

PICO-i.MX6UL 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 – PICO-i.MX6UL board, and software support. Specifically, it walks through the hardware setup 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:

- PICO-i.MX6UL-eMMC System-On-Module (SOM)
- PICO- carrier board (pre-assembled with the SOM)
- 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 USB type-C cable

Serial console:

- USB to RS232 (DB-9 Female) converter cable
- DB-9 RS232 Serial Null Modem (Female to Female)
- Serial console connector with 5 pin-outs for UART

- WiFi antenna (IPEX interface)

3. Getting Familiar with the Development Platform

If you just got your board, please familiarize yourself with the key components of the System-on-Module (PICO-i.MX6UL-eMMC System-On-Module)

PICO-i.MX6UL SYSTEM-ON-MODULE

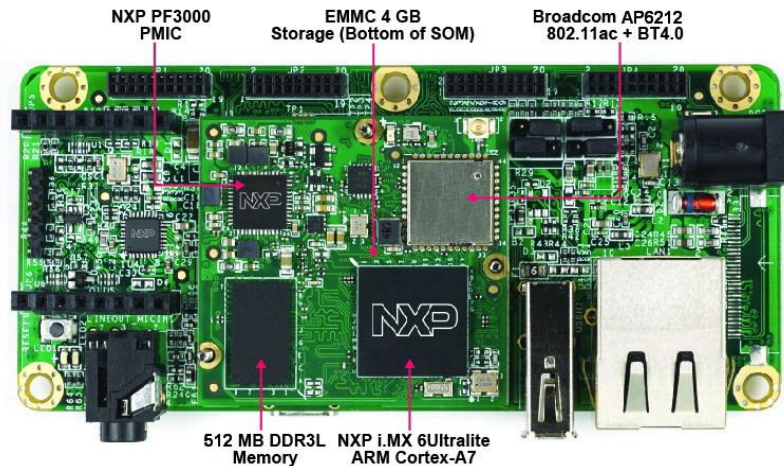


Figure 1. PICO-i.MX6UL-eMMC SOM Components

The key interfaces of the board are shown in Figure 2 and 3:

- 5V power input jack (number 6 in Figure 2).
- Microphone+headphone jack (number 4 in Figure 2)
- WiFi+Bluetooth antenna connector (number 5 in Figure 2)
- 6-pin header for UART console interface (number 2 in Figure 2). This interface is used for serial console interface.

PICO-i.MX6UL BOARD TOP VIEW

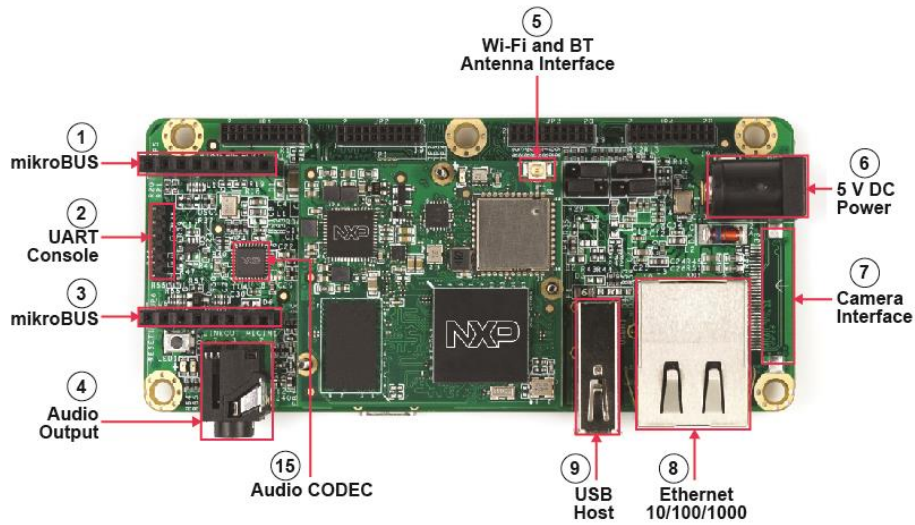


Figure 2. Top view of the PICO-i.MX6UL board

- USB OTG type-C connector (number 11 in Figure 3). This connector is used for ADB and FASTBOOT interface.

PICO-i.MX6UL BOARD FRONT VIEW

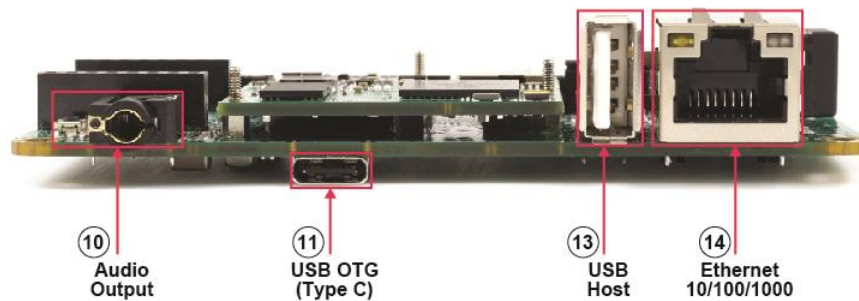


Figure 3. Front view of the PICO-i.MX6UL board

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

PICO-i.MX6UL BOARD JUMPER SETUP (DOWNLOAD MODE)

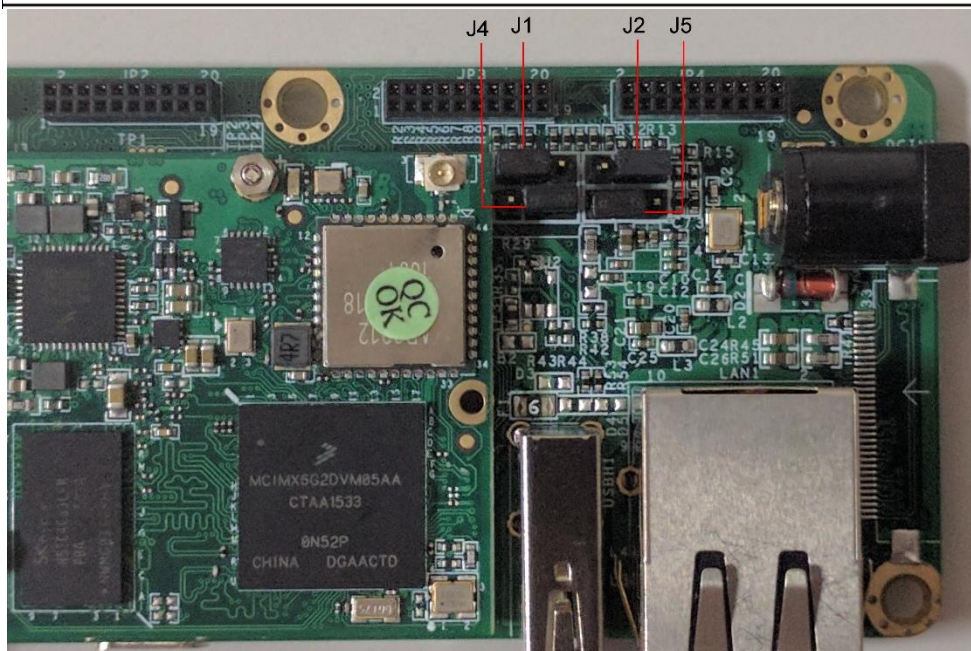


Figure 4a. Jumper Setup (Download Mode)

PICO-i.MX6UL BOARD JUMPER SETUP (BOOT MODE)

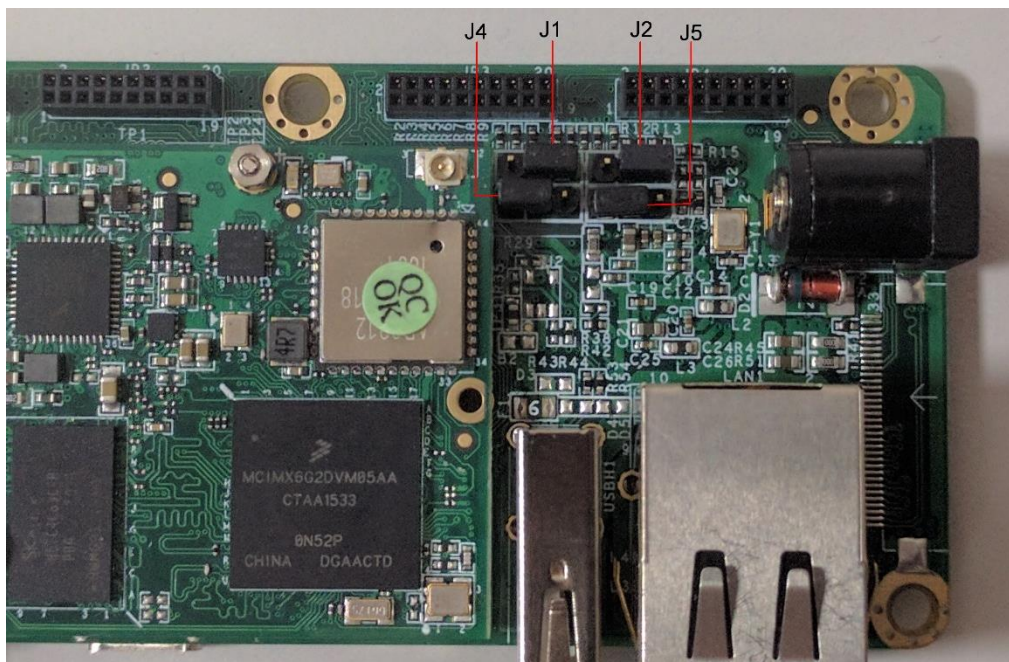


Figure 4b. Jumper Setup (Boot Mode)

4. Connect the board and host computer

1. Connect the USB to RS232 adapter to the DB9 null modem and serial console connector as shown in Figure 5. Connect the USB end of the USB to RS 232 adapter to the PC. For more instructions for serial communication terminal setup, see next section.

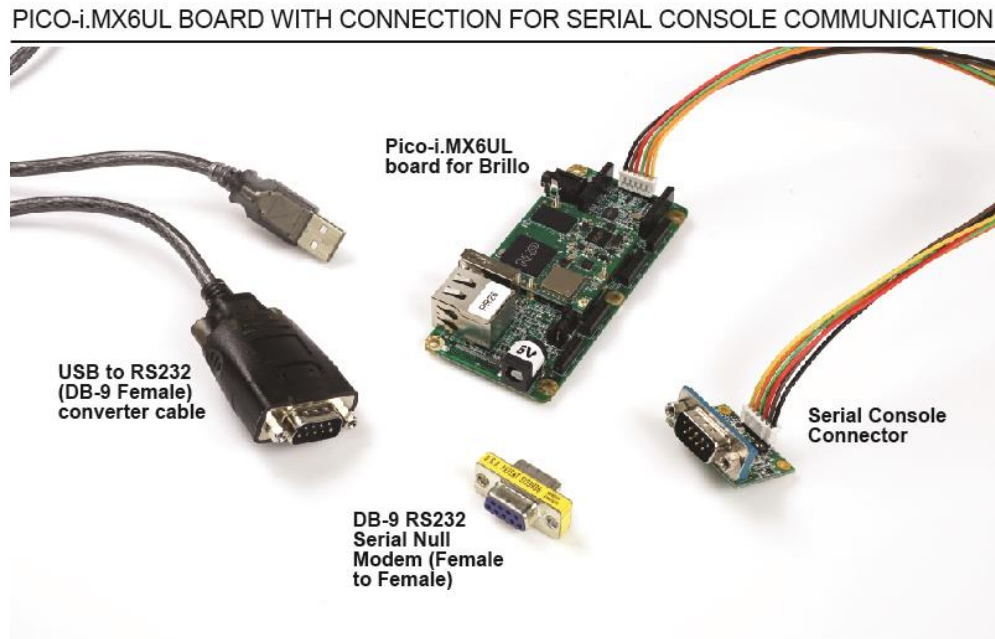


Figure 5. PICO-i.MX6UL board with connection for serial console communication

2. Get a USB type A to USB type C cable. Plug the USB type C end to the USB OTG type C connector (number 11 in Figure 3) for ADB and FASTBOOT interface. Plug the other end of the USB cable to your computer.
Note: the connector is reversible but only one side is connected to the device. If you don't have ADB or FASTBOOT access, try plugging the USB cable with the other side up.
3. Connect the WiFi antenna to connector (number 5 in Figure 2)
4. Plug in the power supply

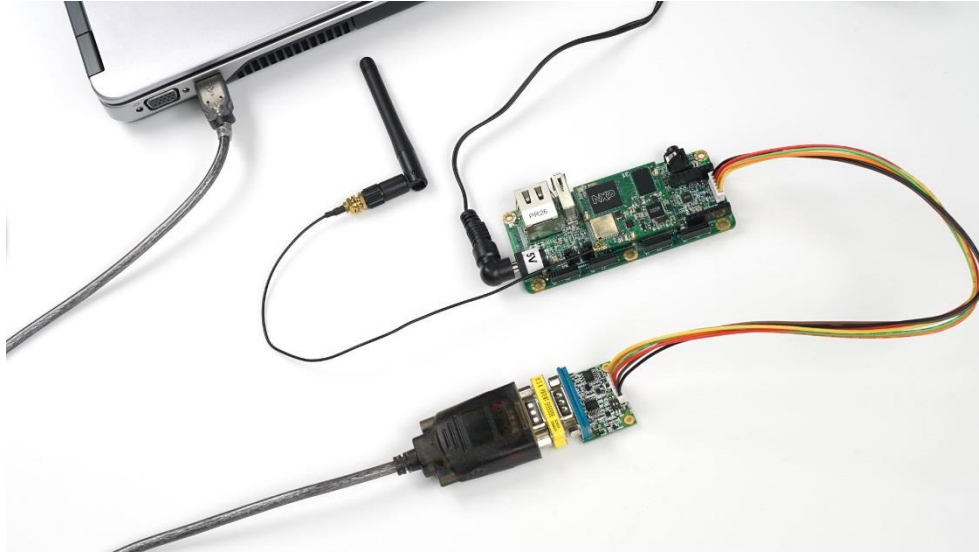


Figure 6. PICO-i.MX6UL board with serial console and Wi-Fi antenna connected

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 it 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, 8 data bits, 1 stop bit, no parity.

Note: The PC needs a driver to enable a virtual COM port through the PC USB port. Please consult www.ftdichip.com/Documents/InstallGuides.htm to download the correct driver.

Set up serial communication terminal in Putty as below:

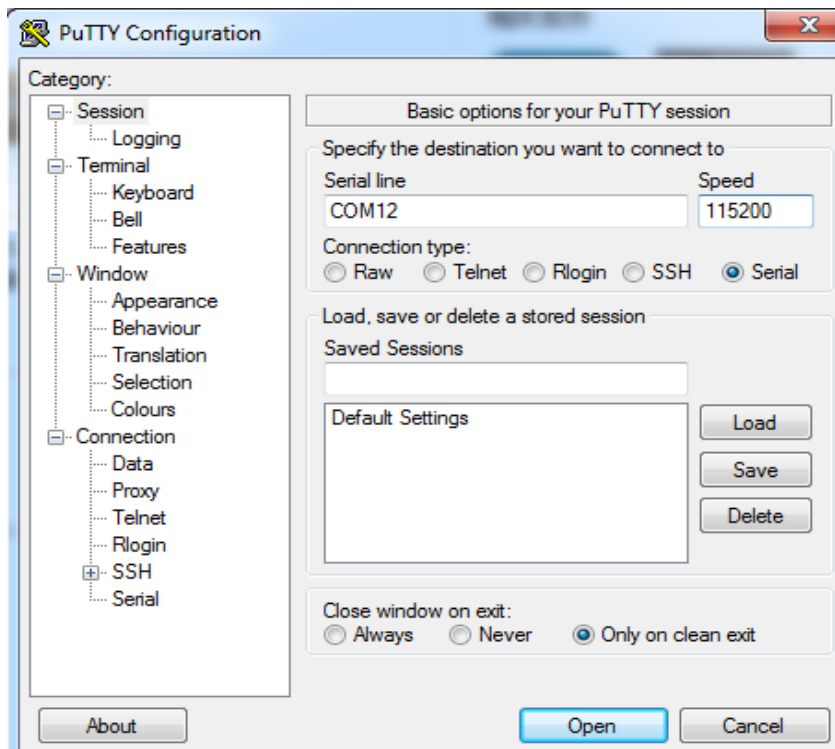


Figure 7a. Screenshot of Putty

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:



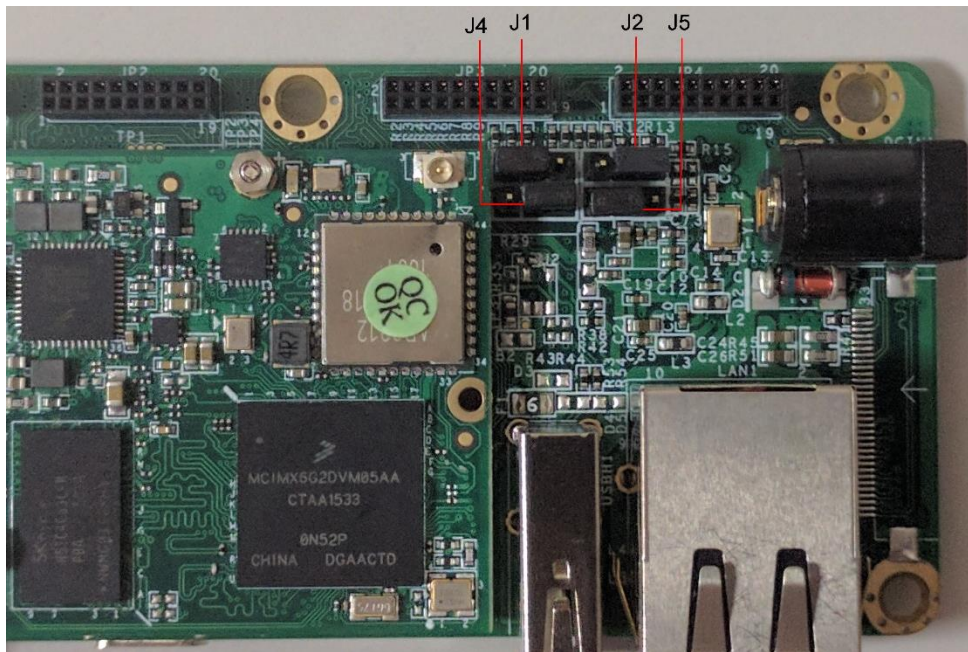
Figure 7b. Screenshot of Minicom

6. Download Mode and Boot Mode

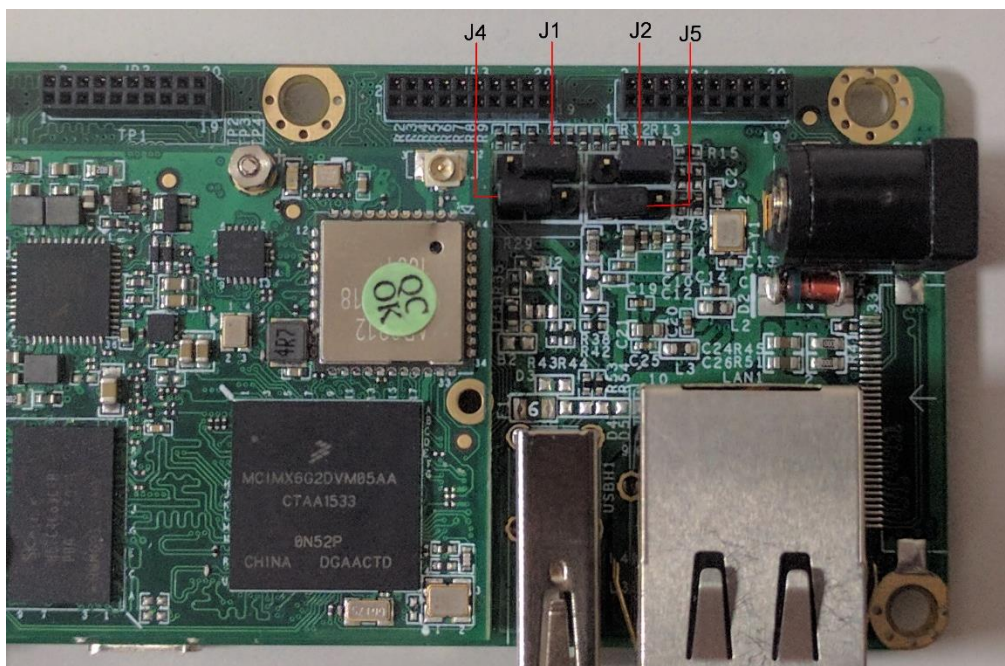
The board is designed as booting from the internal eMMC. There are two modes for the PICO-i.MX6UL board. One is the download mode in which the board will receive the instructions from MFG Tools to

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 jumpers are set as shown:



To make the board in boot mode, please make sure that the four jumpers 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 jumper setting as above.

7. Prepare Android Things Images

The next section gives an introduction to the Android Things Image, using uboot (universal bootloader) and how to use NXP provided MFGtool (Manufacturing tool) to flash the board.

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 Partition
u-boot.imx	The u-boot bootloader image, which is the first code run after the PICO-i.MX6UL 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 on the meta data stored in misc partition	The first boot partition of PICO-i.MX6UL-emmc
partition-table.img	The GUID Partition Table image, which define the partitions in the PICO-i.MX6UL-emmc	gpt partition for PICO-i.MX6UL-emmc
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 PICO-i.MX6UL-emmc boot_b(Slot b's boot partition) for PICO-i.MX6UL-emmc

userdata.img	The Android Things user data image	userdata partition for PICO-i.MX6UL-emmc
system.img	The Android Things system image which includes all Android Things related binarys, librarys, and system configuration files.	system_a partition(Slot a's system partition) for PICO-i.MX6UL-emmc system_b partition(Slot b's system partition) for PICO-i.MX6UL-emmc
gapps.img	The google application image.	gapps_a partition(Slot a's gapps partition) for PICO-i.MX6UL-emmc gapps_b partition(Slot b's gapps partition) for PICO-i.MX6UL-emmc
oem.img	The oem image.	oem_a partition(Slot a's oem partition) for PICO-i.MX6UL-emmc oem_b partition(Slot b's oem partition) for PICO-i.MX6UL-emmc

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

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.

Pico-imx6 board is Android Things compatible and provides full support for ADB and FASTBOOT over USB for debugging.

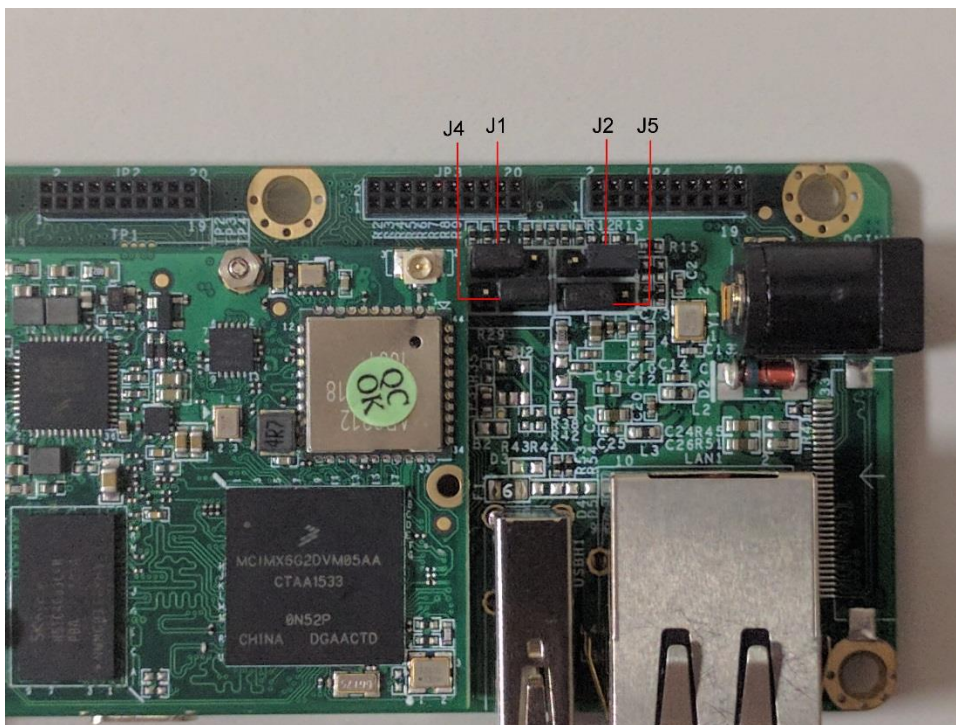
ADB and FASTBOOT are the tools in Android SDK. Please refer to the link <http://developer.android.com/sdk/index.html#Other> to download the latest version of Android SDK

9 Flash Android Things Images

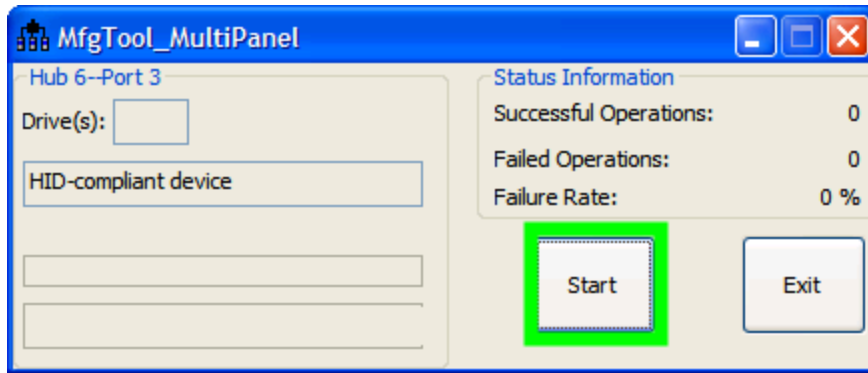
By default, a valid bootloader binary has been flashed into the PICO-i.MX6UL board. It will make the board into FASTBOOT mode if Android Things Images are not been flashed yet. Please refer “11. Instructions to make board into FASTBOOT mode” to check whether your board is into FASTBOOT mode.

9.1 Flash Android Things bootloader binary with MFG Tools

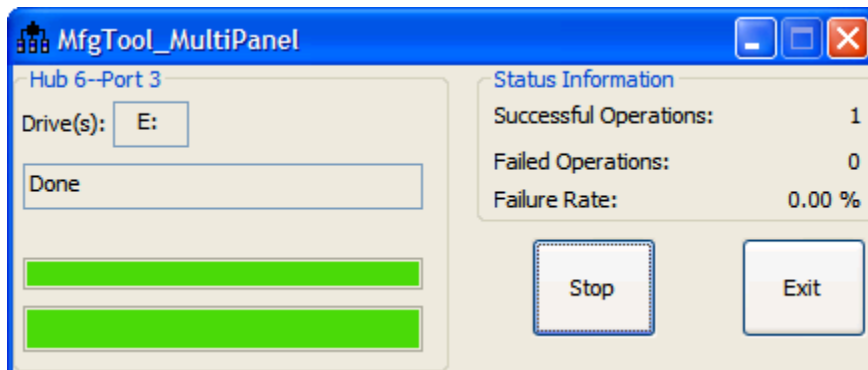
1. Unzip the mfgtools.tar.gz file to a selected location. The directory is named MFGTool-Dir in this example.
2. Make your board into download mode by setting the board's jumper as JP1:12, JP2:23, JP4:23, and JP5:12. The jumper setting shows as below:



3. Power on the board. Using USB cable on the Pico OTG port, connect your WINDOWS/LINUX PC with Pico.
4. On WINDOWS, double click the file "mfgtool2-brillo-mx6ul-pico-emmc-firmware.vbs" to flash only the uboot.imx of Android Things, Click "Start".



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

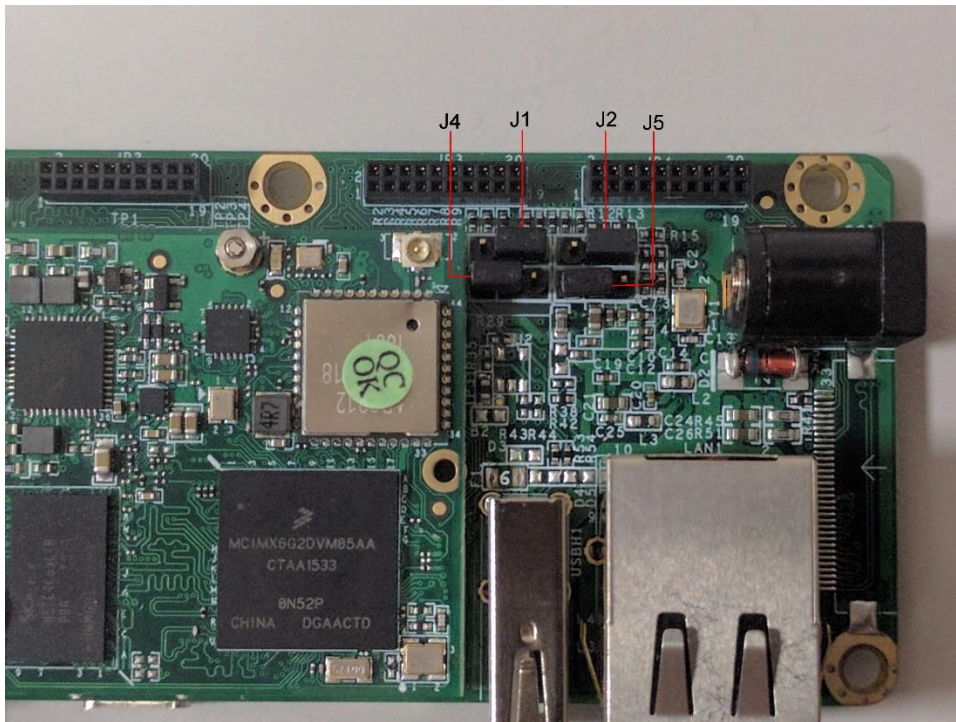


On Linux, run below commands:

```
sudo ./linux-runvbs.sh mfgtool2-brillo-mx6ul-pico-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 with the jumper setting as JP1:23, JP2:23, JP4:12, and JP5:12. Then power on, the board will enter into fastboot mode.



9.2 Provision images with FASTBOOT mode

1. Download Android Things images package for PICO i.MX6UL 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-imx6ul.bat` On WINDOWS PC
 - Execute the shell script `iot-flashall-imx6ul.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	<code>\$fastboot flash bootloader u-boot.imx</code>
partition-table.img	gpt	<code>\$fastboot flash gpt partition-table.img</code>

boot.img	boot_a/boot_b	<i>\$fastboot flash boot_a boot.img \$fastboot flash boot_b boot.img</i>
system.img	system_a/system_b	<i>\$fastboot flash system_a system.img \$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 \$fastboot flash gapps_b gapps.img</i>
oem.img	oem_a/oem_b	<i>\$fastboot flash oem_a oem.img \$fastboot flash oem_b oem.img</i>

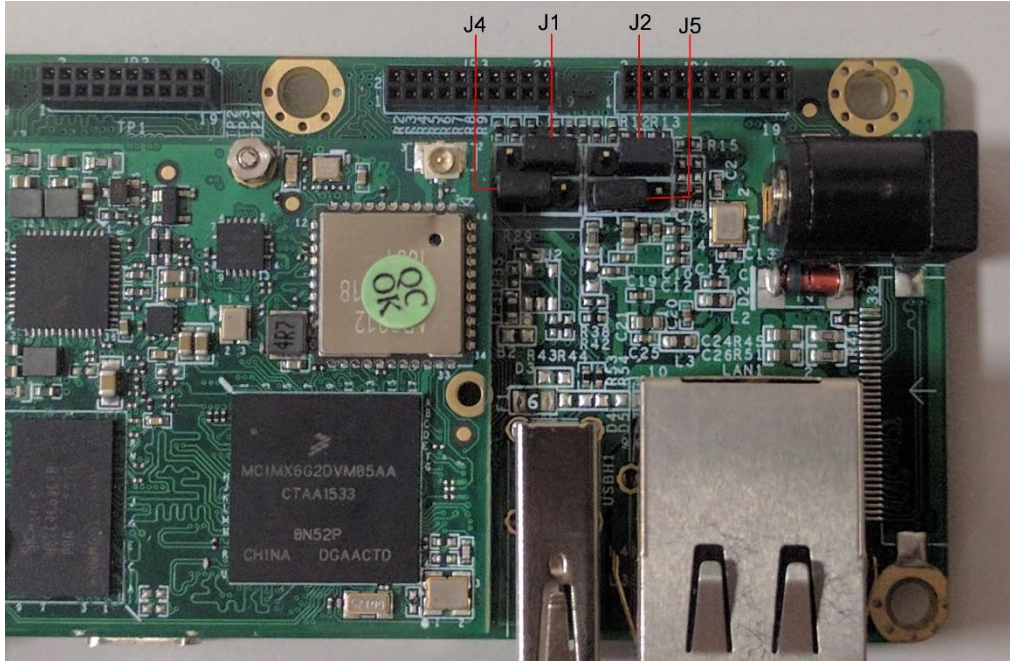
Note: The partitions boot_a, boot_b, system_a, system_b and userdata are defined by the partition-table.img flashed in board's eMMC. The partition-table.img should be flashed into board's eMMC before flashing those partitions.

After flash above partitions, run "*\$fastboot set_active_a*"

- 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. Please make sure the board is in Boot Mode with the jumper setting as JP1:23, JP2:12, JP3:12, and JP4:23.



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=ttyMXC5,115200 init=/init
androidboot.console=ttyMXC5 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 USB type-C 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

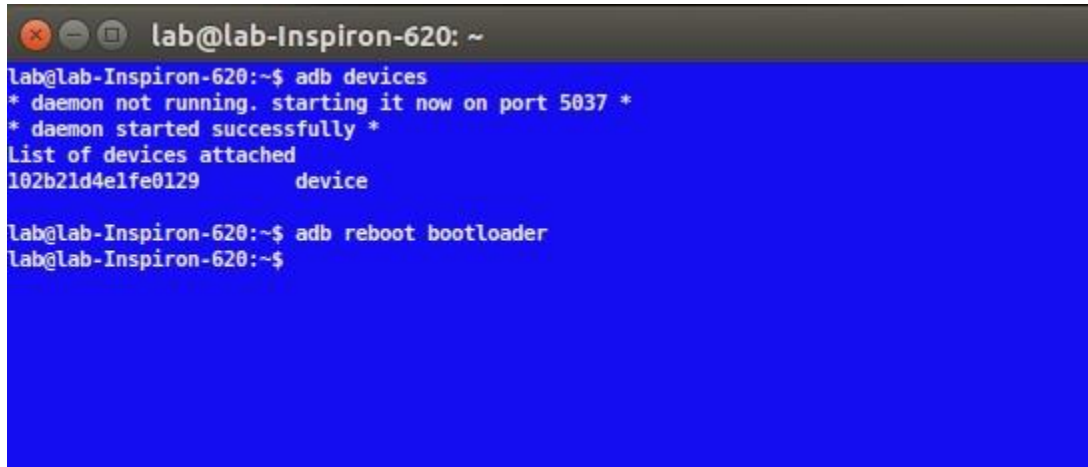
Commands with return string	Device in FASTBOOT mode
<code>~\$fastbootdevices</code> <code>000000f674a400d3 fastboot</code>	Yes
<code>~\$fastbootdevices</code>	No

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:



```
lab@lab-Inspiron-620: ~  
lab@lab-Inspiron-620:~$ adb devices  
* daemon not running. starting it now on port 5037 *  
* daemon started successfully *  
List of devices attached  
102b21d4e1fe0129    device  
  
lab@lab-Inspiron-620:~$ adb reboot bootloader  
lab@lab-Inspiron-620:~$
```

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 similar to the screen shown below:

```
reboot: Restarting system with command 'bootloader'

U-Boot 2015.04-00350-g958d865 (Jan 25 2016 - 09:24:21)

CPU:   Freescale i.MX6UL rev1.0 at 396 MHz
CPU:   Temperature 59 C
Reset  cause: WDOG
Board: PicoSOM i.mx6UL
I2C:   ready
DRAM:  256 MiB
PMIC:  PFUZE300 DEV_ID=0x30 REV_ID=0x11
MMC:   FSL_SDHC: 0
*** Warning - bad CRC, using default environment

In:    serial
Out:   serial
Err:   serial
check_and_clean: reg 100, flag_set 0
Fastboot: Normal
flash target is MMC:0
Net:   FEC1
Error: FEC1 address not set.
```

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

12. Useful Links

Android Things Developer site: <https://developer.android.com/things/index.html>

Android Things Developer Docs pages: <https://developer.android.com/things/guides/index.html>

Android Things Developer Samples pages: <https://developer.android.com/things/samples/index.html>

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

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