MegaRAC Community Edition™

[Getting Started Guide]

Contents

[1 Introduction 2](#_Toc190896353)

[2 Prerequisites 2](#_Toc190896354)

[3 Download and Build 2](#_Toc190896355)

[3.1 AST2600EVB 2](#_Toc190896356)

[3.2 AST2700EVB 2](#_Toc190896357)

[3.3 Nuvoton Arbel: 4](#_Toc190896358)

[4 Firmware Flashing Method 4](#_Toc190896359)

[4.1 Dediprog 4](#_Toc190896360)

[4.2 TFTP Flash 4](#_Toc190896361)

[4.3 Web Flash 5](#_Toc190896362)

[4.4 Phosphor IPMI Flash 6](#_Toc190896363)

[5 Network Configurations 6](#_Toc190896364)

[5.1 U-boot 6](#_Toc190896365)

[5.2 Linux 6](#_Toc190896366)

[5.3 Using ipmi Configure Network Settings 6](#_Toc190896367)

[5.4 Configure Network through Web UI 6](#_Toc190896368)

[6 BMC Debug Console 11](#_Toc190896369)

[6.1 Minicom Configuration 11](#_Toc190896370)

[6.2 Uboot 12](#_Toc190896371)

[6.3 Linux 12](#_Toc190896372)

[6.4 Uboot - Booting to BMC Linux 13](#_Toc190896373)

[7 Supported Features 14](#_Toc190896374)

[7.1 Generic Features 14](#_Toc190896375)

[7.2 SOC Specific 15](#_Toc190896376)

[7.2.1 AST2600 15](#_Toc190896377)

[7.2.2 AST2700 15](#_Toc190896378)

[7.2.3 Arbel 15](#_Toc190896379)

# Introduction

This document provides information about ASPEED and Nuvoton EVB board details, build and flash using MegaRAC Community Edition™ in the OCP.

CE is our production-ready, opensource-based BMC remote management solution developed on core technology from the Linux Foundation OpenBMC project. The solution integrates state-of-the-art technologies across multiple platforms and system-on-chip (SoC) processors, enabling it to be implemented on a wide range of industry use cases. And can be enhanced with proprietary AMI IP Expansion Packs (EPs) and Silicon packs based on the proven MegaRAC firmware stack, trusted for years by leading OEMs and ODMs for its robust, secure remote server management.

For more details: https://www.ami.com/megarac/

# Prerequisites

* The user should **clean** **install** Ubuntu 18.04 or later version as build machine.
* The OpenBMC Linux Foundation (LF) source code was maintained under Git, the user must install the Git on the build machine before fetching the source code.
* Open Linux console on the build machine and run the below command.

$ sudo apt-get install -y git build-essential libsdl1.2-dev texinfo gawk chrpath diffstat

* Install python and pip

sudo apt-get install python

sudo apt-get install python-pip

* Install git python and yaml parser

$ sudo pip install gitpython

$ sudo pip install pyyaml

# Download and Build

## AST2600EVB

* git clone https://github.com/ocp-hm-openbmc-opf-ami/openbmc openbmc; cd openbmc
* git clone https://github.com/ocp-hm-openbmc-opf-ami/meta-common
* git clone https://github.com/ocp-hm-openbmc-opf-ami/meta-ami
* meta-ami/github-gitlab-url.sh
* TEMPLATECONF=meta-ami/meta-evb/meta-evb-aspeed/meta-evb-ast2600/conf/templates/default . openbmc-env
* bitbake obmc-phosphor-image

## AST2700EVB

* git clone https://github.com/ocp-hm-openbmc-opf-ami/openbmc openbmc; cd openbmc
* git clone https://github.com/ocp-hm-openbmc-opf-ami/meta-common
* git clone https://github.com/ocp-hm-openbmc-opf-ami/meta-ami
* meta-ami/github-gitlab-url.sh
* TEMPLATECONF=meta-ami/meta-evb/meta-evb-aspeed/meta-evb-ast2700/meta-ast2700/conf/templates/default . openbmc-env
* bitbake obmc-phosphor-image

## Nuvoton Arbel:

* git clone https://github.com/ocp-hm-openbmc-opf-ami/openbmc openbmc; cd openbmc
* git clone https://github.com/ocp-hm-openbmc-opf-ami/meta-common
* git clone <https://github.com/ocp-hm-openbmc-opf-ami/meta-ami>
* meta-ami/github-gitlab-url.sh
* TEMPLATECONF=meta-ami/meta-evb/meta-evb-nuvoton/meta-evb-npcm845/conf/templates/default . openbmc-env
* bitbake obmc-phosphor-image

# Firmware Flashing Method

**Note:**

For Dediprog and TFTP “xxx**.**mtd” image should be used.

For WEB Flash “xxx.mtd.tar” or “xxx.mtd.tar.all” image should be used.

“xxx” means image name.

## Dediprog

**Hardware Tool**: DediProg\* SF600 or similar tool

**Software**: DediProg Software latest version

**Following are the steps to perform using Dediprog Software.**

1. Take the BMC EEPROM from the BMC socket and place it in DediProg programming socket.
2. Launch the DediProg software and select SPI Device.
3. Upload the firmware image file.
4. Click **Batch** and then the flashing starts.

## TFTP Flash

1. Set the environment value:

Set Tftp server ip and ethaddr and get dynamic ip address

**Example:**

setenv serverip <IP Address>

setenv ethaddr <MAC>

setenv eth1addr <MAC>

setenv ethact <ethaddr>

dhcp

1. Download image from tftp server and flash (ASPEED):

**Example:**

tftp 0x83000000 <image>

sf probe 0

sf update 0x83000000 0x00 <image size in bytes>

reset

1. Download image from tftp server and flash (Nuvoton Arbel):

**Example:**

tftp 0x62080000 <image>

sf probe 0:0

sf update 0x62080000 0x00 <image size in bytes>

reset

## Web Flash

User can navigate to **Operations** à **Firmware** page.

Given below is the screenshot for **Firmware** page.

A screenshot of a computer

Description automatically generated

The **Firmware** page contains the information about following:

* **BMC** - It displays the version of **Running image** and **Backup image.**
* **Host -** It displays the version of **Running** **image** and **Backup** **image**.

**Update Firmware**

1. Click **Add file** and locate the firmware file to upload based on the firmware image.
2. Click **Start Update** and bmc will start to upload the firmware and after some time UI gets logout and BMC will reboot it.
3. 5-10 minutes for the BMC to flash and reset.

## Phosphor IPMI Flash

Phosphor IPMI Flash is a command line tool used to perform the in-band update of BMC firmware using OEM IPMI blob transport. Refer [Phosphor IPMI Flash Tool](#_Phosphor_IPMI_Flash) section for more details and usage information of the tool.

# Network Configurations

## U-boot

**ast>setenv ethaddr xx:xx:xx:xx:xx:xx**

**ast>setenv eth1addr xx:xx:xx:xx:xx:xx**

ast>dhcp

After we get BMC IP, save the environment variables by issuing **save** command.

ast>save

To know more information about booting from uboot to BMC Linux, kindly refer “BMC Debug Console: uboot – booting to Linux”.

## Linux

**~ # ifconfig eth0 down**

**~ # ifconfig eth0 hw ether xx:xx:xx:xx:xx:xx**

**~ # systemctl restart systemd-networkd.service**

## Using ipmi Configure Network Settings

**# ipmitool -H <BMCIP> -U root -P 0penBmc -I lanplus lan set <channel number> macaddr <MAC>**

**# ipmitool -H <BMCIP> -U root -P 0penBmc -I lanplus lan set <channel number> ipsrc <static/dhcp>**

**# ipmitool -H <BMCIP> -U root -P 0penBmc -I lanplus lan set <channel number> ipaddr <IP>**

**# ipmitool -H <BMCIP> -U root -P 0penBmc -I lanplus lan set <channel number> netmask <netmask>**

## Configure Network through Web UI

User can navigate to **Settings** > **Network** page. Given below is the screenshot for **Network** page.

A screenshot of a computer

Description automatically generated

The **Network** page contains the information about following:

Network page helps to configure **BMC network settings.**

**Network Settings**

* **Hostname** – Click Edit hostname icon to add Hostname.
* **Use domain name -** Enableto use domain name**.**
* **Use DNS servers -** Enableto use DNS servers**.**
* **Use NTP servers -** Enableto useNTP servers**.**

User can Configurethe BMC network setting through interfaces **eth0, usb0** and **usb1**. By clicking the respective interfaces, it allows the user to perform the following operations:

It shows the Link status and speed (mbps).

* **Interface settings:**
* **FQDN –** It shows domain name.
* **MAC address** – Click **Edit Mac address** icon to add MAC address.

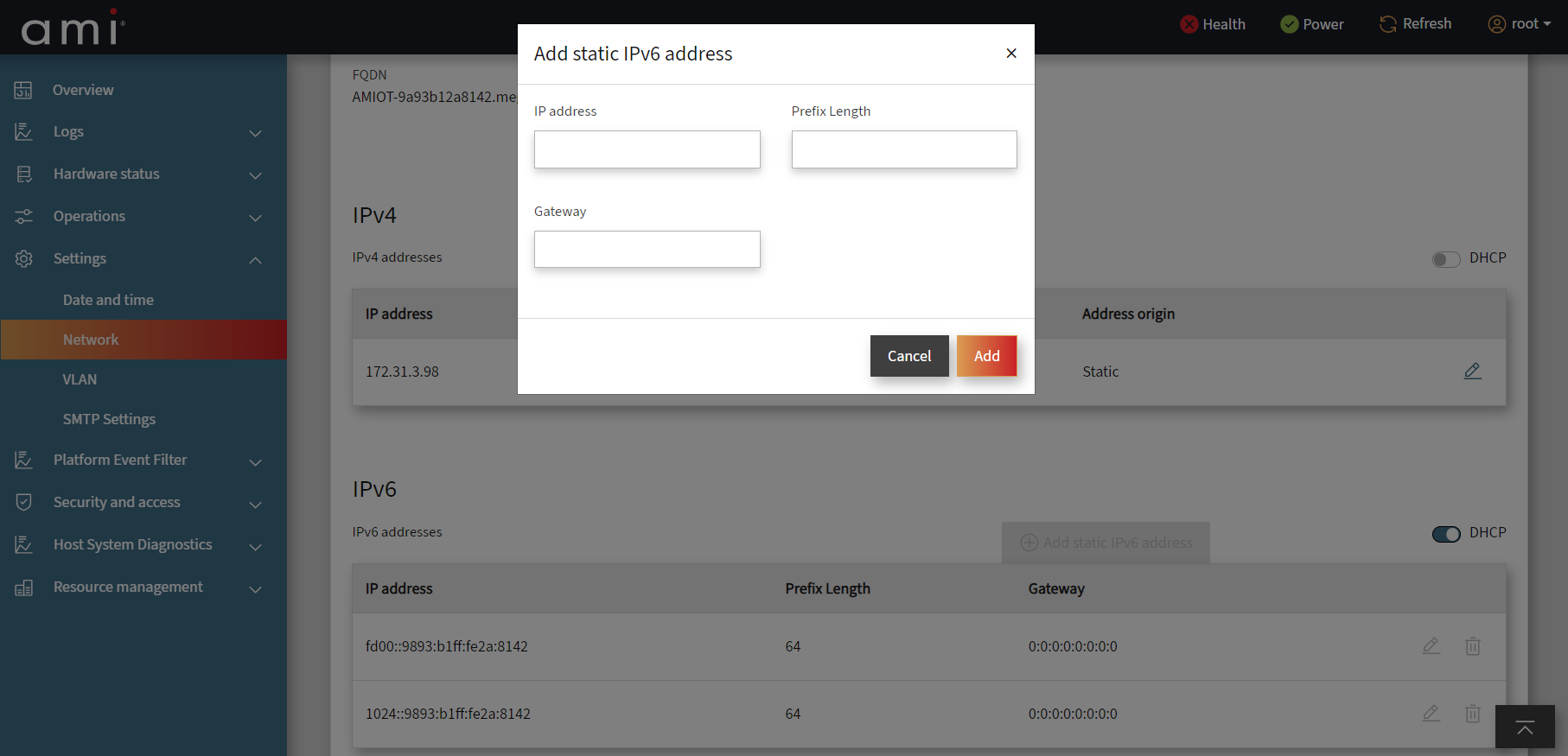
IPv4 addresses – It displays information like IP address, Gateway, Subnet mask and Address origin. And then the user can add static IPv4 address as displayed in the below screenshot.

A screenshot of a computer

Description automatically generated

* **Add static IPV4 address** – Toggle **DHCP** in **Network** pageor **c**lick **Add static IPv4 address** and enter **IP address**, **Gateway,** and **Subnet mask.**
* **Add -** Click **Add** toadd the static IPV4 address.
* Address can be deleted or edited using **Delete** or **Edit** icon in the right corner.

IPv6 addresses – It displays the information like **IP address**, **Prefix Length** and. **Gateway**. And user can a**dd static IPv6 address**.



* **Add static IPv6 address** - Toggle **DHCP** in **Network** pageor **c**lick **Add static IPv6 address** and enter **IP address**, **Prefix Length** and **Gateway.**
* **Add -** Click **Add** toadd the static IPv6 address.
* Added can be deleted or edited using **Delete** or **Edit** icon in the right corner.

Static DNS - Click **Add IP address**, enter the **Static DNS** and click **ADD** to successfully add the Static DNS.

A screenshot of a computer

Description automatically generated

***Note:***

*Open BMC will support only uni-cast and locally administrated MAC address.*

**Range:**

There are 4 ranges of Locally Administered Address Ranges that can be used on a local network:

x2-xx-xx-xx-xx-xx

x6-xx-xx-xx-xx-xx

xA-xx-xx-xx-xx-xx

xE-xx-xx-xx-xx-xx

Where x can be replaced by any hex value

**Establish WEB**

BMC FW supports a web server based on the bmcweb implementation and webui-vue front end. This can be accessed using the BMC IP address.

The fields are explained as follows:

**Username**: Enter your username in this field.

**Password**: Enter your password in this field.

**Log in**: After entering the required credentials, click **Log in** to login to MegaRAC community edition GUI.

Once logged successfully, you will get the BMC Overview page as shown below.

A screenshot of a computer

Description automatically generated

Quick buttons at the top right panel in **Overview** page:

1. **Health** - Click **Health** will redirect to **Event Logs** page.
2. **Power** - Click **Power** will redirect to **Server power operations** page.
3. **Refresh** - It helps to refresh the current window.
4. **Root -** Click **Root** **->** navigate to **Profile Setting** page to find Profile information, Change password, Time zone. and **logout** which will come back to the Login page**.**

# BMC Debug Console

## Minicom Configuration

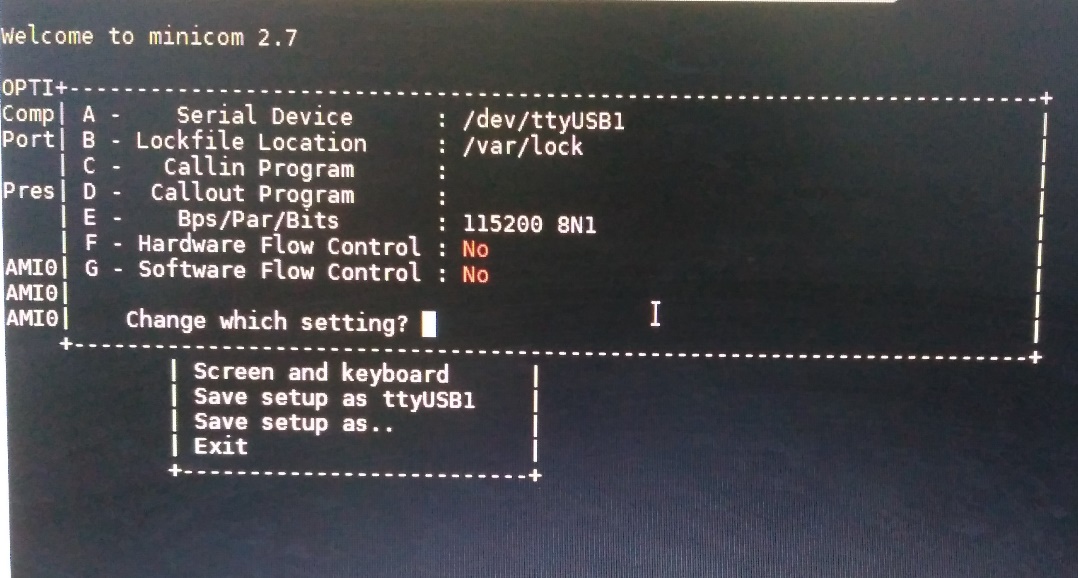
In order to get correct BMC Debug console, user should configure following, apart from ‘TTY to USB serial cable’ connections from server to minicom machine. Here minicom machine can be any machine where user can get access to BMC debug console.

**When using Ubuntu as minicom machine,**

1. Open a terminal and type “dmesg | grep tty” which gives you in which tty device, the TTY to USB cable active.
2. Install ‘minicom’ if not already available. By giving “apt-get install minicom”. If you already have minicom installed, type “minicom ttyUSB<x>” in order to get the debug console.
3. Press ‘Ctrl + a + o’ which will show ‘configuration’ grid. From there, traverse down to “Serial Port Setup” hit enter. Following parameters are observed.

* Press ‘A’ – and set ‘Serial Device’ with ‘/dev/ttyUSB<x>’
* Press ‘E’ – and choose 115200 8N1 as ‘Bps’
* Press ‘G’ – choose ‘Hardware Flow Control’ as ‘No’
* Press ‘H’ – choose ‘Software Flow Control’ as ‘No’

After setting all the parameters, it should look like below.



1. Press ‘Enter’ after setting all the parameters, which will take you back to ‘configuration’ grid, traverse down to ‘Save Setup as ttyUSB<x>’ and save your configurations. And then ‘exit’
2. Now close the minicom session by giving ‘Ctrl + a + q’ and reopen the session. After that users should be having decent BMC Debug Console.

**When using Windows as Minicom Machine,**

1. User needs to install drivers for TTY to USB cable for the first time. Following link can be used for getting drivers.

<https://learn.adafruit.com/adafruits-raspberry-pi-lesson-5-using-a-console-cable/software-installation-windows>

1. After successful installation of drivers, plug in the USB to minicom machine, and go to, ‘Device Manager’ -> ‘Serial COM ports’ and find out on which COM Port, the device is populated.
2. Open Putty terminal chose the last section ‘Serial’ from ‘Category’. On right side, user gets following options

‘Serial line to connect to’ -> mention whatever the COM port observed from step 2.

‘Speed’ -> set as ‘115200’

‘Data bits’ -> set as ‘8’

‘Stop bits’ -> set as ‘1’

‘Parity’-> set as ‘None’

‘Flow Control’ ->set as ‘XON/XOFF’

1. Hit ‘Open’ at the end to start accessing the BMC Debug Console.

## Uboot

By default, auto delay is not configured when debug-tweaks is disabled. User has to enable the debug-tweaks to stop the BMC in –boot.

Uncomment EXTRA\_IMAGE\_FEATURES += "debug-tweaks" in build/conf/local.conf

Users can get to uboot prompt via 2 ways, mainly during AC Power ON or by issuing ‘reboot’ BMC.

1. In order to get uboot prompt upon AC power on, user should press and hold the “Esc or Del” key as soon as AC power is applied, until redirected to uboot prompt.

Sometimes, user may not succeed in getting to uboot as, ‘bootdelay’ environment variable in uboot is ‘0’ initially. Which means as soon as AC power is applied to board, BMC starts booting to Linux, within ‘0’ seconds of delay or in other terms no delay. In such cases, the user should wait till BMC boots to Linux completely. And follow next method mentioned below.

1. Please refer “Section 12.3 BMC Debug Console: BMC Linux shell prompt” to get access to BMC Linux prompt.

From BMC Linux shell prompt, issue ‘reboot’ command and press and hold the “Esc or Del” key until redirected to uboot prompt.

User may optionally set ‘bootdelay’ uboot environment variable with desired delay in seconds.

This defines how much seconds BMC booting should be delayed upon reboot or next AC cycle.

ast> set bootdelay 3; save;

## Linux

Upon BMC successfully booting to Linux, the user will be asked for username and password at ‘login’ to give access to BMC Linux shell prompt. Users can use the credentials below to login.

Username: root

Password: 0penBmc

## Uboot - Booting to BMC Linux

After completing required operations in uboot prompt, user should issue **reset** command to continue booting to BMC Linux.

ast**>reset or bootm <kernel load address>**

Environmental Variables for UBoot

The basic environmental variables for uboot are:

* **Serverip:** The server ip contains the ip of the Linux or Windows system in which tftp server is present and running. The command to set the server ip is below.

**setenv serverip <IP Address>**

* **Ethaddr and Eth1addr:** Both variables are used for setting the mac address for the board. To set the mac address, the commands are

**setenv ethaddr <MAC ADDRESS>**

**setenv eth1addr <MAC ADDRESS>**

* **Ethact:** This variable is used to set the active network interface for the board. The command is given below.

**For eth0 – setenv ethact ethaddr**

**For eth1 – setenv ethact eth1addr**

**For eth2 – setenv ethact eth2addr**

**For eth3 – setenv ethact eth3addr**

# Supported Features

## Generic Features

* IPMI 2.0 and DCMI
* IPMI LAN Interface
* IPMB Support
* IPv4 and IPv6
* Active Directory
* Sensor Monitoring
* FRU Support
* NTP Support
* Network Management
* mDNS and DNS Support
* Redfish Support
* Event Logging
* Firmware Update
* Web UI Support
* LDAP
* User Management
* Certificate Management
* SEL: Linear and Circular SEL Support
* Telemetry Support
* PEF and Alert Management (Email)
* Enhanced Password Policy Support
* Backup Restore Support
* Service Manager
* SSH Configuration
* SLPD
* LLDPD
* Hardware Watchdog
* SNMP
* Remote syslog Support
* Session Management
* SMTP

## SOC Specific

### AST2600

* IPMI KCS Interface
* VLAN
* KVM
* Virtual and Remote Media
* Dual image support
* Factory Reset
* Bonding Support
* UART Logging Support
* Preserve Configuration Support
* SOL
* License Control Support
* RADIUS

### AST2700

* SOL
* VLAN
* KVM
* Virtual and Remote Media
* Factory Reset
* UART Logging Support
* License Control Support
* Preserve Configuration Support

### Arbel

* Bonding Support
* RADIUS