

# **SprloT 6UL Development Platform for Android Things**

**Quick Start Guide**

## 1. Overview

This tutorial helps new developers get started with the NXP based development platform for Android Things – SprIoT 6UL board, and software support. Specifically, it walks through the hardware setup, Android Things image build and board booting process.

Code development, build, and unit testing take place on the developer's host computer. The resulting image is flashed to the target hardware for further integration testing and debugging over USB or Ethernet. Just as Android Things is Android-based, the software development leverages Android development tools including ADB (Android Development Bridge) and FASTBOOT mode to interact with the target.

This development platform together with the Board Support Package software aim to enable faster development of IoT devices based on Android Things, and flexible hardware/software customization needed for the particular device.

## 2. Hardware Requirement

The development kit contains:

- SprIoT 6UL System-On-Module (SOM)
- EVK6UL-Base Board
- DC power supply

The power adapter specification is rated as follows:

- DC output voltage: 5V
- DC output Current: 2A
- DC plug dimensions (mm): 5.5 +/- 0.05 (OD) x 2.1 +/- 0.05 (ID) x 10 +/- 0.3 (L)

Besides, other required materials include:

- Cables:
  - For ADB/FASTBOOT/MFGTool**
    - USB type-A to Micro USB cable
  - Serial console:**
    - USB type-A to Micro USB cable

### 3. Getting Familiar with the Development Platform

The key interfaces of the board are shown in Figure 1:

- Boot mode switch to choose boot mode or download mode for the device.
- 5V power supply interface.
- Power switch.
- Micro USB connector which is used for USB to Serial Console converter for debugging.
- Audio jack.
- USB OTG interface.
- Optional WiFi and BT antenna with U.FL interface.

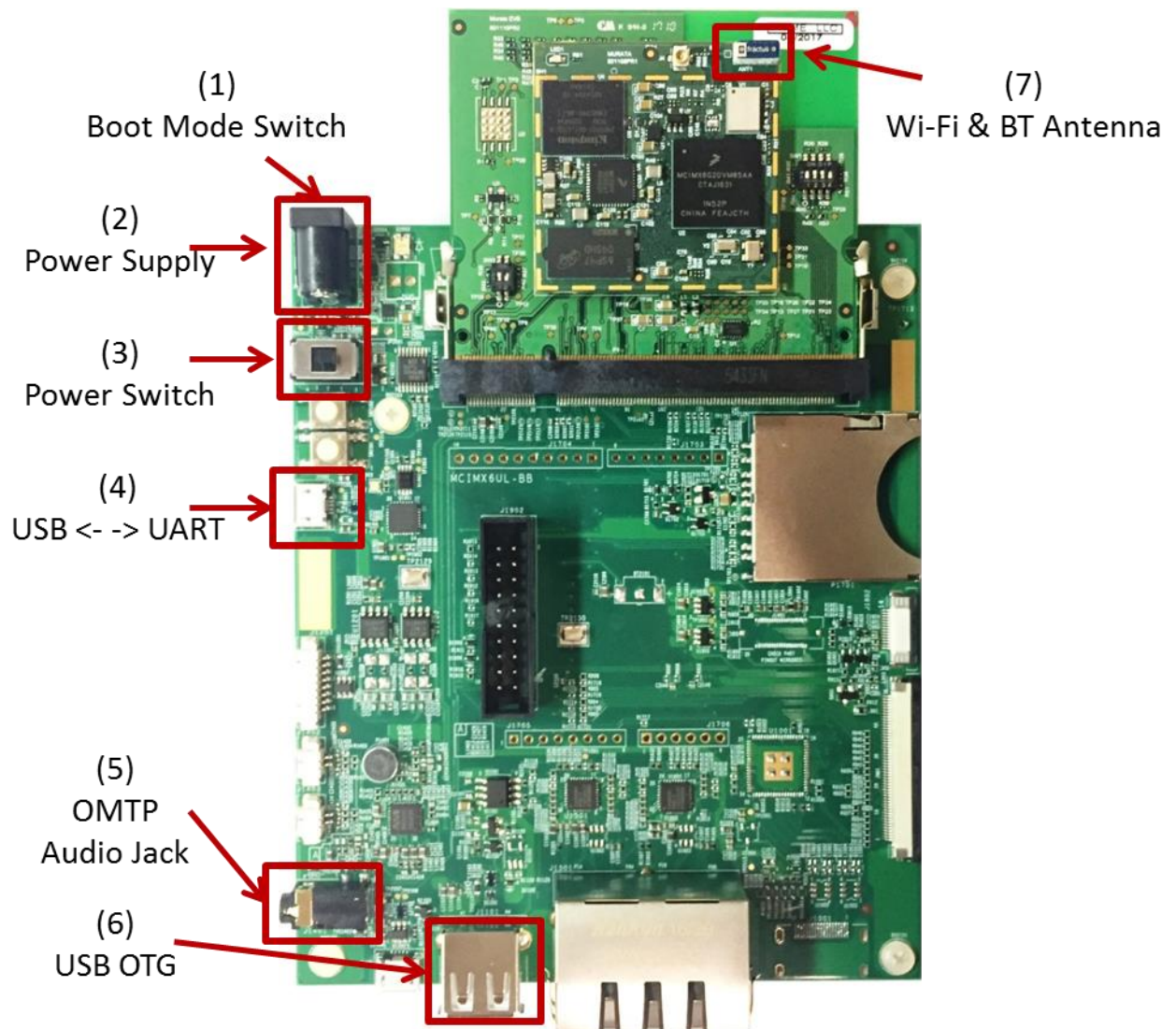


Figure 1. Top view of the SprIoT 6UL SOM and EVK6UL-BB

- Take a close look at the switch on the top view of the board. There are two different setups for download mode and boot mode. Continue reading for further details.

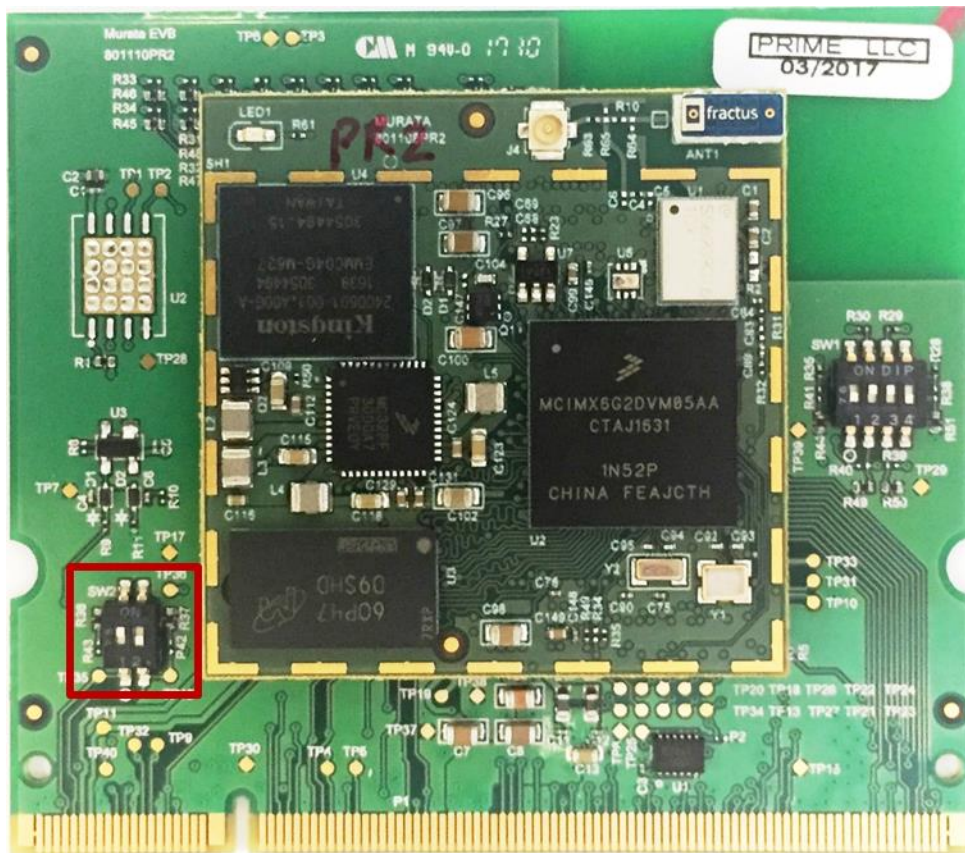


Figure 2a. Switch Setup (Download Mode S1:2'b01)

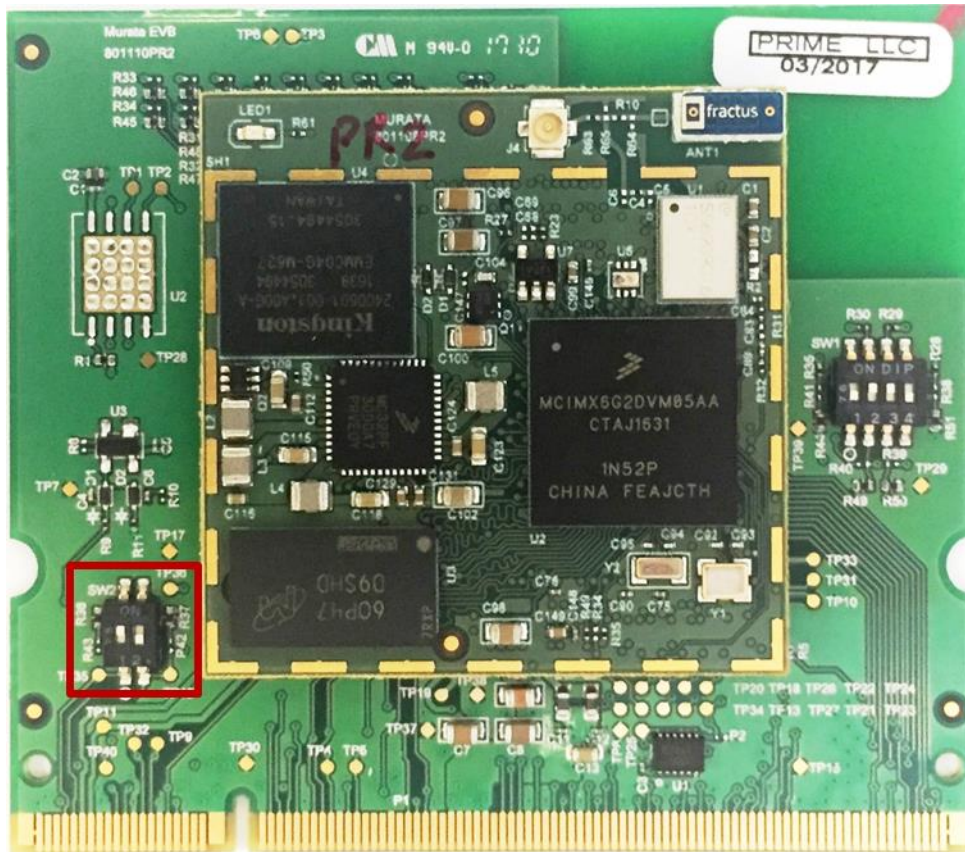


Figure 2b. Jumper Setup (Boot Mode S1:2'b10)

#### 4. Connect the board and host computer

1. Connect the Micro USB to USB type-A wire's Micro USB end to the USB to UART interface (Number 4 in Figure 1). Connect the type-A end to the host PC.
2. Get a USB type-A to micro USB cable. Plug the micro USB end to the micro USB OTG (number 6 in Figure 1) for ADB and FASTBOOT interface. Plug another end of the USB cable to your computer.
3. Plug in the power supply (Number 2 in figure 1) and switch S1(Number 7 in figure 1) to "2'b1b".

#### 5. Instructions to set up the serial console terminal

1. Make sure the you connect to the UART serial console as shown in step 3 in "Connect the board and host computer" section
2. Start the serial communication software
3. Choose operating system of host computer– Window
  - a. Once the PC recognizes the virtual USB to UART device, it can be seen in your PC Device Manager list. You can determine the port number of the virtual COM port by looking under the "Ports" group.



- b. With the serial port driver installed, run your favorite terminal application to view the serial output from i.MX6UL microprocessor's UART.

Recommended tools for serial communication terminal: Putty and Minicom  
**Serial port configuration: 115200 baud rate, 8 data bits, 1 stop bit, no parity.**

Set up serial communication terminal in Putty as below:

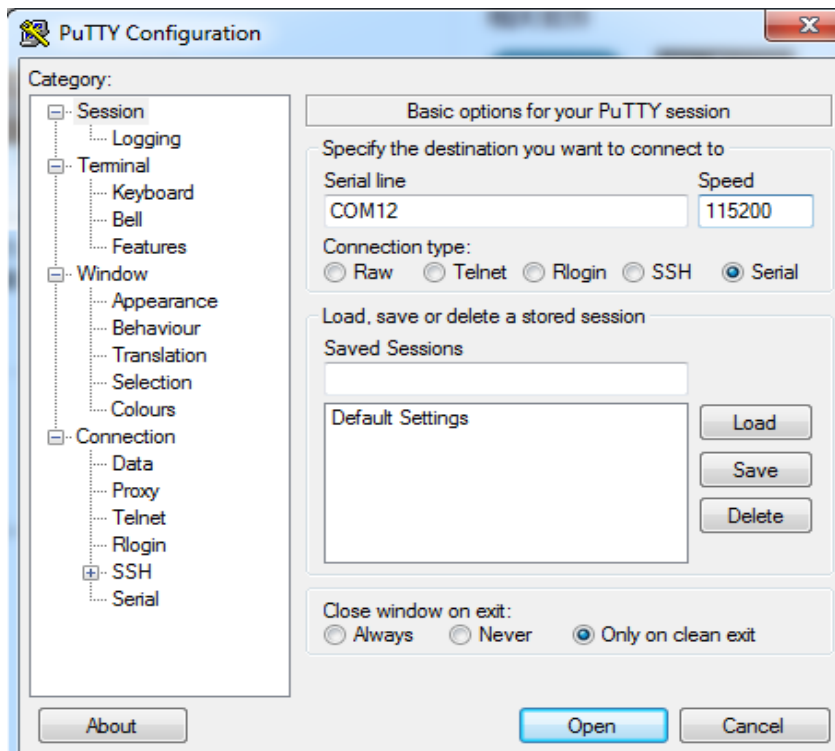


Figure 7a. Screenshot of Putty

4. Choose operating system of host computer – Ubuntu  
Install Minicom on host computer as below commands:

```
$ sudo apt-get install minicom
```

Set up serial communication terminal in Minicom as below:

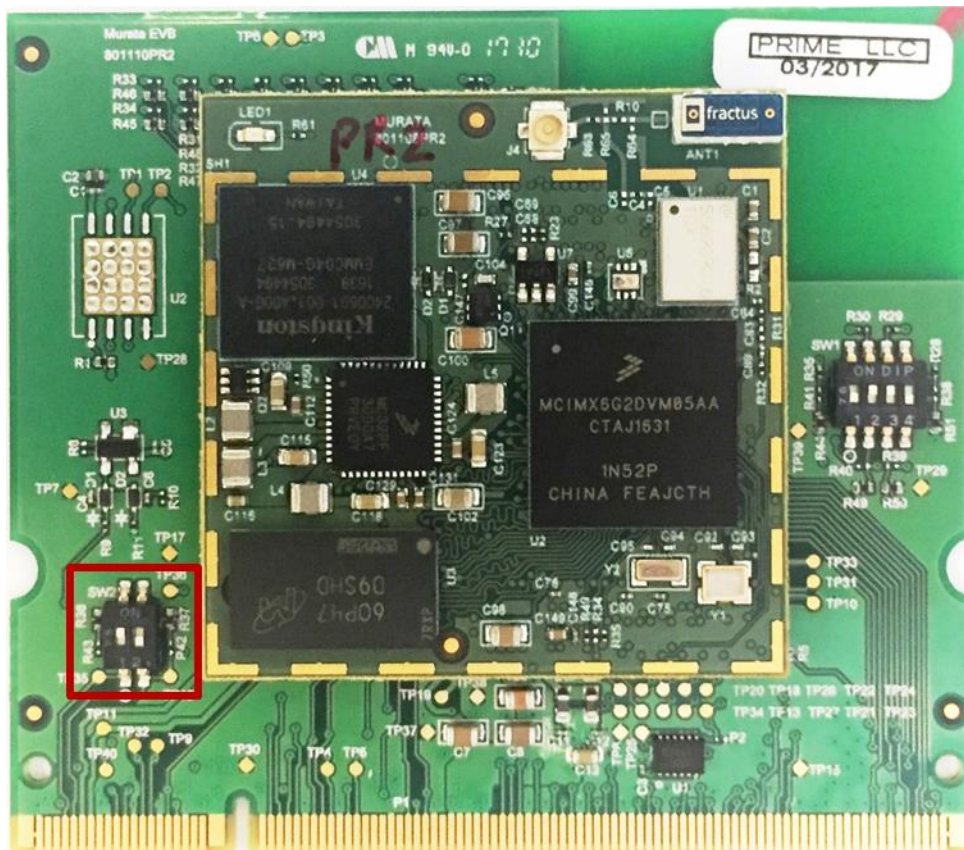
```
ad+-----+
ad| A - Serial Device      : /dev/ttyUSB0
ad| B - Lockfile Location  : /var/lock
nd| C - Callin Program    :
dr| D - Callout Program   :
dr| E - Bps/Par/Bits      : 115200 8N1
dr| F - Hardware Flow Control : No
dr| G - Software Flow Control : No
dr|
dr|   Change which setting? █
dr+-----+
droid work| Screen and keyboard |ONNECTED
```

Figure 7b. Screenshot of Minicom

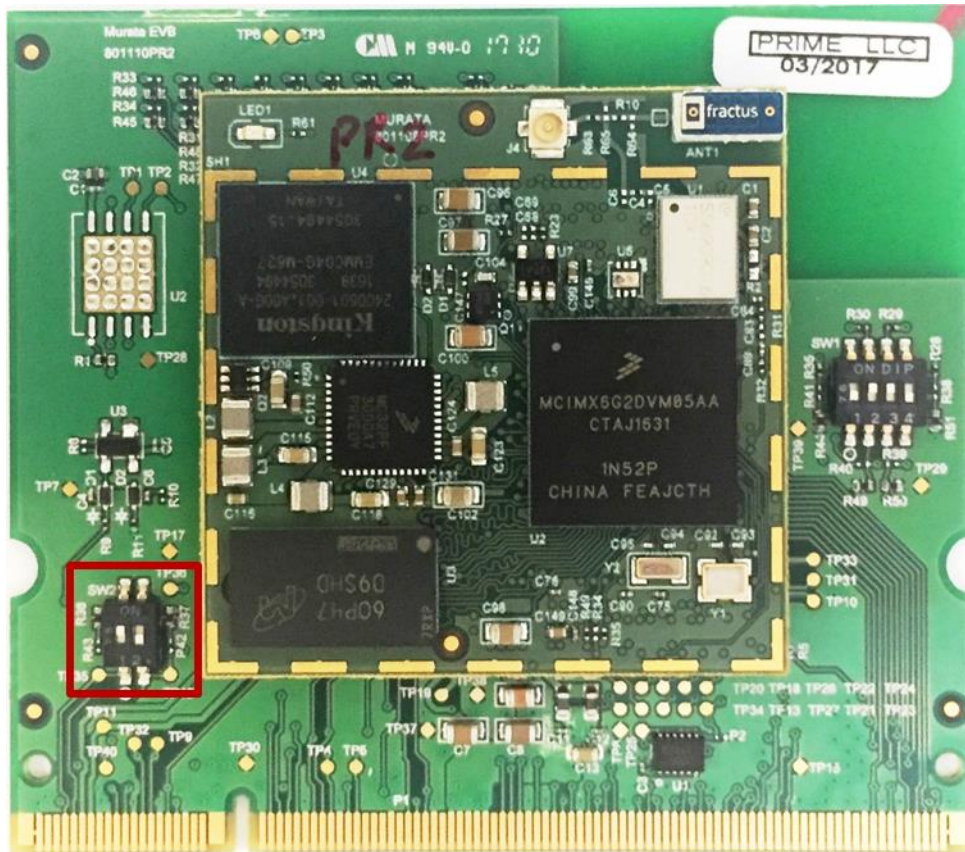
## 6. Download Mode and Boot Mode

The board is designed as booting from the internal eMMC. There are two modes for the SprIoT 6UL board. One is the download mode in which the board will receive the instructions from MFG Tools to flash images to boot storage such as eMMC. The other one is boot mode in which the board will load the image from the boot storage and boot from the image.

To make the board in download mode, please make sure that the four switch S1 is set as shown:



To make the board in boot mode, please make sure that the four switch S1 are set as shown:



The board comes with a working image burned in eMMC. To boot the board from that image, you can boot the board directly with the power supply connected. Please make sure the board is in Boot Mode with the switch S1 setting as above.

## 7. Prepare Android Things Images

### 7.1 Android Things Image Introduction

The table describes Android Things images and the targeted eMMC partition where the Android Things images to be flashed into:

Image Name	Image Description	Target Parition
u-boot.imx	The u-boot bootloader image, which is the first code run after the SprIoT 6UL board hardware reset. It will load and jump to the boot.img either from Slot a's boot partition or Slot b's boot partition, based	The first boot partition of SprIoT 6UL



	on the meta data stored in misc partition	
partition-table.img	The GUID Partition Table image, which define the partitions in the SprloT 6UL	gpt partition(Slot a's boot partition) for SprloT 6UL
boot.img	The Android Things boot image which is composed by Linux kernel zImage, linux kernel dtb(Device Tree Binary) file, Android Things ramdisk image, and linux kernel boot arguments. The code in boot.img will mount the related system.img based on the meta data stored in misc partition.	boot_a partition(Slot a's boot partition) for SprloT 6UL boot_b(Slot b's boot partition) for SprloT 6UL
userdata.img	The Android Things user data image	userdata partition for SprloT 6UL
system.img	The Android Things system image which includes all Android Things related binaries, libraries, and system configuration files.	system_a partition(Slot a's system partition) for SprloT 6UL system_b partition(Slot b's system partition) for SprloT 6UL
gapps.img	The Google application image.	gapps_a partition(Slot a's system partition) for SprloT 6UL gapps_b partition(Slot b's system partition) for SprloT 6UL
oem.img	The oem image.	oem_a partition(Slot a's system partition) for SprloT 6UL oem_b partition(Slot b's system partition) for SprloT 6UL

## 7.2 Prebuilt Android Things Images

The prebuilt Android Things images can be downloaded at <https://developer.android.com/things/preview/download.html>

## 7.3 The mfgtools

The mfgtools can be downloaded at

[http://www.nxp.com/products/software-and-tools/software-development-tools/i.mx-software-and-tools/iot-development-platforms-based-on-i.mx-6ul-processor-and-android-things-os:IOT-DEV-PLATFORMS-i.MX6UL?tab=Design\\_Tools\\_Tab](http://www.nxp.com/products/software-and-tools/software-development-tools/i.mx-software-and-tools/iot-development-platforms-based-on-i.mx-6ul-processor-and-android-things-os:IOT-DEV-PLATFORMS-i.MX6UL?tab=Design_Tools_Tab)

## 8. Testing and Debugging Tools

Unit tests run locally on the developer's host computer and integration tests interact with the target device via ADB.

SprIoT -6UL board is Android Things compatible and provides full support for ADB and FASTBOOT over USB for debugging.

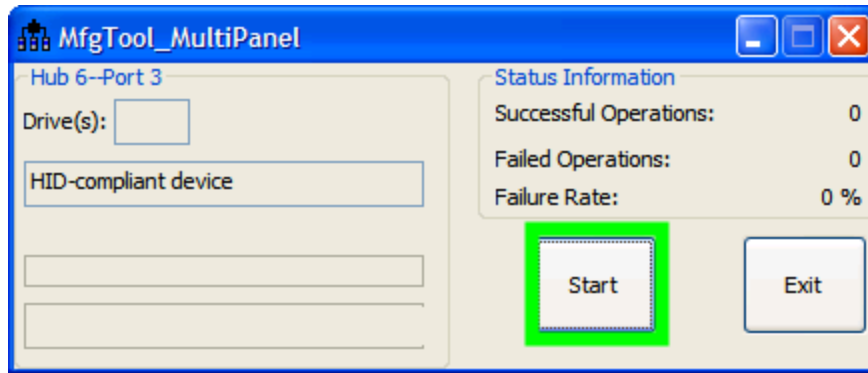
ADB and FASTBOOT are the tools in Android SDK. Please refer to the link <http://developer.android.com/sdk/index.html#Other> to download the latest version of Android SDK

## 9 Flash Android Things Images

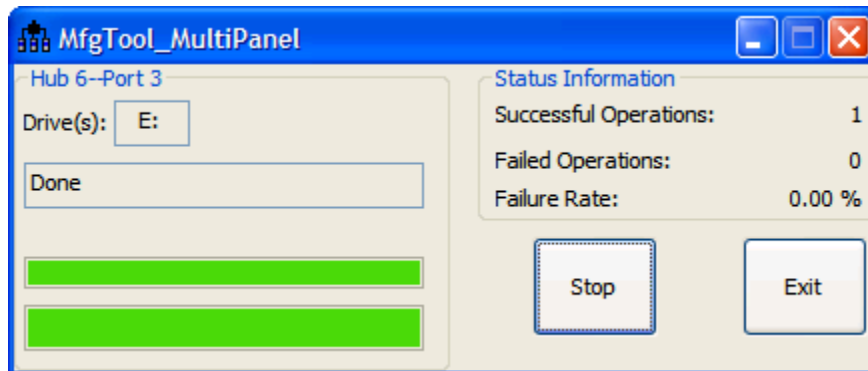
By default, a valid bootloader binary has been flashed into the SprIoT 6UL board. It will make the board into FASTBOOT mode if Android Things Images are not been flashed yet. Please refer "11. Instructions to make board into FASTBOOT mode" to check whether your board is into FASTBOOT mode.

### 9.1 Flash Android Things bootloader binary with MFG Tools

1. Unzip the **mfgtools.tar.gz** file to a selected location. The directory is named MFGTool-Dir in this example.
2. Make your board into download mode
3. Switch power supply to "ON" to power on the board. Using USB cable on the SprIoT OTG port, connect your WINDOWS/LINUX PC with SprIoT.
4. On WINDOWS, double click the file "**mfgtool2-brillo-mx6ul-spriot-emmc-firmware.vbs**" to flash only the uboot.imx of Android Things, Click "Start".



The mage below shows what the tool will become once the download is complete.



On Linux, run below commands:

```
sudo ./linux-runvbs.sh mfgtool2-brillo-mx6ul-spriot-emmc-firmware.vbs
```

Note: If blocked, please plug out the USB OTG cable, then plug in.

5. Power off, set the board is in Boot Mode.

## 9.2 Provision Android Things images with FASTBOOT mode

1. Download Android Things images package for SprIoT 6UL from <https://developer.android.com/things/preview/download.html>, and unzip it.
2. Refer "11. Instructions to make board into FASTBOOT mode" to make the board into FASTBOOT mode.
3. Flash Android Things images with either of the two ways below:
  - 3.1 Flash all images with the shell script in Android Things images package
    - Execute the batch file *iot-flashall-ix6ul.bat* On WINDOWS PC
    - Execute the shell script *iot-flashall-ix6ul.sh* On LINUX PC
  - 3.2 Flash Android Things images with fastboot command

Execute below commands in Linux PC to flash the related images

Image File Name	Partition Name	Fastboot command
u-boot.imx	bootloader	<i>\$fastboot flash bootloader u-boot.imx</i>

partition-table.img	gpt	<i>\$fastboot flash gpt partition-table.img</i>
boot.img	boot_a/boot_b	<i>\$fastboot flash boot_a boot.img</i> <i>\$fastboot flash boot_b boot.img</i>
system.img	system_a/system_b	<i>\$fastboot flash system_a system.img</i> <i>\$fastboot flash system_b system.img</i>
userdata.img	userdata	<i>\$fastboot flash userdata userdata.img</i>
gapps.img	gapps_a/gapps_b	<i>\$fastboot flash gapps_a gapps.img</i> <i>\$fastboot flash gapps_b gapps.img</i>
oem.img	oem_a/oem_b	<i>\$fastboot flash oem_a oem.img</i> <i>\$fastboot flash oem_b oem.img</i>

Note: The partitions boot\_a, boot\_b, system\_a, system\_b and userdata are defined by the partition-table.img flashed in board's eMMC. The partition-table.img should be flashed into board's eMMC before flashing those partitions.

- Run below commands in Linux PC to make the board in lock state, and reboot the board  
*\$fastboot flashing lock*  
*\$fastboot reboot*

## 10. Boot Android Things

After flashing the images, you can boot the board directly with the power supply connected and power switch to "ON". Please make sure the board is in Boot Mode.

### 10.1 Change boot arguments

By default, the u-boot will take the boot arguments stored in Android Things' boot.img. Below is an example in case you need to change the default boot arguments used by u-boot.

```
U-Boot > setenv bootargs console=ttyMXC0,115200 init=/init
androidboot.console=ttyMXC0 androidboot.hardware=imx6ul cma=96M rootwait ro
U-Boot > saveenv
U-Boot > boot
```



## 11. Instructions to make board into FASTBOOT mode

FASTBOOT mode is a state in which the board will respond the commands from host PC FASTBOOT commands to flash Android Things images or query board information. The board should connect with your host PC through USB type-A to micro USB cable.

### 11.1 Check whether the device is into FASTBOOT mode

You can check whether your board is into FASTBOOT mode through fastboot commands on your PC

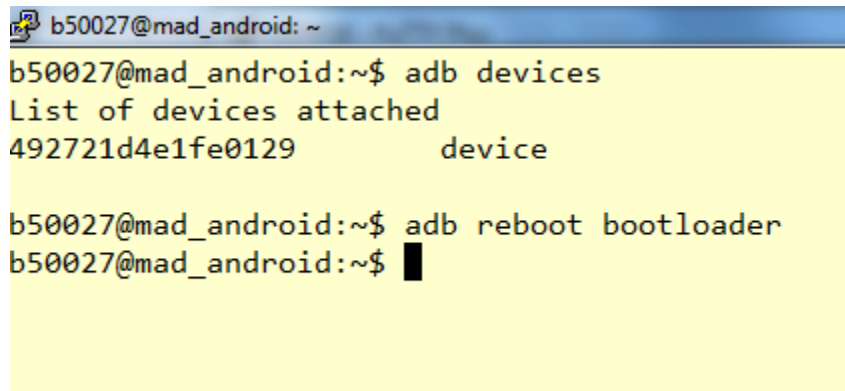
Commands with return string	Device in FASTBOOT mode
<pre>~\$ fastboot devices 000000f674a400d3    fastboot</pre>	Yes
<pre>~\$ fastboot devices</pre>	No

### 11.2 Set the device into FASTBOOT mode

If your device isn't into FASTBOOT mode, you can set the device into FASTBOOT mode with either of the two ways below:

#### 1. Via adb command line

- Once you confirm that you have access the device through adb on your PC, run command “adb reboot bootloader” as shown below:

A terminal window screenshot showing the execution of adb commands. The prompt is b50027@mad\_android: ~. The first command is 'adb devices', which outputs 'List of devices attached' and '492721d4e1fe0129 device'. The second command is 'adb reboot bootloader', which outputs 'b50027@mad\_android:~\$' followed by a black cursor block.

```
b50027@mad_android: ~
b50027@mad_android:~$ adb devices
List of devices attached
492721d4e1fe0129    device

b50027@mad_android:~$ adb reboot bootloader
b50027@mad_android:~$ █
```

#### 2. Via serial console

- Once the board completed booting-up, type the following commands in your serial console window:

```
$su
$reboot bootloader
```

Once the device is in FASTBOOT mode, your serial console will look like the screen shown below:

```
CPU:   Freescale i.MX6UL rev1.1 at 396 MHz
Reset cause: WDOG
Board: Murata MX6UL-Aquila
I2C:   ready
DRAM:  512 MiB
PMIC:  PFUZE3000 DEV_ID=0x30 REV_ID=0x11
MMC:   FSL_SDHC: 0, FSL_SDHC: 1
*** Warning - bad CRC, using default environment

No panel detected: default to TFT43AB
Display: TFT43AB (480x272)
Video: 480x272x24
In:    serial
Out:   serial
Err:   serial
flash target is MMC:1
Net:   FEC0
Error: FEC0 address not set.

Fastboot: Got bootloader commands!
CTRL-A Z for help |115200 8N1 | NOR | Minicom 2.5
```

To get the device out of FASTBOOT mode, run command “fastboot reboot” from your PC.

## 12. Useful Links

<http://developer.android.com/tools/help/adb.html>

<https://www.kingoapp.com/help/fastboot-mode.htm>