

**Firmware User Manual**

**iMX8M-HMI Reference Design**

|  |  |
| --- | --- |
| **Version**  **Status**  **Date** | 0.2  Baseline  21-Dec-2018 |
|  |  |

**Confidentiality Notice**

Copyright (c) 2018 eInfochips. - All rights reserved

This document is authored by eInfochips and is eInfochips intellectual property, including the copyrights in all countries in the world. This document is provided under a license to use only with all other rights, including ownership rights, being retained by eInfochips. This file may not be distributed, copied, or reproduced in any manner, electronic or otherwise, without the express written consent of eInfochips

**Contents**

[Document Details 5](#_Toc533169400)

[ Document History 5](#_Toc533169401)

[ Definition, Acronyms and Abbreviations 5](#_Toc533169402)

[ References 5](#_Toc533169403)

[Introduction 6](#_Toc533169404)

[ Purpose of the document 6](#_Toc533169405)

[ About the System 6](#_Toc533169406)

[ Before You Start 7](#_Toc533169407)

[ Get the firmware package 7](#_Toc533169408)

[ Flash the firmware image to SD Card 7](#_Toc533169409)

[ Hardware Installation 7](#_Toc533169410)

[ Open board's terminal- console(minicom) on x86 host pc 8](#_Toc533169411)

[Running-Demos 9](#_Toc533169412)

[1. Ethernet demo 9](#_Toc533169413)

[2. HDMI Demo 9](#_Toc533169414)

[3. Dual Display Demo 10](#_Toc533169415)

[4. HDMI2 Touch panel Demo 12](#_Toc533169416)

[5. Mezzanine DSI Display Demo 13](#_Toc533169417)

[6. Camera Demo 14](#_Toc533169418)

[7. Audio Codec Demo 15](#_Toc533169419)

[8. LTE Demo 17](#_Toc533169420)

[9. USB Hub demo 18](#_Toc533169421)

[10. USB OTG as host 18](#_Toc533169422)

[11. USB OTG as Devices 19](#_Toc533169423)

[12. Bluetooth 21](#_Toc533169424)

[13. EEPROM 22](#_Toc533169425)

[14. Zigbee Demo 23](#_Toc533169426)

[15. USER LED 26](#_Toc533169427)

[16. Low Power Expansion GPIO 27](#_Toc533169428)

[17. CAN Interface demo 29](#_Toc533169429)

[18. NOR Flash demo 29](#_Toc533169430)

[Known Issues and Limitations 29](#_Toc533169431)

[Contact US 29](#_Toc533169432)

**Figures**

[Figure 1 iMX8M\_HMI Platform Connectors 6](#_Toc533164260)

[Figure 2 iMX8M\_HMI Platform UART Connections 8](#_Toc533164261)

[Figure 3 iMX8M\_HMI Platform Mezzanine DSI OLED 13](#_Toc533164262)

[Figure 4 USB Mass Storage on HOST system 20](#_Toc533164263)

**Tables**

[Table 1: Documents History 5](#_Toc533164257)

[Table 2: Definition, Acronyms and Abbreviations 5](#_Toc533164258)

[Table 3: References 5](#_Toc533164259)

# Document Details

## Document History

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Version** | **Author** | | **Reviewer** | | **Approver** | |
| **Name** | **Date**  **(DD-MM-YYYY)** | **Name** | **Date**  **(DD-MM-YYYY)** | **Name** | **Date**  **(DD-MM-YYYY)** |
| *0.1* | *Rutvij Trivedi* | *17-Dec-2018* | Prajose John | *17-Dec-2018* | Bhavin Patel | *17-Dec-2018* |
| 0.2 | Rutvij Trivedi | *21-Dec-2018* | Prajose John | *21-Dec-2018* | Bhavin Patel | *21-Dec-2018* |
|  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Version** | **Description Of Changes** |
|
| *0.1* | *initial draft* |
| 0.2 | Added new features as mentioned in release note 0.2 |
|  |  |

Table 1: Documents History

## Definition, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Definition/Acronym/Abbreviation** | **Description** |
| Cd | Change directory |
| scp | Secure copy over the network |
| Dfl | Default |
| Wi-Fi | Wireless fidelity |
| LTE | Long-Term Evolution |
| BLE | Bluetooth low energy device |
| DSI | Display Serial Interface |
|  |  |
|  |  |

Table 2: Definition, Acronyms and Abbreviations

## References

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Document** | **Version** | **Remarks** |
| 1 | Refer the release note V0.2 | 0.2 |  |

Table 3: References

# Introduction

## Purpose of the document

* Purpose of this documentis to use/understand/flash/demonstrate interfaces on iMX8M-HMI PLATFORM firmware.

## About the System

* This system contains iMX8M reference design with multiple interfaces, can be used for Human-machine interface experience.

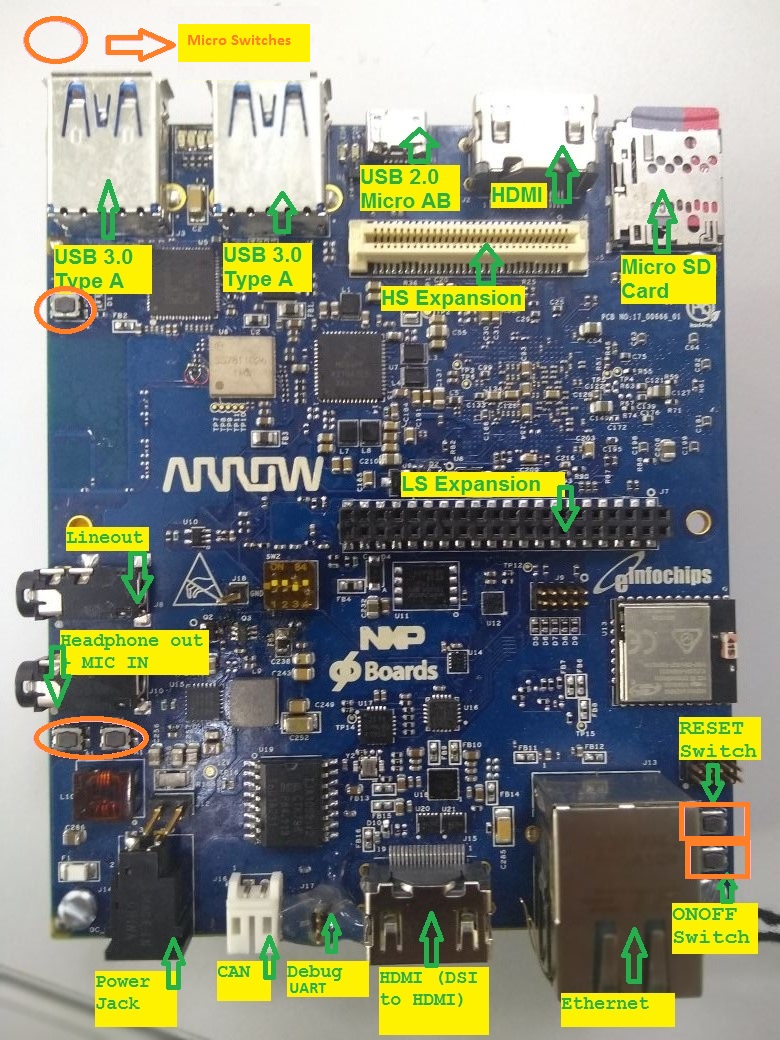


Figure 1 iMX8M\_HMI Platform Connectors

## Before You Start

* Ensure you have x86 host system having Linux Ubuntu 16.04 LTS installed
* Basic understanding of Linux commands

## Get the firmware package

* Download the provided SD card (sdcad.bz2) image in Linux pc
* Open terminal in host pc from left desktop panel or using keyboard shortcut (**ctrl + t)**
* From command terminal traverse the location where downloaded firmware image is residing using **cd** command

**ex. cd /home/user/download/imximages/**

* use **ls** command to verify the existence of downloaded image

**ex**. **ls –l**

* *Verify md5 check sum of downloaded image which should be provided over the share point*

***md5sum <image name>.*sdcard.bz2**

* Extract the provided **.bz2** image using **bunzip2** command, which will take couple of minutes.

**bunzip2 -dkf <image\_name>.sdcard.bz2**

* Once done, will end with **.sdcard** image in the same directory and can again be verified using **ls -l** command.

## Flash the firmware image to SD Card

* Plugin micro SD card into x86 host pc
* *Verify the node created for SD card into /dev directory*

***ex. ls -l /dev/sd\****

* Open terminal and traverse the location where downloaded firmware image is residing using **cd** command
* Ensure the extracted firmware image's file format is **.sdcard** using **ls -l** command
* Apply below command for flashing if the SD card’s entry in Linux is **/dev/sdb**
* **sudo dd if=<image\_name>.sdcard of=/dev/sdb bs=1M conv=fsync ;sync**
* Above command will take couple of minutes or more (depending upon host machine config) to flash the SD card
* Once done plug-out and replug-in the sd card, two drives will get mounted if the above gets successful, named <boot> and <rootfs>
* **Eject (safely remove) SD card from host pc and plug it into board's sd card slot**

## Hardware Installation

* Place hardware board on statically clean place
* Plug flashed SD card to J5 SD card slot.
* Plug serial cable's micro end to board's J10 Connector (near Ethernet connector) and USB end to host x86 pc's usb connector.
* Plug Ethernet cable to board's Ethernet connector J12.
* Apply 12V-5A power supply (provided with board) to board on J14 **DC\_IN** connector. After all the other hardware setup is done and required interfaces are connected to board.

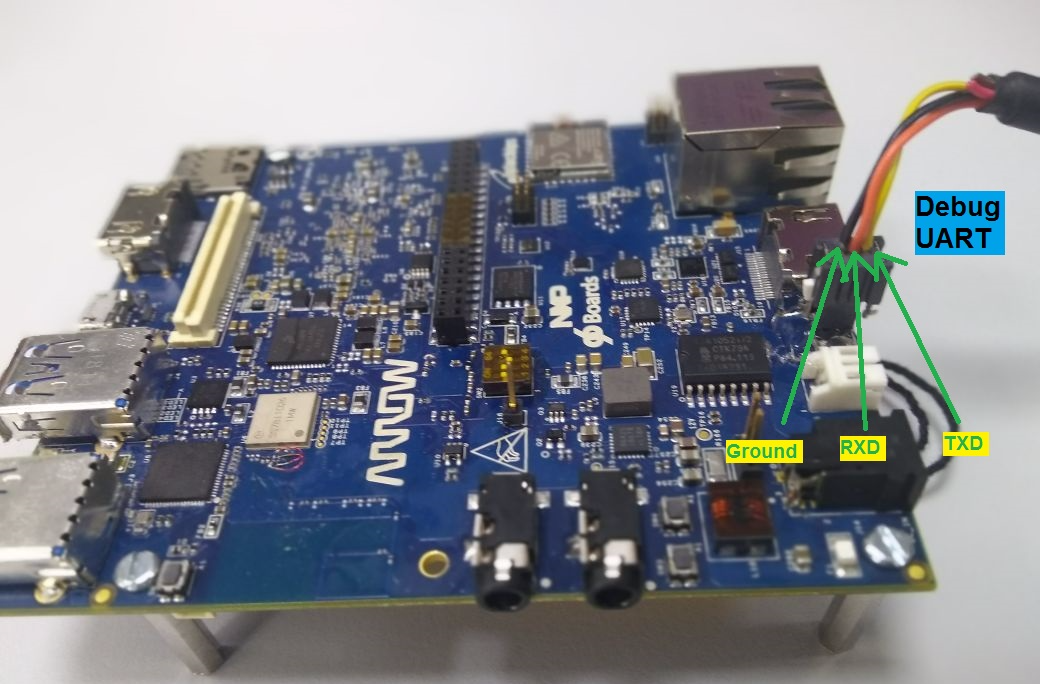


Figure 2 iMX8M\_HMI Platform UART Connections

## Open board's terminal- console(minicom) on x86 host pc

* Ensure SD card is flashed and serial cable is plugged in into board as per mentioned in hardware setup.
* Attached serial cable's USB end to host x86 PC's USB.
* Ensure **minicom** is installed in x86 Ubuntu pc
* Apply below command to open serial command's setting.

**sudo minicom –s**

* **set baud rate and other setting as per below**
* baud rate 115200 , parity none , hardware flow control/software flow control none , serial device /dev/ttyUSB0 ,save setup as dfl.
* Once board gets powered up , above configured terminal will show logs on x86 and can interact with board using this open terminal

# Running-Demos

## Ethernet demo

* Plug in Ethernet cable to target board as per above figure.
* Power up the board.
* Once board gets booted, apply below command using console (minicom require)
* Ifconfig
* Ping <any server ip>

## HDMI Demo

* Ensure board is not powered up and SD card is flashed with the latest provided image.
* Insert HDMI cable into board's J2 HDMI connector.
* Apply power to board and go to terminal of x86 host system and open board’s console as mentioned above
* Hold boot on u-boot screen by pressing any key on host machine keyboard(immediate after boot within 3 seconds)
* Apply dtb file changes as per below command on u-boot console
  + **setenv fdt\_file fsl-imx8mq-evk.dtb**
  + **saveenv**
  + **boot**
* Console will show booting logs
* Once booting is completed, console will hold on login prompt where user can enter username as **root**. (no password)
* At this time connected HDMI display will show grey image of desktop and should stop complaining about "No Signal"

**Play Video Test pattern on HDMI display**

* Go to board's console and type below command from x86 minicom
* **gst-launch-1.0 videtestsrc ! autovideosink**
* Above command will show color strips on HDMI display

**Play local videos on HDMI display with audio**

* Ensure Ethernet is connected with board
* Go to board's console and type below command from x86 minicom
* **ifconfig**
* Get the ip address of Ethernet eth0 interface and note down.
* Go to x86 host system and download sample mp4 video with audio.
* Locate to video location from command line in x86 (no minicom require)
* Apply below command
* **scp ./Sample\_Video\_with\_audio.mp4 root@<noted ip address of board>:/home/root/**
* This will copy the video file from host x86 to board's /home/root location
* Go to board's console (require minicom) and ensure video got copied using ls -l command, will show you Sample\_Video\_with\_audio.mp4 in current directory.
* Be in the board's console and apply below command to play video over HDMI Display with audio (HDMI Display should have support of audio )
* **gst-launch-1.0 filesrc location=/home/root/video.mp4 ! decodebin name=dec ! videoconvert ! autovideosink dec. ! audioconvert ! audioresample ! alsasink device=plughw:3,0**
* Above command will print logs on console of board and will be played over HDMI display with audio.

## Dual Display Demo

* Ensure board is not powered up and SD card is flashed with the latest provided image.
* Insert one HDMI cable into board's J2 HDMI connector, another HDMI cable to HDMI2 on j15 connector
* Apply power to board and go to terminal of x86 host system and open board’s console as mentioned above
* Apply dtb file changes as per below command on u-boot console
  + **setenv fdt\_file fsl-imx8mq-evk-dual-display-b3.dtb**
  + **saveenv**
  + **boot**
* Console will show booting logs
* Login to board using root username with no password.

**Play local videos on HDMI display(Dual)**

* Copy local sample\_video.mp4 & sample\_video2.mp4 videos using **scp** command to on board sd card as mentioned in above demo.
* Go to board's console (require minicom) and ensure video got copied using ls -l command, will show you sample\_video.mp4 in current directory.
* Apply below command to play video over HDMI(j2) Display
* **gst-launch-1.0 filesrc location=sample\_video.mp4 typefind=true ! video/quicktime ! qtdemux ! queue max-size-time=0 ! vpudec ! queue max-size-time=0 ! kmssink sync=true &**
* With not much delaying apply below command to play video over HDMI2 (j15) Display
* **gst-launch-1.0 -v filesrc location=sample\_video2.mp4 typefind=true ! video/quicktime ! aiurdemux ! queue max-size-time=0 ! vpudec ! waylandsink &**
* The both the connected HDMI will show dual video demo playback.

**Play Network stream and local videos on HDMI display (Dual)**

* Ensure the network bandwidth.
* To play network stream on DSI (HDMI2 – j15) display, create an rtsp server on any host machine (i.e. x86) and play video over network, (one can use vlc player to do the same) and note IP address and port number for rtsp server(host machine).
* For how to create server on vlc and play stream can be seen from below link.
  + [**https://support.spinetix.com/wiki/Tutorial:Streaming\_using\_VLC**](https://support.spinetix.com/wiki/Tutorial:Streaming_using_VLC)
* In board’s console apply below command to play the same stream over the network
* **gst-launch-1.0 playbin uri=rtsp://<Ip add>:<port>/<video name without format >uridecodebin0::source::latency=300 &**
* Play simultaneously local video over HDMI1 (j2) using below command.
* **gst-launch-1.0 filesrc location=sample\_video.mp4 typefind=true ! video/quicktime ! qtdemux ! queue max-size-time=0 ! vpudec ! queue max-size-time=0 ! kmssink sync=true &**

## HDMI2 Touch panel Demo

**Play local videos on HDMI display**

* Copy local sample\_video.mp4 videos using scp command to on board sd card as mentioned in above demo.
* Insert HDMI cable into board's J15 HDMI2 connector and touch USB to board’s USB.
* Apply power to board and go to terminal of x86 host system and open board’s console as mentioned above
* Hold boot on u-boot screen by pressing any key on host machine keyboard(immediate after boot within 3 seconds)
* Apply dtb file changes as per below command on u-boot console
  + **setenv fdt\_file fsl-imx8mq-evk-dcss-adv7535-b3.dtb**
  + **setenv mmcargs 'setenv bootargs ${jh\_clk} console=${console} root=${mmcroot} video=HDMI-A-1:1280x800-12@70'**
  + **saveenv**
  + **boot**
* Console will show booting logs
* Once booting is completed, console will hold on login prompt where user can enter username as **root**. (no password)
* Once boot completes apply below commands
  + **systemctl stop weston**
* Start playback using below command
* **gst-launch-1.0 filesrc location=video.mp4 typefind=true ! video/quicktime ! aiurdemux ! queue max-size-time=0 ! vpudec ! queue max-size-time=0 ! autovideosink**

**Touch Demo**

* Ensure touch panel is connected to board using USB and is powered up.
* Go to board's console (require minicom) , log in and apply below commands
  + **ls /dev/input/ -l**
* Find entry as per below
  + **lrwxrwxrwx 1 root     0      6 Nov 23 12:35 touchscreen0 -> event1**
* Once received node, cat that node using below command,
  + **cat /dev/input/touchscreen0 | hexdump**
* Touch on the screen and get the events on console to validate touch

## Mezzanine DSI Display Demo

* Attach dsi display MX8\_DSI\_OLED to high-speed mezzanine connector.

**Play video on mezzanine DSI display**

* After verification on mezzanine connection with h/w, power up the board**.**
* Go to board's console (require minicom) and immediately stop at **u-boot autoboot** console by pressing any key.
* Apply below commands for changing dtb file
  + **setenv fdt\_file  fsl-imx8mq-evk-dcss-rm67191-b3.dtb**
  + **saveenv**
  + **boot**
* Copy local sample\_video.mp4 videos using scp command to on board sd card as mentioned in above demo.
* Play above copied video on the OLED Display using below command
* **gst-launch-1.0 filesrc location= sample\_video.mp4 typefind=true ! video/quicktime ! aiurdemux ! queue max-size-time=0 ! vpudec ! queue max-size-time=0 ! autovideosink**



Figure 3 iMX8M\_HMI Platform Mezzanine DSI OLED

## Camera Demo

**Play live video stream from camera on HDMI display**

* Attach camera module to high-speed mezzanine connector.
* To watch live stream over the HDMI, connect HDMI Display too.
* Power up the board
* Go to board's console (require minicom) and immediately stop at **u-boot autoboot** console by pressing any key.
* Apply below commands for changing dtb file
  + **setenv fdt\_file  fsl-imx8mq-evk-mipi-csi.dtb**
  + **saveenv**
  + **boot**
* Once boot completes apply below commands
  + **systemctl stop weston**
* Display will get blank and will be black only
* Then apply below command
  + **/unit\_tests/V4L2/mx8\_v4l2\_cap\_drm.out -cam 1**
* This will show live streaming over the attached HDMI for couple of minutes

**Capture image from camera**

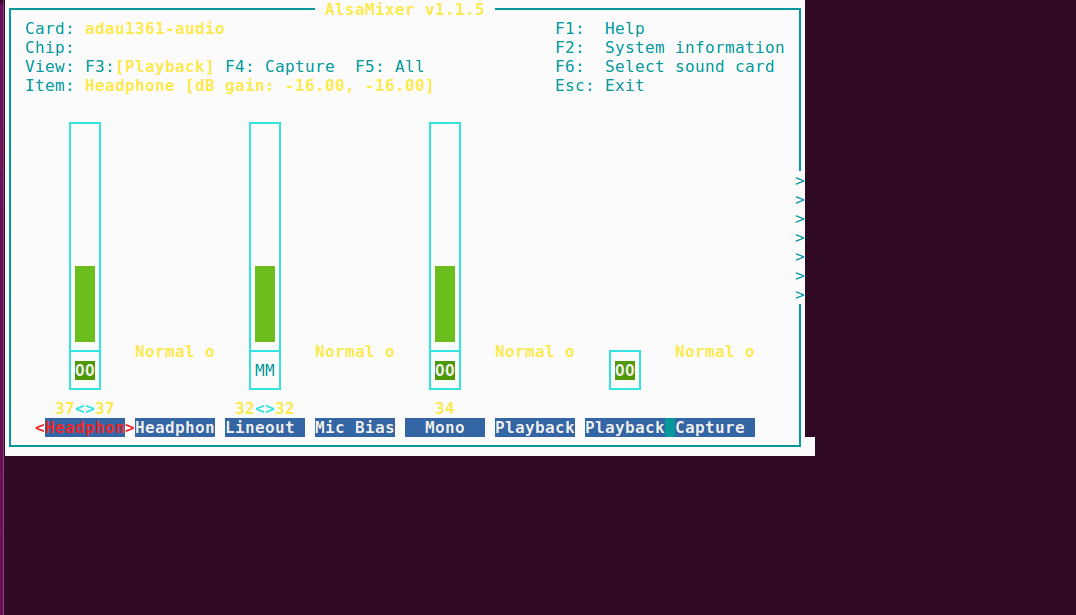
* Go to board's console (require minicom) and power up the board with above mentioned dtb change configuration.
* Ensure Ethernet is plugged-in to get image from board to local x86 host pc.
* Apply below command to capture image from camera.
* **gst-launch-1.0 v4l2src num-buffers=1 ! jpegenc ! filesink location= /home/root/test.jpg**
* Above command will capture image named test.jpg in /home/root/ location

Copy image from board to local pc using below command

* **scp test.jpg <user name of host pc >@<ip of host pc>:/home/user/Desktop**
* Go to local pc's /home/user/Desktop and watch image into image viewer to verify captured image from board's camera.

## Audio Codec Demo

* Go to board's console (require picocom) and power up.
* Install picocom in host machine using below command.
  + **sudo apt-get install picocom**
* After installing picocom open console use of below command
  + **sudo picocom -b 115200 -r –l /dev/ttyUSB0**
* From the above command you will get the imx8mq hmi board’s tty console
* Type alsamixer command in console
* Following screen will get on console.



* You can select a **different sound card by pressing F6**. It will bring up a menu that shows the known sound cards on imx8mq hmi system.
* The default you see above is the “Playback” view. You can choose “Capture” by pressing F4 and “All” (which includes “Playback” and “Capture”) by pressing F5. Return to “Playback” with F3.Move right and left, respectively, through those options by pressing the Left and Right arrow keys.
* Adjust each **volume with Down**to reduce the volume of a channel and **Up to increase** the volume.
* You can **mute** and **unmute** any channel by **pressing m.**
* To play and capture the audio you need to all setting must be unmute and gain of all channel are not to be zero.
* After done all setting press ESC button to close alsamixer utility.
* Enter below command to check the audio codec playback probed
  + **aplay –l**
* find the below log from the list

**adau1361audio [adau1361-audio], device 0: adau1x61 adau-hifi-0 []**

**Subdevices: 1/1**

**Subdevice #0: subdevice #0**

* Enter below command to check audio codec capture driver probed

**arecord –l**

* Find below logs from the list

**adau1361audio [adau1361-audio], device 0: adau1x61 adau-hifi-0 []**

**Subdevices: 1/1**

**Subdevice #0: subdevice #0**

* To playback audio enter below command
* **aplay -Dplughw:0,0 SAI/TEST.wav**
* To record audio enter below command
* **arecord –Dplughw:0,0 record.wav**

## LTE Demo

* Connect Quectel module with target board.
* Go To Board’s console and apply below command
* **pppd call quectel-ppp &;**
* Edit /etc/resolv.conf as per below

**nameserver 59.144.127.117**

**nameserver 59.144.144.46**

* Save above file and apply below command
* **Ifconfig ppp0**
* **Ping** [www.google.com](http://www.google.com) **–I ppp0**
* Will be able to ping to google.com

## USB Hub demo

* Connect USB device disk to USB port of target board
* Go to board ‘s console and apply below command
* **lsusb**
* On Connecting usb pendrive

usb 2-1.2: new SuperSpeed USB device number 4 using xhci-hcd  
**usb-storage 2-1.2:1.0: USB Mass Storage device detected**  
scsi host0: usb-storage 2-1.2:1.0  
**scsi 0:0:0:0: Direct-Access     SanDisk  Ultra Fit        1.00 PQ: 0 ANSI: 6  
sd 0:0:0:0: [sda] 30031872 512-byte logical blocks: (15.4 GB/14.3 GiB)**

* On Disconnecting usb pendrive

**usb 2-1.2: USB disconnect, device number 4**

## USB OTG as host

* Power up the board
* Go to board's console (require minicom) and immediately stop at **u-boot autoboot** console by pressing any key.
* Apply below commands for changing dtb file
  + **setenv fdt\_file  fsl-imx8mq-evk-otg-host.dtb**
  + **saveenv**
  + **boot**
* Once boot completes
* Connect USB device disk to USB OTG port of target board
* Go to board ‘s console and apply below command as same as USB hub
  + **lsusb**
* On Connecting usb pendrive

usb 2-1.2: new SuperSpeed USB device number 4 using xhci-hcd  
**usb-storage 2-1.2:1.0: USB Mass Storage device detected**  
scsi host0: usb-storage 2-1.2:1.0  
scsi 0:0:0:0: Direct-Access     SanDisk  Ultra Fit        1.00 PQ: 0 ANSI: 6  
sd 0:0:0:0: [sda] 30031872 512-byte logical blocks: (15.4 GB/14.3 GiB)

* On Disconnecting usb pendrive

**usb 2-1.2: USB disconnect, device number 4**.

## USB OTG as Devices

* Connect USB cable (same like debug uart cable) USB OTG port of target board
* Run the below command

**#dd if=/dev/zero of=/mass\_storage bs=1M seek=256 count=0  
#mkfs.fat /mass\_storage   
#cat <<EOT | sfdisk --reorder  /mass\_storage   
 ,,c  
 EOT**

**#mkfs.vfat /mass\_storage**

**# chmod 777 /mass\_storage   
#mount -o loop  /mass\_storage /mnt/  
#mount  
#modprobe /lib/modules/$(uname -r )/kernel/drivers/usb/gadget/legacy/g\_mass\_storage.ko file=/mass\_storage**

* Disconnect and connect the USB cable
* User will see the drive on host machine.

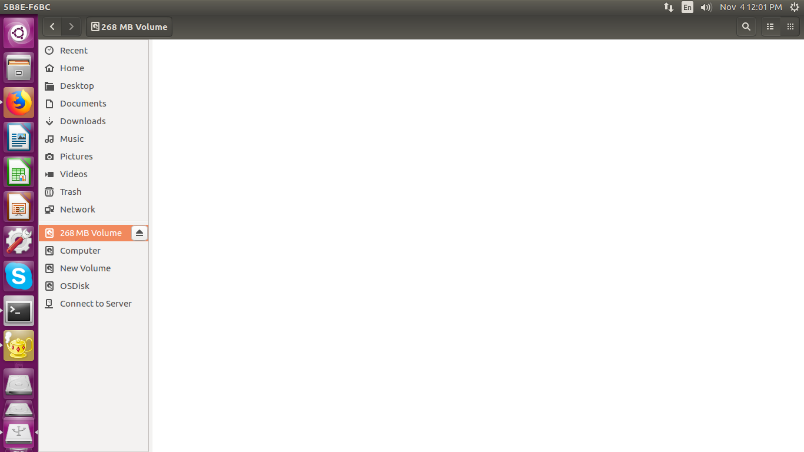
****

Figure 4 USB Mass Storage on HOST system

## Bluetooth

* Go to board ‘s console and apply below command

**#hciattach /dev/ttymxc1 bcm43xx 3000000 flow -t 20**

* Wait until the complete the command response

**# hciconfig hci0 up**

**# hciconfig hci0 -a**

* User will get the hcio interface
* Run the “bluetoothctl” utility

**#bluetoothctl  
[bluetooth]# power on  
[bluetooth]# agent on  
[bluetooth]# default-agent  
[bluetooth]# pairable on  
[bluetooth]# scan on  
  
Copy mac address  
  
[bluetooth]# scan off  
[bluetooth]# pair <mac address>  
  
Approve pairing on Device if required  
  
[bluetooth]# trust <mac address>  
[bluetooth]# connect <mac address>  
[bluetooth]# quit**

* Sending file command.

**#export $(dbus-launch)  
#/usr/libexec/bluetooth/obexd &  
  
#obexctl  
[obex]# connect <mac addr>  
[<mac addr>]# send <file>  
  
[<mac addr>]# disconnect  
[<mac addr>]# quit**

* Play the audio over BT commands
* Collect the audio file from the support package folder.
* Get the Bluetooth headset or Bluetooth speaker.

**# aplay -D bluealsa:HCI=hci0,DEV=<mac addr>,PROFILE=a2dp play\_audio.wav**

* Get the Mobile headset,
* Connect mobile with our modem using above **bluetoothctl** command.
* play the music on mobile player
* run below command to capture the audio from Bluetooth

**# arecord -D bluealsa:HCI=hci0,DEV=<mac addr>,PROFILE=a2dp record\_audio.wav**

* copy recovered file in your host PC and verify with any player on host PC

## EEPROM

* Run below command to test EEPROM

**$echo hello > /sys/bus/i2c/devices/i2c-1/1-0050/eeprom**

**$cat /sys/bus/i2c/devices/i2c-1/1-0050/eeprom | hexdump -C**

## Zigbee Demo

* Steps to test Zigbee as below :
* Copy host zigbee application to device using scp command via Ethernet if already not on the board at path /home/root/zigbee/ Z3GatewayHost .
* Use below command to set executable flags for binary.

**# chmod 777 Z3GatewayHost**

* Run Z3GatewayHost as below.(Please flash ncp-spi and bootloader if not flashed). Note for board which are shipped, are already flashed.

**# ./Z3GatewayHost**

[../../../platform/base/hal/micro/unix/host/spi-protocol-linux.c] traceMask = 0xFF

[../../../platform/base/hal/micro/unix/host/spi-protocol-linux.c] Opened nHOST\_INT device 75.

[../../../platform/base/hal/micro/unix/host/spi-protocol-linux.c] Opened SPI device /dev/spidev1.0.

[../../../platform/base/hal/micro/unix/host/spi-protocol-linux.c] Opened nCS device 8.

[../../../platform/base/hal/micro/unix/host/spi-protocol-linux.c] Opened nRESET device 132.

[../../../platform/base/hal/micro/unix/host/spi-protocol-linux.c] Opened nWAKE device 74.

[../../../platform/base/hal/micro/unix/host/spi-protocol-linux.c] Cannot write to /sys/class/gpio/export.

[../../../platform/base/hal/micro/unix/host/spi-protocol-linux.c] Cannot write to /sys/class/gpio/export.

[../../../platform/base/hal/micro/unix/host/spi-protocol-linux.c] Cannot write to /sys/class/gpio/export.

[../../../platform/base/hal/micro/unix/host/spi-protocol-linux.c] Cannot write to /sys/class/gpio/export.

[../../../platform/base/hal/micro/unix/host/spi-protocol-linux.c] traceMask = 0xFF

[../../../platform/base/hal/micro/unix/host/spi-protocol-linux.c] Opened nHOST\_INT device 75.

[../../../platform/base/hal/micro/unix/host/spi-protocol-linux.c] Opened SPI device /dev/spidev1.0.

[../../../platform/base/hal/micro/unix/host/spi-protocol-linux.c] Opened nCS device 8.

[../../../platform/base/hal/micro/unix/host/spi-protocol-linux.c] Opened nRESET device 132.

[../../../platform/base/hal/micro/unix/host/spi-protocol-linux.c] Opened nWAKE device 74.

Reset info: 11 (SOFTWARE)

ezsp ver 0x07 stack type 0x02 stack ver. [6.4.0 GA build 385]

Ezsp Config: set source route table size to 0x0064:Success: set

Ezsp Config: set security level to 0x0005:Success: set

Ezsp Config: set address table size to 0x0002:Success: set

Ezsp Config: set TC addr cache to 0x0002:Success: set

Ezsp Config: set stack profile to 0x0002:Success: set

Ezsp Config: set MAC indirect TX timeout to 0x1E00:Success: set

Ezsp Config: set max hops to 0x001E:Success: set

Ezsp Config: set tx power mode to 0x8000:Success: set

Ezsp Config: set supported networks to 0x0001:Success: set

Ezsp Policy: set binding modify to "allow for valid endpoints & clusters only":Success: set

Ezsp Policy: set message content in msgSent to "return":Success: set

Ezsp Value : set maximum incoming transfer size to 0x00000052:Success: set

Ezsp Value : set maximum outgoing transfer size to 0x00000052:Success: set

Ezsp Config: set binding table size to 0x0010:Success: set

Ezsp Config: set key table size to 0x0000:Success: set

Ezsp Config: set max end device children to 0x0020:Success: set

Ezsp Config: set aps unicast message count to 0x000A:Success: set

Ezsp Config: set broadcast table size to 0x000F:Success: set

Ezsp Config: set neighbor table size to 0x0010:Success: set

NCP supports maxing out packet buffers

Ezsp Config: set packet buffers to 255

Ezsp Config: set end device poll timeout to 0x0005:Success: set

Ezsp Config: set end device poll timeout shift to 0x0006:Success: set

Ezsp Config: set zll group addresses to 0x0000:Success: set

Ezsp Config: set zll rssi threshold to 0xFF80:Success: set

Ezsp Config: set transient key timeout to 0x00B4:Success: set

Ezsp Endpoint 1 added, profile 0x0104, in clusters: 8, out clusters 19

Ezsp Endpoint 242 added, profile 0xA1E0, in clusters: 0, out clusters 1

Found 0 files

Z3GatewayHost>**network leave**

leave 0x70

Z3GatewayHost> **plugin network-creator-security open-network**

NWK Creator Security: Open network: 0x01

Z3GatewayHost>plugin network-creator start 1

NWK Creator: Form: 0x00

NWK Creator Security: Start: 0x00

NWK Creator: Form. Channel: 20. Status: 0x00

NWK Creator: Stop. Status: 0x00. State: 0x00

EMBER\_NETWORK\_UP 0x0000

Z3GatewayHost>

Now turn on a zigbee end device and it will be start broadcasting and will connect above network and on host side below log will occur.

Z3GatewayHost>Trust Center Join Handler: status = UNsecured join, decision = use preconfigured key (00), shortid 0x2229

T0000007E:RX len 3, ep FF, clus 0x0003 (Identify) FC 01 seq 00 cmd 01 payload[]

T0000007E:RX len 4, ep 01, clus 0x0500 (IAS Zone) FC 08 seq 00 cmd 04 payload[00 ]

T0000007F:RX len 15, ep 01, clus 0x0500 (IAS Zone) FC 08 seq 01 cmd 01 payload[10 00 00 F0 16 BC 1E FE FF 9F FD 90 ]

T0000007F:RX len 12, ep 01, clus 0x0019 (Over the Air Bootloading) FC 01 seq 01 cmd 01 payload[00 31 11 24 10 21 51 00 23 ]

QueryNextImageRequest mfgId:0x1131 imageTypeId:0x1024, fw:0x23005121

T0000007F:RX len 20, ep 01, clus 0x0500 (IAS Zone) FC 08 seq 02 cmd 01 payload[00 00 00 30 00 01 00 00 31 15 00 02 00 00 19 20 00 ]

T0000008F:RX len 7, ep 01, clus 0x0500 (IAS Zone) FC 09 seq 02 cmd 01 payload[15 00 31 11 ]

Sent enroll response with responseCode: 0x00, zoneId: 0x00, status: 0x00

Below log is for door sensor device which indicates open<**24**> and close<**25**>:

T00000092:RX len 9, ep 01, clus 0x0500 (IAS Zone) FC 09 seq 03 cmd 00 payload**[24** 00 00 00 00 00 ]

T0000009A:RX len 9, ep 01, clus 0x0500 (IAS Zone) FC 09 seq 04 cmd 00 payload**[25** 00 00 00 00 00 ]

T000000A0:RX len 9, ep 01, clus 0x0500 (IAS Zone) FC 09 seq 05 cmd 00 payload[24 00 00 00 00 00 ]

T000000A2:RX len 9, ep 01, clus 0x0500 (IAS Zone) FC 09 seq 06 cmd 00 payload[25 00 00 00 00 00 ]

T000000A5:RX len 9, ep 01, clus 0x0500 (IAS Zone) FC 09 seq 07 cmd 00 payload[24 00 00 00 00 00 ]

T000000A7:RX len 9, ep 01, clus 0x0500 (IAS Zone) FC 09 seq 08 cmd 00 payload[25 00 00 00 00 00 ]

T000000A8:RX len 9, ep 01, clus 0x0500 (IAS Zone) FC 09 seq 09 cmd 00 payload[24 00 00 00 00 00 ]

T000000A9:RX len 9, ep 01, clus 0x0500 (IAS Zone) FC 09 seq 0A cmd 00 payload[25 00 00 00 00 00 ]

T000000B2:RX len 9, ep 01, clus 0x0500 (IAS Zone) FC 09 seq 0B cmd 00 payload[21 00 00 00 00 00 ]

T000000B4:RX len 9, ep 01, clus 0x0500 (IAS Zone) FC 09 seq 0C cmd 00 payload[25 00 00 00 00 00 ]

T000000C4:RX len 3, ep FF, clus 0x0003 (Identify) FC 01 seq 0D cmd 01 payload[]

T000000C8:RX len 3, ep FF, clus 0x0003 (Identify) FC 01 seq 0E cmd 01 payload[]

## USER LED

* Run the below command to control the Led

#BT\_LED

echo 96 > /sys/class/gpio/export

echo out > /sys/class/gpio/gpio96/direction

cat /sys/class/gpio/gpio96/value

echo 1 > /sys/class/gpio/gpio96/value

# WIFI\_LED

echo 97 > /sys/class/gpio/export

echo out > /sys/class/gpio/gpio97/direction

cat /sys/class/gpio/gpio97/value

echo 1 > /sys/class/gpio/gpio97/value

#LED\_1

echo 117 > /sys/class/gpio/export

echo out > /sys/class/gpio/gpio117/direction

cat /sys/class/gpio/gpio117/value

echo 1 > /sys/class/gpio/gpio117/value

# LED\_2

echo 118 > /sys/class/gpio/export

echo out > /sys/class/gpio/gpio118/direction

cat /sys/class/gpio/gpio118/value

echo 1 > /sys/class/gpio/gpio118/value

# LED\_3

echo 124 > /sys/class/gpio/export

echo out > /sys/class/gpio/gpio124/direction

cat /sys/class/gpio/gpio124/value

echo 1 > /sys/class/gpio/gpio124/value

# LED\_4

echo 125 > /sys/class/gpio/export

echo out > /sys/class/gpio/gpio125/direction

cat /sys/class/gpio/gpio125/value

echo 1 > /sys/class/gpio/gpio125/value

## Low Power Expansion GPIO

* Run the below command to control the Led

#LS\_GPIO2\_A

echo 42 > /sys/class/gpio/export

echo out > /sys/class/gpio/gpio42/direction

cat /sys/class/gpio/gpio42/value

echo 1 > /sys/class/gpio/gpio42/value

echo 0> /sys/class/gpio/gpio42/value

# LS\_GPIO2\_B

echo 43 > /sys/class/gpio/export

echo out > /sys/class/gpio/gpio43/direction

cat /sys/class/gpio/gpio43/value

echo 1 > /sys/class/gpio/gpio43/value

echo 0 > /sys/class/gpio/gpio43/value

# LS\_GPIO3\_C

echo 88 > /sys/class/gpio/export

echo out > /sys/class/gpio/ gpio88/direction

cat /sys/class/gpio/ gpio88/value

echo 1 > /sys/class/gpio/gpio88/value

# LS\_GPIO3\_D

echo 84 > /sys/class/gpio/export

echo out > /sys/class/gpio/ gpio84/direction

cat /sys/class/gpio/ gpio84/value

echo 1 > /sys/class/gpio/gpio84/value

# LS\_GPIO2\_E

echo 39 > /sys/class/gpio/export

echo out > /sys/class/gpio/ gpio39/direction

cat /sys/class/gpio/ gpio39/value

echo 1 > /sys/class/gpio/gpio39/value

# LS\_GPIO3\_F

echo 85 > /sys/class/gpio/export

echo out > /sys/class/gpio/ gpio85/direction

cat /sys/class/gpio/ gpio85/value

echo 1 > /sys/class/gpio/gpio85/value

# LS\_GPIO2\_G

echo 40 > /sys/class/gpio/export

echo out > /sys/class/gpio/ gpio40/direction

cat /sys/class/gpio/ gpio40/value

echo 1 > /sys/class/gpio/gpio40/value

# LS\_GPIO3\_H

echo 86 > /sys/class/gpio/export

echo out > /sys/class/gpio/ gpio86/direction

cat /sys/class/gpio/ gpio86/value

echo 1 > /sys/class/gpio/gpio86/value

# LS\_GPIO3\_I

echo 76 > /sys/class/gpio/export

echo out > /sys/class/gpio/ gpio76/direction

cat /sys/class/gpio/ gpio76/value

echo 1 > /sys/class/gpio/gpio76/value

# LS\_GPIO1\_J

echo 77 > /sys/class/gpio/export

echo out > /sys/class/gpio/gpio77/direction

cat /sys/class/gpio/gpio77/value

echo 1 > /sys/class/gpio/gpio77/value

# LS\_GPIO3\_K

echo 5 > /sys/class/gpio/export

echo out > /sys/class/gpio/gpio5/direction

cat /sys/class/gpio/gpio5/value

echo 1 > /sys/class/gpio/gpio5/value

# LS\_GPIO1\_L

echo 3 > /sys/class/gpio/export

echo out > /sys/class/gpio/gpio3/direction

cat /sys/class/gpio/gpio3/value

echo 1 > /sys/class/gpio/gpio3/value

## CAN Interface demo

* CAN interface can be tested by communicating between two boards.
* Connect 2 boards with supplied CAN cable.
* Enable CAN in both the boards with below command with board’s console.
  + **ip link set can0 type can bitrate 125000;ifconfig can0 up**
* Configure one board as receiver as below command.
  + **candump can0 &**
* Configure another board as sender and send data using below command
  + **cansend can0 18FC2A00#0100000000000000**
* Observe on receiver side board receiving data

## NOR Flash demo

* Go to Board’s console and create text file and write some data into it by below command
  + **vi write.txt**
* Once done writing save and quit the above file by below command.
  + **<ESC><:><wq>**
* Check for the Nor flash node by below command
  + **ls –l /dev/mtd0**
* Erase NOR flash using below command
  + **flash\_eraseall /dev/mtd0**
* Write the created file into NOR flash using below command.
  + **time dd if=write.txt of=/dev/mtd0**
* Read from NOR flash from the same location
  + **dd if=/dev/mtd0 of=read.txt**
  + **cat read.txt**
* Compare the read.txt, it should be same as write.txt

# Known Issues and Limitations

* Please refer the Release note V0.2

# Contact US

*For any queries related to product, please contact us at arrow.imx8HMi@einfochips.com*