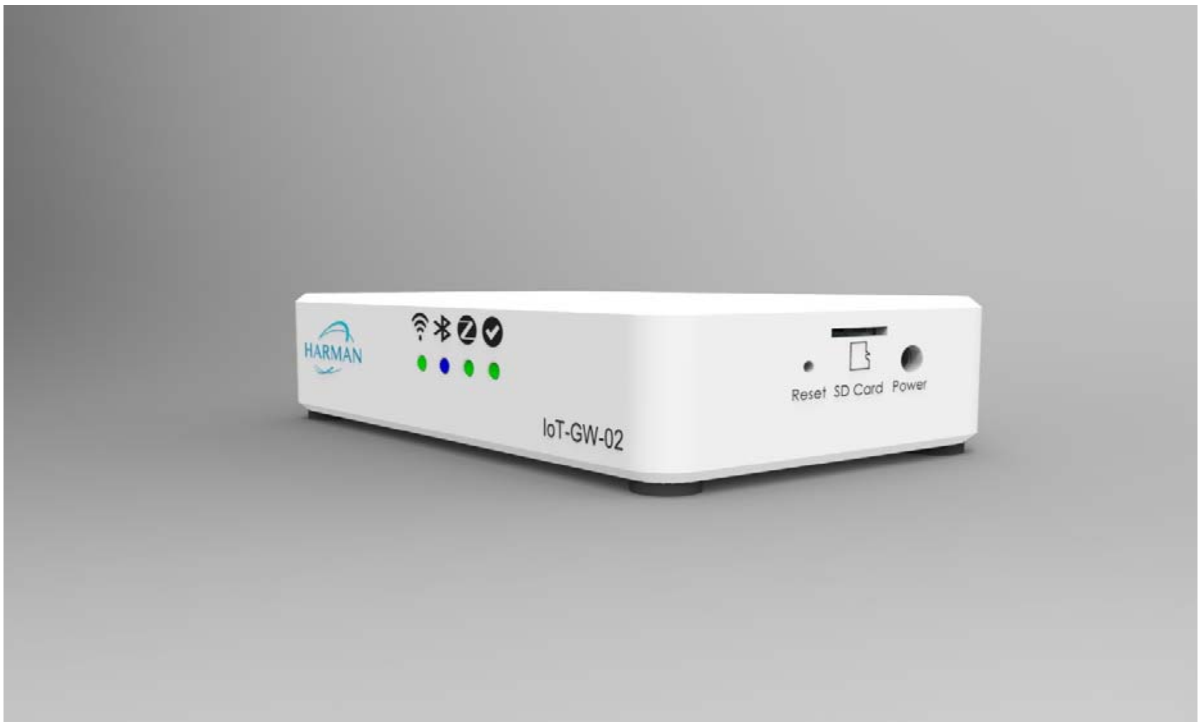
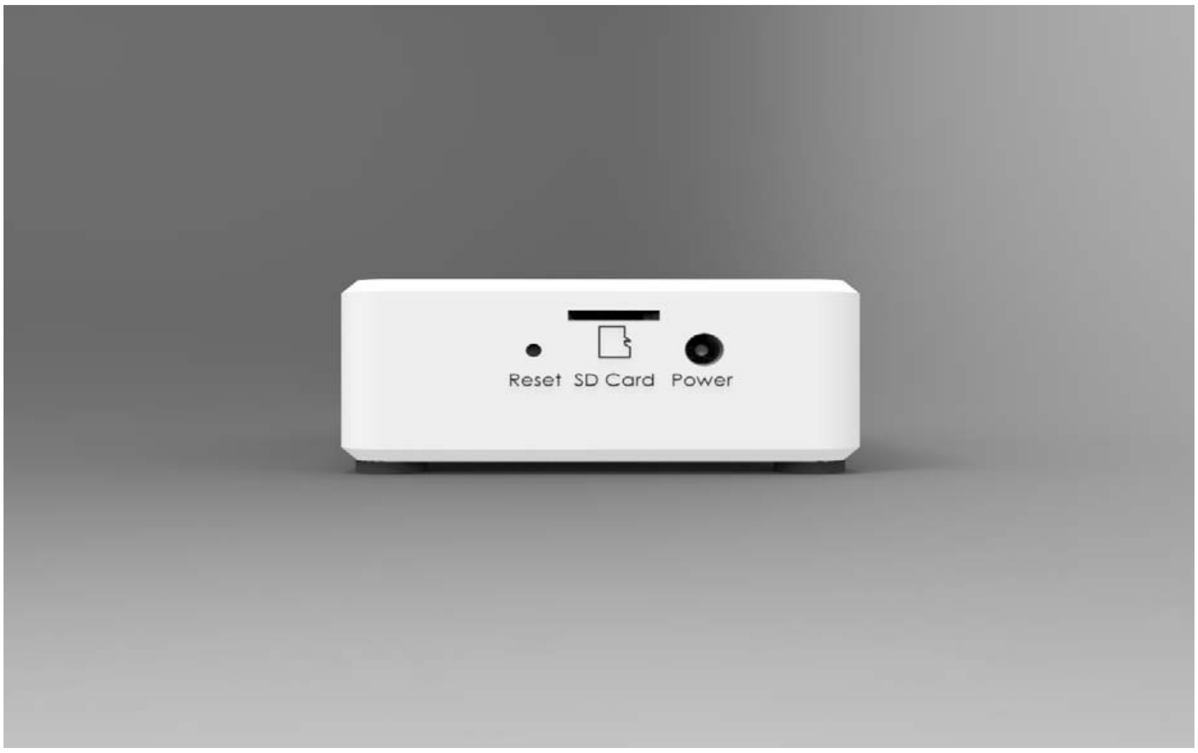
STCT_IOG1_USER_GUIDE

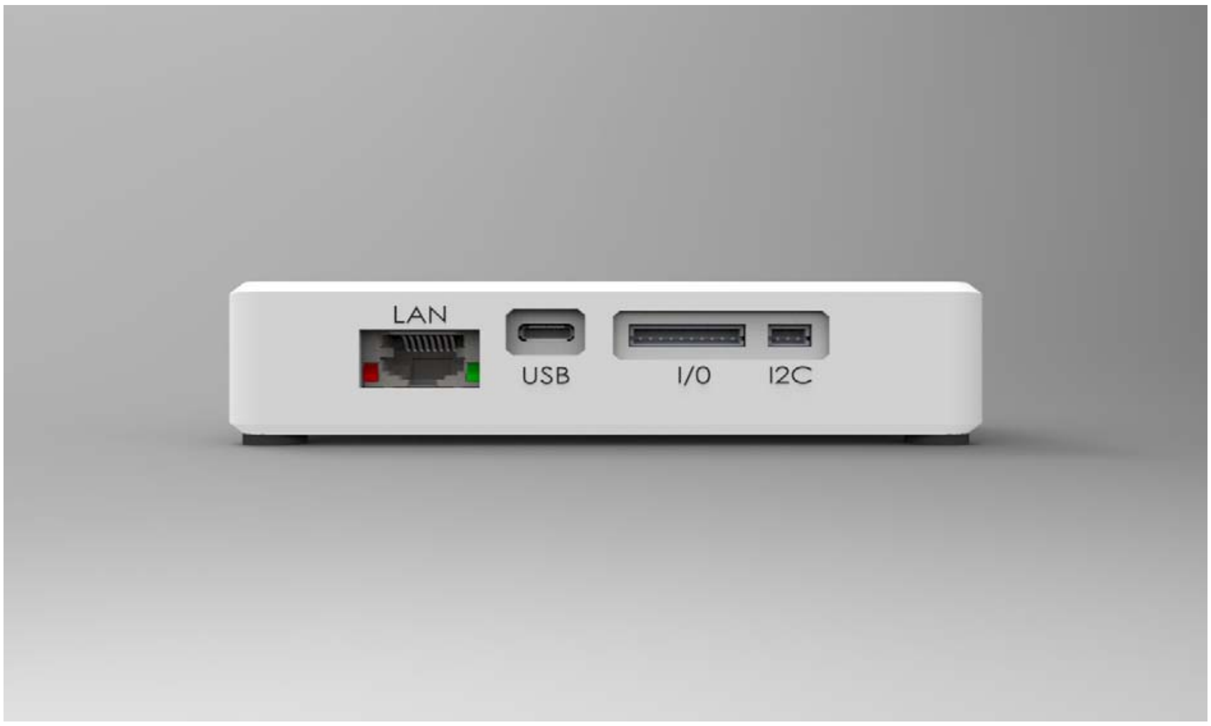
Rev: C1

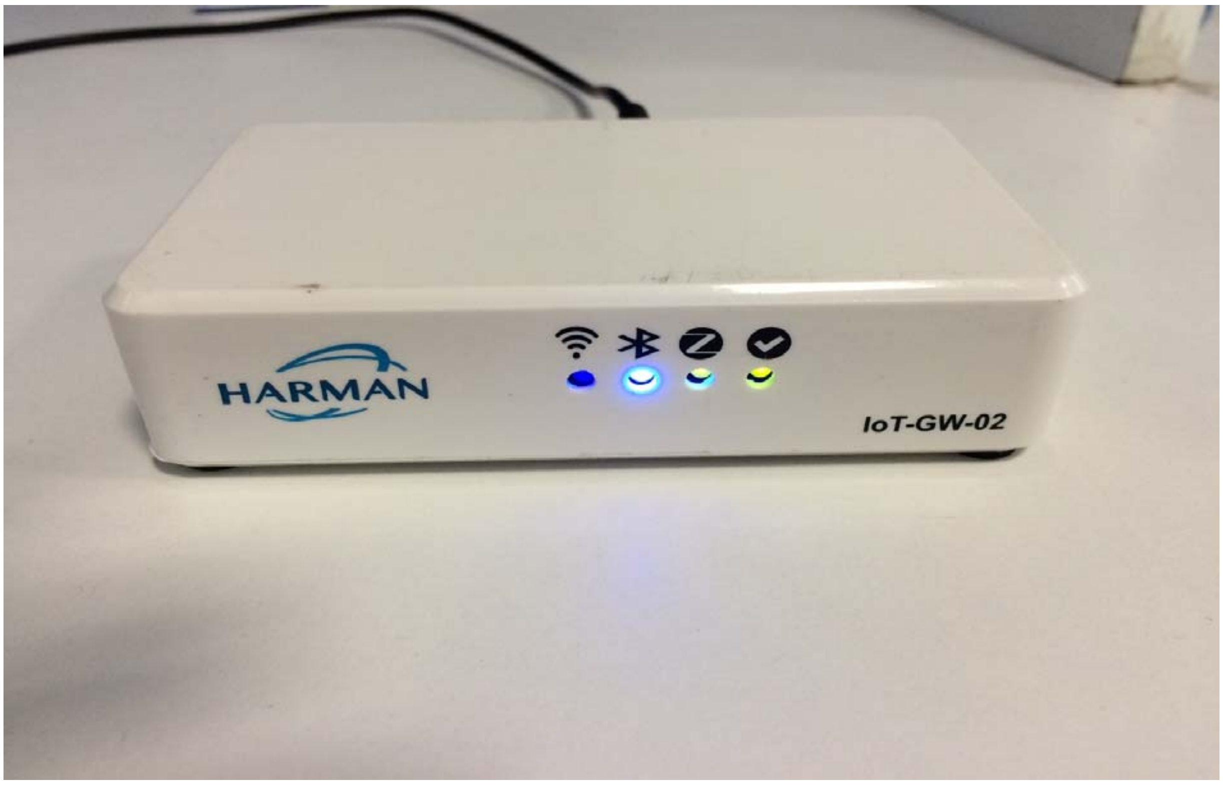
15 AUG 2016

LAN
USB
I/O
I2C


HARMAN
IoT-GW-02
Reset
SD Card
Power


Reset
SD Card
Power


LAN
USB
I/0
I2C


HARMAN
IoT-GW-02

Revision History:

Date	**Rev No.**	**Description**	**By**
16 Feb 2015	A0-01	Initial document	Anbu.Sankar
17 Feb 2015	A0-02	Updated after internal review & Added testing interfaces, Compilation procedures	Anbu.Sankar
25 Feb 2015	A0-03	Added steps for tftpboot of ramdisk images	BibinBalachandran
16 Mar 2015	A0-04	Updated commands for 64MB SPI flash	AlokPawar
18 May 2015	A0-05	Added SD card boot mode	AlokPawar
19 May 2015	A0-06	Updated diagram	Alok Pawar
15 Aug 2016	C1	Added USB Hub, I2C, 8 I/O GPIO	Alok Pawar

Certification

FCC CE

FCC Statement

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

-Reorient or relocate the receiving antenna.

-Increase the separation between the equipment and receiver.

-Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.

-Consult the dealer or an experienced radio/TV technician for help.

To assure continued compliance, any changes or modifications not expressly approved by the party responsible for compliance could void the user’s authority to operate this equipment.

(Example- use only shielded interface cables when connecting to computer or peripheral devices)

FCC Radiation Exposure Statement

This equipment complies with FCC RF radiation exposure limits set forth for an uncontrolled

Environment. This equipment should be installed and operated with a minimum distance of 20 centimeters between the radiator and your body.

This transmitter must not be co-located or operating in conjunction with any other Antenna or transmitter.

The antennas used for this transmitter must be installed to provide a separation distance of at least 20 cm from all persons and must not be co-located or operating in conjunction with any other antenna or transmitter.

This equipment complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

(1) This device may not cause harmful interference, and

(2) This device must accept any interference received, including interference that may cause undesired operation.

Caution!

The manufacturer is not responsible for any radio or TV interference caused by unauthorized modifications to this equipment. Such modifications could void the user authority to operate the equipment.

Table of Contents

Table of Figures

1 Introduction

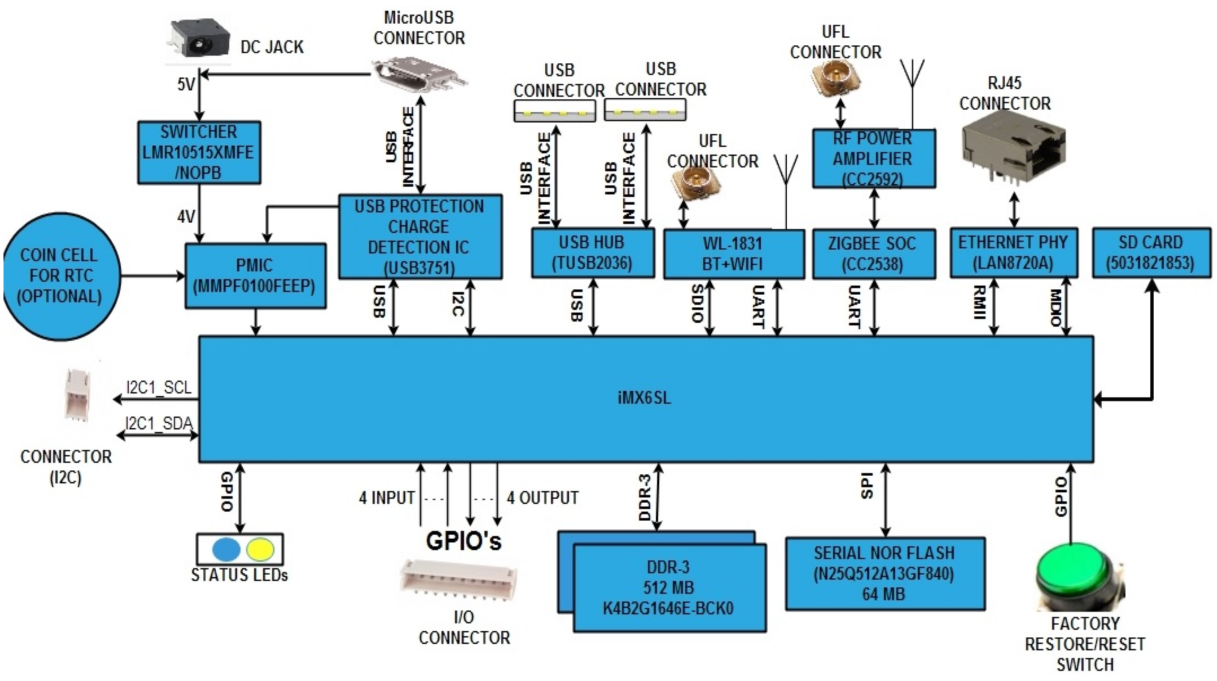


Fig 1: Hardware Block Diagram

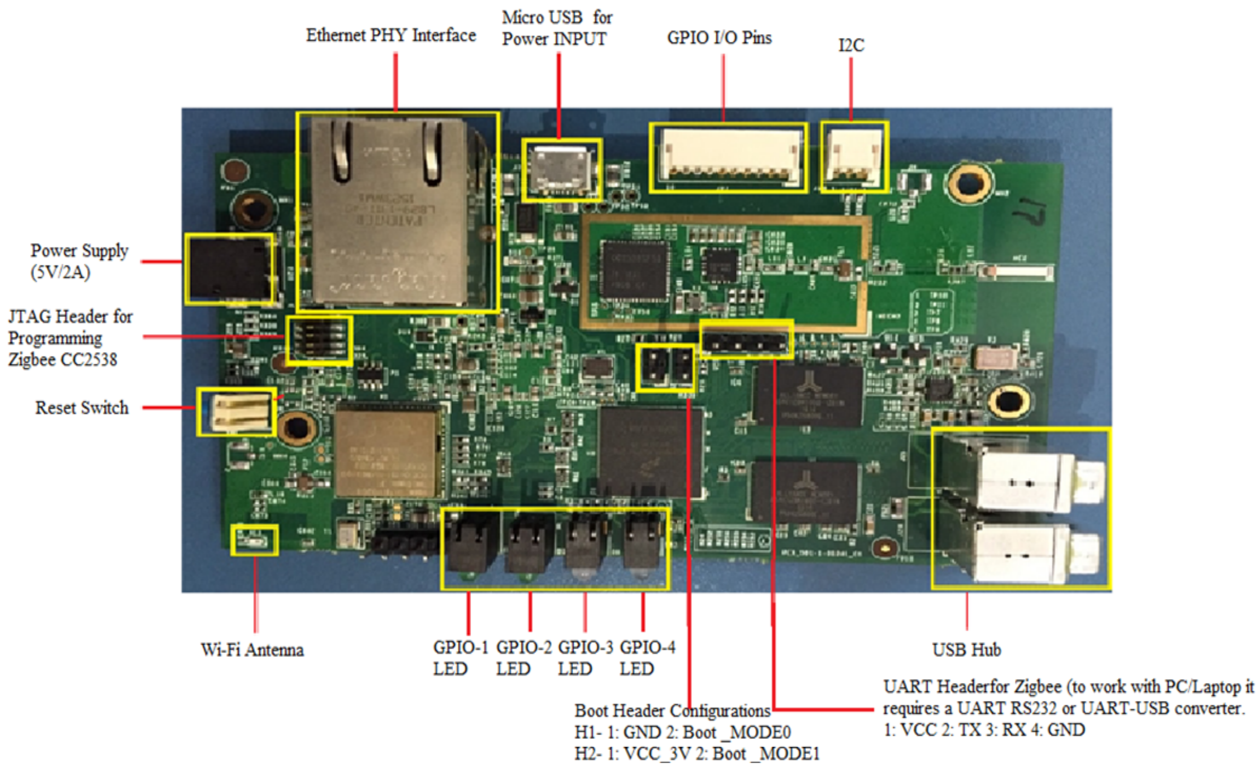

Ethernet PHY Interface
Micro USB for Power INPUT
GPIO I/O Pins
I2C
Power Supply (5V/2A)
JTAG Header for Programming Zigbee CC2538
Reset Switch
Wi-Fi Antenna
GPIO-1 LED
GPIO-2 LED
GPIO-3 LED
GPIO-4 LED
USB Hub
UART Headerfor Zigbee (to work with PC/Laptop it requires a UART RS232 or UART-USB converter.
1: VCC 2: TX 3: RX 4: GND
Boot Header Configurations
H1- 1: GND 2: Boot _MODE0
H2- 1: VCC_3V 2: Boot _MODE1

Fig 2: Board Details

The STCT_IOG1 board can act as an automation solution. IOT Gateway functions as a gateway between a ZigBee network and an IP network through Ethernet and Wi-Fi. On the ZigBee network side, it acts as a node talking to a ZigBee enabled devices. On the IP network side, Gateway acts as a control node for the ZigBee devices from smart devices like phones/tablets.

2 Bringup Procedures

The board can be booted in 2 modes

1. USB boot mode

2. SPI boot mode

Boot mode is selected by the headers H1 and H2. If user need usb boot mode he need to place jumpers on both H1 and H2 headers.

If no jumpers are placed on H1 and H2 the board will be in SPI boot mode.

Four images needs to be programmed to the board.

Images	Descriptions
iog1_uboot	Boot loader image
iog1_uImage	Linux Kernel Image
iog1_ramdisk	File system image
iog1_dtb	Device tree image

2.1 USB boot mode

This mode is used to boot the board from USB. User will be able to boot up to bootloader(u-boot) using this mode. This boot mode is intended for first time programming of the board.

2.1.1 Pre-requirements

- Micro usbcable :- This is used for powering the board initially and loading image for USB boot mode.
- A linuxPC :- 'imx_usb' tool needs to be executed on linux PC for booting. (Tested in Ubuntu 12.04 & Ubuntu 14.04LTS)
- Ethernet cable :- For using TFTP in order to program linux images from uboot console. Default IP address of board is 192.168.1.200.
- UART console cable:- serial console cable for accessing the board console in PC(through terminal). Baud rate is 115200.

2.1.2 Board detection in PC

If the board is connected to the PC in USB boot mode it will listed as "Freescale Semiconductor, Inc" in the output of lsusb command.

2.1.3 Loading Uboot image

An image for uboot is loaded into RAM, using usb boot mode and imx_usb_loader tool.

Procedures:

1. Open two terminal. In one terminal open imx_usb_loader tool and in another terminal open minicom for serial console of board.

2.To load uboot image, following command is used.
This command should be executed in imx_usb_loader tool folder.(It is available at tools directory in the source code).

sudo ./imx_usb<path_for_uboot_image>

3.In other terminal(minicom), We can see the process of uboot loading. Stop the uboot process by pressing any button, at the end stage of uboot while it is waiting for few seconds(default 3 or 5 sec). Now we can get uboot prompt. Otherwise uboot will try to load other images. If user forget to stop uboot at this stage, no issue. You can get the prompt by pressing any button. Better stop the uboot within that waiting time.

2.1.4 Writing images into flash

Now uboot image is loaded into RAM only. We have towrite uboot and other images into flash memory permanently in order to eliminate usb boot mode further.

To write images into flash memory tftp server is needed in PC. All images should be present intftp folder. (For installing tftp follow this link:https://mohammadthalif.wordpress.com/2010/03/05/installing-and-testing-tftpd-in-ubuntudebian/. This is only for example. User can follow any tftp installation method).To write images into flash memory following commands are used.

Programming to Flash :

==

Writing uboot image

==

tftp 0x80800000 iog1_uboot
sf probe
sf erase 0x0 0x40000
sf write 0x80800000 0x400 0x40000

After writing uboot image, reboot the board by power off/on. Now uboot automatically will be loaded. Stop uboot at the end stage as discussed earlier.

==

Writing kernel image

==

tftp 0x82000000 iog1_uImage
sf probe
sf erase 0x100000 0x400000
sf write 0x82000000 0x100000 0x400000

==

Writing ramdisk (file system) image

==

tftp 0x82600000 iog1_ramdisk
sf probe
sf erase 0x500000 0x1400000
sf write 0x82600000 0x500000 0x1400000

==

Writing device tree image

==

tftp 0x88000000 iog1_dtb
sf probe
sf erase 0x1900000 0x20000
sf write 0x88000000 0x1900000 0x20000

==

##NOTE::: In these commands, names and memory ranges may vary later based on image name and size.

2.1.5 Setting environment variables

After writing all images into flash, we have to set environment variables according to written images. Based on these environment variables only, uboot will load all other images.

```
================================================================
setenv bootargs 'console=ttymxc0,115200 root=/dev/ram rw ramdisk_size=80000'

Setenv loadimages 'sf probe;sf read 0x82600000 0x500000 0x1400000;sf read
0x88000000  0x1900000 0x20000;sf read 0x82000000 0x100000 0x400000'

Setenv bootcmd 'run loadimages;bootm 0x82000000 0x82600000 0x88000000'

Setenv ipaddr ‘192.168.1.200’;setenv serverip ‘192.168.1.10’;setenv
ethaddr ’00:50:C2:BC:C0:F1’

saveenv
================================================================
```

After setting all environment variables,reboot the board by power off/on. Now images will automatically load. Finally we can get, gateway command prompt. Give input for username as **root**. Then prompt will be displayed as **root@imx6slevk:~#.**

2.2 SPI boot mode

In this mode board will boot from the images on SPI NOR flash.The boards once programmed with bootloader images will boot in this mode. User can update linux images from uboot prompt using tftp.

2.2.1 Pre-requirements

- Micro usb cable :- This is used for only powering the board in thismode.
- UART console cable :- serial console cable for accessing the board in PC(through terminal). Baud rate is 115200.
- Ethernet cable :- For upgrading images using tftp ,if required. Default IP address of board is 192.168.1.200.

2.2.2 Loading images from flash

If the board is programmed with all images, then images will be loaded and gateway command prompt will be appeared automatically as already board is programmed when the board is powered up.

2.3 SD boot mode

In this mode, board will boot from the images on SPI NOR flash and file system will be load from SD card.The boards once programmed with bootloader images will boot in this mode. User can update linux images from uboot prompt using tftp.

2.3.1 Loading images from flash and filesystem from SD card

If the board is programmed with all images, then images will be loaded and gateway command prompt will be appeared automatically as already board is programmed when the board is powered up.

2.3.2 Setting environment variables for SD card

```
================================================================

Setenv bootargs 'console=ttymxc0,115200 root=/dev/mmcblk0p1 rootfstype=ext3 rootwait rw'

setenv bootcmd_sd 'sf probe; sf read 0x82000000 0x100000 0x400000;sf read 0x88000000 0x1900000 0x20000;bootm 0x82000000 - 0x88000000'

setenv bootcmd 'run bootcmd_sd'

saveenv

================================================================
```

2.3.3 File system in SD card

Following are the steps for copying ‘filesystem’ in SD card

- Format the SD card in ext3 format using application.
  $ sudo gparted
  { Note: Format the card using ‘gparted’ and if not installed then install using
  $ sudo apt-get install gparted
  }
- After format in ext3 type, remove and mount SD card again to Linux system.
- Go to the image directory s*ource/yocto/build/tmp/deploy/images/imx6slzbha/*
- Check the SD card partition and untar the filesystem inside the SD card

$ tar –xvf*core-image-minimal-imx6slzbha.tar.bz2 –C /media/<card_partition>*

- After untar the filesystem, run
  $ Sync
- Then unmount the SD card

3 Testing Interfaces

1.WiFi Interface

2.Bluetooth Interface

3.GPIO (LED) Interface

4.ZigBee Interface

5. jffs2 partition

3.1 WiFi Interface

To check WiFi interface run the IOTG_WIFI.sh acript at /home/root/ using following command.

sh ./ IOTG_WIFI.sh

It will create a WiFi access point (AP). We can get AP name as**SoftAP_iog1** while searching for WiFi networks. Default ip address is 10.4.30.34.

3.2 Bluetooth Interface

In the command prompt ***root@imx6slevk*****:~#**, execute following commands.

hciconfig hci0 up

/usr/sbin/bluetoothd –n &

hciconfig hci0 piscan&> /dev/null

hcitool scan

Now, Bluetooth will scan all the devices and will display all the Bluetooth devices in the terminal.
To check the range, keep Bluetooth devices in some distance, search again.

3.3 GPIO Interface

To access any GPIO pin by user, user need to export particular gpio to user space. Then user must set direction (in & out) for particular pin based on requirement. Then user can set the values to particular gpio pin.

For testing the GPIO pins, execute following commands.

```
echo XX > /sys/class/gpio/export
echo "out / in" > /sys/class/gpio/gpioXX/direction
echo 0 > /sys/class/gpio/gpioXX/value
echo 1 > /sys/class/gpio/gpioXX/value
```

Note:- XX is calculated using the GPIO bank number.(Bank-1)*32+GPIO no in the bank.

3.3.1 LED GPIO

GPIO**X**_IO**YY**→ X- bank number; YY- gpio number;

GPIO1_IO14 - Green Led
GPIO1_IO15 - Green Led
GPIO1_IO16 - Blue Led
GPIO1_IO17 & GPIO1_IO18 - Bicolour Led

Example commands for GPIO1_IO14

XX=(1-1)*32+14 = 14

```
echo 14 > /sys/class/gpio/export
echo "out" > /sys/class/gpio/gpio14/direction
echo 0 > /sys/class/gpio/gpio14/value
echo 1 > /sys/class/gpio/gpio14/value
```

0 – for turn off LED; 1 – for turn on LED;

If values(1 or 0) will be set to pin numbers 17 and 18 alternatively, then the Bi color led will change the color.

3.3.2 Input GPIO

User can access 4 input gpio, which are following

GPIO4_IO21 : Gpio number 117
GPIO4_IO19 : Gpio number 115
GPIO3_IO22 : Gpio number 86
GPIO3_IO25 : Gpio number 89

Example of Input GPIO4_IO21

XX=(4-1)*32+21 = 117

For checking the value of Input Gpio :

Cat /sys/class/gpio117/value

3.3.3 Output GPIO

Use can access 4 output Gpio, which are following

GPIO3_IO20	Gpio number 84
GPIO3_IO29	Gpio number 93
GPIO4_IO02	Gpio number 98
GPIO5_IO05	Gpio number 101

Example commands for GPIO3_IO29

XX=(3-1)*32+29 = 93

```
echo 193 > /sys/class/gpio/export
echo "out" > /sys/class/gpio/gpio93/direction
echo 0 > /sys/class/gpio/gpio93/value
echo 1 > /sys/class/gpio/gpio93/value
```

3.4ZigBee Interface

To check ZigBee interface open two telnet sessions to from the PC to board. (board default IP is 192.168.1.200).

1. On the first telnet window change the directory to /home/root/zigbee/servers.

Run the script using following command

```
sh ./zigbeeHAgwbbb
```

2. On the second terminal execute change the directory to home/root/zigbee/servers.

Run the Zigbee testing application using the command

```
./start_application
```

Now a Demo app will be displayed which can be used to view the zigbee devices and control them. For Help type '?' on the demo app.

Note: Some Useful commands for providing access and permission for zigbee over telnet. Need to be run on command prompt **root@imx6slevk:~#**,

ln –sf /bin/login /usr/bin/login

chmod 777 –R zigbee/

3.5 *jffs2* Partition

In SPI flash there are a 41MB jffs2 partition which can be used as storage for files. And file will be present there on every boot till then we delete those from jffs2 partition.

For mounting that partition following commands need to run from command prompt **root@imx6slevk:~#**,

mount –t jffs2 /dev/mtdblock6 /mnt

cd /mnt

Note: If you want to erased full jffs2 partition then following command should run from uboot

sf probe

sf erase 0x1a00000 0x2400000

3.6 USB HUB

Board contains 2 USB host port, for USB port testing connect the pendrive or any device for verifying the port. After adding the device on USB port some prints will be shown on console.

3.7 I2C Interface

I2C-2 is verified using the i2c-tools on kernel level using i2cdetect ommand. Following is the command

```
$ i2cdetect –y –a 2
```

4 Compilation procedures

This source package is compiled and tested with Ubuntu 12.04 and 14.04LTS. Following packages should be installed in your PC.

gawk, wget, git-core, diffstat,
unzip, texinfo, gcc-multilib,
build-essential, chrpath,
libsdl1.2-dev, apt-file

1. Go to source/yocto directory

   cd source/yocto/

2. Give command

   MACHINE=imx6slzbha source ./setup-environment build

This command sets the build environment and automatically you will be moved to build directory.

3. Give following command in build directory inorder to compile the images.

   bitbake core-image-minimal

NOTE: Yocto first time build will take more time and you need a working internet connection for this build.

You will get images in the directory
source/yocto/build/tmp/deploy/images/imx6slzbha/

The images are

u-boot.imx- Boot loader image
uImage – Linux Kernel image
uImage-imx6sl-evk-ldo.dtb – Device tree image
core-image-minimal-imx6slzbha.tar.bz2 – File system image(This need to be converted to ramdisk format)

You can copy the uboot, uImage and dtb images to tftpboot folder in order to flash them. While copying rename them as follows for matching with the nor flash commands explained in section 2.1.4.

rename 'u-boot.imx' to 'iog1_uboot'
rename 'uImage' to 'iog1_uImage'
rename 'uImage-imx6sl-evk-ldo.dtb' to 'iog1_dtb'

The file system image needs to be converted to ramdisk format. For this change directory to tools.

Steps are as follows

1.Go to directory *tools*

2. you create a directory called *ramdisk* in the directory *tools.*

3. untar the filesystem image to this ramdisk folder using the command

sudo tar -xjf ../source/yocto/build/tmp/deploy/images/imx6slzbha/core-image-minimal-imx6slzbha.tar.bz2 -C ramdisk/

4.Run command

sh create_ramdisk.sh

5. you will get *uramdisk.image.gz* in tools directory.

You can copy this image to tftpfolder for programming. While copying rename this file as *iog1_ramdisk* for flashing to board.

5. **Creating Ramdisk image with extra applications and booting this ramdisk using tftpboot.**

Create the ramdisk using the steps mentioned below

1.Go to directory *tools*

2. you create a directory called *ramdisk* in the directory *tools.*

3. untar the filesystem image to this ramdisk folder using the command

sudo tar -xjf ../source/yocto/build/tmp/deploy/images/imx6slzbha/core-image-minimal-imx6slzbha.tar.bz2 -C ramdisk/

4. Copy the necessary files and binaries to the target file system destinations in the ramdisk folder.

5. Now edit the *create_ramdisk.sh* file to change the count parameter. Set the *'count'*parmeter to the required size in Mbyte.

Example:- for creating ramdiskfilesystem of size 256MB edit the create_ramdisk.sh file as follows.

#!/bin/bash

```
set -e

dd if=/dev/zero of=initrd.imgbs=1M count=256

mke2fs -F -v -m0 initrd.img

mkdir -p tmp_mnt

sudo mount -o loop initrd.imgtmp_mnt/

sudocp -raramdisk/* tmp_mnt

sudoumounttmp_mnt/

gzip -9 initrd.img

mkimage -A arm -T ramdisk -C gzip -d initrd.img.gz uramdisk.image.gz

rm initrd.img.gz

rm -rftmp_mnt
```

6. Now run command
   sh create_ramdisk.sh
you will get *uramdisk.image.gz* in tools directory.

7. Copy this image to tftpfolder for programming. While copying rename this file as 'iog1_ramdisk'.

4.1 Booting Board using tftpboot

For **Booting the board using tftpboot** follow these steps.

1. Copy all other images uImage and uImage-imx6sl-zbha-ldo.dtb fromsource/yocto/build/tmp/deploy/images/imx6slzbha/

totftpfolder of the PC for tftpboot. Rename the files as following

'uImage' to 'iog1_uImage'

'uImage-imx6sl-zbha-ldo.dtb' to 'iog1_dtb'

2. Boot the board and stop at uboot(bootloader) prompt by pressing any keys on the PC keyboard. On uboot prompt change the environment variables.

Note:-Following commands are for booting with 256Mbyte ramdisk image. If you want to change the ramdisk image size set the 'ramdisk_size' to required size in Kbytes.

```
Setenv tftpimages 'tftp 0x82600000 iog1_ramdisk;tftp 0x88000000 iog1_dtb;tftp 0x82000000 iog1_uImage'

Setenv bootcmd 'run tftpimages;bootm 0x82000000 0x82600000 0x88000000'

Setenv bootargs 'console=ttymxc0,115200 root=/dev/ram rw ramdisk_size=256000'

saveenv
```

3. Now boot the board by giving 'boot' command and it will boot with the new ramdisk images that are created.

4.2 Patch on SDK

For applying patch on SDK following are the command, need to be run on command prompt ***root@imx6slevk*:~#**,

```
patch –p1  < <patch_file_name.patch>
```