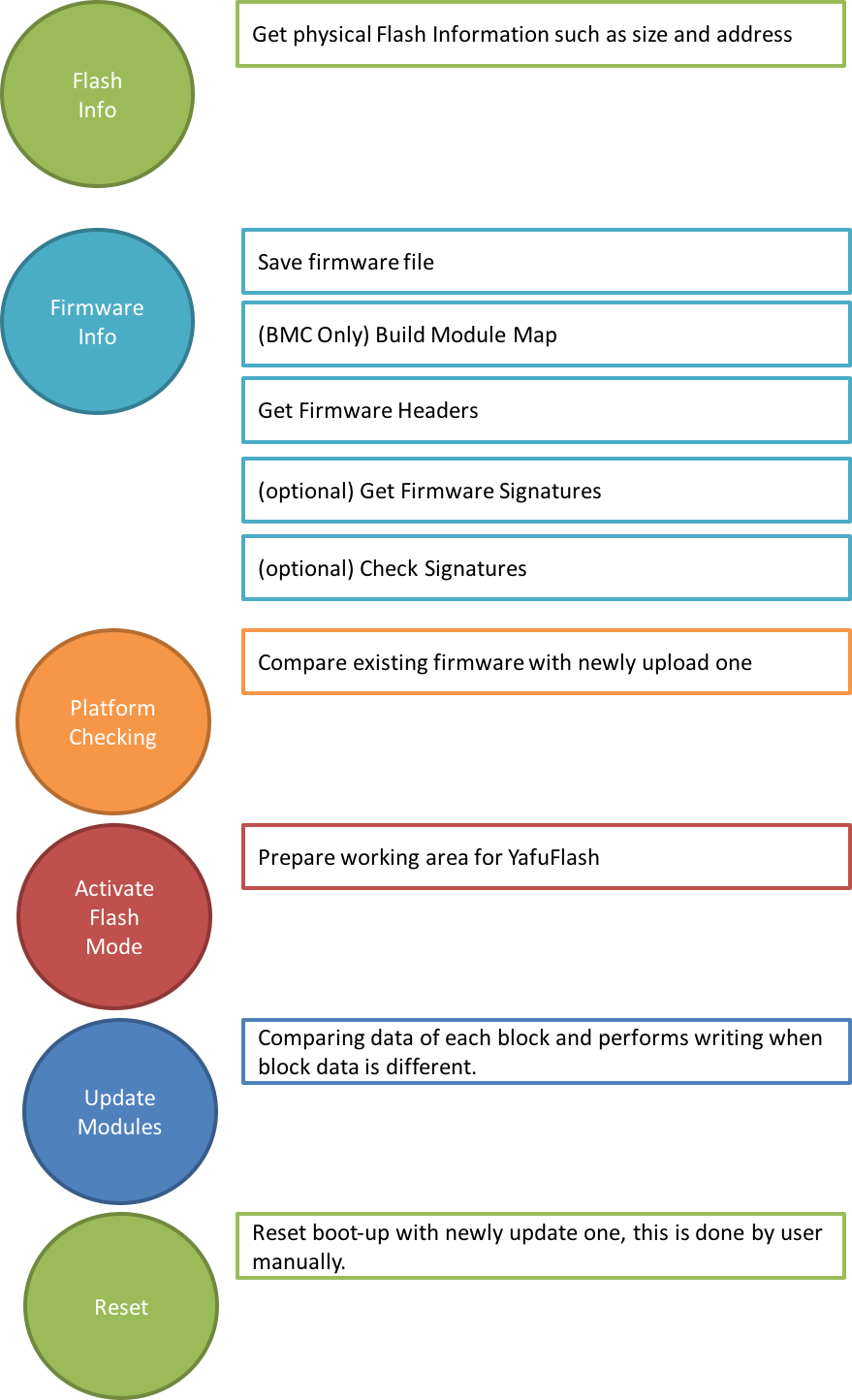


Figure 29: The procedure of upgrading firmware with YafuFlash

YafuFlash is executed as interactive mode and users are able to choose the preserved configuration during the execution.

#### WebUI

To update firmware via WebUI. Following step should be performed.

Go to the “maintenance tab” and choose “Dual Firmware Update”

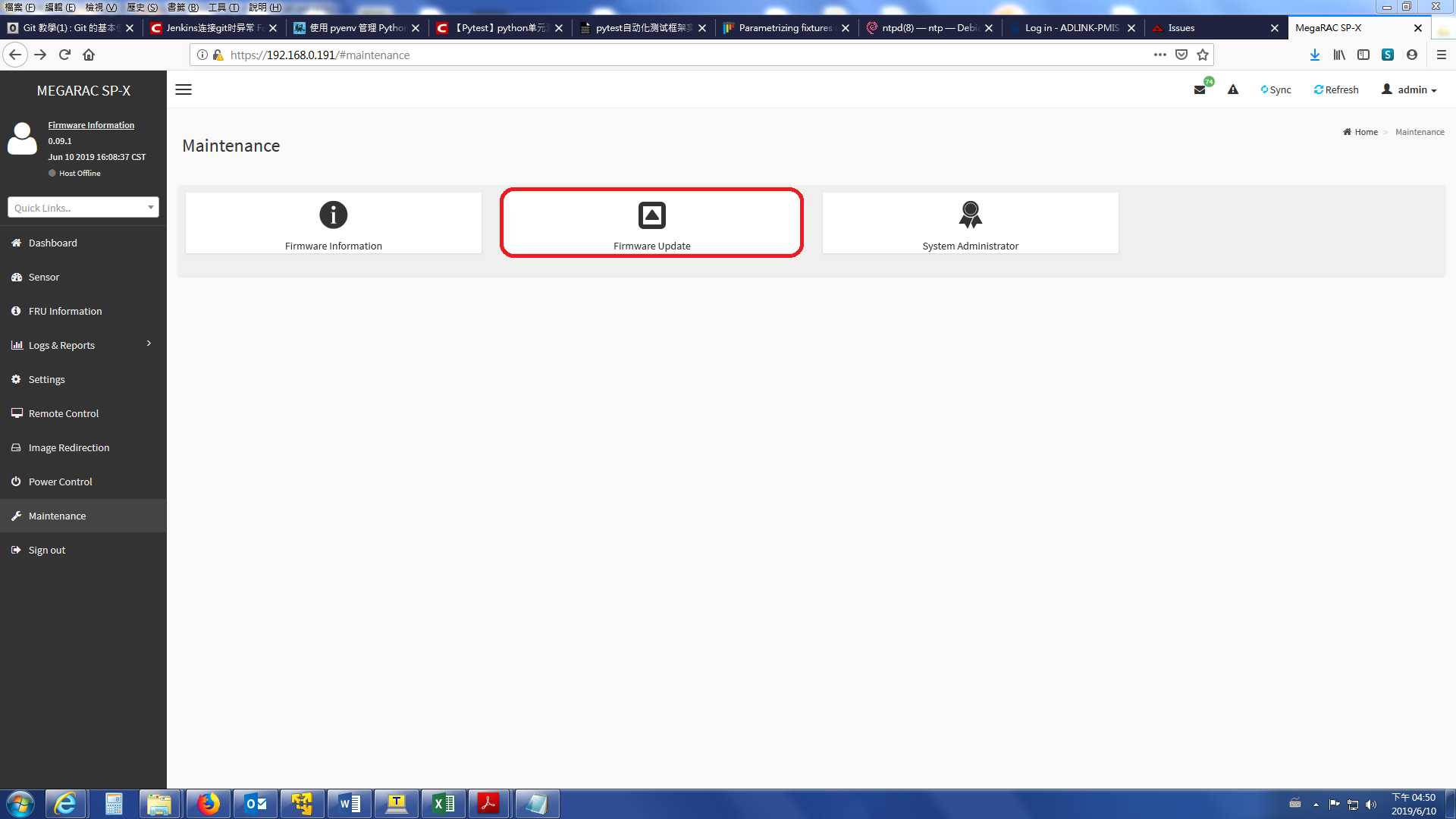


Figure 30: The procedure of upgrading firmware via WebUI

Edit the preserve configuration, upload the image and then start to upgrade.

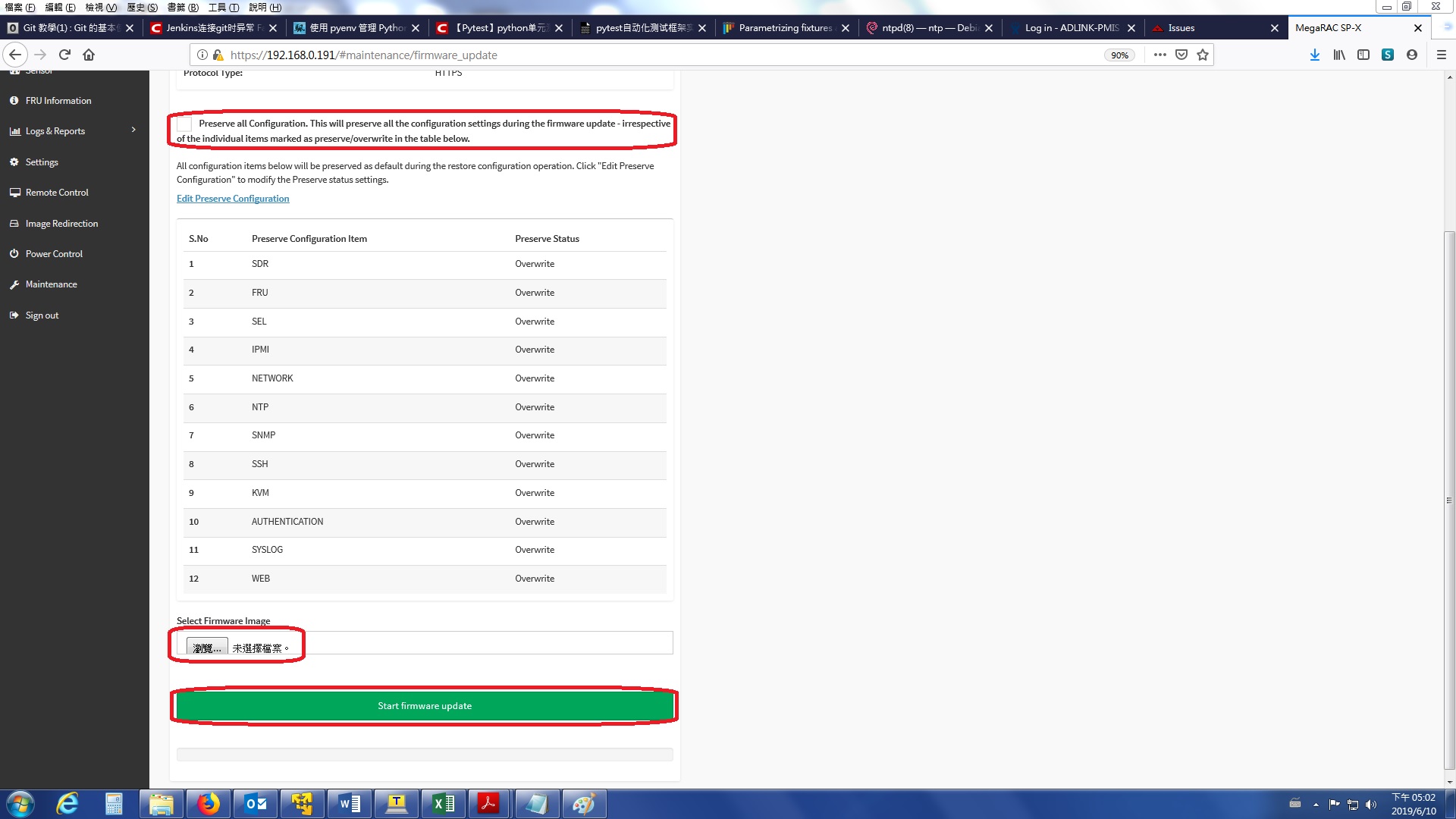


Figure 31: The procedure of upgrading firmware via WebUI

## Audit Logs

The audit logs give the complete login/logout details. Following interface would be logged into the system audit log at "/var/log/audit.log".

|  |  |
| --- | --- |
| Interface | Action |
| WEBUI | Login Successful  Login Failed  Logout  Auto Logout (timeout) |
| SSH | Login Successful  Login Failed  Logout  Auto Logout (timeout) |
| KVM | Login Successful  Logout |

Table 14: Actions of WebUI, SSH, and KVM interfaces

Following information would be recorded

|  |  |
| --- | --- |
| Field | Description |
| Timestamp | The current timestamp which is aligned with the current system time. |
| Target | Usually it is localhost |
| Service Name | SSHD, KVM or https |
| Severity | WARNING or INFO |
| Type | Login  Login Failed  Logout  Timeout |
| IP | IP address |
| Username | The username |

Table 15: The content of audit logs

## Reset to factory default

For debug purpose, the firmware provides several ways to restore all or partial settings into factory default. Please use it for debug only and do not apply to a working machine.

### Reset BMC to default

Following IPMI raw command could be used for restoring the factory default values.

*ipmitool -I lanplus -H <BMC\_IP> -U admin -P admin raw 0x32 0x66*

This command won’t affect to the MAC address.

## BMC Health Monitoring

To make sure the BMC works correctly, the BMC has several daemons to monitor the health state of the BMC itself including the process monitoring, free memory monitoring and disk space monitoring.

# Appendix A Error Log Collection

When error occurs, for example, the firmware upgrading fails. It is able to collect useful information by retrieving system logs.

## Enable the Remote Logs

The remote log could be enabled by setting up the remote log server on the Web Server.

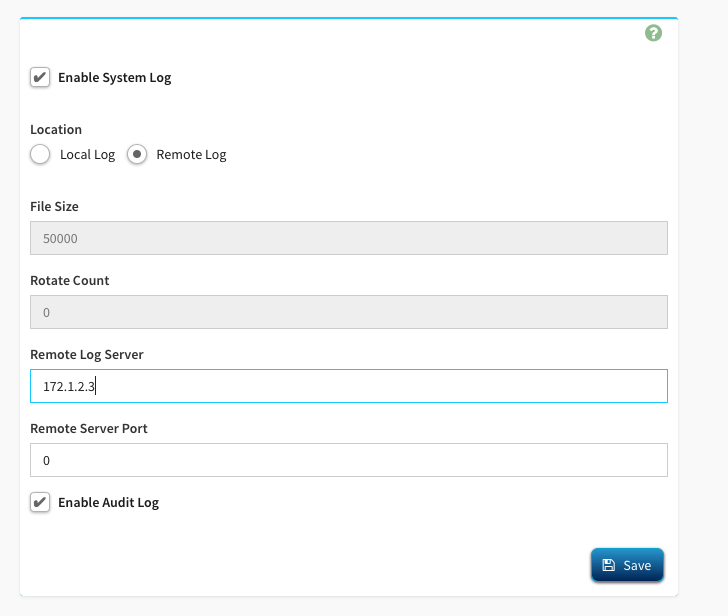


Figure 32: The remote log could be set by clicking following icons with the order “Settings”->“Log Settings”->“Advanced Log Settings”.

The received log entries from the BMC remotely, a syslogd service should be enabled on the machine. When error occurs, all possible logs in the /var/log should be sent to the server provider for debugging. The location of remote logs might not be located in the /var/log, please check the settings of the syslogd on the remote server to find out the correct location of logs.

## Retrieve logs on the BMC

If the remote log service is not enabled, users are still able to send logs to the provider. If the BMC is still accessible via the SSH connection or the debug console, users can export all logs by following command.

*tftp -pr <filename> <remote server>*

The tftp command on the BMC does not support transferring files with the wild card, users should manually transfer all file one by one.

## Supported command table

|  |  |
| --- | --- |
| IPM Device “Global” Commands | Comment |
| Get Device ID | N/A |
| Cold Reset | N/A |
| Warm Reset | N/A |
| Get Self Test Results | N/A |
| Set ACPI Power State Command | N/A |
| Get ACPI Power State Command | N/A |
| Get Device GUID | N/A |
| Get NetFn Suppot | N/A |
| Get Command Support | N/A |
| Get Command Sub-function Support | N/A |
| Get Configurable Commands | N/A |
| Get Configurable Sub-functions | N/A |
| Set Command Enables | The response returns unknown means attempt to enable an unsupported or un-configurable command. |
| Get Command Enables | N/A |
| BMC Watchdog Timer Commands |  |
| Reset Watchdog Timer | N/A |
| Set Watchdog Timer | N/A |
| Get Watchdog Timer | N/A |
| IPMI Messaging Support Commands |  |
| Set BMC Global Enables | N/A |
| Get BMC Global Enables | N/A |
| Clear Message Flags | N/A |
| Get Message Flags | N/A |
| Get Message | N/A |
| Send Message | N/A |
| Get System GUID | N/A |
| Get Channel Authentication Capabilities | N/A |
| Get Session Challenge | N/A |
| Activate Session | N/A |
| Set Session Privilege Level | N/A |
| Close Session | N/A |
| Get Session Info | N/A |
| Set Channel Access | N/A |
| Get Channel Access | N/A |
| Get Channel Info | N/A |
| Set User Access | N/A |
| Get User Access | N/A |
| Set User Name | N/A |
| Get User Name | N/A |
| Set User Password | N/A |
| Activate Payload | N/A |
| Deactivate Payload | N/A |
| Set User Payload Access | N/A |
| Get User Payload Access | N/A |
| Master Write-Read | N/A |
| Get Channel Cipher Suites | N/A |
| Chassis Commands |  |
| Get Chassis Capabilities | N/A |
| Get Chassis Status | N/A |
| Chassis Control | N/A |
| Chassis Identify | N/A |
| Set Chassis Capabilities | N/A |
| Set Power Restore Policy | N/A |
| Get System Restart Cause | N/A |
| Set System Boot Options | N/A |
| Get System Boot Options | N/A |
| Get POH Counter | N/A |
| Event Commands |  |
| Set Event Receiver | N/A |
| Get Event Receiver | N/A |
| Platform Event Message | N/A |
| PEF and Alerting Commands |  |
| Get PEF Capabilities | N/A |
| Set PEF Configuration Parameters | N/A |
| Get PEF Configuration Parameters | N/A |
| Set Last Processed Event ID | N/A |
| Get Last Processed Event ID | N/A |
| Sensor Device Commands |  |
| Get Device SDR Info | N/A |
| Get Device SDR | N/A |
| Reserve Device SDR Repository | N/A |
| Get Sensor Reading Factors | N/A |
| Set Sensor Hysteresis | N/A |
| Get Sensor Hysteresis | N/A |
| Set Sensor Thresholds | N/A |
| Get Sensor Thresholds | N/A |
| Set Sensor Event Enable | N/A |
| Get Sensor Event Enable | N/A |
| Re-arm Sesnor Event | N/A |
| Get Sensor Event Status | N/A |
| Get Sesnor Reading | N/A |
| FRU Inventory Device Commands |  |
| Get FRU Inventory Area Info | N/A |
| Read FRU Data | N/A |
| Write FRU Data | N/A |
| SDR Repository Device Command |  |
| Get SDR Repository Info | N/A |
| Reserve SDR Repository | N/A |
| Get SDR | N/A |
| Partial Add SDR | N/A |
| SEL Device Commands |  |
| Get SEL Info | N/A |
| Get SEL Allocation Info | N/A |
| Get SEL Entry | N/A |
| Add SEL Entry | N/A |
| Partial Add SEL Entry | N/A |
| Delete SEL Entry | N/A |
| Clear SEL | N/A |
| Get SEL Time | N/A |
| Set SEL Time | N/A |
| IPMI LAN Commands |  |
| Set LAN Configuration Parameters | N/A |
| Get LAN Configuration Parameters | N/A |
| SOL Configuration Commands (IPMI v2.0) |  |
| Set SOL Configuration Parameters | N/A |
| Get SOL Configuration Parameters | N/A |