HSBUV Board with Nuvoton NPCM7mnx RunBMC module
Quick (Standalone) Setup Guide

This Quick Setup guide describes how to set up the NPCM7mnx HSBUV Board + Nuvoton RunBMC module.

A. HSBUV + Nuvoton RunBMC module Overview

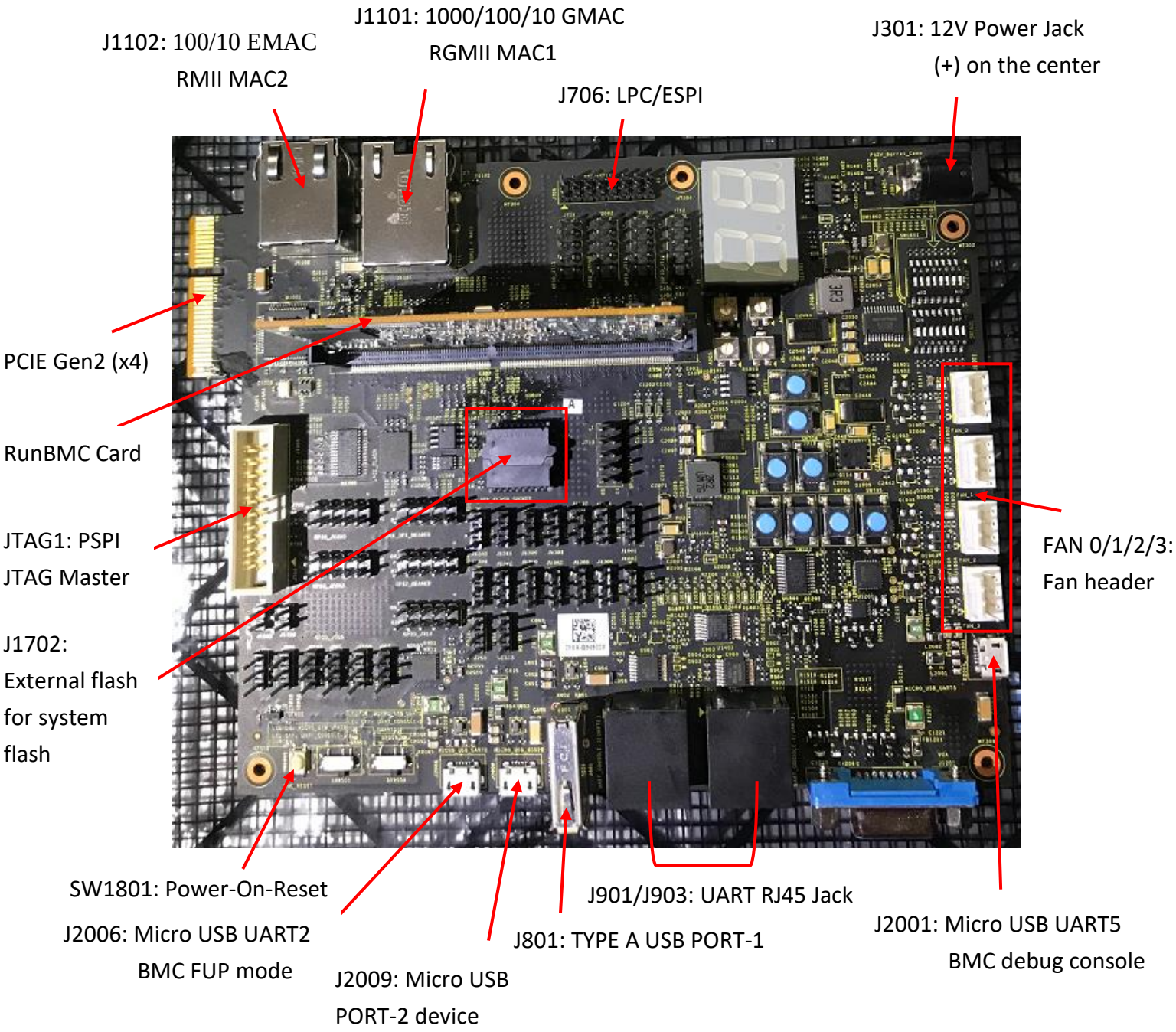


Figure 1: Connectors on the HSBUV Board

Note: Instruction refer to Figure 1, above.

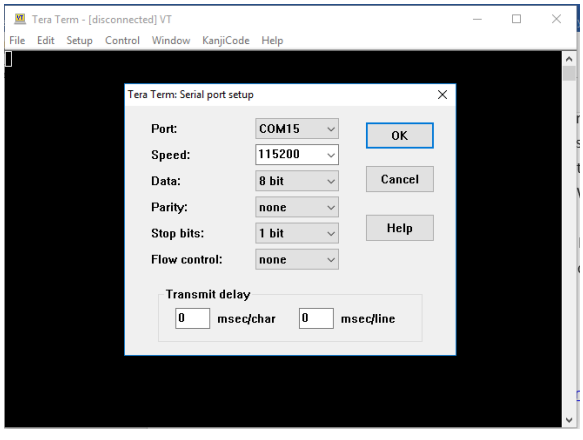
- Power-On and Reset:
  a. Connect the 12V power supply to power jack J301. The power supply should be 12V and at least 2A; the jack should be 2.5x 5.5 x9.5mm in diameter
  b. Press and release PWR-ON-RST (SW1801) push-button.

- USB-to-UART5 for BMC debug console:
  a. Download and install the USB-to-UART driver from:
  http://www.ftdichip.com/Drivers/VCP.htm
  according to the host OS.
  b. Connect a mini-USB cable between the PC host and HSBUV J2001. J2001 is Micro_USB_UART5.
  connector to the Serial Interface (SI2) of the BMC. Uboot and Linux terminal messages are sent though this port.
  c. Wait for the FTDI driver to be installed automatically. The COM port of number is assigned automatically.
  d. Verify that one green power LED (D2008) is ON.

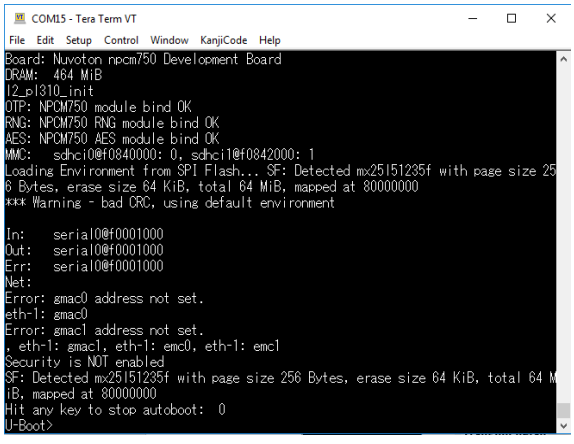
- Terminal:
  a. Open a terminal (e.g., Tera Term version 4.87) and set the correct COM port number assigned by the FTDI driver (in Step 2c).
  The COM port should be configured as follows:
  115200 Kbps, 8 bit, 1 stop-bit, no parity no flow control.
  b. Press and release the PWR-ON-RST (SW1801) push-button to issue a Power-On reset.
  c. Verify that the boot block, Uboot and Linux versions are up-to-date. Check with Nuvoton support for the most recent versions.

Figure 2: boot into boot block, Uboot

Tera Term Serial port setting:



boot into Uboot:

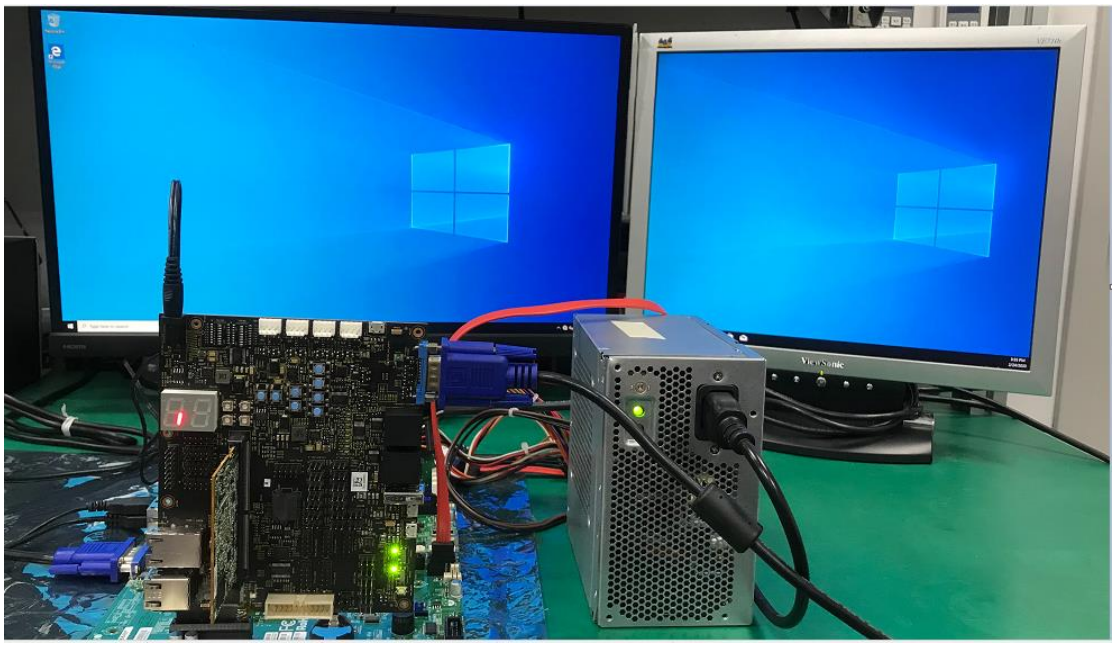


- The PCI-Express Interface which supports a PCIE Gen 2 (x4) connection (Note:). This interface shall be insert the system MB of PCIe slot for VGA display mailbox function. These signals are expected to be dedicated to PCIe functionality and should not offer a secondary function.

Note:

- Only x1 lane is been used.
- RunBMC can be used as a secondary video card since RunBMC card does not include on-board VGA BIOS and since MB does not include Matrox VGA BIOS.

Figure 3: HSBUV + RunBMC module boot into Win 10, the VGA display is ok



B. Build OpenBMC

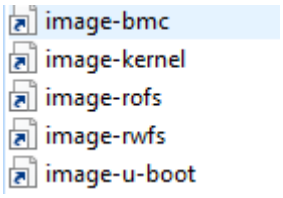
https://github.com/Nuvoton-Israel/openbmc/tree/runbmc

- **How to Build**

- Ubuntu 18.04 as example

```
$ sudo apt-get install -y git build-essential libsdl1.2-dev texinfo gawk chrpath diffstat
$ git clone -b runbmc --single-branch https://github.com/Nuvoton-Israel/openbmc
$ cd opebnmc
$ export TEMPLATECONF=meta-evb/meta-evb-nuvoton/meta-buv-runbmc/conf
$ . openbmc-env
$ bitbake obmc-phosphor-image
```

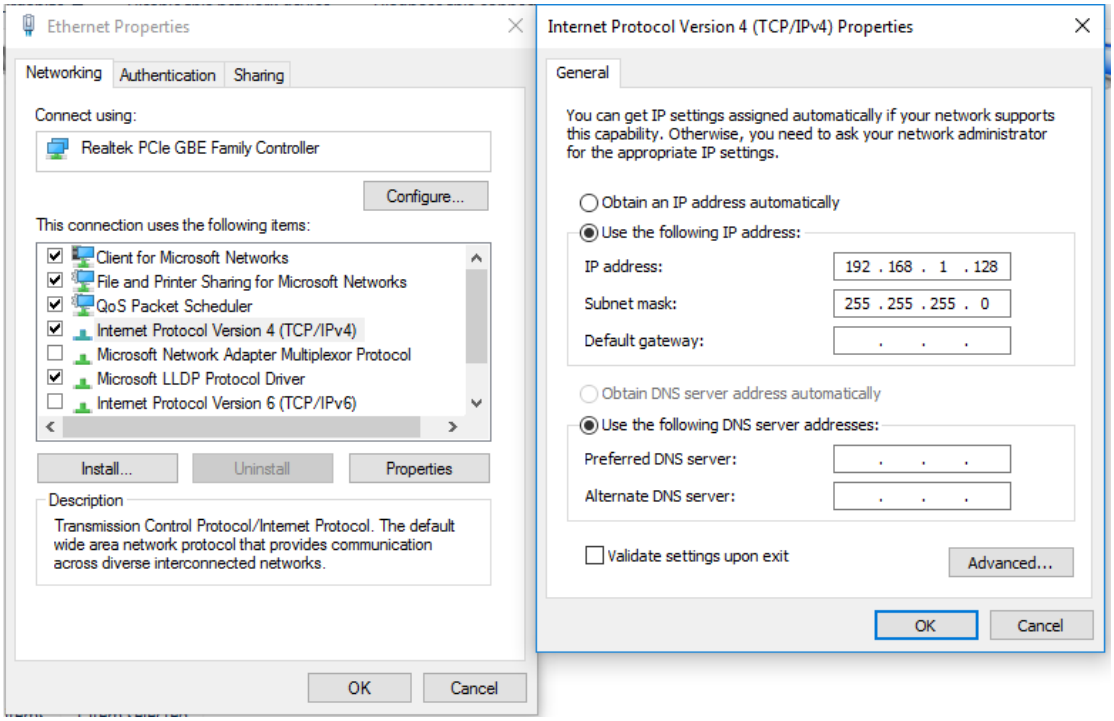
- If built successfully, you will find images in openbmc/build/tmp/deploy/images/buv-runbmc/



➢ **How to flash image-bmc**

- BMC FW update over u-boot TFTP
  - Setup IP Environment:

    NB IP: 192.168.1.128



    HSBUV IP: 192.168.1.15

    Open a Tera Term, press some commands under Uboot as below:

    Command as attached:

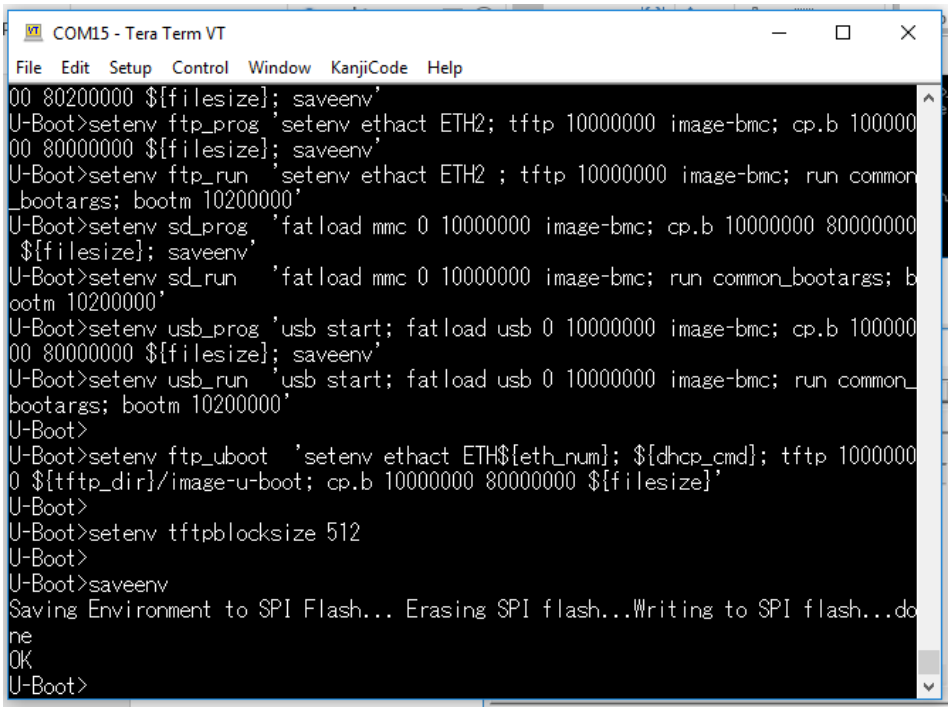
```
env default -a

echo Set the MAC address as printed on your board
echo ========================================
setenv mac_offset 01DC

setenv mac_base 00:00:F7:A0
setexpr byte ${mac_offset} / 100;setexpr nibh ${byte} / 10;setexpr nibl ${byte} % 10;setenv
mac_base ${mac_base}:${nibh}${nibl}
```

```
setexpr byte ${mac_offset} % 100;setexpr nibh ${byte} / 10
setexpr nibl ${byte} % 10;setenv ethaddr ${mac_base}:${nibh}${nibl}
setexpr nibl ${nibl} + 1;setenv eth1addr ${mac_base}:${nibh}${nibl}
setexpr nibl ${nibl} + 1;setenv eth2addr ${mac_base}:${nibh}${nibl}
setexpr nibl ${nibl} + 1;setenv eth3addr ${mac_base}:${nibh}${nibl}
setenv byte; setenv nibh; setenv nibl; setenv mac_base; setenv mac_offset

echo Those setting are according to your specific network parameters
echo ============================================================
setenv gatewayip           192.168.1.2
setenv nfspath             192.168.71.153:/home/ubuntu/shared/nfs
setenv serverip            192.168.1.128
setenv ipaddr              192.168.1.15
setenv netmask             255.255.0.0

echo choose either dhcp_on or dhcp_off by prefex 'echo' before the undesired option
echo ============================================================================
setenv dhcp_cmd dhcp
echo setenv dhcp_cmd

echo Update the images names and path if needed
echo ==========================================
setenv tftp_dir .



echo Select the network device for uboot: 0: RMII1, 1: RMII2, 2: RGMII1, 3: RGMII2
echo ============================================================================
setenv eth_num 2

echo Those settings are usefull for booting Poleg linux
echo ==================================================
setenv autoload  no
setenv autostart no
setenv baudrate 115200
setenv bootcmd 'run romboot'
setenv bootdelay 2
setenv common_bootargs 'setenv bootargs earlycon=${earlycon} root=/dev/ram0 console=${console}
mem=${mem} ramdisk_size=48000'
setenv console 'ttyS0,115200n8'
```

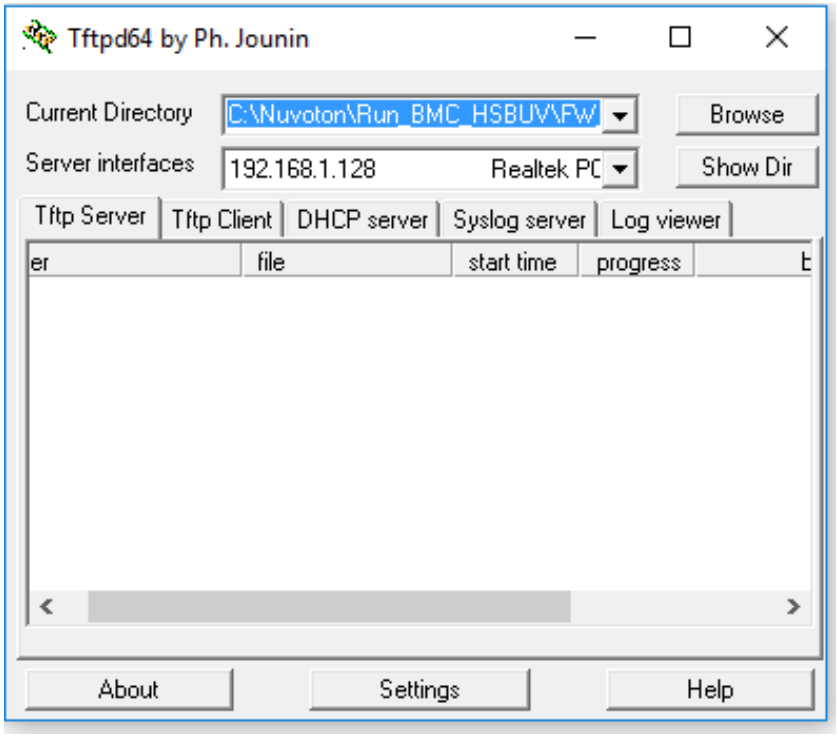
```
setenv earlycon 'uart8250,mmio32,0xf0001000
setenv mem 476M
setenv nfsboot 'setenv ethact ETH${eth_num};${dhcp_cmd};run common_bootargs;setenv bootargs
${bootargs} root=/dev/nfs init=/sbin/init nfsroot=${nfspath};tftp 10000000 ${tftp_dir}/image-
bmc; bootm 10200000'
setenv romboot 'run common_bootargs; echo Booting Kernel from flash; echo +++ uimage at
0x${uimage_flash_addr}; echo Using bootargs: ${bootargs};bootm ${uimage_flash_addr}'
setenv stderr serial
setenv stdin serial
setenv stdout serial
setenv uimage_flash_addr 80200000

setenv ftp_m 'setenv ethact ETH2; tftp 10000000 merged_1FF.bin; cp.b 10000000 80000000
${filesize}; saveenv'
setenv ftp_k 'setenv ethact ETH2; tftp 10000000 image-kernel; cp.b 10000000 80200000
${filesize}; saveenv'
setenv ftp_prog 'setenv ethact ETH2; tftp 10000000 image-bmc; cp.b 10000000 80000000
${filesize}; saveenv'
setenv ftp_run  'setenv ethact ETH2 ; tftp 10000000 image-bmc; run common_bootargs; bootm
10200000'
setenv sd_prog  'fatload mmc 0 10000000 image-bmc; cp.b 10000000 80000000 ${filesize};
saveenv'
setenv sd_run   'fatload mmc 0 10000000 image-bmc; run common_bootargs; bootm 10200000'
setenv usb_prog 'usb start; fatload usb 0 10000000 image-bmc; cp.b 10000000 80000000
${filesize}; saveenv'
setenv usb_run  'usb start; fatload usb 0 10000000 image-bmc; run common_bootargs; bootm
10200000'

setenv ftp_uboot  'setenv ethact ETH${eth_num}; ${dhcp_cmd}; tftp 10000000 ${tftp_dir}/image-
u-boot; cp.b 10000000 80000000 ${filesize}'

setenv tftpblocksize 512

saveenv
```

command.txt

```
echo ================================================================================
setenv eth_num 2

echo Those settings are usefull for booting Poleg linux
echo ================================================
setenv autoload  no
setenv autostart no
setenv baudrate 115200
setenv bootcmd 'run romboot'
setenv bootdelay 2
setenv common_bootargs 'setenv bootargs earlycon=${earlycon} root=/dev/ram0 console=${console} mem=${mem} ramdisk_size=48000'
setenv console 'ttyS0,115200n8'
setenv earlycon 'uart8250,mmio32,0xf0001000
setenv mem 476M
setenv nfsboot 'setenv ethact ETH${eth_num};${dhcp_cmd};run common_bootargs;setenv bootargs ${bootargs} root=/dev/nfs init=/sbin/init nfsroot=${nfspath};tftp 10000000 ${tftp_dir}/image-bmc; bootm 10200000'
setenv romboot 'run common_bootargs; echo Booting Kernel from flash; echo +++ uimage at 0x${uimage_flash_addr}; echo Using bootargs: ${bootargs};bootm ${uimage_flash_addr}'
setenv stderr serial
setenv stdin serial
setenv stdout serial
setenv uimage_flash_addr 80200000

setenv ftp_m 'setenv ethact ETH2; tftp 10000000 merged_1FF.bin; cp.b 10000000 80000000 ${filesize}; saveenv'
setenv ftp_k 'setenv ethact ETH2; tftp 10000000 image-kernel; cp.b 10000000 80200000 ${filesize}; saveenv'
setenv ftp_prog 'setenv ethact ETH2; tftp 10000000 image-bmc; cp.b 10000000 80000000 ${filesize}; saveenv'

setenv ftp_run  'setenv ethact ETH2 ; tftp 10000000 image-bmc; run common_bootargs; bootm 10200000'
setenv sd_prog  'fatload mmc 0 10000000 image-bmc; cp.b 10000000 80000000 ${filesize}; saveenv'
setenv sd_run   'fatload mmc 0 10000000 image-bmc; run common_bootargs; bootm 10200000'
setenv usb_prog 'usb start; fatload usb 0 10000000 image-bmc; cp.b 10000000 80000000 ${filesize}; saveenv'
setenv usb_run  'usb start; fatload usb 0 10000000 image-bmc; run common_bootargs; bootm 10200000'

setenv ftp_uboot  'setenv ethact ETH${eth_num}; ${dhcp_cmd}; tftp 10000000 ${tftp_dir}/image-u-boot; cp.b 10000000 80000000 ${filesize}'

setenv tftpblocksize 512

saveenv
```



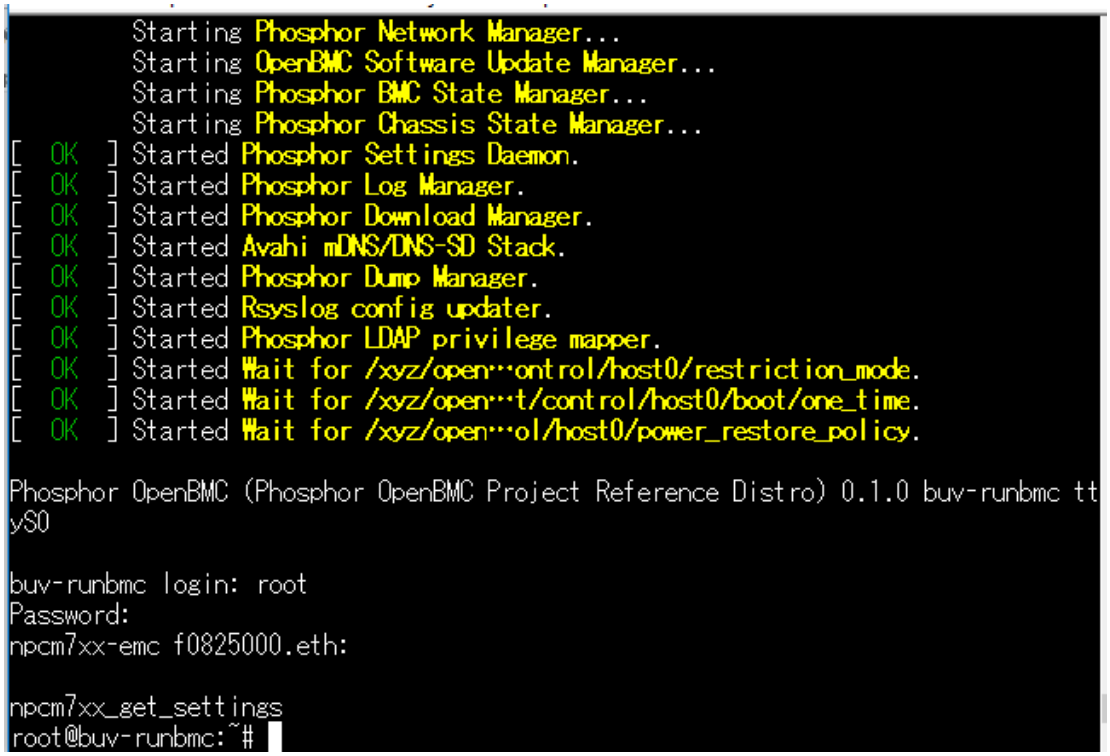
- Put image-bmc into your tftp server IP:



- Update u-boot env with https://github.com/Nuvoton-Israel/nuvoton-info/blob/master/npcm7xx-poleg/evaluation-board/sw_deliverables/uboot_env_parameters.txt
  (please make sure the serverip is your tftp server)
- Input command in u-boot console:
  U-Boot> run ftp_prog; reset

```
U-Boot>run ftp_prog; reset
mdio_register: non unique device name 'emc0'
mdio_register: non unique device name 'emc1'
Speed: 1000, full duplex
Using gmac0 device
TFTP from server 192.168.1.128; our IP address is 192.168.1.15
Filename 'image-bmc'.
Load address: 0x10000000
Loading: ###################################################################
         ###################################################################
```

- Update fw done and boot into Linux kernel:



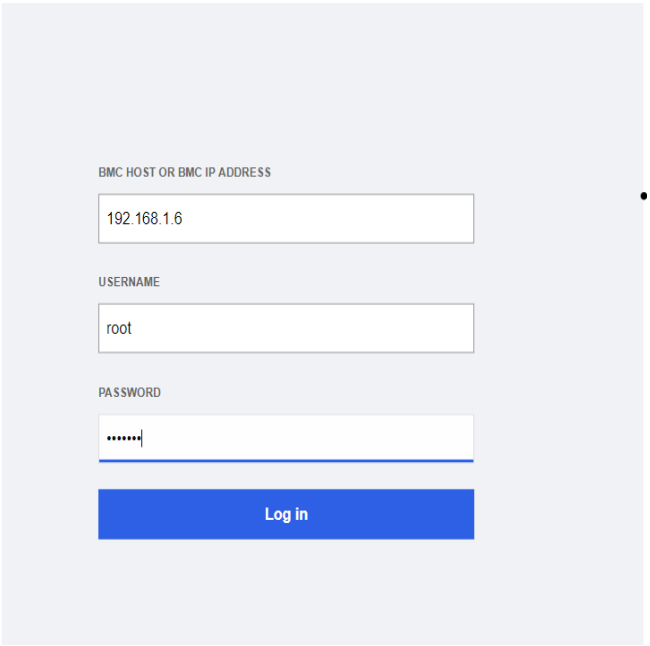
- BMC FW update over OpenBMC
  https://github.com/Nuvoton-Israel/openbmc/tree/runbmc/meta-quanta/meta-olympus-nuvoton#bmc-firmware-update

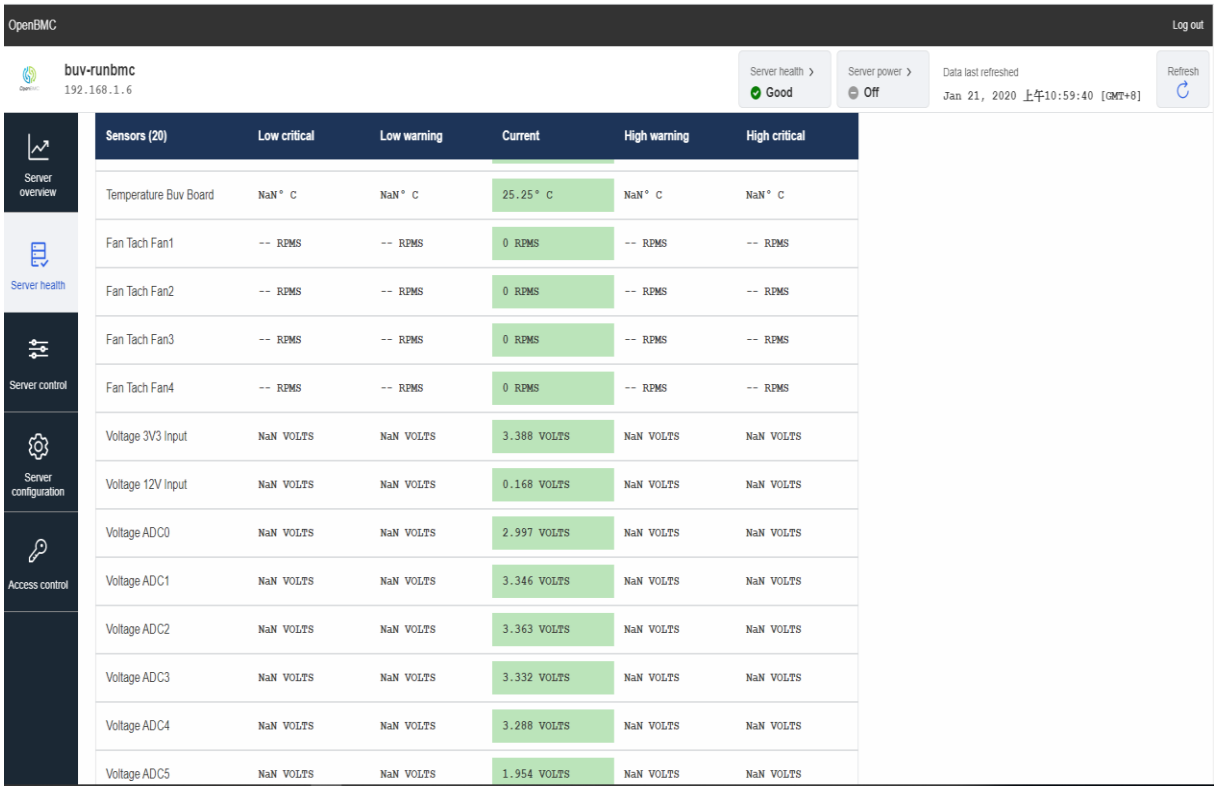
C. Enjoy OpenBMC

https://<BMC_IP>

Username: root

Password: openBmc

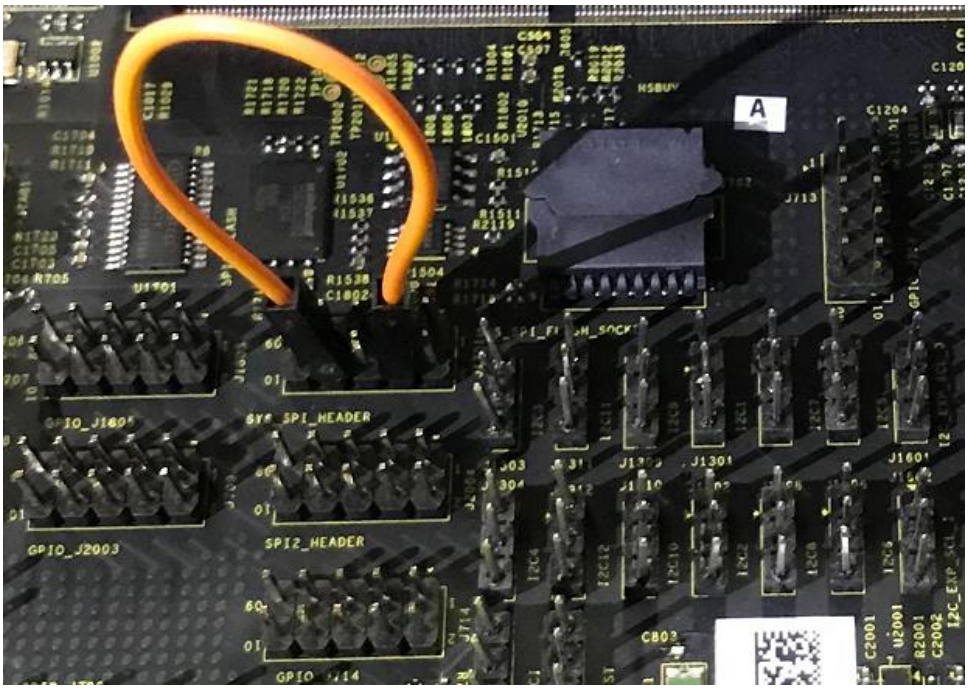




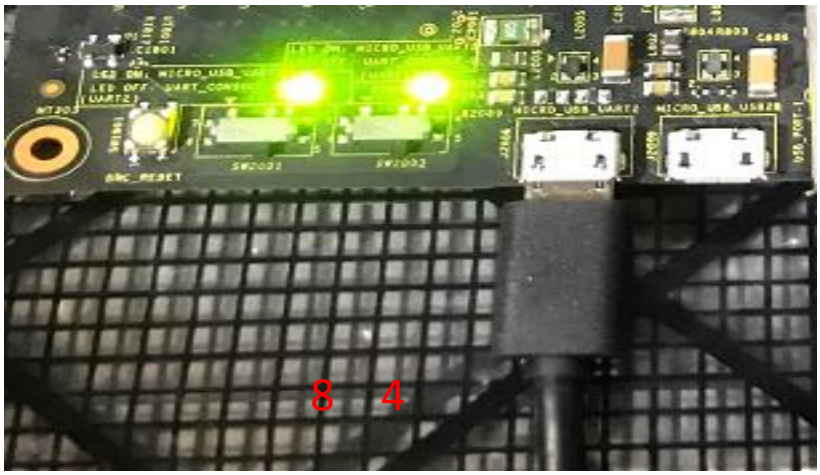
D. FUP mode for emergency firmware update (Boot-Block and Uboot)

Steps:

1. Remove HSBUV board AC Power 12V(J301) and Micro USB UART - BMC Debug UART (J2001)
2. Connects pin8 and pin4 of J1701 header

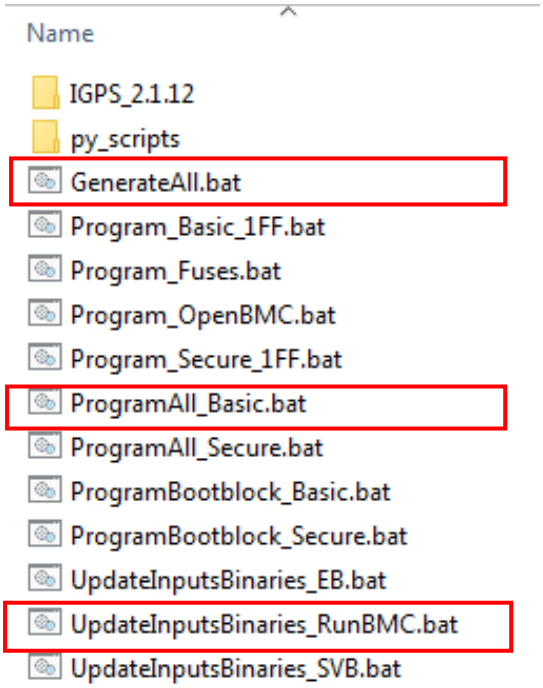


3. Connects USB cable to Micro USB UART - BMC FUP mode (J2006)



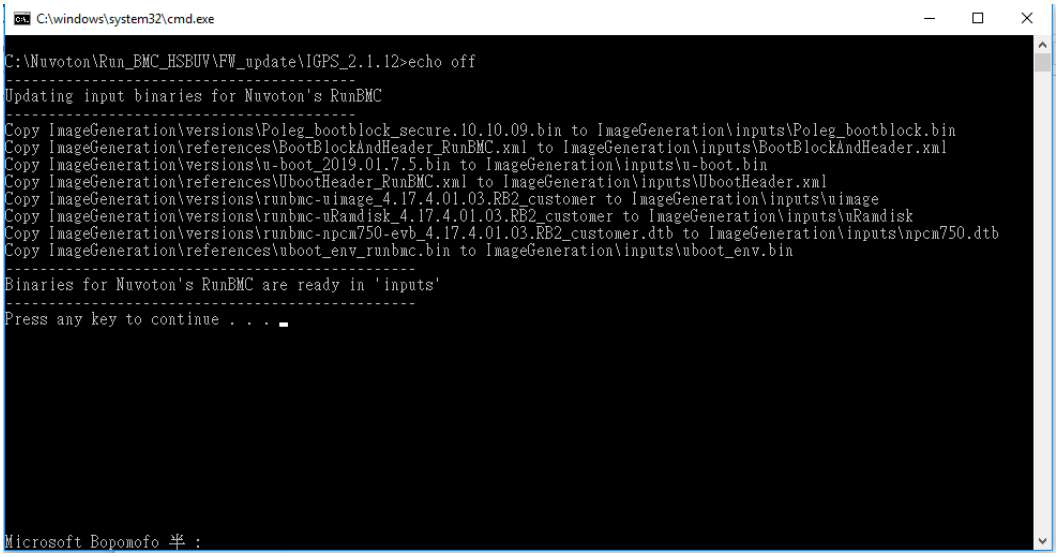
4. Recovery BMC over IGPS:
   https://github.com/Nuvoton-Israel/igps

Programmer fw list (.bat file)

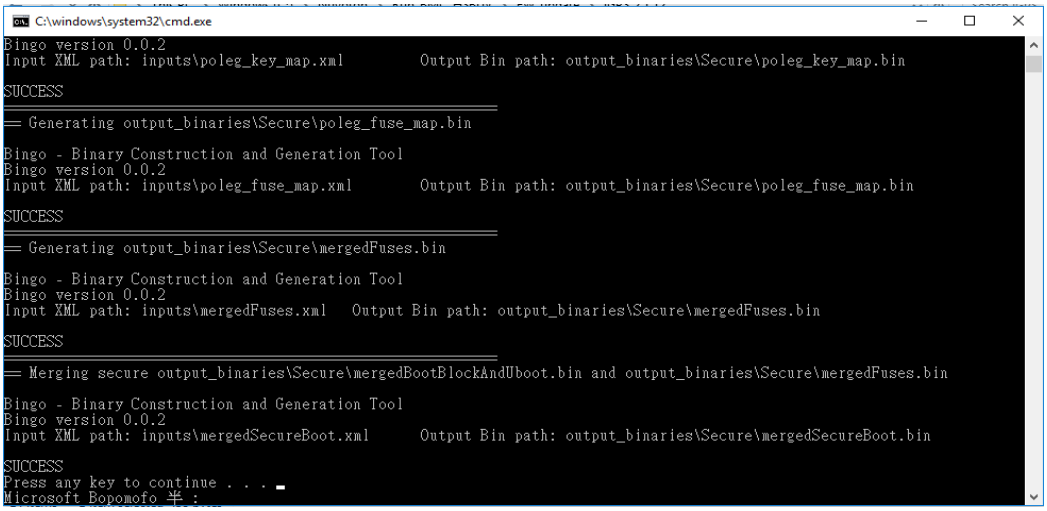


Steps:

(1) Run UpdateInputsBinaries_RunBMC.bat



(2) Run GenerateAll.bat



(3) Run ProgramAll_Basic.bat