everyday genius

MT2712 Android® User Guide
- For Bringup software

Version:	EA1810
Release date:	2018-10-26

Specifications are subject to change without notice.

Document Revision History

Revision	Date	Description
2018.07.13	2018-07-13	First release
EA1810	1018-10-26	Second release

Table of Contents

Lists of Tables and Figures

1   Overview

This document introduces how to build and install images for MT2712 EVB (Evaluation Board) by using MT2712 Android® BSP (Board Support Package), how to setup the Android Project build environment and how to flash images to get Android run on EVB. It also introduces the features that MT2712 Android® BSP is planned to offer.

1.1   Prepare tools and drivers

Table 1-1 lists the required tools and drivers, which can be obtained from your MediaTek Customer Project Manager (CPM) window or downloaded from the Internet.

Table 1-1. Tools and Drivers

Software	Description
Android ADB-tools	https://developer.android.com/studio/index.html#command-tools
MediaTek USB cable driver	The MediaTek USB driver that enables communication between a host machine and the MT2712 device Notes: please contact MediaTek CPM to obtain the driver
USB-RS232 driver	Prolific USB-to-RS232 driver that enables serial connections for UART debugging http://www.prolific.com.tw/US/ShowProduct.aspx?pcid=41&showlevel=0041
python 2.7.x	https://www.python.org/ftp/python/2.7.2/python-2.7.2.msi
pySerial 3.x	https://pypi.python.org/pypi/pyserial (e.g. pyserial-3.3.tar.gz) Notes: Unzip and execute setup.py to install it. Exceptionally, for Windows XP, please install pySerial 3.0.1 (https://pypi.python.org/pypi/pyserial/3.0.1)

1.2   Introduction

MT2712 Android® BSP is a collection of binary files, source codes, and SDK that supports users to develop Android® applications and build images of bootloaders, TEE, Linux Kernel and the Android images (such as: system.img ,cache.img, vendor.img,userdata.img) for MT2712 platform.

Chapter-2 Fetching and building introduces the information on how to set up the Linux host machine, how to fetch Mediatek source code from Mediatek git01 server and how to generate a full android system images.

Chapter-3 BSP Features introduces the MT2712 Android® BSP features on MT2712 EVB now.

2   Fetching and building

This chapter introduces the Android Project and gives you an idea of what you need to prepare to get started. You can find information on how to set up your host machine and build also from google website.

2.1   Requirement

Before you download and build the Android source, ensure your system meets the following requirements. Then see next chapter for the installation instructions for details.

2.1.1   Hardware requirements

- A 64-bit environment is required.
- At lease 100GB free disk space for source code and extra 150GB for building, if you conduct multiple builds or employ ccache, you will need even more free disk space.
- If you are running Linux in a virtual machine, you need at least 16GB RAM/swap.

2.1.2   Software requirements

The recommended workstation must have the software list below:

Host and JDK
The host machine use one of these Operating System: Ubuntu 14.04 or Mac OS V10.10 or later with Xcode4.5.2 and Command line tools
Mediatek use Ubuntu 14.04 64-bit, to align the same base, Ubuntu is recommended too.

For the Java Development Kit, master branch of Android in AOSP using a prebuilt version of OpenJDK; so no additional installation is required.

Key packages
- Python 2.6/2.7 from python.org
- GNU make 3.81/3.82 from gnu.org
- Git 1.9 from git-scm.org

Build toolchain
Android O support only CIANG/LLVM for building Android platform
For the Native Development Kit and legacy kernels, GCC4.9 included in the AOSP master branch may also be used

JDK
Android O: Ubuntu- OpenJDK 8 , Mac OS-jdk 8u45 or newer

2.2   Set Up a build environment

The Android build is routinely tested in house on recent versions of Ubuntu LTS (14.04), but most distributions should have the required build tools available.

2.2.1   Host Setup

MediaTek uses Ubuntu 14.04 64-bit (LTS) as the standard host environment for building the images.

Installing the JDK

The Android in the Android Open Source Project (AOSP) comes with a prebuilt version of OpenJDK in /platform/prebuilts/jdk/jdk8. So no additional installation is required.
But a 64-bit version of Ubuntu is needed. (Ubuntu 14.04 is recommended)

> *$ sudo apt-get install git-core gnupg flex bison gperf build-essential zip curl zlib1g-dev gcc-multilib g++-multilib libc6-dev-i386 lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z-dev ccache libgl1-mesa-dev libxml2-utils xsltproc unzip device-tree-compiler*

Optimizing a build environment (optional)

It is optionally to tell the build to use the ccache compilation tool, which is a compiler cache for C and C++ that can help make builds faster. It is especially useful for build servers and other high-volume production environments. Ccache acts as a compiler cache that can be used to speed up rebuilds. This works very well if you use make clean often, or if you frequently switch between different build products.

To use ccache, issue these commands in the root of the source tree:

> *export USE_CCACHE=1*
> *export CCACHE_DIR=/<path_of_your_choice>/.ccache*
> *prebuilts/misc/linux-x86/ccache/ccache -M 50G*

The suggested cache size is 50-100G.

Put the following in your .bashrc (or equivalent):

> *export USE_CCACHE=1*

By default the cache will be stored in ~/.ccache. If your home directory is on NFS or some other non-local filesystem, you will want to specify the directory in your .bashrc file too.

2.2.2   Setting Up the Repo Utility

Repo is a tool that help the users to manage multiple Git repositories easier. It is useful in managing the software projects because there are huge number of files (directories) in Android project. Following steps are performed to set up the repo utility:

```
$ mkdir ~/bin
$ PATH=~/bin:$PATH
$ curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
$ chmod a+x ~/bin/repo
```

2.3   Fetching code from Mediatek server and Google

Following steps are performed to download the MediaTek Android P BSP from MediaTek git01 server the Android P EAC is exclude and the google P EAC source code should fetch from google directly.
To cowork with MediaTek BSP please fetch the same P EAC version: PPR1.180610.007
Notes: please contact MediaTek CPM to obtain your account: username and password

1. Edit ~/.netrc
    1.1. Add the credential at the end of the ~/.netrc file
          machine git01.mediatek.com login <username> password <password>
    1.2. Change the pwd file to owner read/write only
          $ chmod 600 ~/.netrc
2. Sync Codebase
    2.1. Create and enter working folder
          $ mkdir MT2712-android
          $ cd MT2712-android

    2.2. To fetch latest codebase initialize command should be:
$ ~/bin/repo init -u https://git01.mediatek.com/android-car/platform/manifest -b car-release-master-ivi.android.p -m 2018−10−26_09−56−14.xml --no-repo-verify

    2.3. Sync code from Git server to your working folder
          $ ~/bin/repo sync -j12

After completing the above steps, the source code will be checked out into the working folder "MT2712-android". To integrate P EAC and MediaTek BSP together the whole Android P source is ready.

It is suggested to perform repo synchronization (step 2.3) to update to the latest code. If errors occur during *repo init* (step 2.2), try deleting the .repo directory and running the *repo init* command again.

2.4   Building Images

MediaTek follow the google Android standard flow to generate images.
But first of all some patches need apply before building, which only need to run at the first time to build.

0. Apply MediaTek patches
    $ cd MT2712-android
    $ ./device/mediatek/mt2712/patch/apply_patches.sh

    Now start to build:
1. Use the below command to build an image.

1.1. Full Build
$ source build/envsetup.sh
$ lunch car_mt2712-userdebug
$make –j12 2>&1 | tee build.log

To start a clean build, remove the images output folder, using:
$ rm -rf out

1.2. Partial Build
1.2.1. kernel
$ source build/envsetup.sh
$ lunch car_mt2712-userdebug
$ make bootimage 2>&1 |tee build.log
1.2.2. LK
$ source build/envsetup.sh
$ lunch car_mt2712-userdebug
$ make lk 2>&1 |tee build.log

2.5   Running Buildings

After building, the generated image resides in MT2712-android/out/target/product/mt2712/ folder. Please copy these images and necessary tool to one folder together.

2.5.1   Prepare Host Environment

Confirm the host environment:

1. Host PC must be windows OS.
2. python 2.7.x must be installed.

   Package link: https://www.python.org/ftp/python/2.7.2/python-2.7.2.msi
3. pySerial 3.x must be installed.

   Package link: https://pypi.python.org/pypi/pyserial (e.g. pyserial-3.3.tar.gz). Unzip and execute setup.py install to install it.
   Notes: For Windows XP, please install pySerial 3.0.1, https://pypi.python.org/pypi/pyserial/3.0.1
4. To flash the firmware image to device, Android ADB tool is required. It is recommended to install the Android SDK Platform-tools from the Android SDK manager
   https://developer.android.com/studio/index.html#command-tools
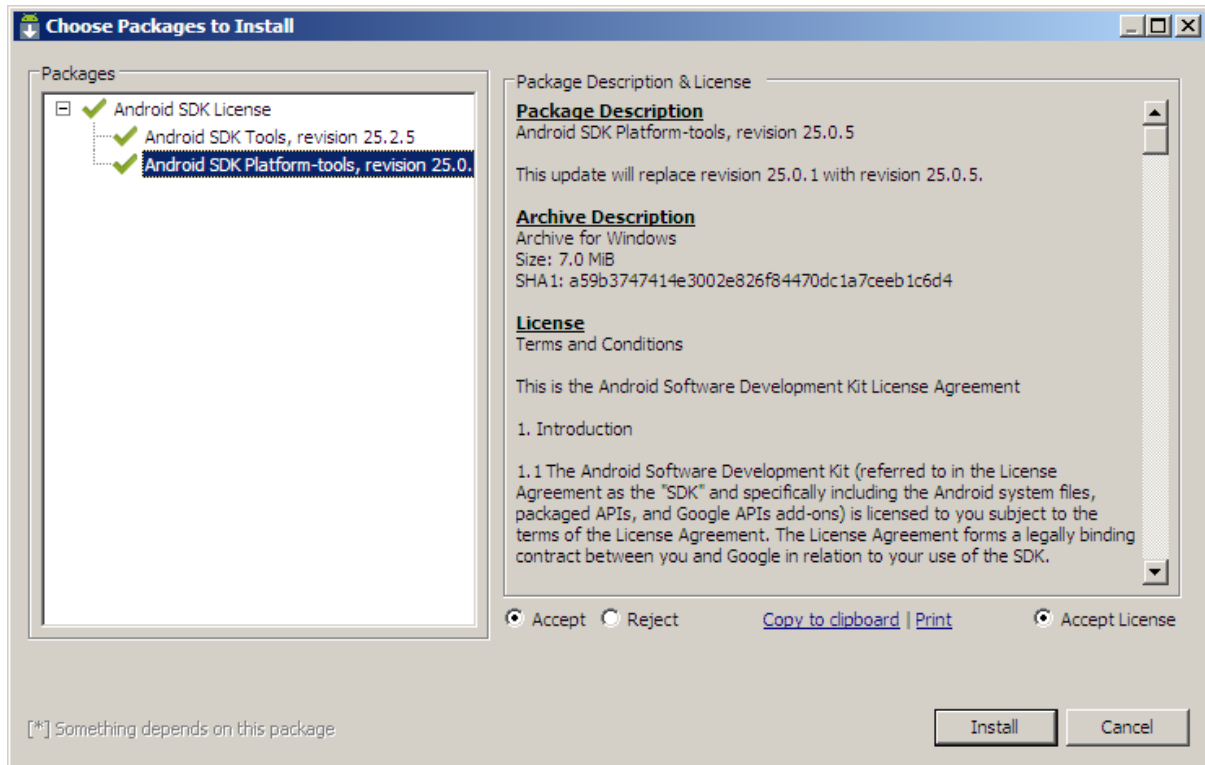
Figure 2-1. Install Android SDK Platform-tools

2.5.2 Prepare Fastboot

Fastboot is a special diagnostic and engineering protocol that you can boot your device into. In MT2712, bootROM and bootloader both support fastboot protocol but with minor difference.
In bootROM, it can receive a bootloader which has JTAG control information in the header. bootROM will setup the JTAG accordingly. Also, bootROM can boot device with the bootloader which supports fastboot protocol and flash capabilities.
In bootloader fastboot mode, you can directly flash the file system in your device over a USB connections.
Please contract MediaTek CPM to get fastboot tools package:

For **Windows**: fastboot.exe
For **Linux x86_64**: fastboot-x86_64
For **Linux arm**: fastboot-arm

### 2.5.3 Install MediaTek USB Cable driver

To recognize the MT2712 board as a device, unzip the SP_Drivers.rar and double click the DriverInstall.exe to install the MediaTek USB cable driver to host Windows PC.

2.5.4 MT2712 Platform USB ID Registration

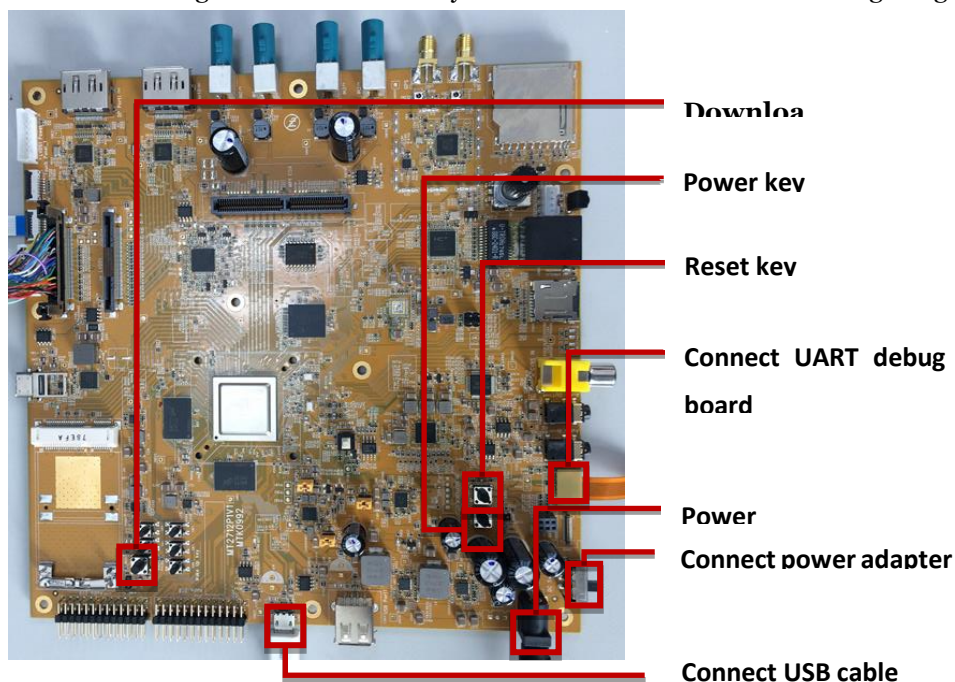1) In the **windows device manager**, find the unknown USB device

2) Right click on the unknown device, select **"Properties"** for the device.

3) In the window, choose **details > Property > Hardware Ids** to confirm hardware ID as below:

   USB\VID_18D1&PID_E447&REV_0100
   USB\VID_18D1&PID_E447

4) In the USB driver installation path: *C:\Program Files\MediaTek\SP Driver\drv\Android*

5) Edit file ***android_winusb.inf*** and add entries as below

   ;MT2712
   %CompositeAdbInterface%   = USB_Install, USB\VID_18D1&PID_E447&REV_0100
   %CompositeAdbInterface%   = USB_Install, USB\VID_18D1&PID_E447

6) Right click on the unknown device, select **"Update Driver Software"** to update your USB driver as below:
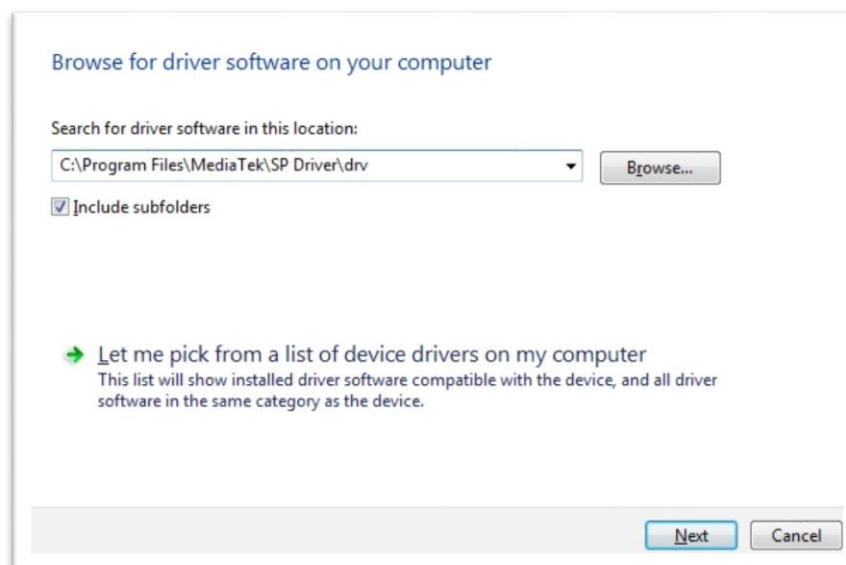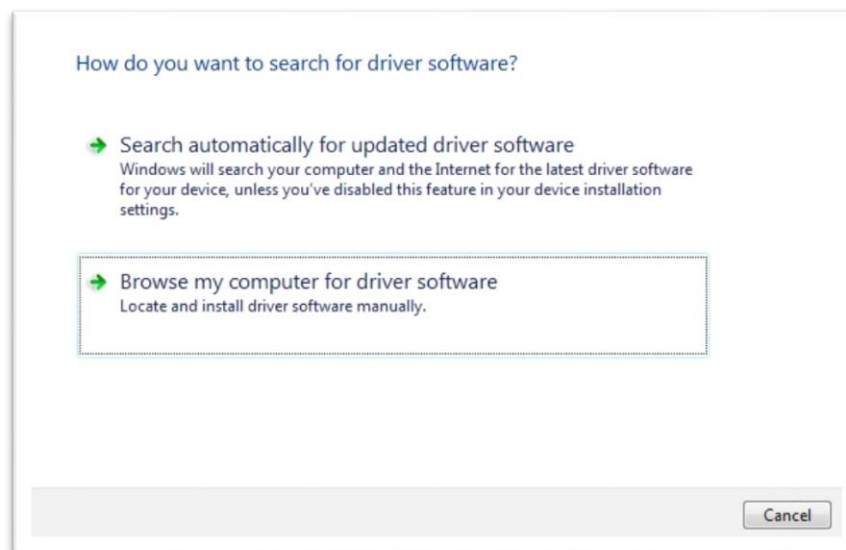
2.5.5 Flash Images

The following sections introduces two methods to flash the images to board via fastboot from Windows PC. Figure 2-2 shows the keys and cable connections for flashing images on MT2712 EVB.

- Downloa
- Power key
- Reset key
- Connect UART debug board
- Power
- Connect power adapter
- Connect USB cable

Figure 2-2. Keys and Cable Connections for Flashing Images on MT2712 EVB

2.5.6   Flash All Images

1. Connect 12V power adapter and turn on power switch.
2. Connect Host PC with USB cable. Notes: the MediaTek USB cable driver shall be installed correctly in advance.
3. Press power key up that turn off the power.
4. Copy images to one folder, execute flashimg-init.bat on windows OS host PC.
5. Press and hold download key on the target board. Notes: Do not release download key until following step say it.
6. Press power key down that turn on the power. Then some messages will be shown in the console.
7. After "sending 'BOOTIMG' (***** KB)..." is shown in host PC console, download key can be released.
8. When "Press any key to continue ..." is shown in the console, image flashing is finished. You can press reset key to reboot target board.

flashimg-init.bat will flash all images below:
  MBR_NOR
  MBR_NOR_EMMC
  lk.img
  tz.img
  boot.img
  cache.img
  recovery.img
  system.img
  userdata.img
  vendor.img

2.5.7   Flash Individual Image

1. Boot into fastboot mode
    1.1. Press and hold download key on the EVB board
    1.2. power on (or press the "Reset" button on the debug board if it is power on already)
    1.3. plugin the USB cable
    1.4. Open **Command Prompt** of Windows PC, and change the working directory to Android SDK platform tools installation directory
    1.5. Use fastboot command to check whether the device is in bootROM or LK fastboot mode
        $ cd D:\tools\android-sdk\android-sdk\platform-tools
        $ fastboot.exe devices
        If serial number is show, device enter fastboot mode successfully.

2. Default storage nor+eMMC flashing commands

    Note: it is necessary to execute step 1 and 2 below prior to any commands (e.g Execute step 1->2->9 in order for upgrading kernel boot.img)

    2.1. Enter Brom Fastboot mode
        $ fastboot flash EMPTY lk.img

2.2. Brom jump: fastboot continue (press download key before you hit ENTER, and only release it when you see lk entering fastboot mode)

2.3. Erase nor boot region

    $ fastboot erase nor0

2.4. Erase eMMC boot region

    $ fastboot erase mmcboot0

2.5. Erase eMMC user region

    $ fastboot erase mmc0

2.6. Flash boot region GPT

    $ fastboot flash nor0 MBR_NOR

2.7. Flash Bootloader

    $ fastboot flash bl1 lk.img

2.8. Flash EMMC user region GPT

    $ fastboot flash mmc0 MBR_NOR_EMMC

2.9. Flash trustzone

    $ fastboot flash tee1 tz.img

2.10. Flash devive tree overlay

    $ fastboot flash dtbo device_id.dtbo

2.11. Flash kernel

    $ fastboot flash boot boot.img

2.12. Flash other android images

    $ fastboot flash system system.img

    $ fastboot flash cache cache.img

    $ fastboot flash recovery recovery.img

    $ fastboot flash userdata userdata.img

   *$ fastboot flash vendor vendor.img*

2.6   Serial Console Connection

The MT2712 platform uses UART and an USB-RS232 adapter to establish serial communication with a Windows PC. The following items are required:

1. MT2712 EVB
2. MediaTek Debug Board and Debug Port Connection Cable
3. USB Type-A Male to Type-B Male Cable
4. USB-RS232 Driver for Windows PC
5. A Serial Terminal Program for Windows PC

Figure 2-3 shows how to connect the MT2712 EVB with MediaTek Debug Board and Debug Port Connection Cable.
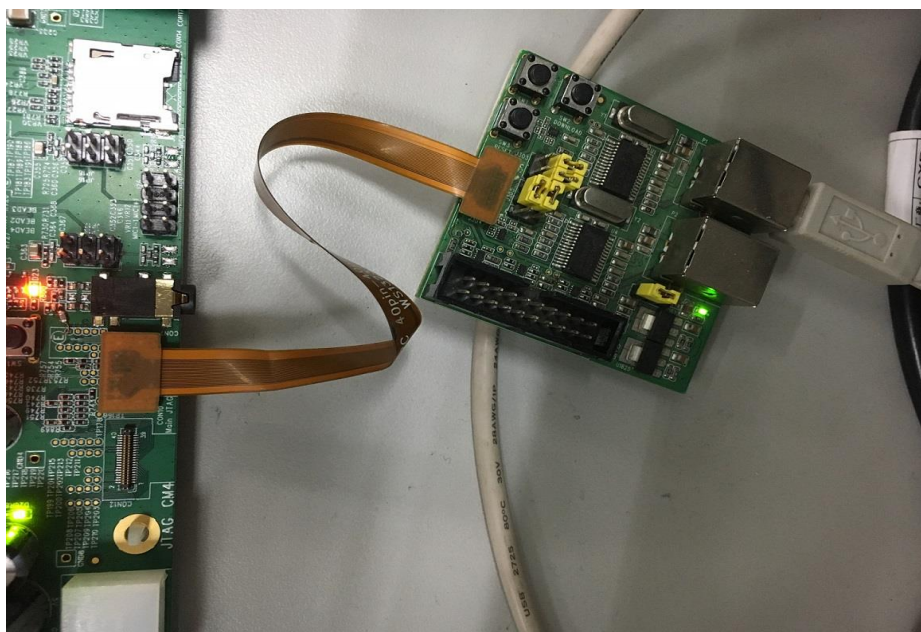
*Figure 2-3. The Connection of the MT2712 EVB and MediaTek Debug Board*

2.6.1 Debug Board Setup

MT2712 Platform fetches log from UART1. Please connect a USB cable with the debug board to get the debug feature.
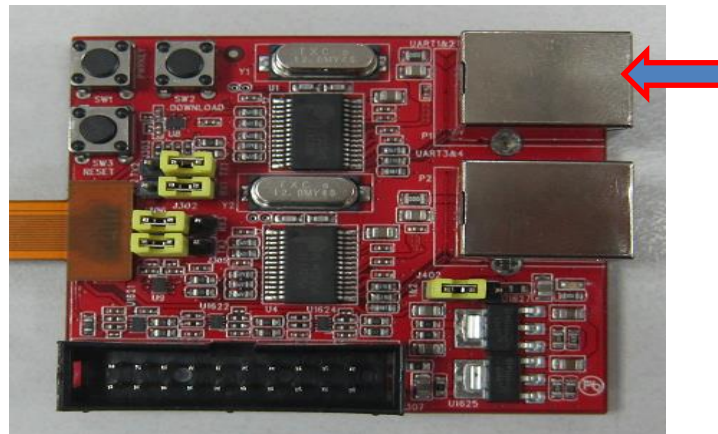
There are three types of MediaTek Debug Board that can be used with MT2712 EVB and shown in the following three figures. The jumper setting and cable connection setup are also shown.

Figure 2-4. MTK0028 Debug Board Setup
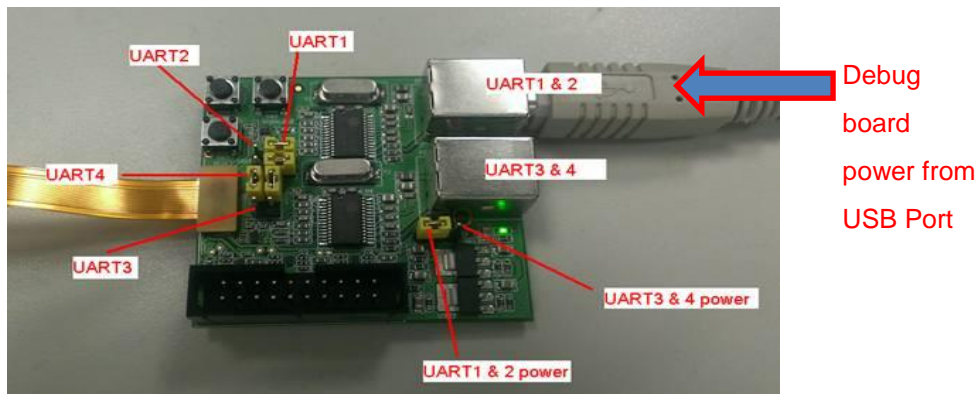
Figure 2-5. MD0843 Debug Board Setup

Figure 2-6. MD0961 Debug Board Setup

2.6.2   Install USB-RS232 Driver

The USB-RS232 Driver shall be installed on a Windows PC to enable the serial console connection with MT2712 EVB via MediaTek Debug Board.

To verify whether the driver is installed successfully, in **Device Manager**, to check if **Prolific USB-to-Serial Comm Port** is available.
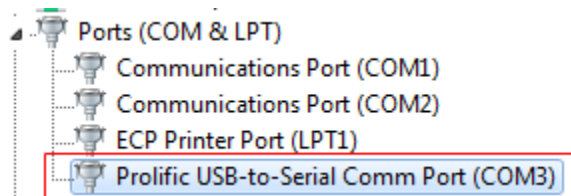
Figure 2-7. The Prolific USB-to-Serial Device in Device Manager

2.6.3   Serial Console Setting

A Serial Terminal Program on Windows PC can be used with the following serial console setting to establish the serial communication with MT2712 EVB.

1. Port: COM*x* (check the COM port number of **Prolific USB-to-Serial Comm Port** in **Device Manager)**
2. Baud Rate: 921600
3. Data Bits: 8
4. Parity: None
5. Stop Bits: 1
6. Flow Type: None

3   BSP Features

Here lists the features available in MT2712 Android BSP® for MT2712P1V1 EVB now.
- boot to home screen within 23S;
- audio playback;
- video playback (need file manager and push video file to internal storage);
- mouse / touch panel;

- Others
  RTC/brightness level/night mode/ calendar so on, setting OK