PICO-i.MX7D 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.MX7D 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:

- PICO-i.MX7D-eMMC System-On-Module (SOM)
- PICO- carrier board (pre-assembled with the SOM)

Besides, other required materials include:

- Cables:
  For ADB/FASTBOOT/MFGTool
  - USB type-A to USB type-C cable

  Serial console:
  - USB type-A to micro USB
- WiFi antenna (IPEX interface)

3. Getting Familiar with the Development Platform

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

- USB to serial console convertor interface (Number 1 in figure 1)
- WiFi+Bluetooth antenna connector (Number 5 in figure 1)
- Microphone and headphone jack (Number 3 in figure 1)

- USB OTG (USB Type-C) and power supply interface (Number 4 in figure 1)

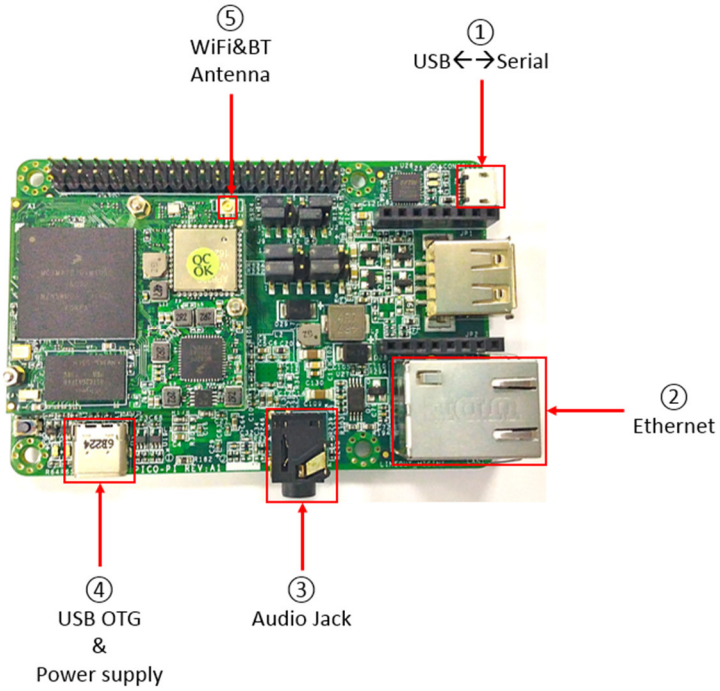


Figure 1. Top view of the PICO-i.MX7D board

- 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.

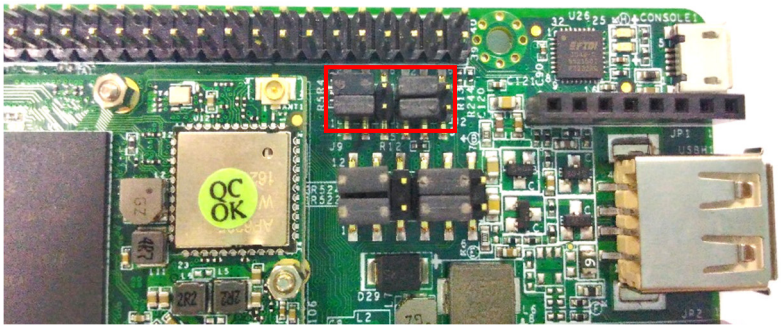


Figure 2a. Jumper Setup (Download Mode)

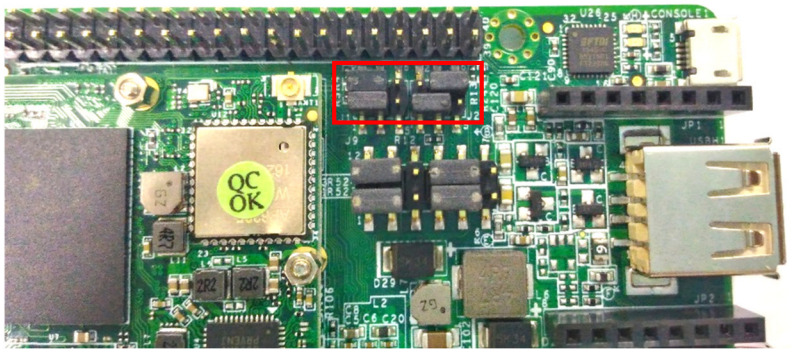


Figure 2b. Jumper Setup (Boot Mode)

4. Connect the board and host computer

1. Connect the USB type-A to micro USB's micro USB end to the micro USB interface (number 1 in figure 1).
2. Connect the WiFi antenna to connector (number 5 in figure 1).
3. Get a USB type A to USB type C cable. Plug the USB type C end to the USB OTG type C connector (number 4 in Figure 1) for ADB and FASTBOOT interface. Plug the other end of the USB cable to your computer. This interface also be the power supply for the board.

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.MX7D microprocessor's UART.

Recommended tools for serial communication terminal: Putty and Minicom
Serial port configuration: 115200 baudrate, 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:

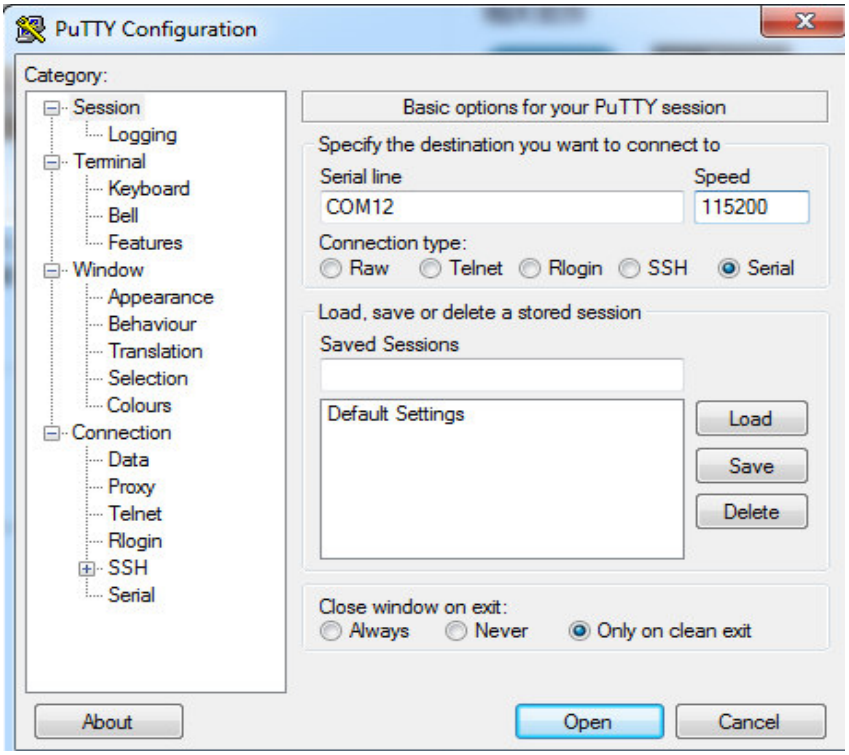


Figure 7a. Screenshot of Putty

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.MX7D 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.
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

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 PICO-i.MX7D 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.MX7D-emmc
partition-table.img	The GUID Partition Table image, which define the partitions in the PICO-i.MX7D-emmc	gpt partition(Slot a's boot partition) for PICO-i.MX7D-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.MX7D-emmc boot_b(Slot b's boot partition) for PICO-i.MX7D-emmc

userdata.img	The Android Things user data image	userdata partition for PICO-i.MX7D-emmc
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 PICO-i.MX7D-emmc system_b partition(Slot b's system partition) for PICO-i.MX7D-emmc
gapps.img	The Google application image.	gapps_a partition(Slot a's system partition) for PICO-i.MX7D-emmc gapps_b partition(Slot b's system partition) for PICO-i.MX7D-emmc
oem.img	The oem image.	oem_a partition(Slot a's system partition) for PICO-i.MX7D-emmc oem_b partition(Slot b's system partition) for PICO-i.MX7D-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

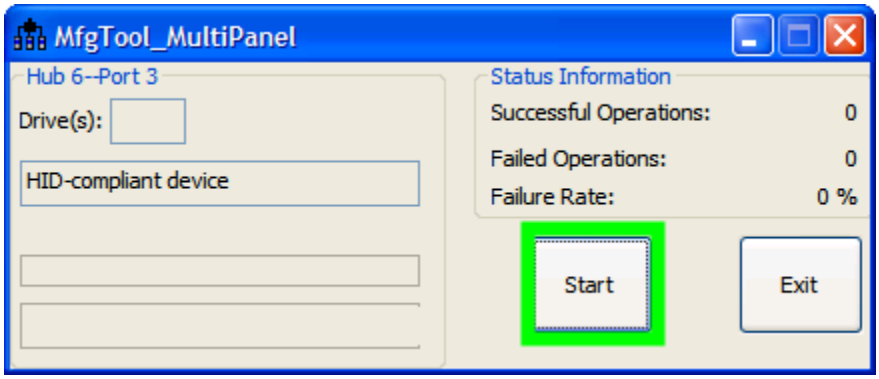
9 Flash Android Things Images

By default, a valid bootloader binary has been flashed into the PICO-i.MX7D 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.

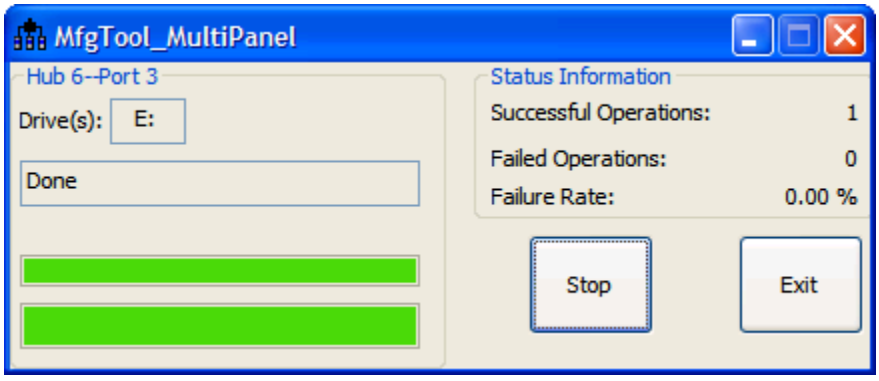
9.1 Flash Android Things bootloader binary with MFG Tools

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.
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-mx7d-pico-emmc-firmware.vbs"** to flash only the uboot.imx of Android Things, or double click the file **"mfgtool2-brillo-mx7d-pico-emmc.vbs"** to flash all Android Things images. Then click "Start".



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



5. Program images in Linux OS:
   a: In Linux, run below commands to flash the uboot.imx image of Android Things.
   sudo ./linux-runvbs.sh mfgtool2-brillo-mx7d-pico-emmc-firmware.vbs

b: In Linux, run below commands to flash all the images of Android Things.
sudo ./linux-runvbs.sh mfgtool2-brillo-mx7d-pico-emmc.vbs

Note: If blocked, please plug out55 the USB OTG cable, then plug in.

6. 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 PICO i.MX7D 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-imx7d.bat* On WINDOWS PC
   - Execute the shell script *iot-flashall-imx7d.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	*$fastboot flash bootloader u-boot.imx*
partition-table.img	gpt	*$fastboot flash gpt partition-table.img*
boot.img	boot_a/boot_b	*$fastboot flash boot_a boot.img* *$fastboot flash boot_b boot.img*
system.img	system_a/system_b	*$fastboot flash system_a system.img* *$fastboot flash system_b system.img*
userdata.img	userdata	*$fastboot flash userdata userdata.img*
gapps.img	gapps_a/gapps_b	*$fastboot flash gapps_a gapps.img* *$fastboot flash gapps_b gapps.img*
oem.img	oem_a/oem_b	*$fastboot flash oem_a oem.img* *$fastboot flash oem_b oem.img*

Note: The paritions 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 parititions.

4. 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

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=ttymxc4,115200 init=/init
androidboot.console=ttymxc4 androidboot.hardware=imx7d vt.global_cursor_default=0
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
~$ fastboot devices *000000f674a400d3 fastboot*	Yes
~$ fastboot devices	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:

```
b50027@mad_android: ~
b50027@mad_android:~$ adb devices -l
List of devices attached
00000377f4a400d3       device usb:3-2 product:iot_imx7d_pico

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 similar to the screen shown below:

```
U-Boot 2015.04-00079-g2c14d01 (Feb 10 2017 - 16:52:42)

CPU:   Freescale i.MX7D rev1.2 at 792 MHz
CPU:   Temperature 37 C
Reset cause: POR
Board: i.MX7D PICOSOM
I2C:   ready
DRAM:  512 MiB
PMIC: PFUZE300 DEV_ID=0x30 REV_ID=0x11
MMC:   FSL_SDHC: 0, FSL_SDHC: 1
No panel detected: default to EJ050NA
Display: EJ050NA (800x480)
Video: 800x480x24
In:    serial
Out:   serial
Err:   serial
flash target is MMC:1
Net:   FEC0
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