Chipsee

Industrial PC

# Android 4 OS on iMX6Q User Manual

For iMX6Q Products

Content can change at anytime, check our website for latest information of this product.
**www.chipsee.com**

# Contents

# Android 4 OS

**Android 4 OS User Manual**

This manual is used to provide users with a fast guide of Chipsee Industrial Computers (Abbreviated as IPC) about Android 4 OS development. Through this manual, users can quickly understand the hardware resources; users can build a complete compilation of Android development environment; users can debug Android 4 OS via serial, USB OTG and Internet.

| Revision | Date | Author | Description |
|---|---|---|---|
| V1.0 | 2021-12-30 | Randy | Initial Version |

**SUPPORTED BOARDS:**

> *CS10600F070 CS12800F101 CS14900F190 CS19108F215*

**PREBUILT FILES PACKAGE:**

Prebuilt files for the various industrial PCs can be found in the OS Downloads.
Below are the links to the prebuilt files for each industrial PC model.

- CS10600F070
- CS12800F101
- CS14900F190
- CS19108F215

System Features

| Feature | Comment |
| --- | --- |
| System | Android 4 |

# Preparation

You will need to prepare the following items before you can start using the Prebuilt Files Package to re-flash the system.

Power Supply Unit (PSU) with the appropriate voltages, as follows:

- These products: CS14900F190 and CS19108F215 requires a 15V to 36V power adapter.
- The CS10600F070 product needs a 6V to 36V power adapter.
- The CS12800F101 product needs a 12V to 36V power adapter.

## Hardware Requirements

- Chipsee Industrial PC
- PSU according to the instructions above
- USB-to-serial or other serial cable for debugging
- USB A-A cable (used only if the hardware configured as OTG)
- Windows 7 PC

## Software Requirements

- Android 4 OS Prebuilt Files Package (from the link above)
- ADT for Windows
- Android USB driver (for Windows)
- MFGTools

# Getting Started and Tests

> **Note**
>
> Throughout this section, the user can use both the pre-built Android 4 image files and the MFGTools software to burn files to the system, boot system and perform necessary software and hardware test.

## Boot Switch Configuration

CS-IMX6 has a boot configuration select switch, as shown on the figure below. You can use the boot select switch to change between two modes, namely

- eMMC Boot
- Download

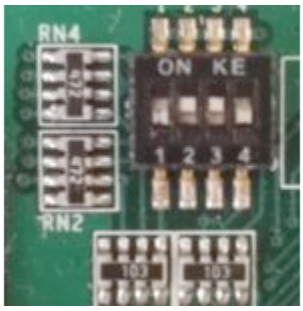



Figure 129: Boot Mode Setup

| SW Mode | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| eMMC | 1 | 1 | 0 | 1 |
| Download | 0 | 1 | 1 | 0 |

Table 41 Boot Configuration Selection

## Prepare Manufacturing Tool and Image

The manufacturing tool, referred to as **MFGTool**, is a tool that runs on a Windows PC. You can use it to download pre-built images to the eMMC on a Chipsee board. The tools directory contains the `tar.gz` file.

| MFGTool | Windows download tool |
|---|---|
| Kernel Image | Boot.img |
| U-boot Image | u-boot.bin |
| Recovery Image | Recovery.img |
| Android File System | System.img |

| MFGTool | Windows download tool |
|---|---|
| **Industrial Computer** | One |
| **USB OTG Cable** | One |
| **12V-2A power adapter** | One |

## Downloading Images by using MFGTool

Chipsee IPC supports booting from an integrated eMMC.

### Configuring MFGTool

To configure MFGTools, follow these steps:

- Unzip `Mfgtools-Rel-xxx_xxxxxx_MX6Q_UPDATER_Vxx.tar.gz` file.
- Open the extracted folder `Mfgtools-Rel-xxx_xxxxxx_MX6Q_UPDATER_Vxx` and edit `cfg.ini` file.

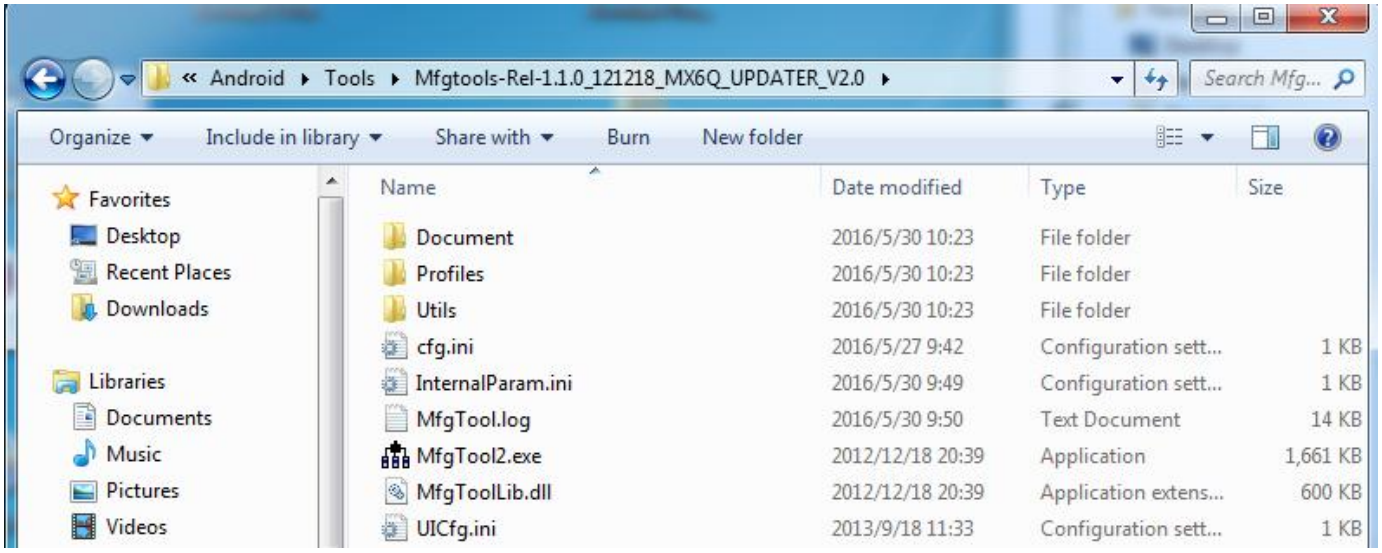



Figure 130: *Extracted folder content*

- In the `cfg.ini` file, ensure the `name` variable is set to `Android-SabreSD-eMMC`, as shown on the figure below.

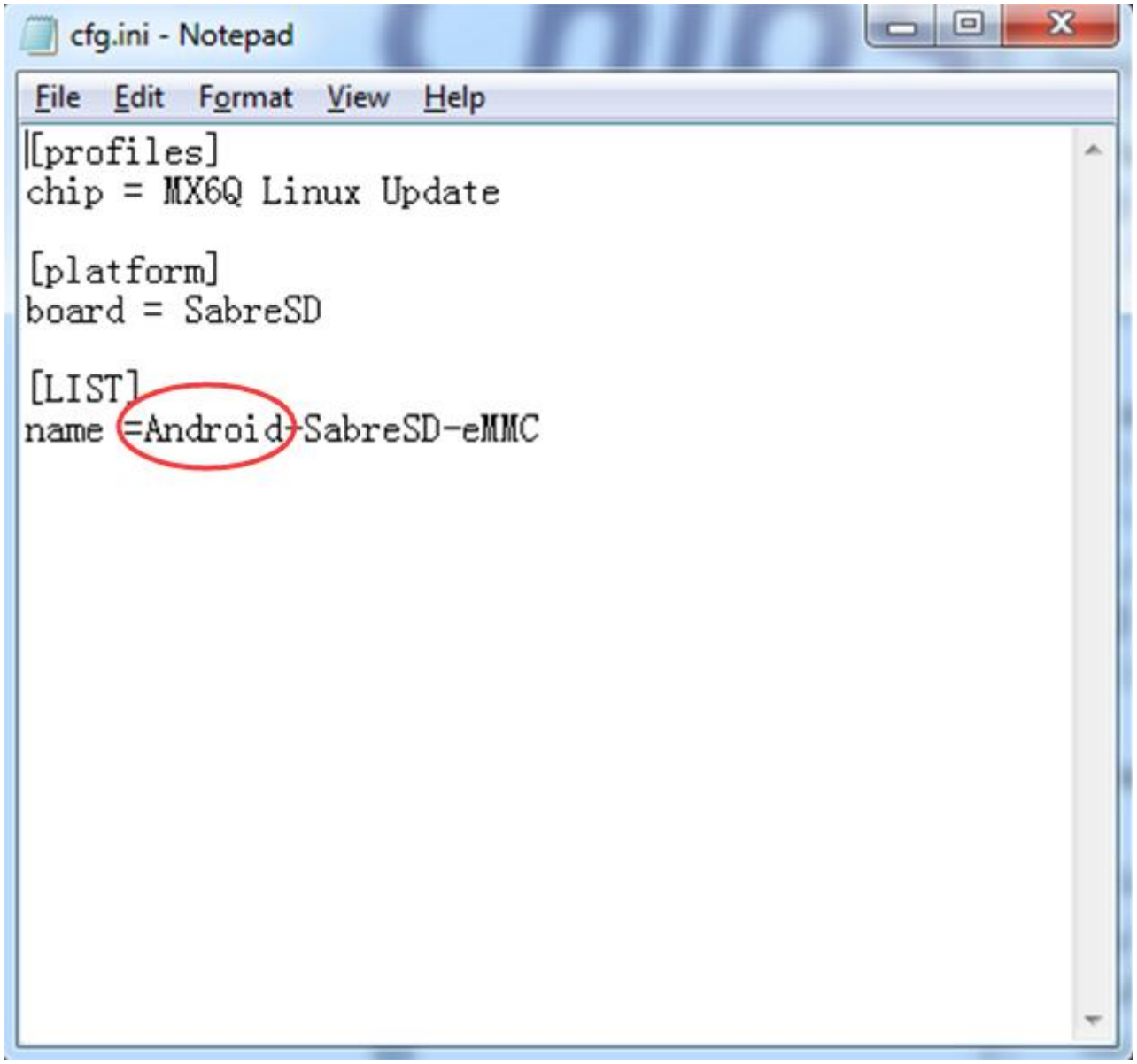



Figure 131: *Cfg.ini file*

### Copy Image To Android Directory

Follow these steps to copy image to Android directory:

- Unzip `prebuilt-csXXXXXfXXX-vX-android-YYYYMMDD.rar` file. The extracted folder will contain these files: `boot.img`, `recovery.img`, `system.img`, and `u-boot.bin`.
- Copy the files in the extracted folder to *Mfgtools-Rel-XXX_XXXXXX_MX6Q_UPDATER_VXX\Profiles\MX6Q Linux Update\OS Firmware\files\android* directory.

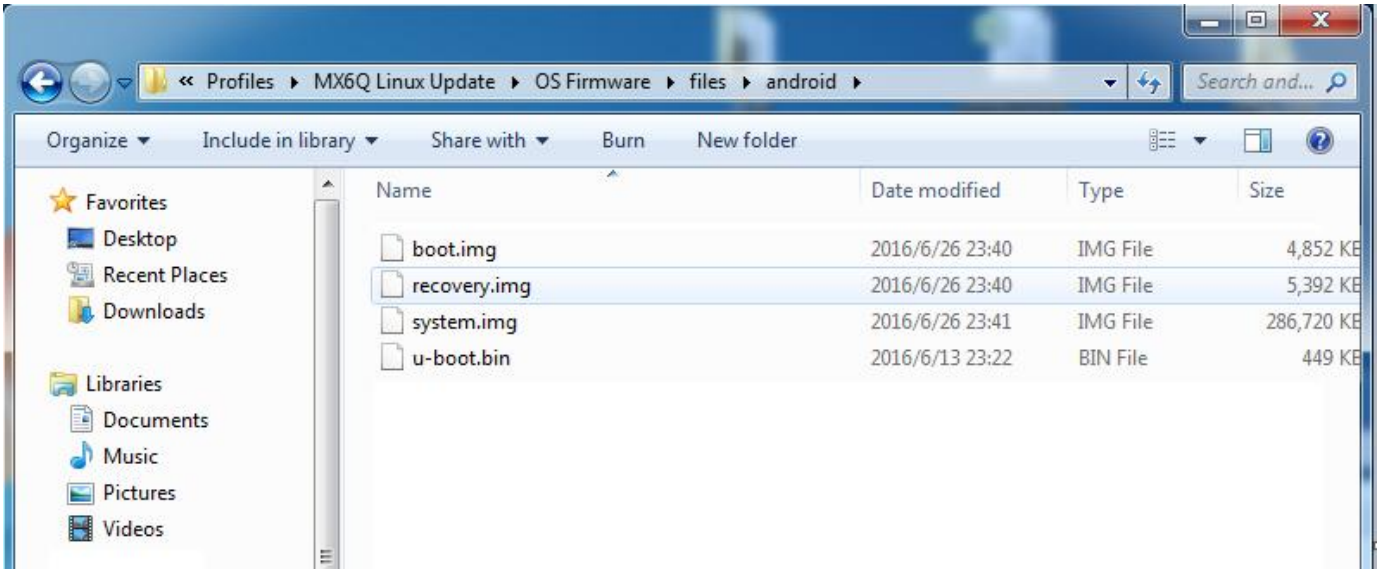



Figure 132: *Extracted folder with files*

### Using MFGTool

1. Connect a USB OTG cable from a Windows PC to the USB OTG port on the IPC.
2. **Change the boot select configuration to 0 1 1 0, as shown on the figure below.**

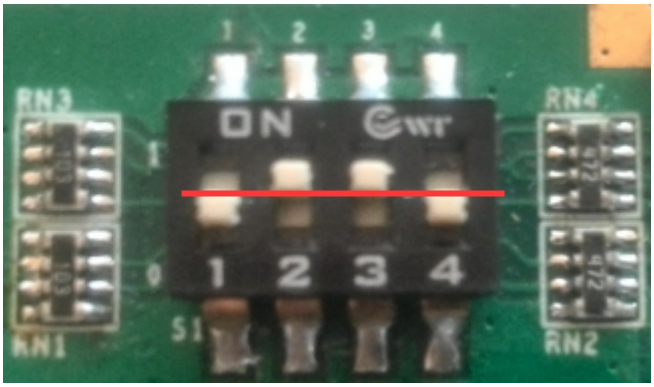



Figure 133: *Boot Switch Config*

3. Connect a 12V-2A power adapter to the IPC and power ON.
4. **On your Windows PC, open the `Mfgtools-Rel-XXX_XXXXXX_MX6Q_UPDATER_VXX` directory and run the `MfgTool2.exe` file, as shown on the figure below.**

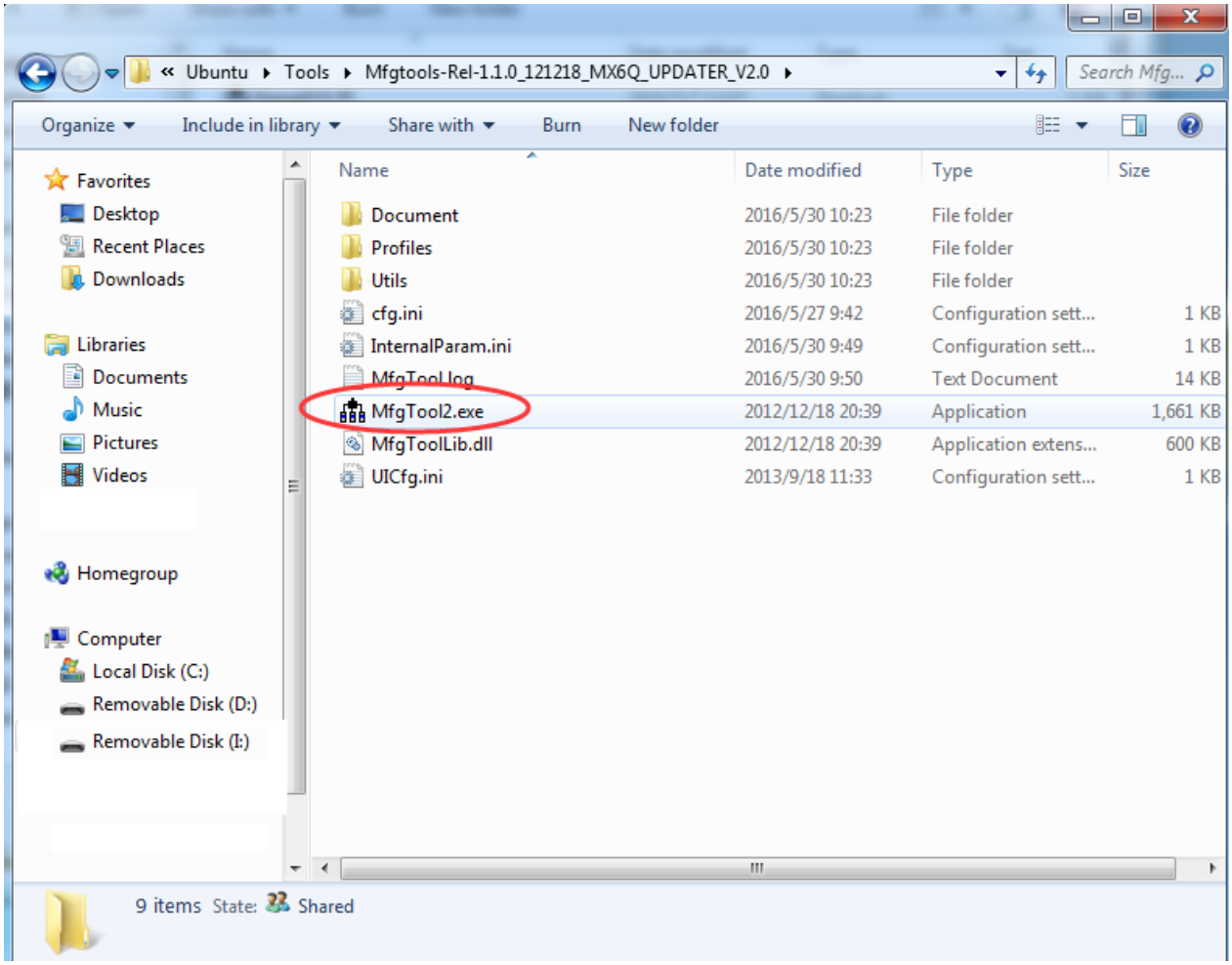



Figure 134: *Run* **MfgTools2.exe** *file*

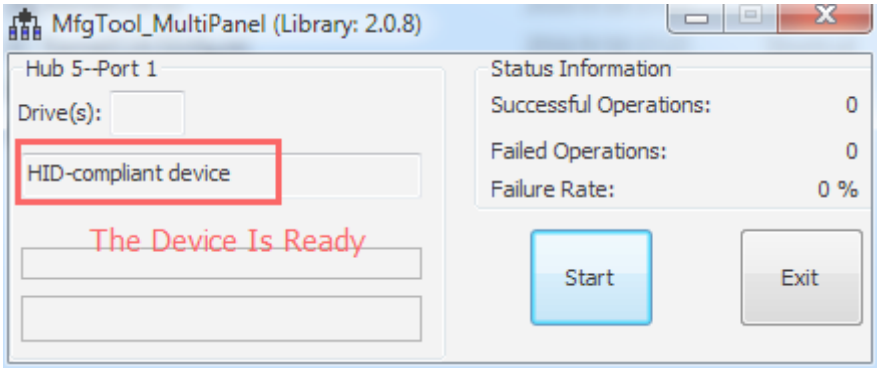



Figure 135: *Prepare to start*

**Note**

If you get a message saying *No Device Connected*, check the USB-OTG cable to ensure it is ready.

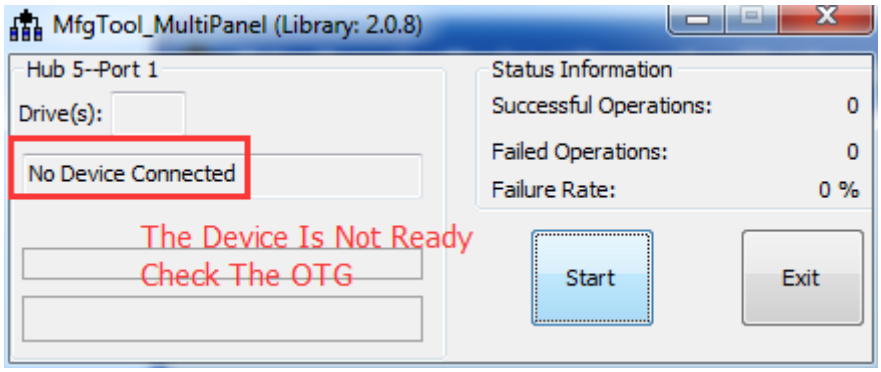


Figure 136: *The USB-OTG cable is not connected correctly.*

5. **Click on Start button to download the Image.**

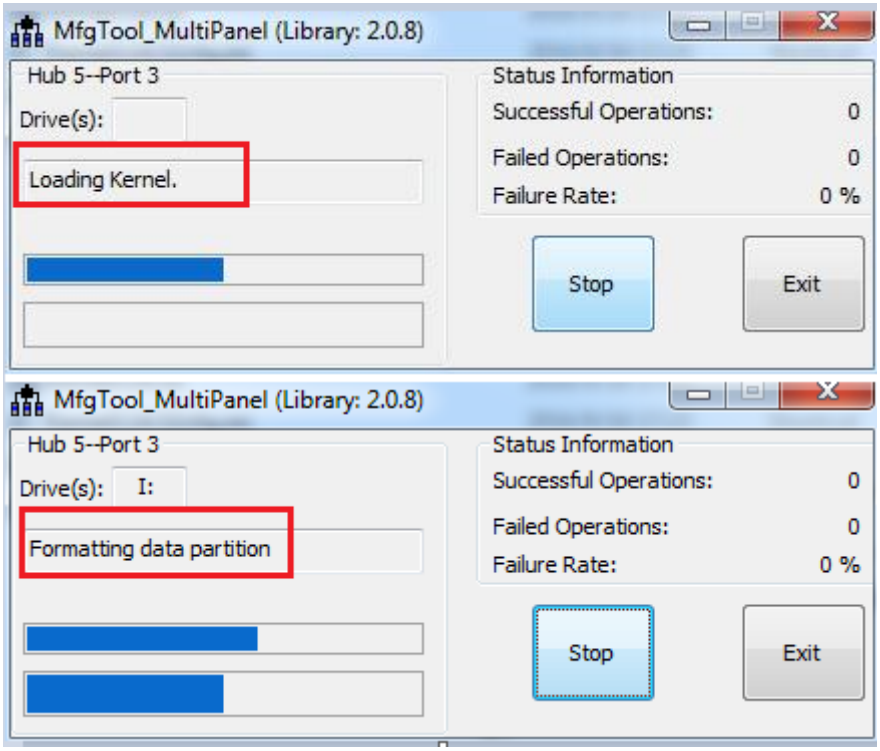
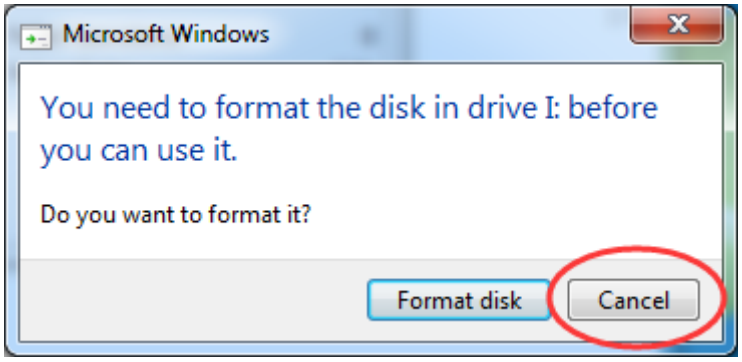



Figure 137: *Loading Kernel and Formatting rootfs partition*

**Note**

If you are using a Window 7 PC, you will receive a prompt that asks you to format the disk. Please ignore or cancel it.




6. **When the process is complete, you click the Stop button to stop downloading Image and exit.**

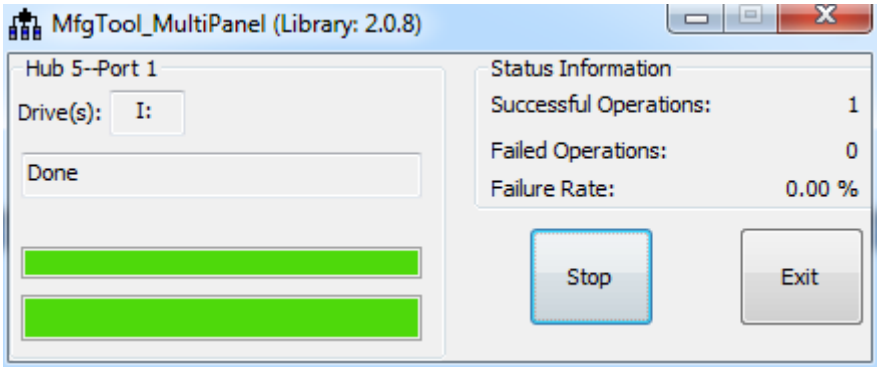



Figure 138: *Download Image is finished*

## Booting Android OS And Test(Using 7inch as example)

The first time you start Android 4 OS on the industrial PC will take a little time. But after the first time, Android 4 OS will start quickly. It is a successful start if you see the Android 4 OS desktop such as the one shown in the figure below:


Figure 139: *Android Desktop Screen*

### SD Test

The IPC supports SD card hot-plug. The figure below shows the message when you plug the SD card into IPC, but the IPC will not mount the SD Card.

```
root@sabresd_6dq:/ #
root@sabresd_6dq:/ # mmc1: new high speed SDHC card at address 59b4
mmcblk1: mmc1:59b4 SS08G 7.40 GiB
 mmcblk1: p1 p2
```

Figure 140: *Serial Message*

You can mount the plugged SD card by using the following command.

```
# mount -t vfat /dev/block/mmcblk1p1 /mnt/extsd/
```

```
root@sabresd_6dq:/ # mount -t vfat /dev/block/mmcblk1p1 /mnt/extsd/
root@sabresd_6dq:/ # ls /mnt/extsd/
MLO
boot.scr
u-boot.img
uImage
ubi.img
root@sabresd_6dq:/ #
```

### USB Flash Disk Test

The USB Flash Disk is like the SD Card, except for the mount point. The IPC mounts the USB Flash Disk to */mnt/udisk* .

```
root@sabresd_6dq:/ # ehci_fsl_bus_resume begins, Host 1
ehci_fsl_bus_resume ends, Host 1
usb 2-1.1: new high speed USB device number 3 using fsl-ehci
scsi0 : usb-storage 2-1.1:1.0
scsi 0:0:0:0: Direct-Access     EAGET    F30              1.00 PQ: 0 ANSI: 6
sd 0:0:0:0: [sda] 30326784 512-byte logical blocks: (15.5 GB/14.4 GiB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Mode Sense: 23 00 00 00
sd 0:0:0:0: [sda] Write cache: disabled, read cache: disabled, doesn't support DPO or FUA
 sda: sda1
sd 0:0:0:0: [sda] Attached SCSI removable disk
```


Figure 141: *USB flash disk test*

### SATA Test

The SATA does not support hot-plug, but the Android 4 OS supports NTFS. You can use the NTFS-3g tool to mount SATA disk partitions by running this command:

```
# ntfs-3g /dev/block/sda1 /mnt/shm/
```

```
root@sabresd_6dq:/ # ntfs-3g /dev/block/sda1 /mnt/shm
root@sabresd_6dq:/ # ls /mnt/shm/
360SANDBOX
360榛樿€熸簮杞 AUTOEXEC.BAT
CONFIG.SYS
Documents and Settings
IO.SYS
InstallC112
Intel
MSDOS.SYS
MSOCache
MaxReport
NTDETECT.COM
Program Files
RECYCLER
System Volume Information
WINDOWS
bar.emf
boot.ini
bootfont.bin
dell
iNodeLog
ntldr
pagefile.sys
testbookmarks.html
tmp
root@sabresd_6dq:/ #
```

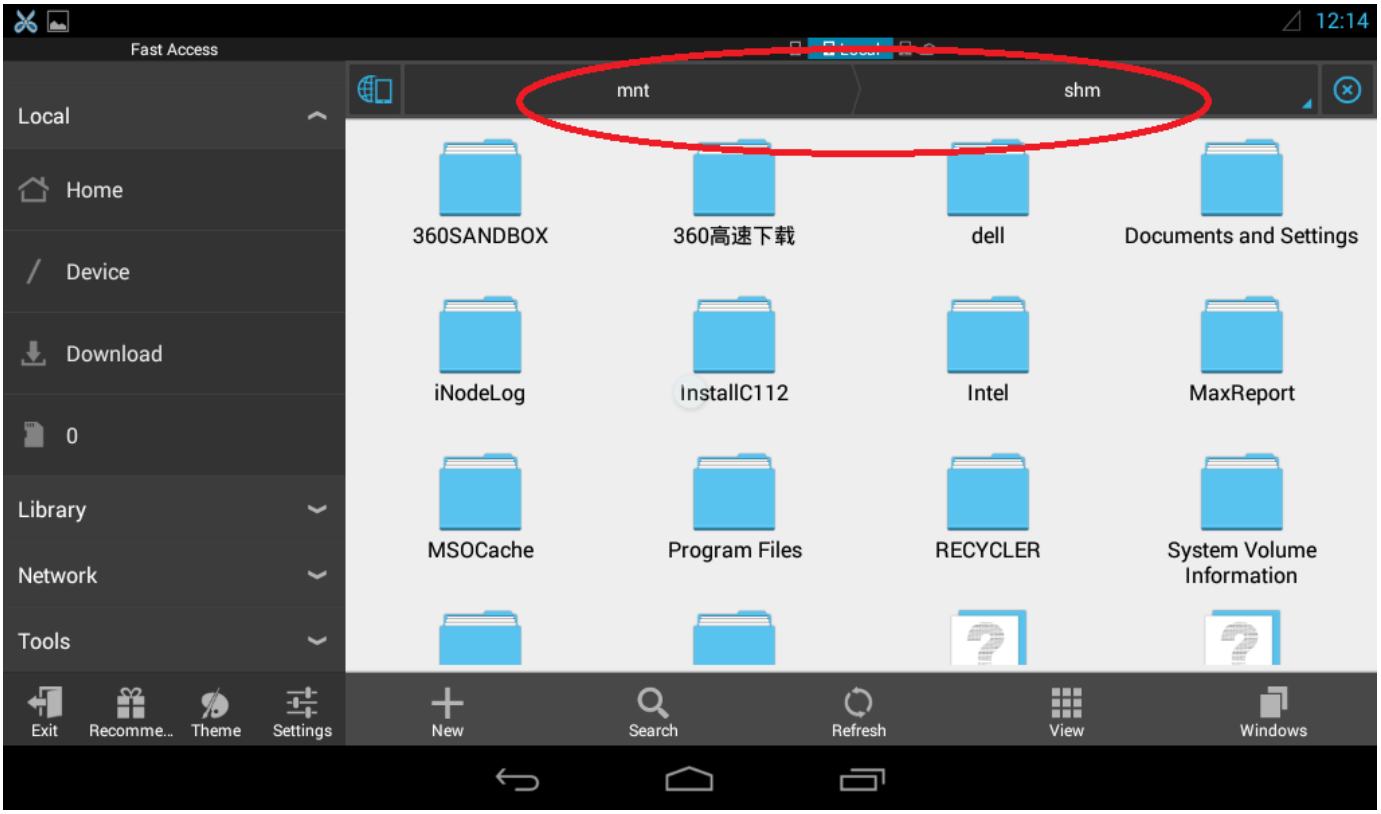
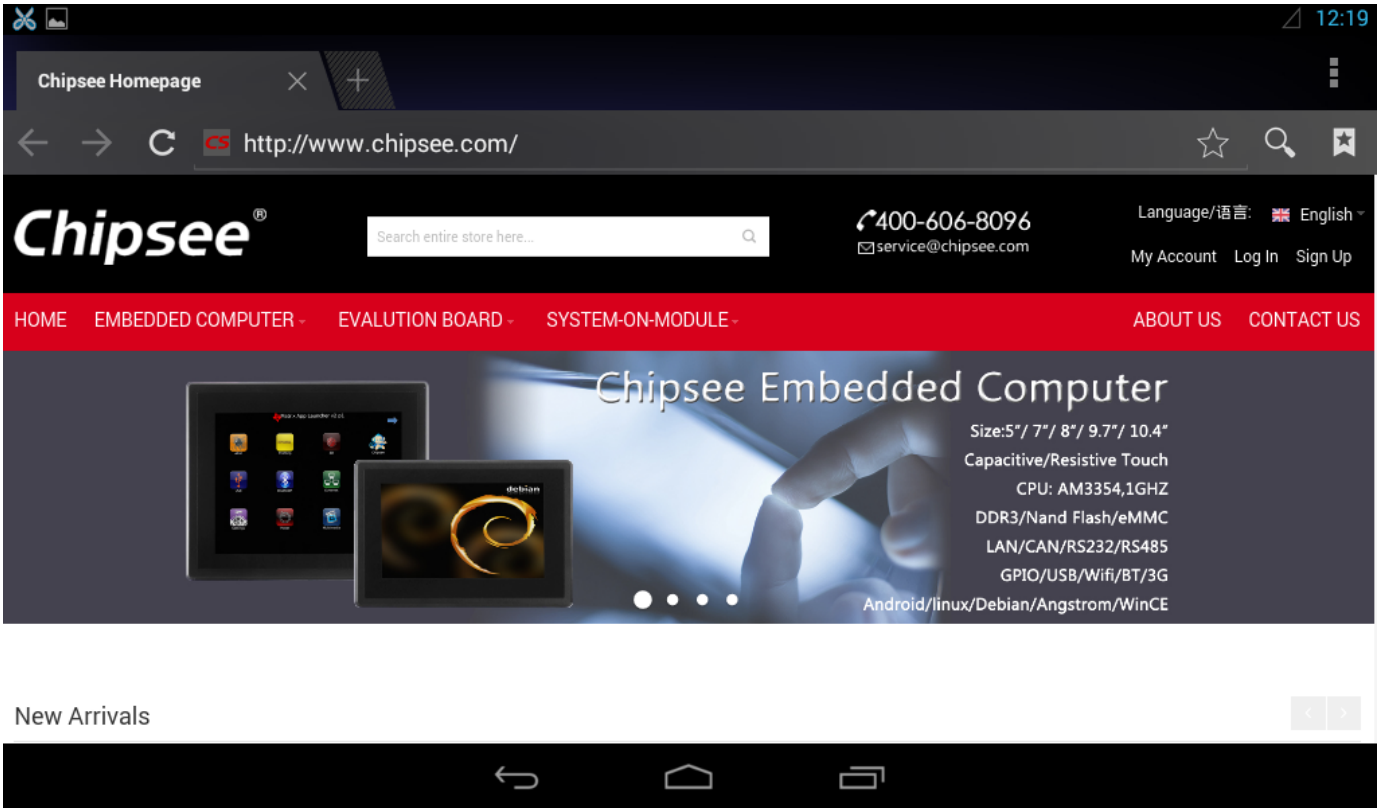



Figure 142: *SATA test*

### Network Test

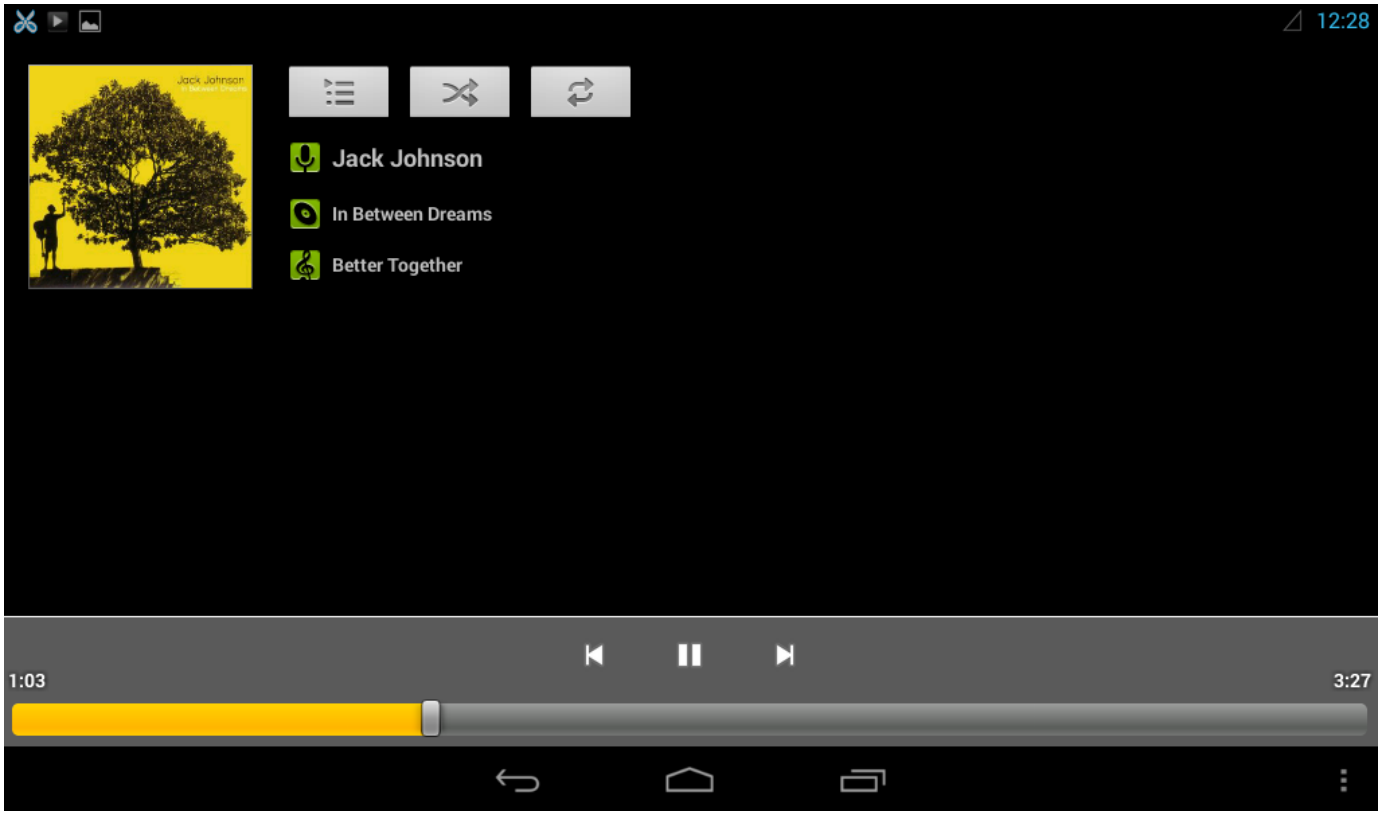
The network uses DHCP to get IP Addresses. You can get network access if you connect a LAN cable from the IPC to a router.
You can change the IP Address with this command.

```
# netcfg eth0 dhcp
```



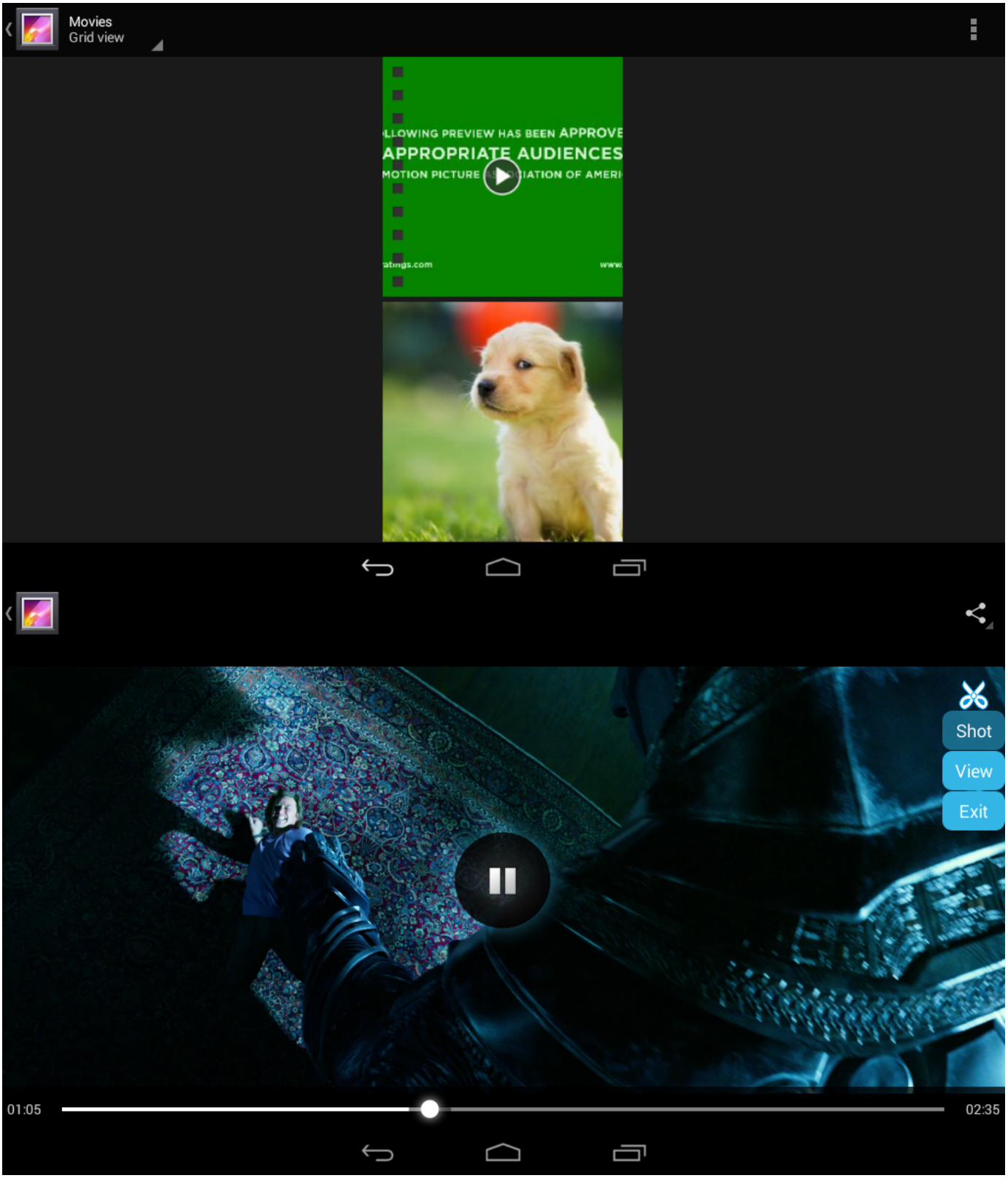
### Sound Card Test

Please open an audio file to test the Sound Card.



### Video Test

Please open a video file to test the Video.




### HDMI Test

You can reference this document, `IMX6Q U-boot Setting HDMI Output For Android.pdf`, to learn about performing HDMI tests.

> **Note**
>
> HDMI does not support hot-plug. The sound comes from the HDMI monitor, neither the speaker nor the headset on board.

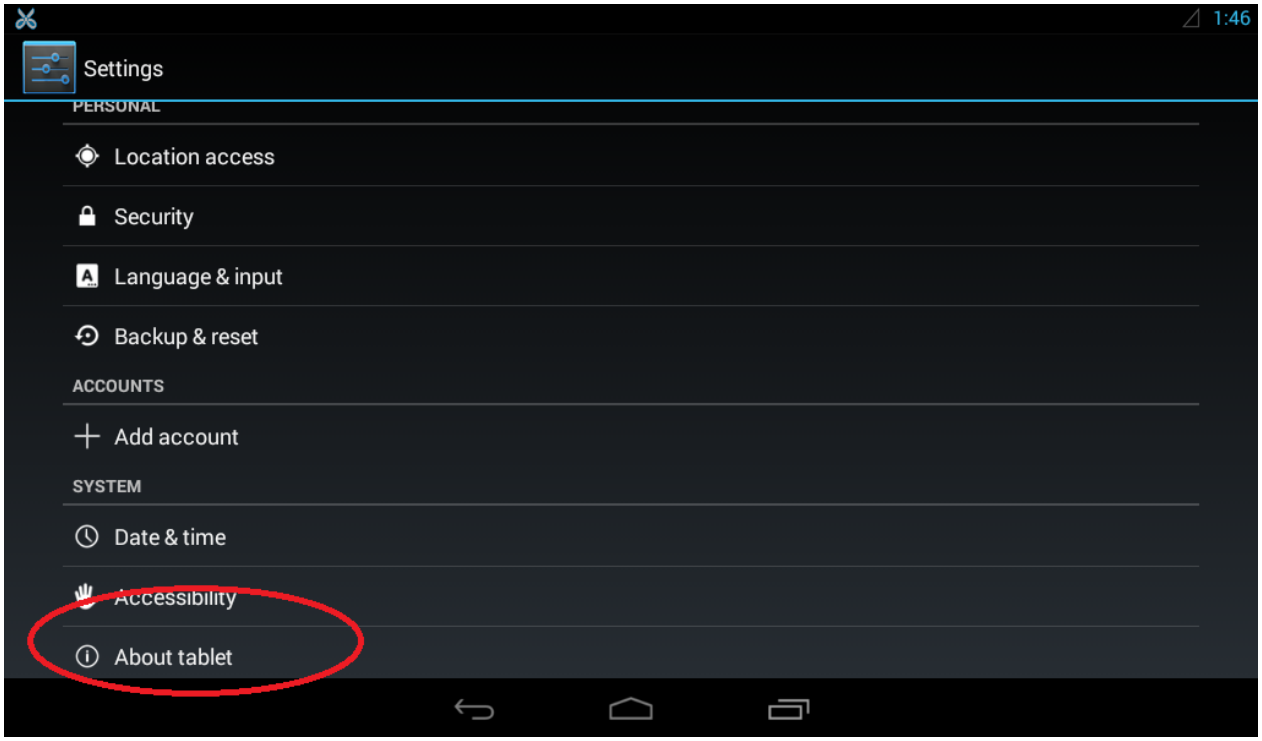
### WIFI Test

You must ensure the IPC has an SDIO Wi-Fi module integrated before performing the Wi-Fi test. If the IPC has an SDIO Wi-Fi Module, you can connect to the Wi-Fi and open a browser to test.

### ADB Test

You must enable USB debugging on IPC before performing ADB test.|br| Follow these steps to enable USB debugging.

- **Goto the Settings and tap on About tablet tab, as shown on the figure below.**




- **Tap on the Build number repeatedly until you enable Developer Options, as shown on the figure below.**

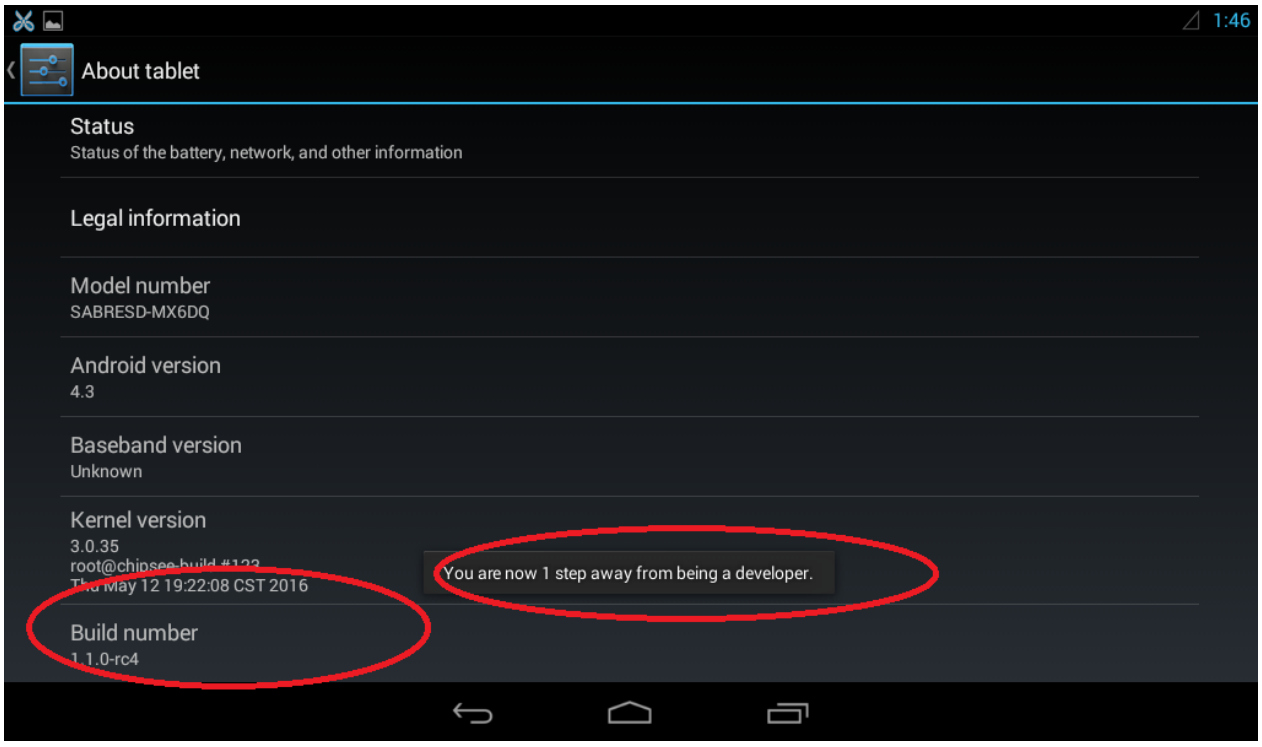



Figure 143: *Enable Developer Options*

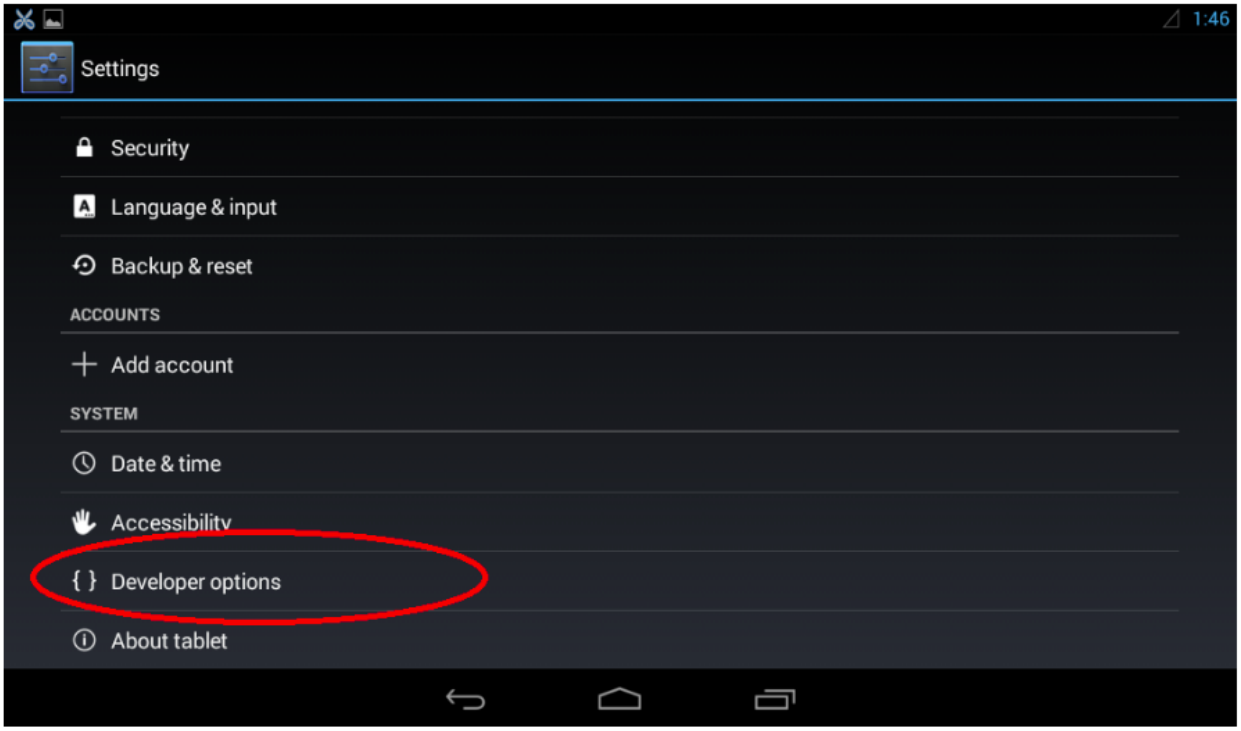



Figure 144: *Developer Options*

- **Move back to Settings and tap on Developer Options. You can enable** *USB debugging* **by ticking the checkbox next to it.**

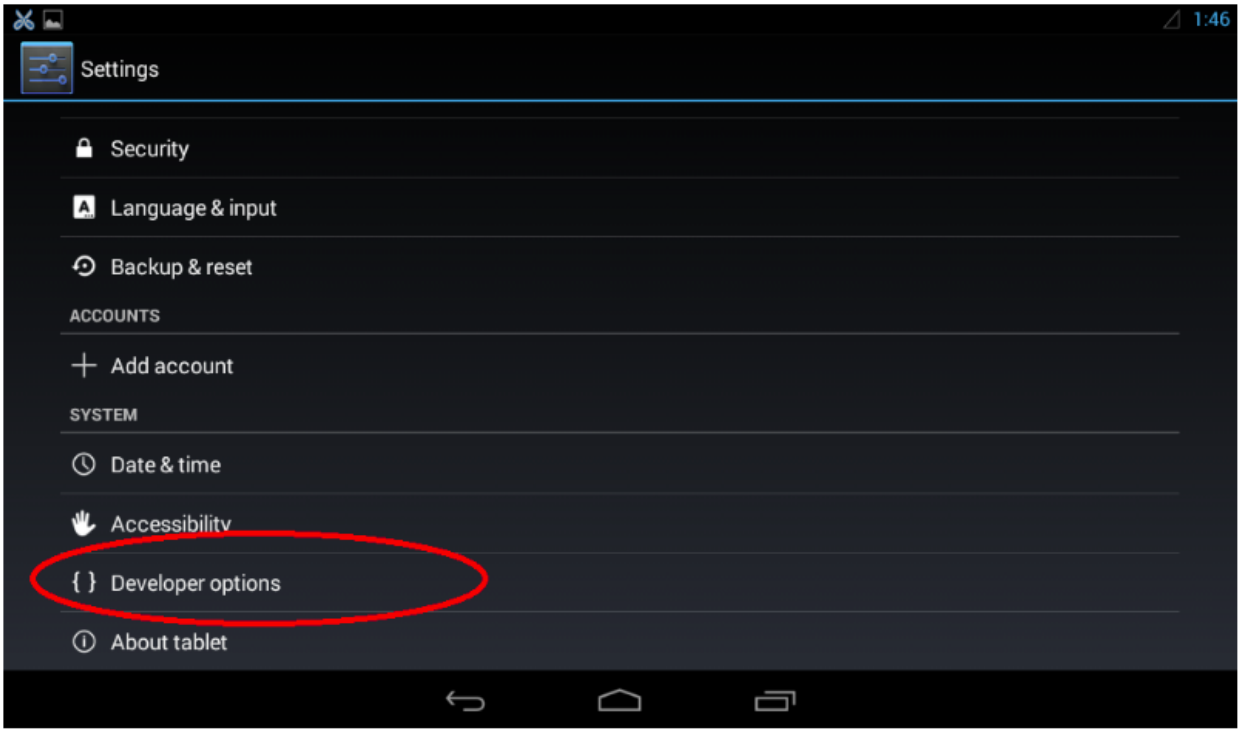



Figure 145: *Open USB Debugging*

Also, you can use the ADB tool in the tools directory to test the ADB.

- **Unzip it to the root directory of your Windows PC (Drive C), as shown on the figure below.**

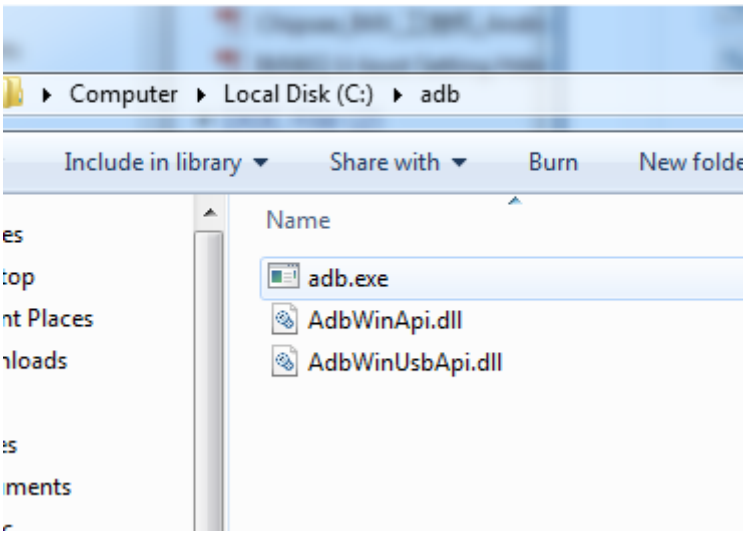



Figure 146: *Unzip adb.rar to c:*

- **You need to add path of the ADB directory to system's environment variable. Follow the steps described in the figures below to set the environment variable.**

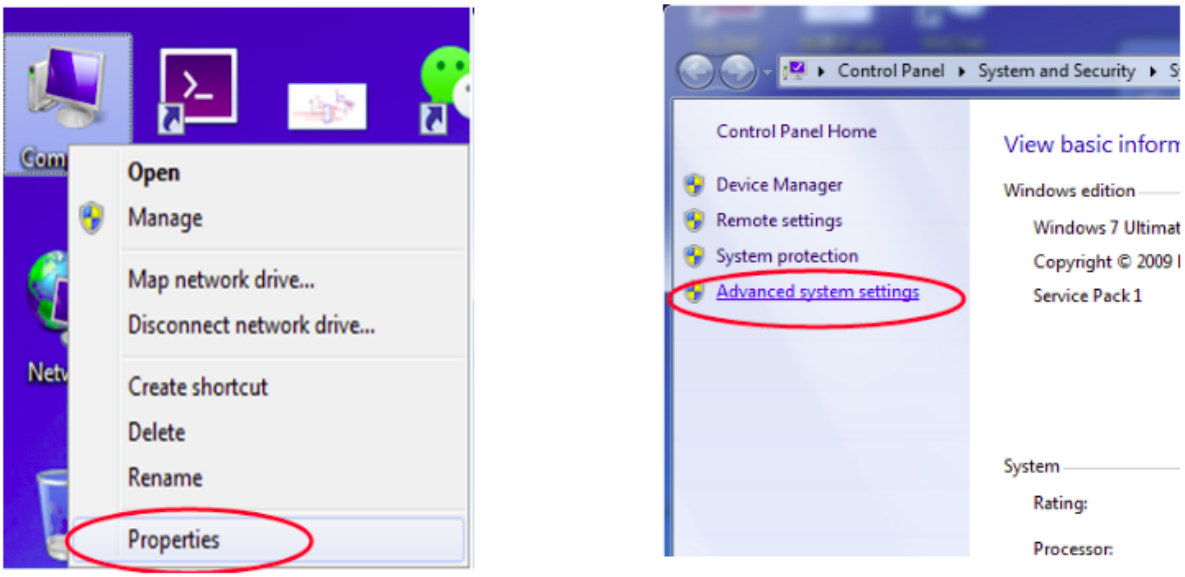



Figure 147: *Open Advance system settings*

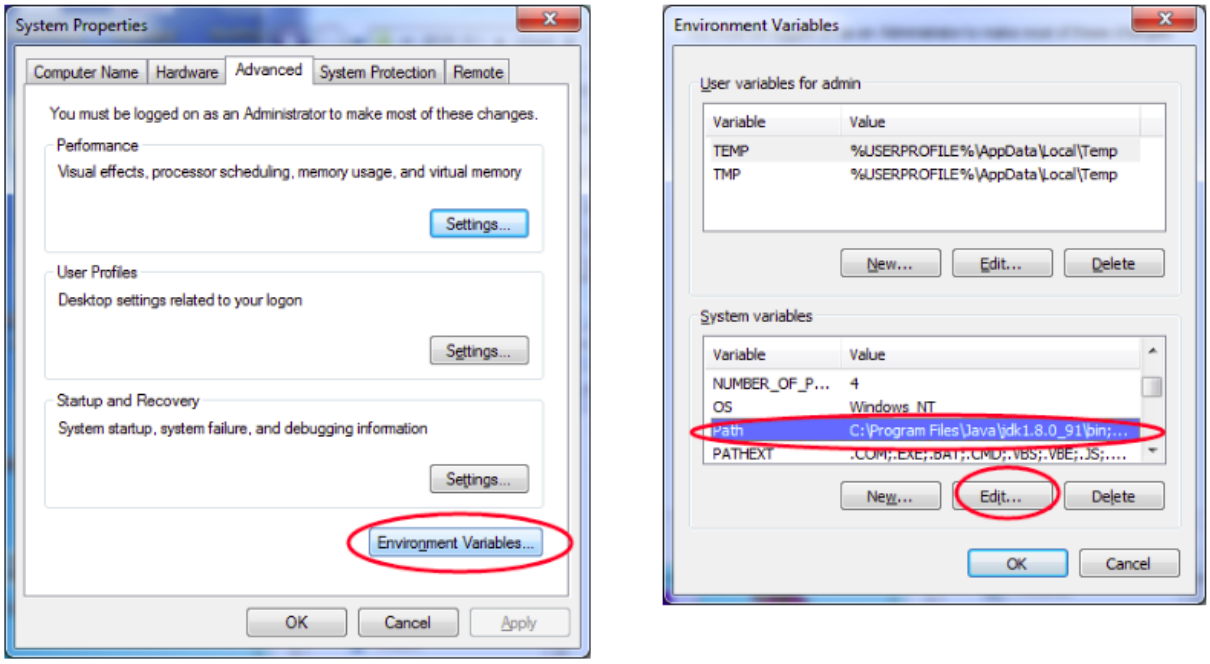



Figure 148: *Open and edit the **Path* system variable*

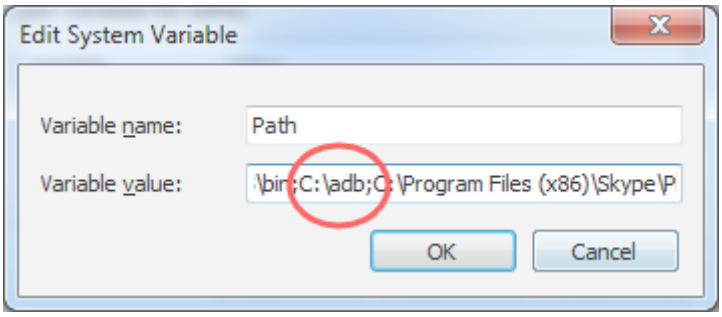



Figure 149: *Add path of the ADB directory to the* **Path** *system variable*

- **Open the command-prompt on Windows and enter this command** *adb version* **, as shown on the figure below. The process is successful, if the command-prompt displays the version number of ADB.**

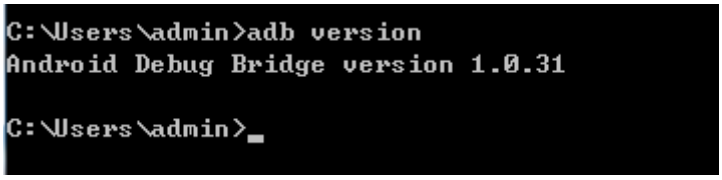



Figure 150: *ADB tool is working*

- **Connect the USB-OTG cable from the Windows PC to IPC. You will get a message Allow USB debugging?. Please select Always allow from this computer and click Ok.**

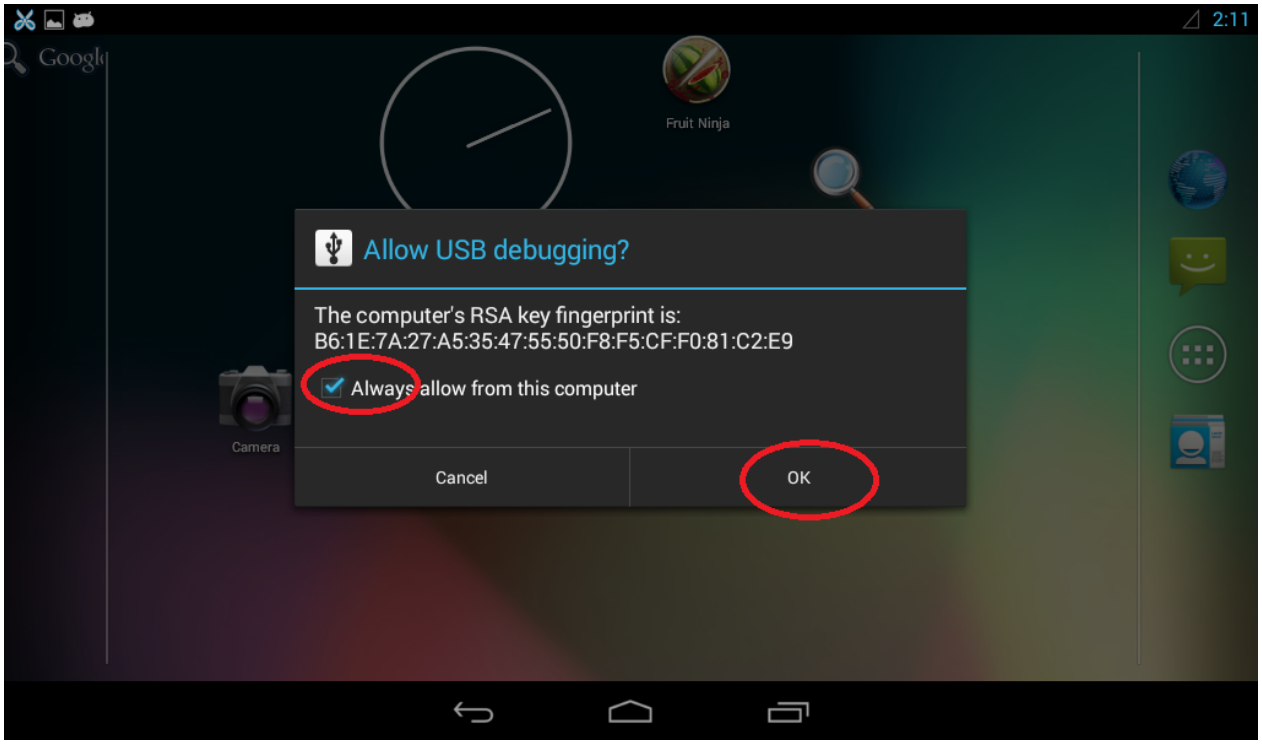



Figure 151: *Enable USB debugging*

You can list the devices attached to the Windows PC with this command.

```
$ adb devices
```

```
C:\Users\admin>adb devices
List of devices attached
0123456789ABCDEF        device
```

Figure 152: *Checking devices attached*

You can install an android app from the Windows PC onto the IPC with this command.

```
$ adb  install  XXX.apk
```

```
C:\Users\admin>adb install E:\share\APK\com.rovio.angrybirds-1.apk
1305 KB/s (29000466 bytes in 21.685s)
        pkg: /data/local/tmp/com.rovio.angrybirds-1.apk
Success

C:\Users\admin>
```

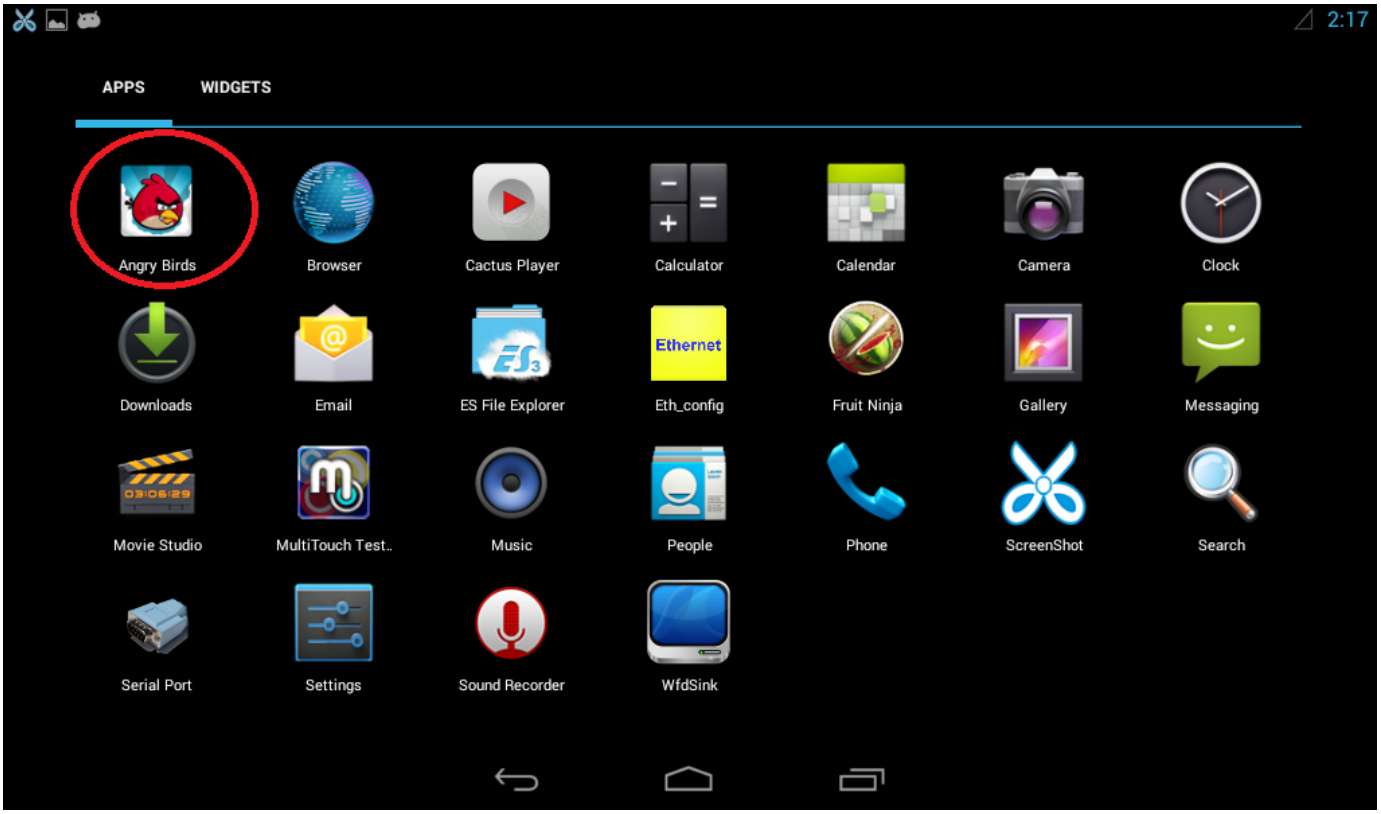



Figure 153: *Install android app*

### Touch Screen Test

Run **MultiTouch** Tester App.
The screen will show the number and position of the touch point when touching the screen.

> **Note**
>
> - Resistive screen expansion board only supports single-touch, and capacitive screen expansion board supports five-point touch as described in the figure below.
> - The 21.5”, 19”, and 17” capacitive screen supports a ten-point touch.

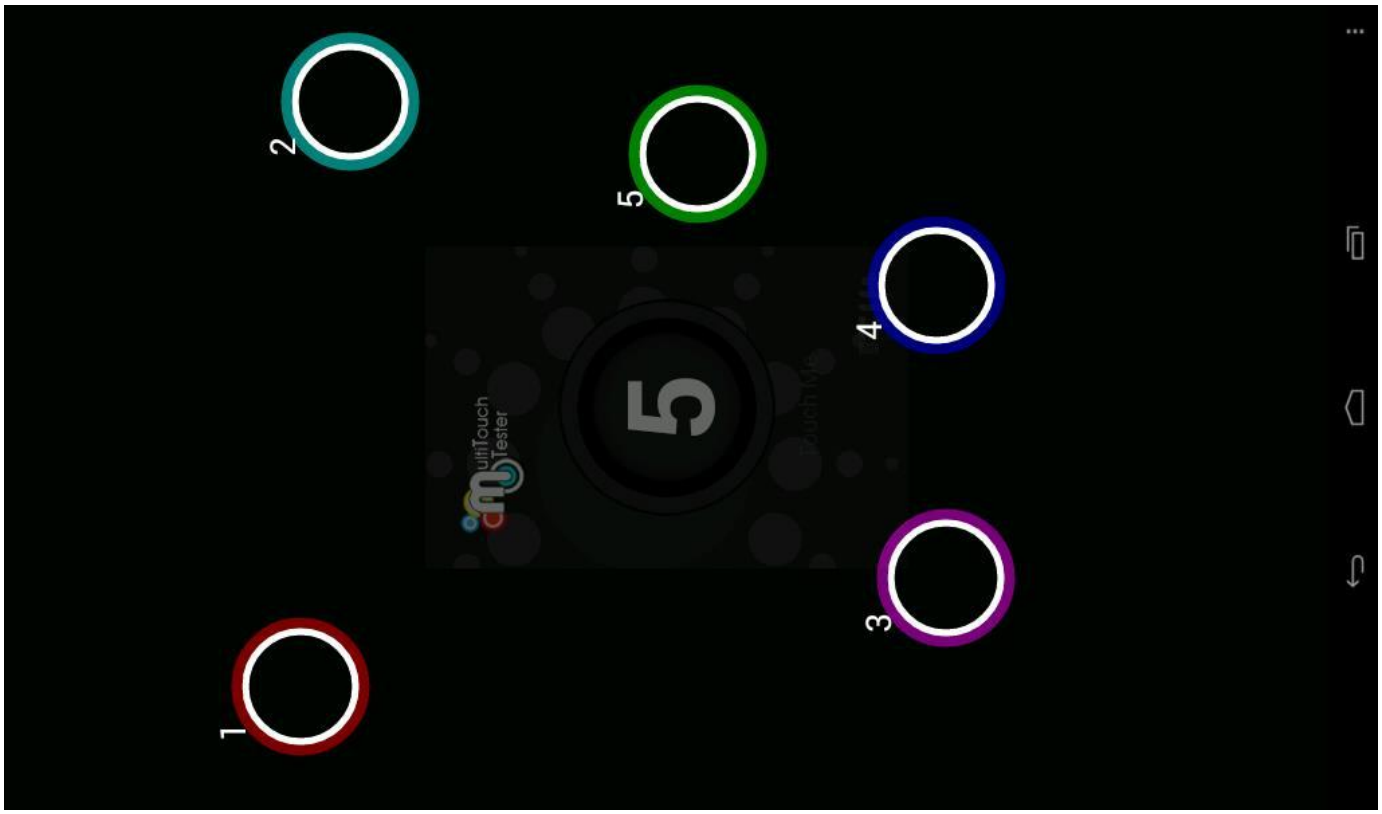



Figure 154: Touch screen test (Capacitive touch)

### Serial Test

There are five serial ports on the Chipsee IPC: 2 x RS232 and 3 x RS485 (can be customised). Refer to the table below for the available serial device nodes.

| Ports | Device Node |
|---|---|
| COM1(RS232, Debug) | /dev/ttymxc0 |
| COM2(RS485) | /dev/ttymxc1 |
| COM3(RS232) | /dev/ttymxc2 |
| COM4(RS485) | /dev/ttymxc3 |
| COM5(RS485) | /dev/ttymxc4 |

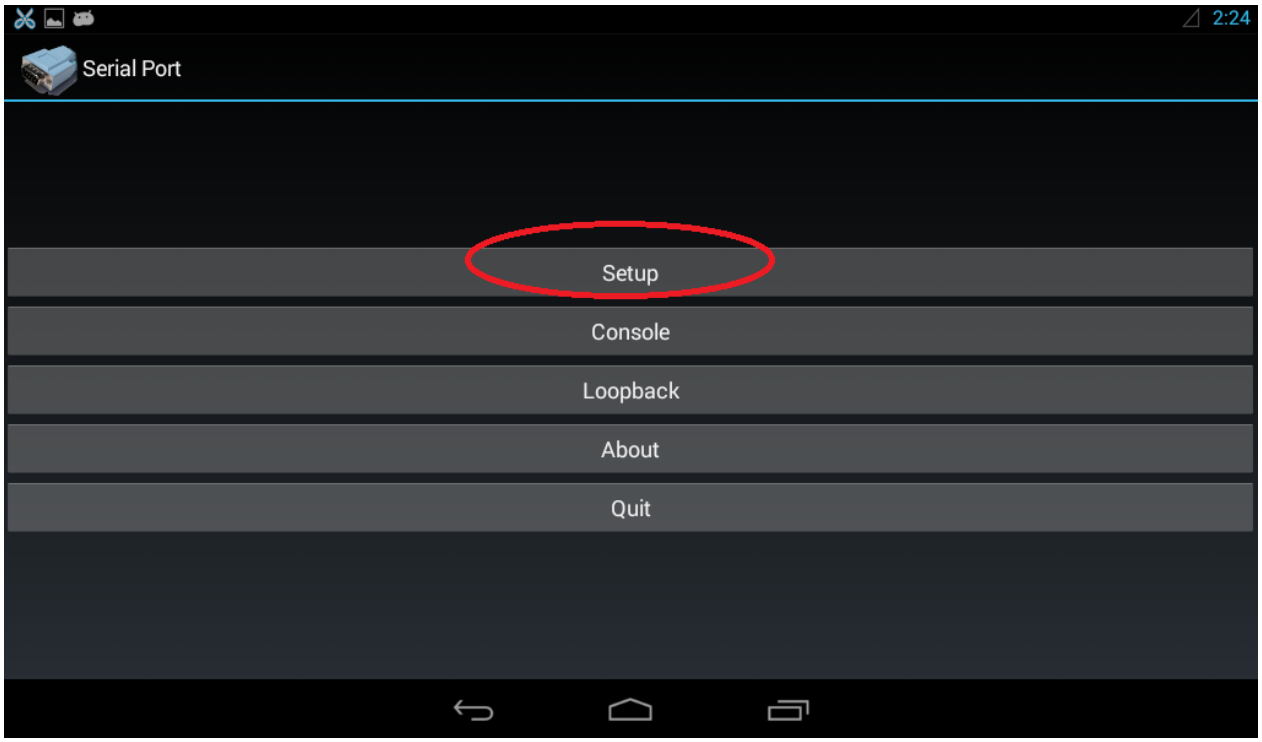
Table 42 Serial Ports Nodes on the System

**Note**

If you use COM2(RS485), you can't use Bluetooth because COM2(RS485) share pin with Bluetooth.

You can install the **SecureCRT** or **Putty** software on a Windows 7 PC to test the serial ports by following these steps:

- Connect COM1 on industrial PC board to Windows 7 PC.
- **Run Serial Port API App to communicate with Windows 7 PC, as shown on the figure below.**



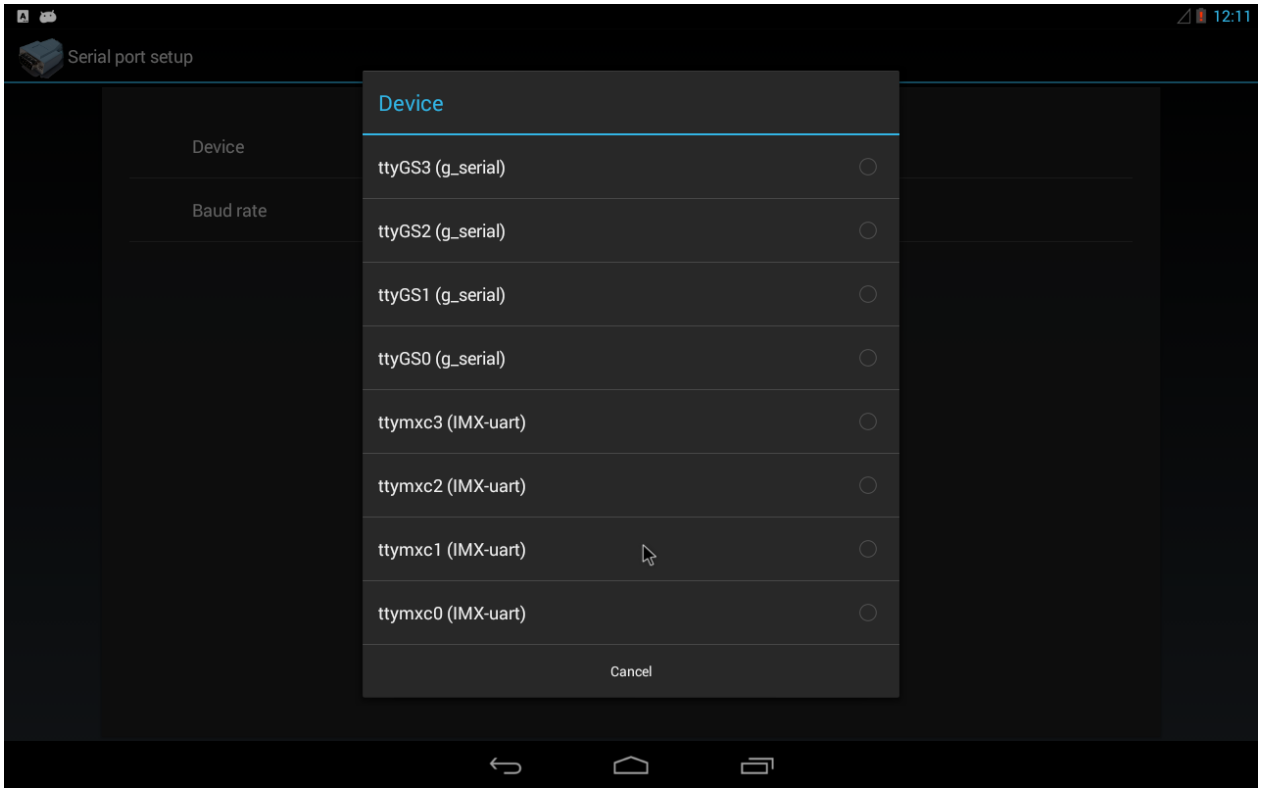



Figure 155: *Serial settings*

- **Push the button with the label “Send 01010101”, you will see something on the Windows 7 PC that looks similar to the figure below.**

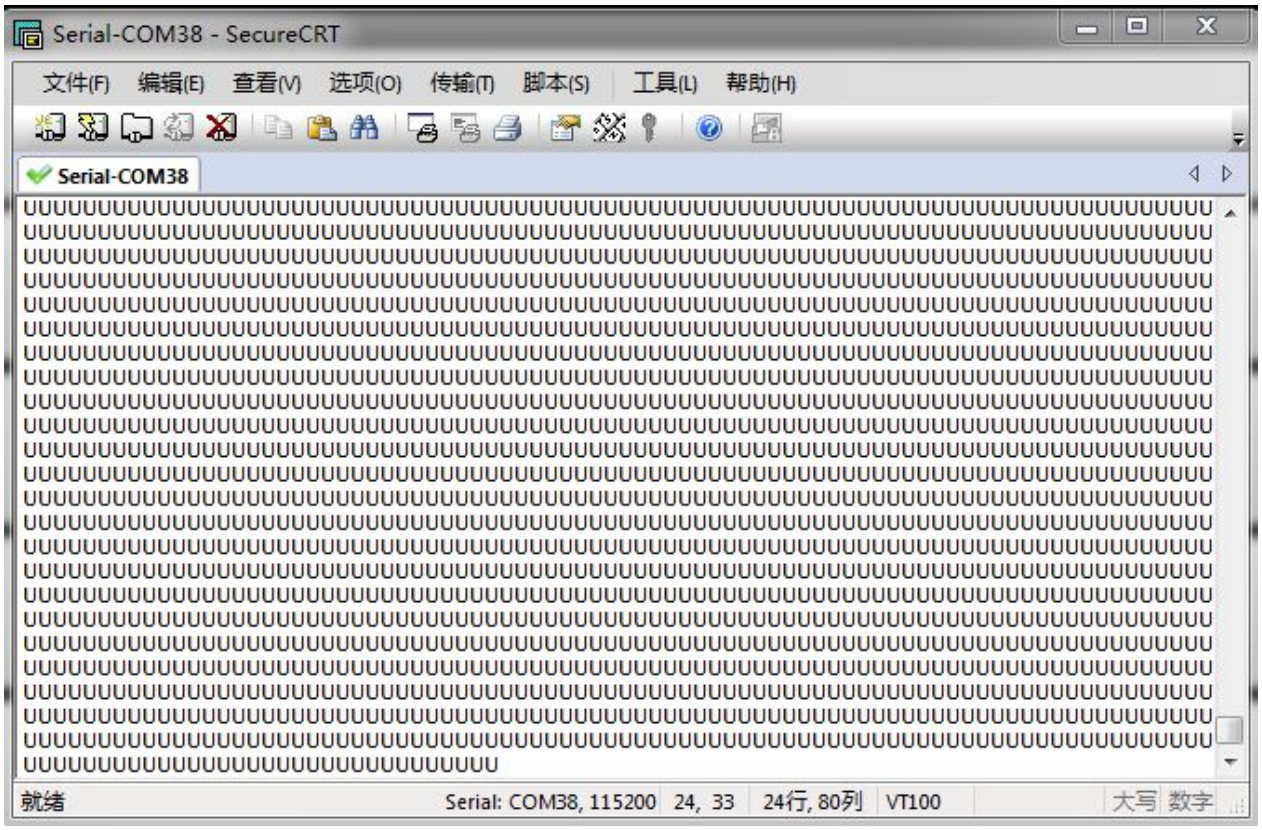



Figure 156: *Serial send test*

- **Push the button with the label “Console”, to send whatever you like as shown on the figure below.**

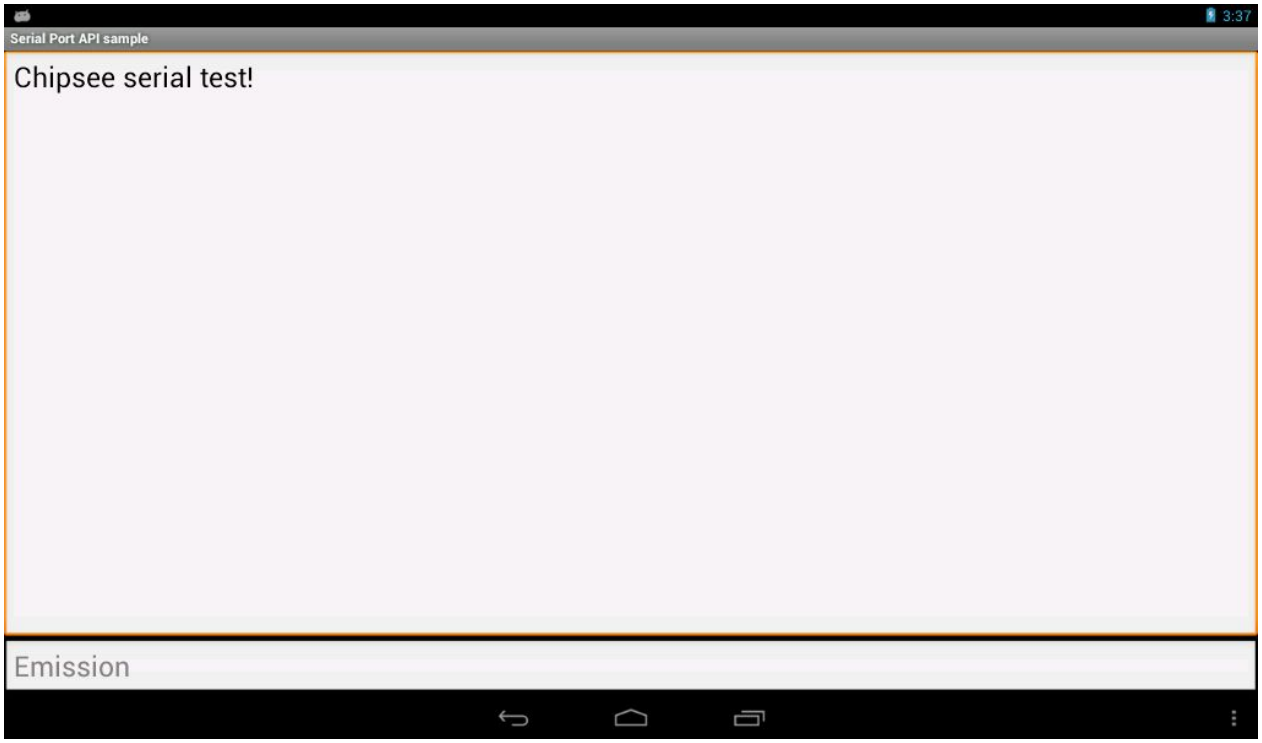



Figure 157: *Serial receive test*

### GPIO

For the **CS12800F101** IPC, there are 8 GPIO ports that you can set as output or input with LOW as 0V; the HIGH as 3.3V.
Please check the **GPIO Connector** section in CS12800F101 to know the position of the GPIO Connector. Refer to the table below for the available GPIO nodes on system.

| PIN Number | GPIO Number |
|---|---|
| 3 | gpio106 |
| 4 | gpio30 |
| 6 | gpio95 |
| 7 | gpio87 |
| 8 | gpio29 |
| 9 | gpio28 |
| 11 | gpio94 |
| 12 | gpio130 |

Table 43 GPIO Nodes on the System

**Example:** We are going to set **gpio106** as output, and let it output HIGH or output LOW.

```
# echo 106 > /sys/class/gpio/export                //export gpio106
# echo out > /sys/class/gpio/gpio106/direction    //set gpio106 output
# echo 1 > /sys/class/gpio/gpio106/value          //set gpio106 output HIGH
# echo 0 > /sys/class/gpio/gpio106/value          //set gpio106 output LOW
```

**Example:** We are going to set **gpio30** as input.

```
# echo 30 > /sys/class/gpio/export                //export gpio30
# echo in > /sys/class/gpio/gpio30/direction      // set gpio30 input
```

**Example:** We are going to set **gpio30** as unexport.

```
# echo 30 > /sys/class/gpio/unexport       //unexport gpio30
```

# Android 4 system debug in Windows

In this section, we will discover how to view the Android 4 system via the serial port and debug the system via USB OTG.
Also, we will discover how to install and uninstall applications via USB OTG.
The following operation is under the Windows 7 x64 environment, similar to other Windows platforms.

## View Android 4 system via the serial port

Install the **SecureCRT** or **Putty** software on a Windows 7 PC to view the Android 4 system via the serial ports.
Follow these steps to view Android 4 system via the serial port:

- Connect COM1 on the industrial PC board to Windows 7 PC.
- **Open the SecureCRT or Putty software on the Windows 7 PC and configure it as shown on the figure below.**

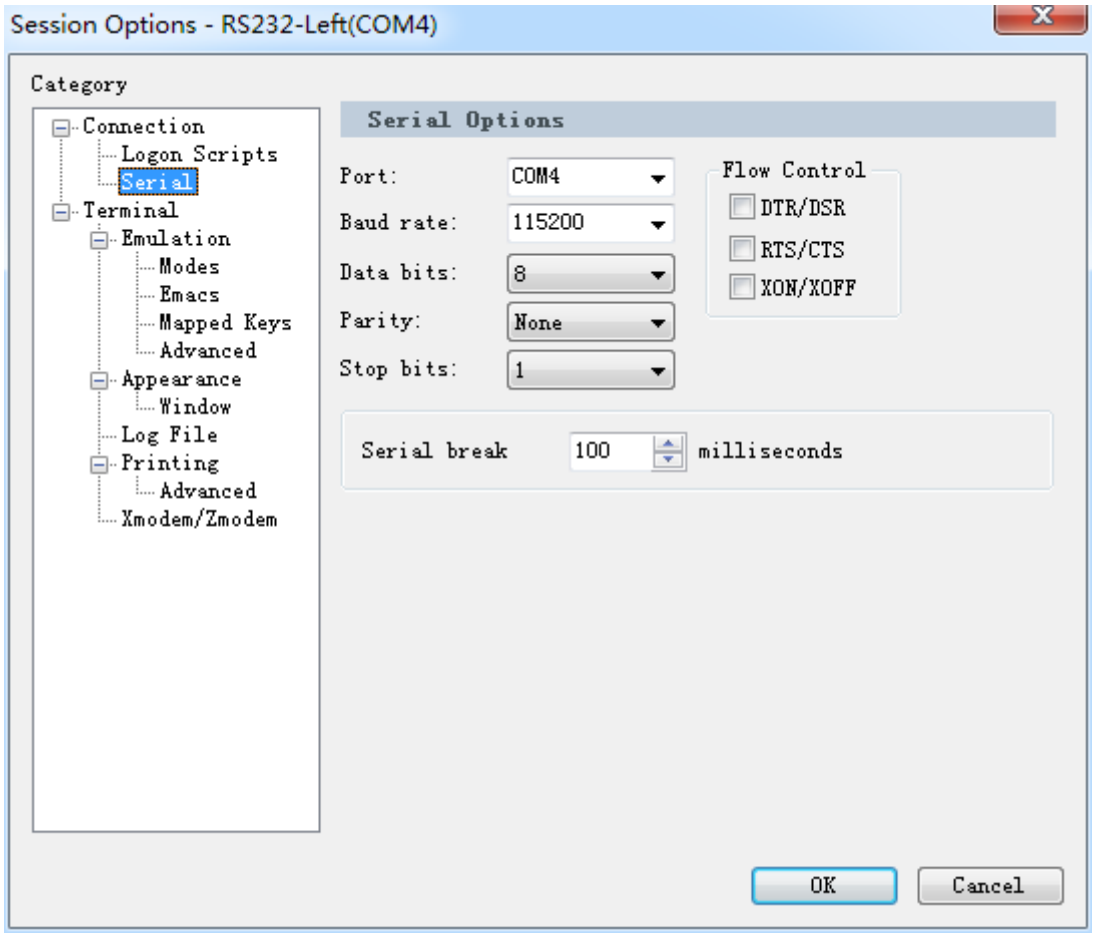



Figure 158: *SecureCRT configuration*

- **Power ON the industrial PC. You will see the serial output information as shown on the figure below.**

```
Hit any key to stop autoboot:  0
mmc3(part 0) is current device

MMC read: dev # 3, block # 2048, count 12288 ... 12288 blocks read: OK
## Booting kernel from Legacy Image at 10800000 ...
   Image Name:   Linux-3.0.35-2508-g54750ff
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    4805952 Bytes =  4.6 MB
   Load Address: 10008000
   Entry Point:  10008000
   Verifying Checksum ... OK
   Loading Kernel Image ... OK
OK

Starting kernel ...

Uncompressing Linux... done, booting the kernel.
Linux version 3.0.35-2508-g54750ff (root@chipsee-build) (gcc version 4.6.2 20110630 (prerelease) (Freesc
CPU: ARMv7 Processor [412fc09a] revision 10 (ARMv7), cr=10c53c7d
CPU: VIPT nonaliasing data cache, VIPT aliasing instruction cache
Machine: Freescale i.MX 6Quad/DualLite/Solo Sabre-SD Board
Ignoring unrecognised tag 0x54410008
Memory policy: ECC disabled, Data cache writealloc
CPU identified as i.MX6Q, unknown revision
PERCPU: Embedded 7 pages/cpu @8c80e000 s5440 r8192 d15040 u32768
Built 1 zonelists in Zone order, mobility grouping on.  Total pages: 462848
Kernel command line: console=ttySAC0,115200 video=mxcfb0:dev=ldb,LDB-WXGA,if=RGB24,bpp=32 ldb=sin0 videc
kernel time=10:10:07,date=Jun 23 2016
PID hash table entries: 4096 (order: 2, 16384 bytes)
Dentry cache hash table entries: 262144 (order: 8, 1048576 bytes)
Inode-cache hash table entries: 131072 (order: 7, 524288 bytes)
Memory: 1568MB 256MB = 1824MB total
Memory: 1836188k/1836188k available, 260964k reserved, 262144K highmem
Virtual kernel memory layout:
    vector  : 0xffff0000 - 0xffff1000   (   4 kB)
    fixmap  : 0xfff00000 - 0xfffe0000   ( 896 kB)
    DMA     : 0xf4600000 - 0xffe00000   ( 184 MB)
    vmalloc : 0xea800000 - 0xf2000000   ( 120 MB)
    lowmem  : 0x80000000 - 0xea000000   (1696 MB)
    pkmap   : 0x7fe00000 - 0x80000000   (   2 MB)
    modules : 0x7f000000 - 0x7fe00000   (  14 MB)
      .init : 0x80008000 - 0x8003d000   ( 212 kB)
      .text : 0x8003d000 - 0x80c44bf0   (12319 kB)
      .data : 0x80c46000 - 0x80cccbe0   ( 539 kB)
       .bss : 0x80cccc04 - 0x80d252ec   ( 354 kB)
SLUB: Genslabs=13, HWalign=32, Order=0-3, MinObjects=0, CPUs=4, Nodes=1
Preemptible hierarchical RCU implementation
```

Figure 159: *Serial output information*

- You can communicate with the system when system boot is complete.

## Adb connect via USB OTG

Please refer to the ADB Test chapter to learn how to set the ADB, how to install an app via ADB, and how to debug with ADB.
You can use the following command to log in to the board and communicate with it.

```
> adb shell
```

```
C:\Users\Chipsee>adb shell
shell@sabresd_6dq:/ $ su
su
root@sabresd_6dq:/ # ls
ls
acct
cache
charger
config
d
data
default.prop
dev
device
etc
file_contexts
fstab.freescale
init
init.chipsee.sh
init.environ.rc
init.freescale.i.MX6DL.rc
init.freescale.i.MX6Q.rc
init.freescale.i.MX6QP.rc
init.freescale.rc
init.freescale.usb.rc
init.rc
init.recovery.imx6.rc
init.trace.rc
init.usb.configfs.rc
init.usb.rc
init.zygote32.rc
```

Figure 160: *ADB Shell*

### Use ADB command to install user APP

Use the `adb` command to install an Android App: for example SogouInput.apk. If there is a **SUCCESS** message, as shown on the figure below, then the app installation was successful.

```
> adb install SogouInput.apk
```

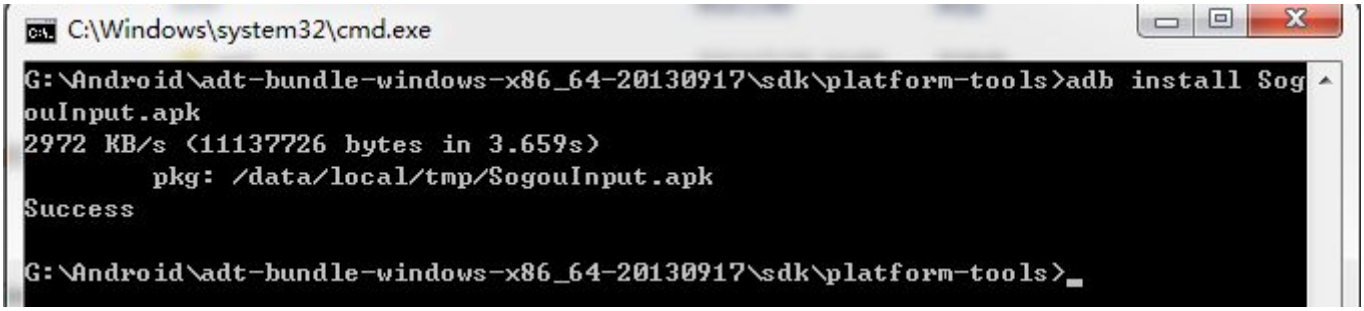



Figure 161: Install App

### Use ADB command to uninstall user APP

Use `adb` command to uninstall an Android app: for example AngryBirds.apk. Follow these commands to uninstall an app.

```
> adb shell pm list packages
> adb uninstall com.rovio.angrybirds
```

- The *pm list* command gets the full name of the app, as shown on the figure below.

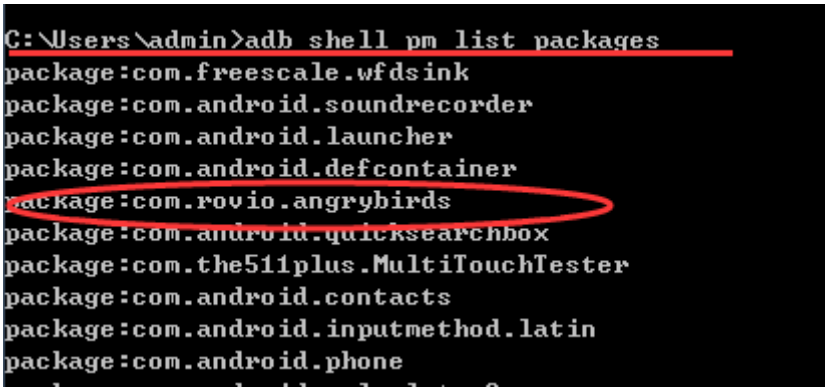



Figure 162: *Uninstall user app*

- The `uninstall` command uninstalls the app from the Android system.
- Delete the apk file for the app by using these commands:

```
> adb shell
# cd /system/app/
# ls
# rm Browser.apk
```

### Use ADB command to uninstall default APP

Use `adb` command to uninstall an Android app: for example *Email.apk*. Follow these commands to uninstall a default app.

```
> adb shell
```

```
$ su
su
```

```
# cd /system/app
cd /system/app
```

```
# rm Email.apk
```

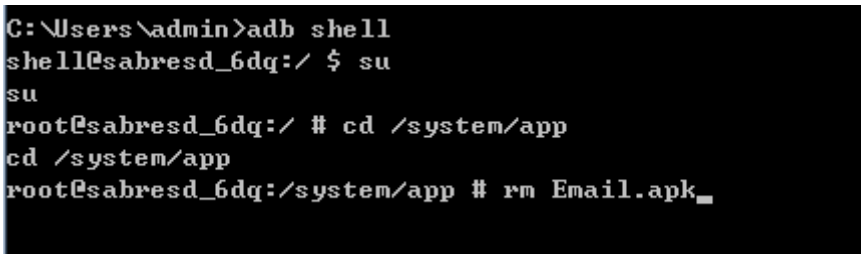



Figure 163: *Uninstall default app*

### Use ADB command to uninstall default APP

Use `adb` command to transport files between the industrial PC and Windows 7 PC.

- **Transfer file from the industrial PC to Windows 7 PC using** *adb pull* **command.**

```
> adb pull <pathTo_file_on_board> <pathTo_store_file_on_PC>
```

- **Transfer file from the Windows 7 PC to the industrial PC using** *adb push* **command.**

```
> adb push <pathTo_file_on_PC> <pathTo_store_file_on_board>
```

For example, copy `<ADT>\sdk\platform-tools\chipsee.txt` from Windows PC to IPC:

```
> adb push chipsee.txt /chipsee.txt
```

Copy `/testFile.txt` from IPC to Windows PC:

```
> adb pull /testFile.txt testFile.txt
```

## Adb connect via internet

1. **The Ethernet port on the industrial PC and the host machine (Windows 7 PC) should connect to the network. Check Ethernet configuration for the industrial PC using the command below.**

```
# netcfg
lo       UP            127.0.0.1/8   0x00000049 00:00:00:00:00:00
can0     DOWN          0.0.0.0/0   0x00000080 00:00:00:00:00:00
eth0     UP            192.168.6.176/24  0x00001043 1e:ed:19:27:1a:b3
```

2. **If the industrial PC's Ethernet is not configured, configure the Ethernet using the `ifconfig` / `netcfg` command as shown below.**

```
# netcfg eth0 dhcp
```

3. **Configure the ADB Daemon to use an Ethernet connection using the `setprop` command, as shown below.**

```
# setprop service.adb.tcp.port 5555
```

4. **If the network is configured successfully using the steps above, then Restart service `adbd` on the Windows 7 PC.**

```
# stop adbd
# start adbd
```

5. **On the host machine (Windows 7 PC) use the following commands to establish the `adb` connection.**

```
$ adb kill-server
$ adb start-server
$ adb connect :5555
```

6. **Verify the device connectivity, by executing the following commands. If connected, find the device name listed as``IPADDRESS:PORT``.**

```
$ adb devices
List of devices attached
192.168.6.176:5555      device
```

7. **An example of using the `adb` command to install software for Android. Make sure the `"**".apk` file is at the current folder, and export the adb path.**

    • Use the argument `–s` to assign the device to use over the internet.

```
$ adb –s 192.168.1.117:5555 install "**".apk
```

# Android App Development

In this section, we will introduce the development of an Android app with Eclipse on Windows. We assume that the USB is OTG model and the driver is already installed. (See Adb connect via USB OTG)

## Preparation

1. Download and install Eclipse IDE.
2. **Go to the `/eclipse` folder, start eclipse.exe.**

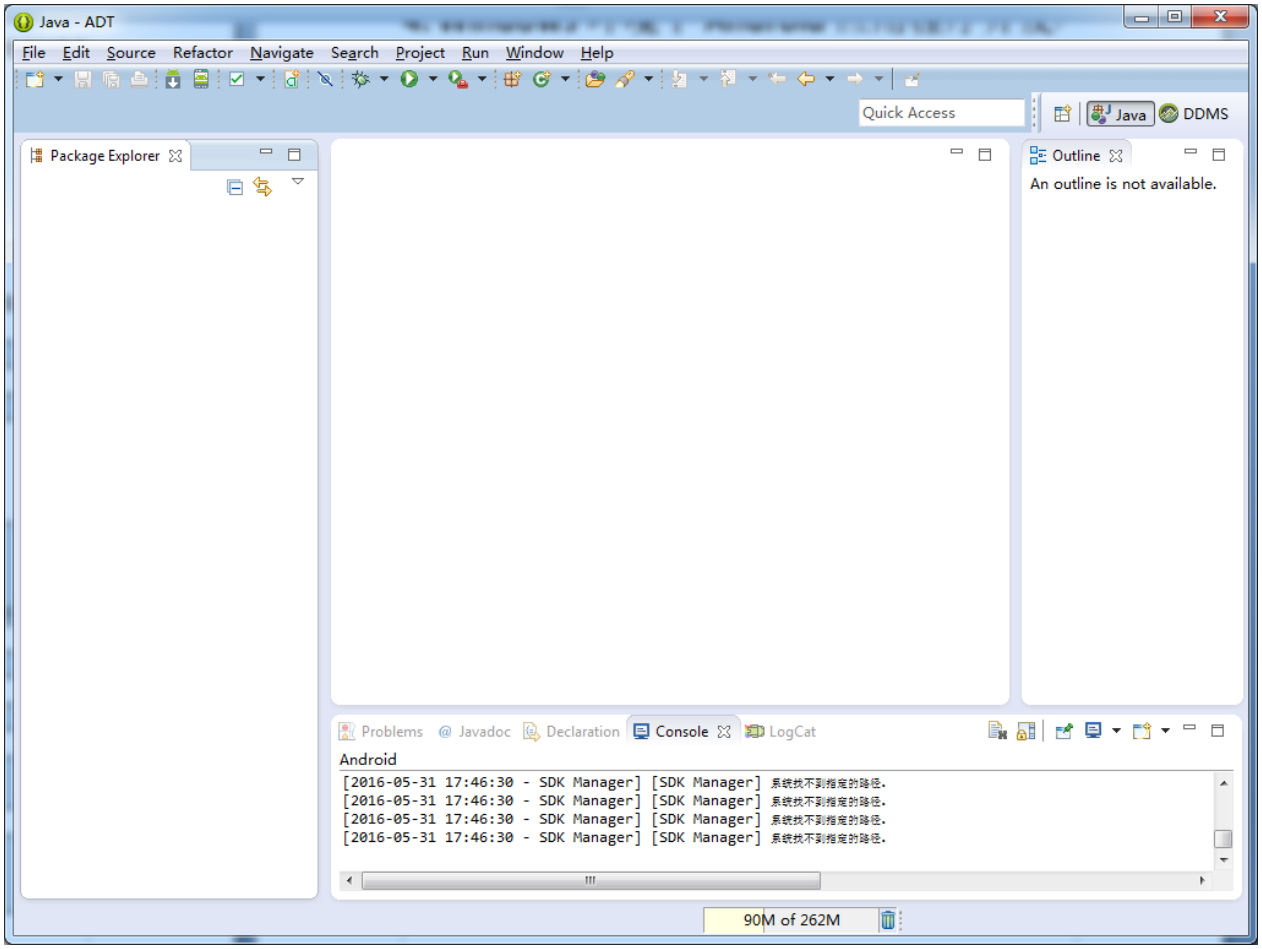



Figure 164: *Start eclipse*

3. **Click Windows–>Android SDK Manager to open SDK Manager.**

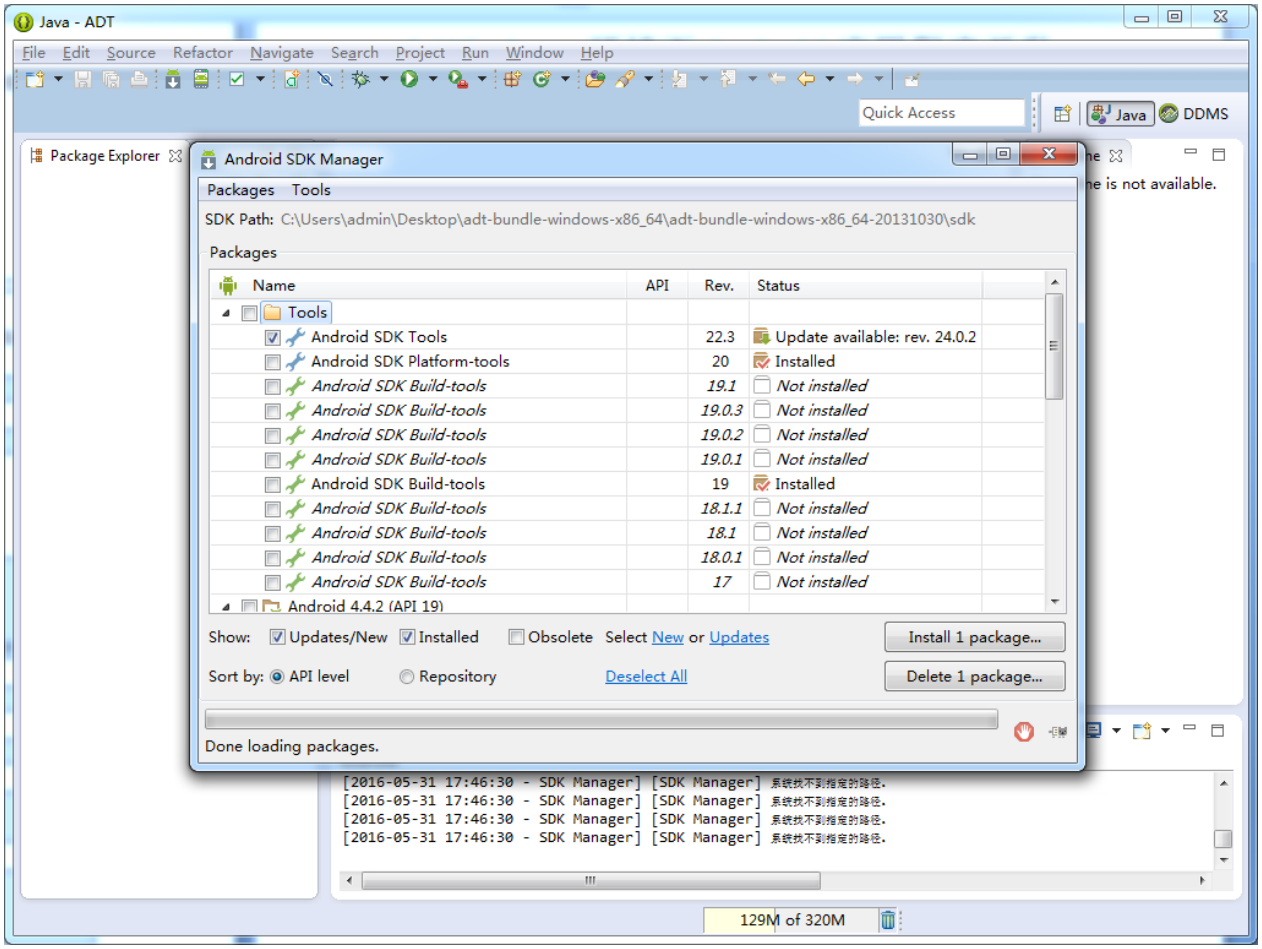



Figure 165: *Android SDK Manager*

4. **Click Tools–>Options, check the `Force` box and click close.**

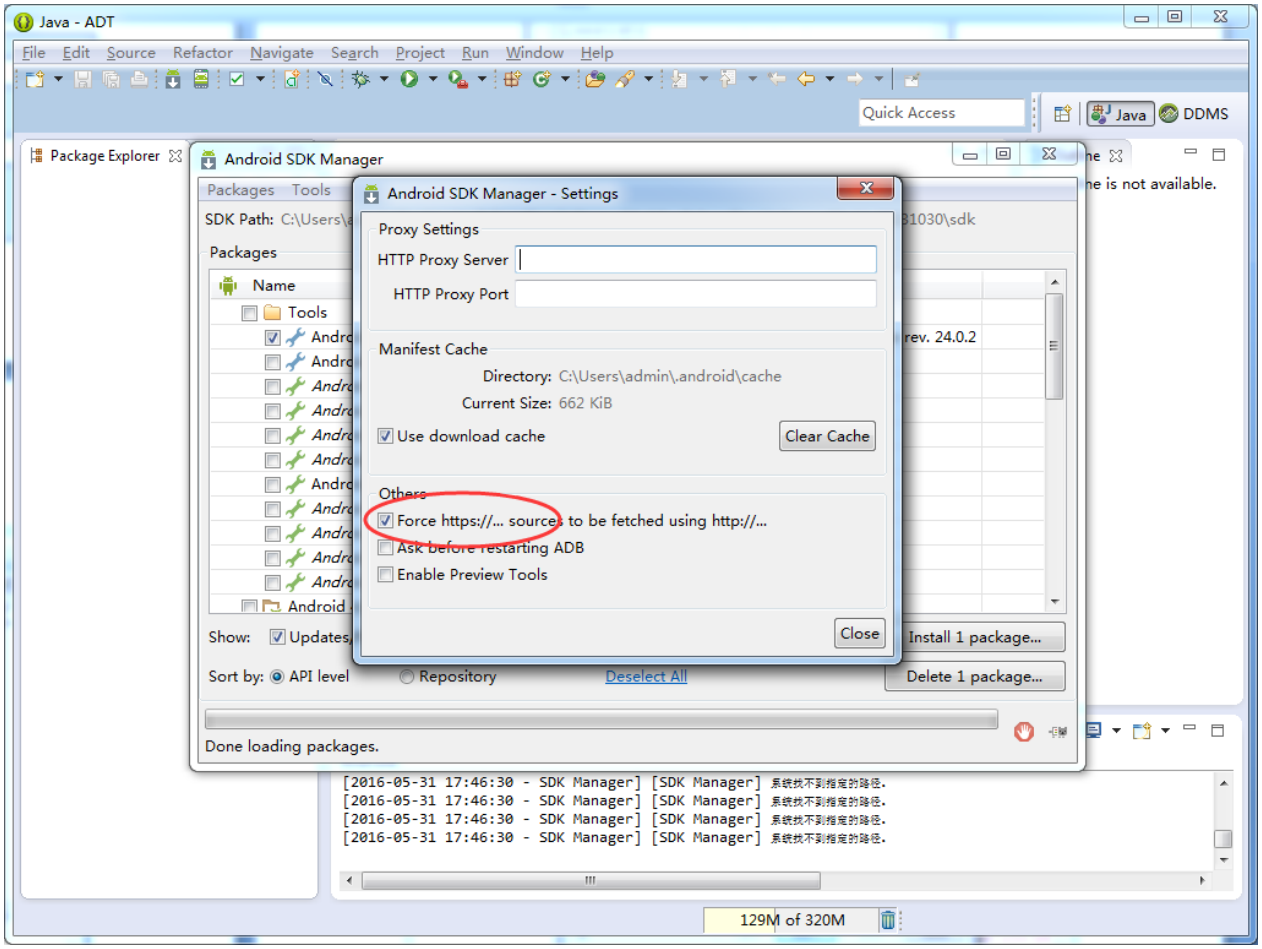



Figure 166: *Android SDK Manager Settings*

5. **Choose the API, such as Android4.3(API 18), then click the Install packages button to start the download and installation of API packages.**

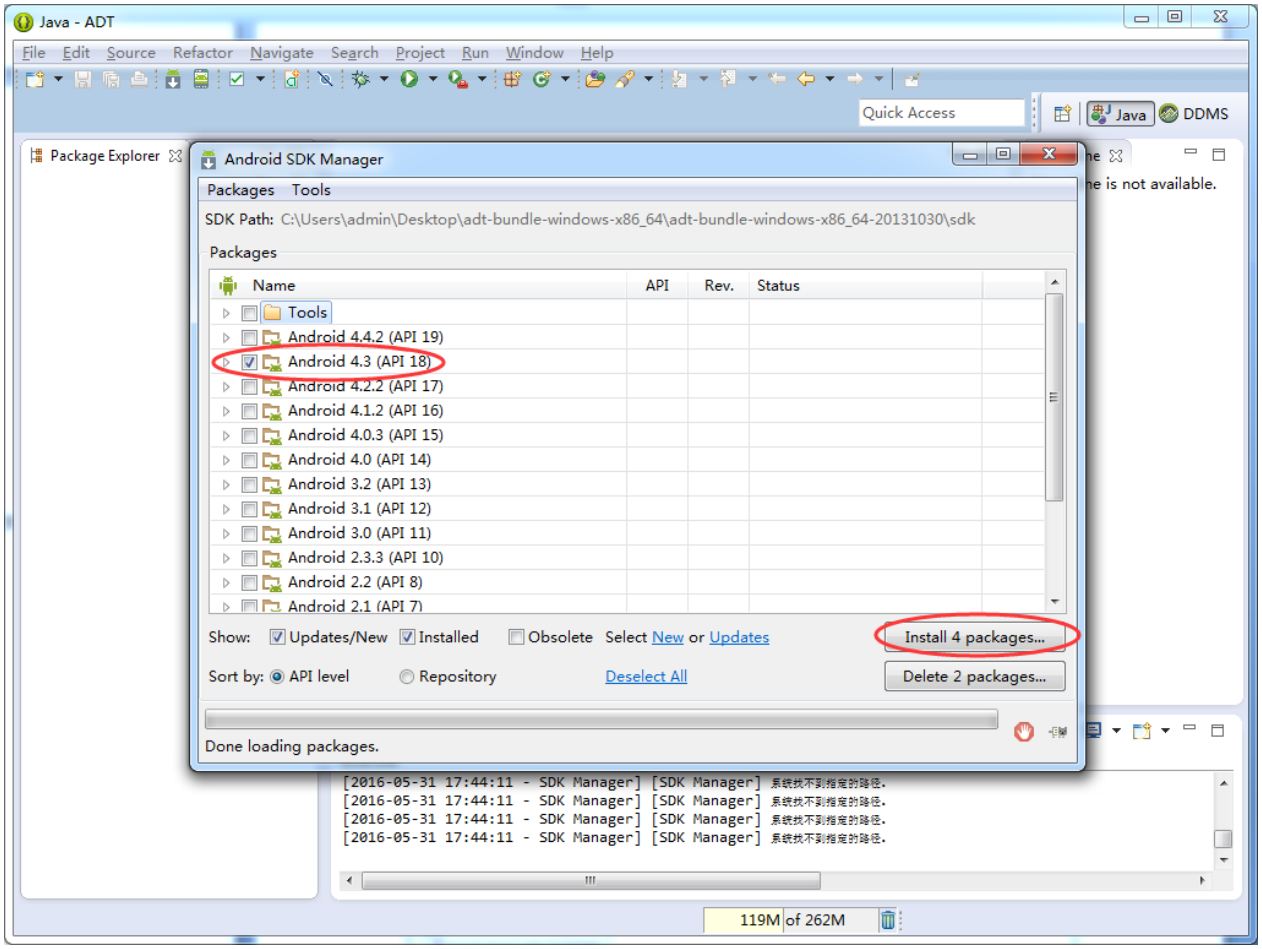



Figure 167: *Install API packages*

6. Downloading and installing the packages will take some time. When the process completes, close the Android SDK Manager.

## Example — Develop a `HelloWorld` Program

1. **Click File–>New–>Android Application Project**

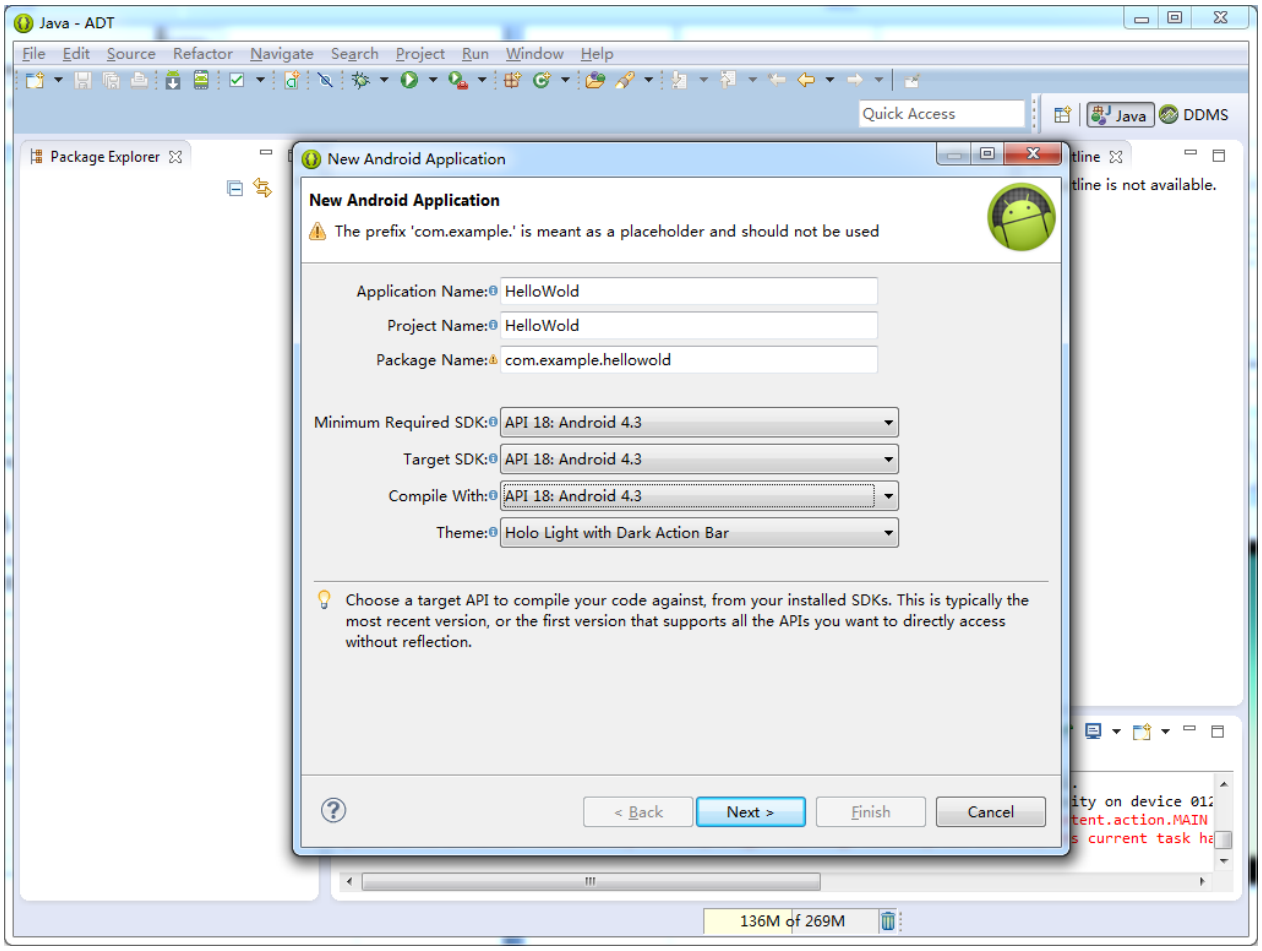



Figure 168: *New Application*

2. **Click on the Next button until the app project is created. Connect the industrial PC to Windows 7 PC via the USB cable (A-A). If the connection is successful, you will see the device in the DDMS window (Windows–>Open Perspective–>Other–>DDMS)**

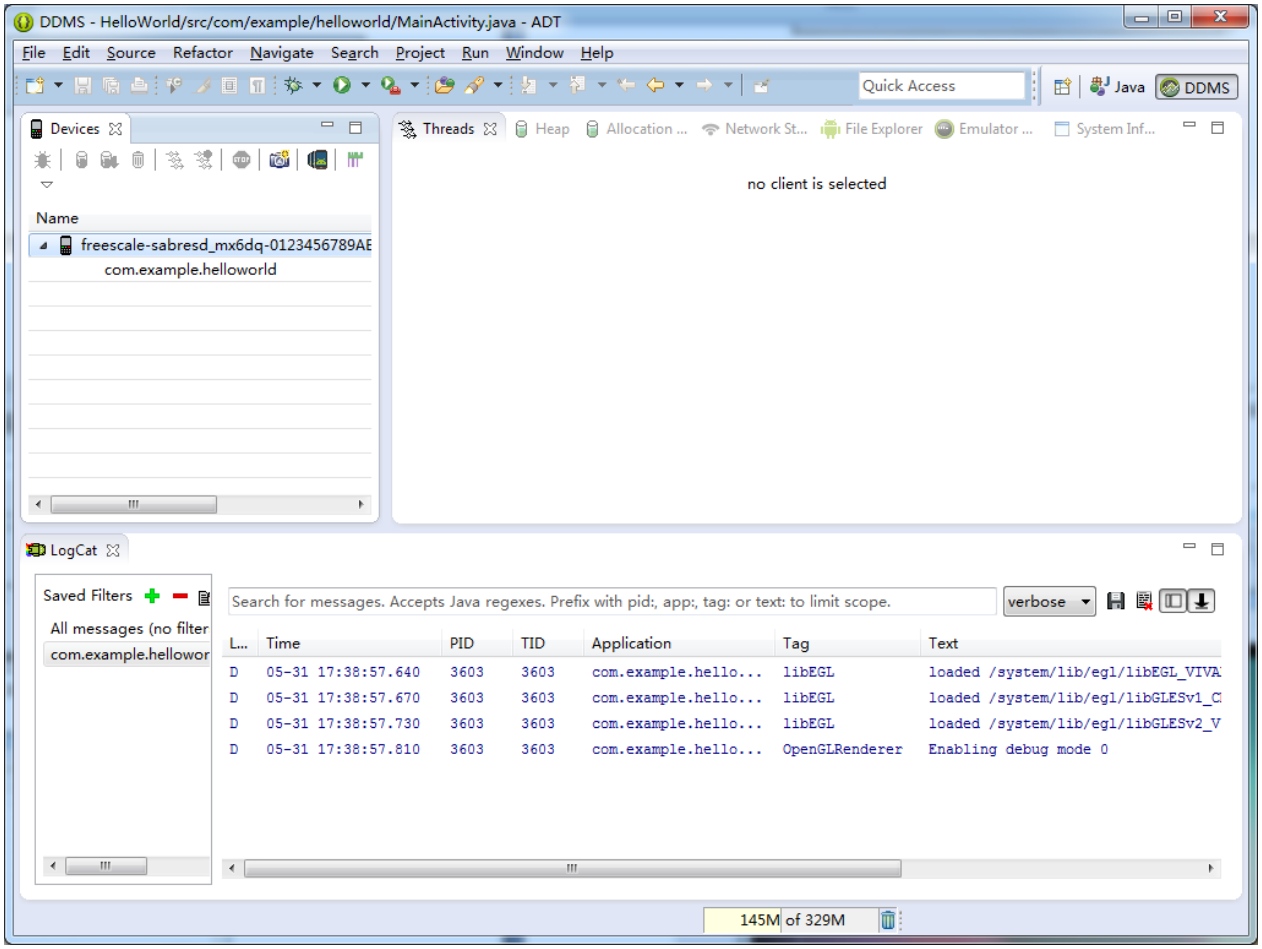



Figure 169: *DDMS*

## 3. **You can capture the desktop of Android**

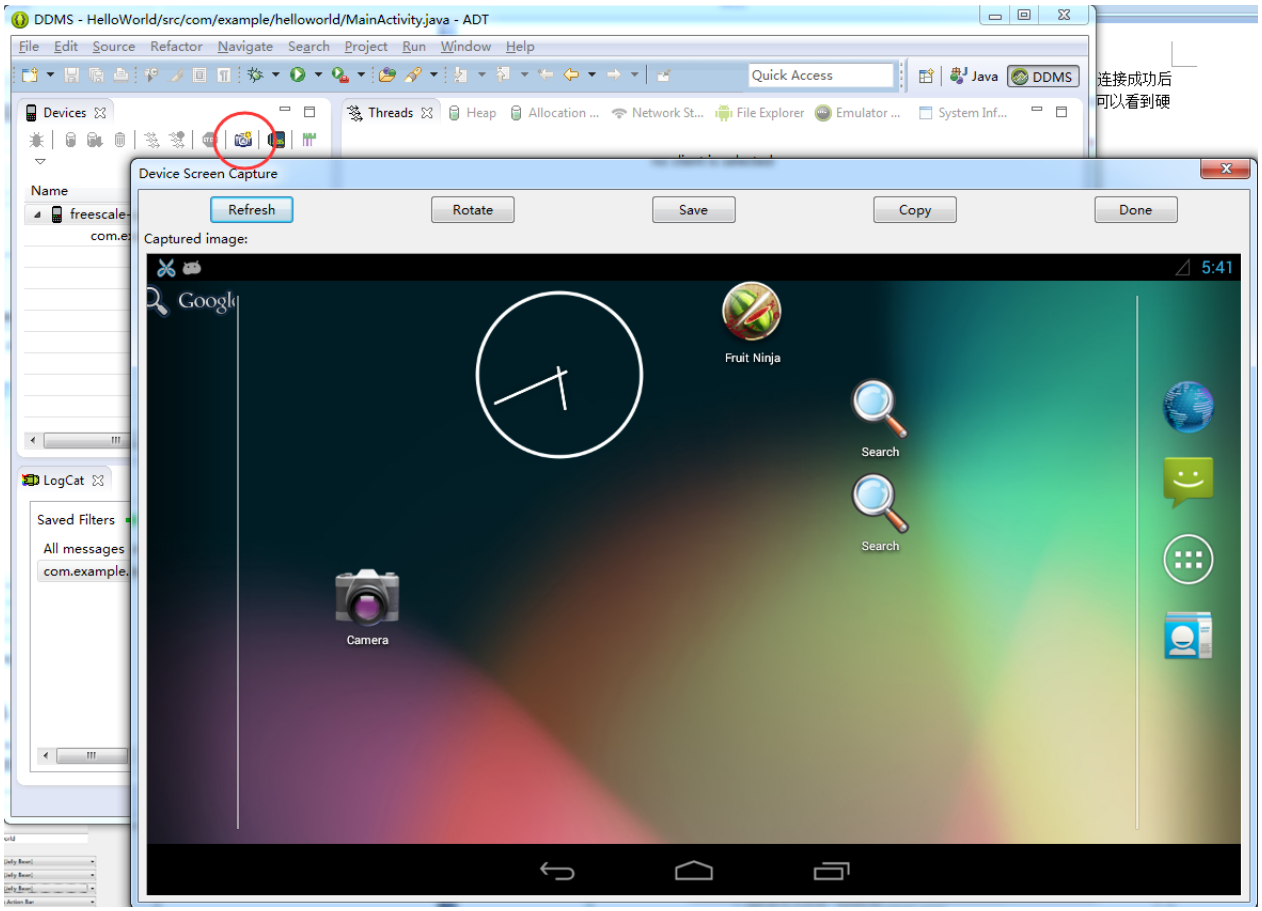


Figure 170: *Capture Android Desktop*

4. **Click run, and choose the device**

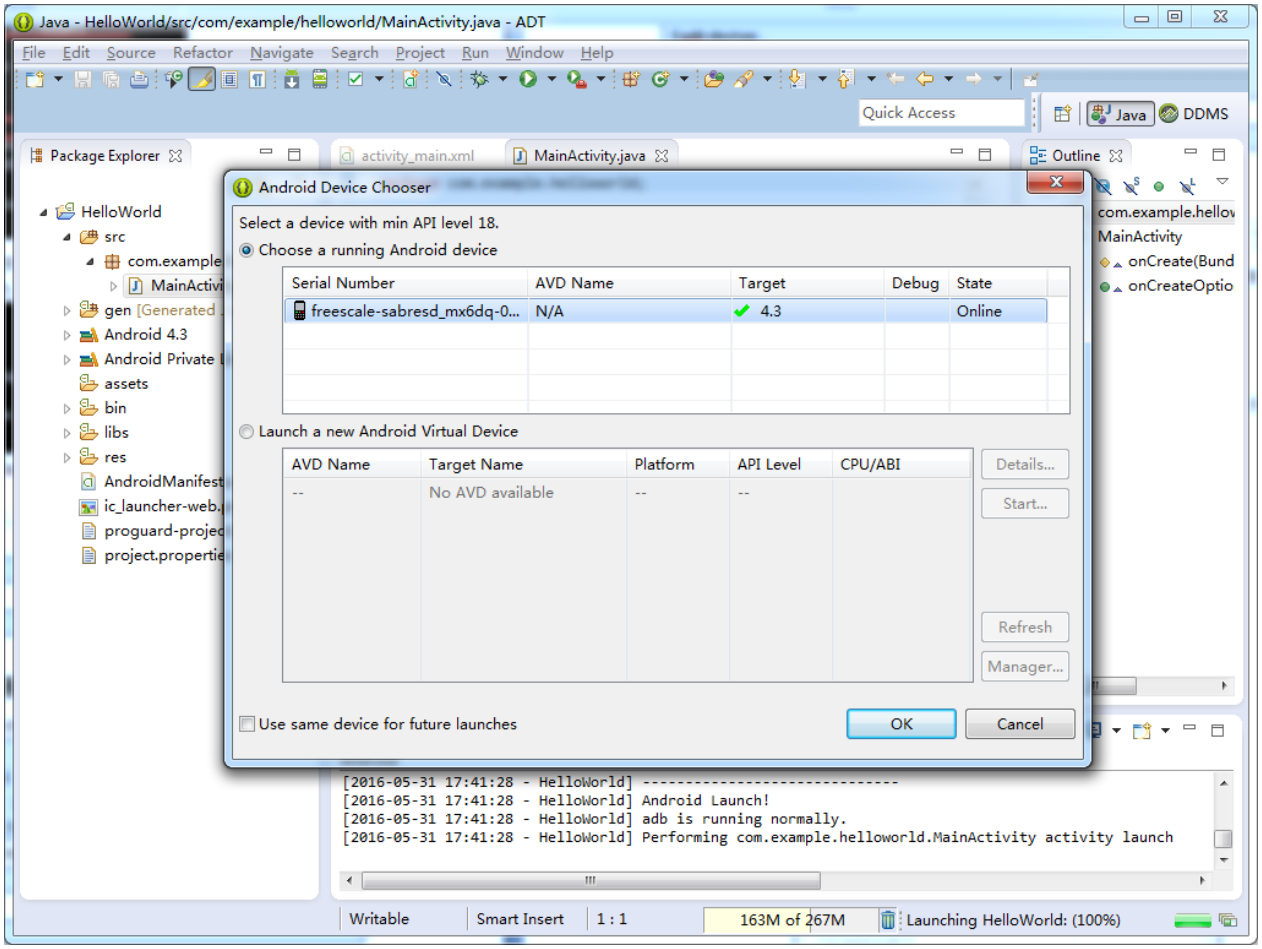



Figure 171: *Run HelloWorld Program*

5. **Result**

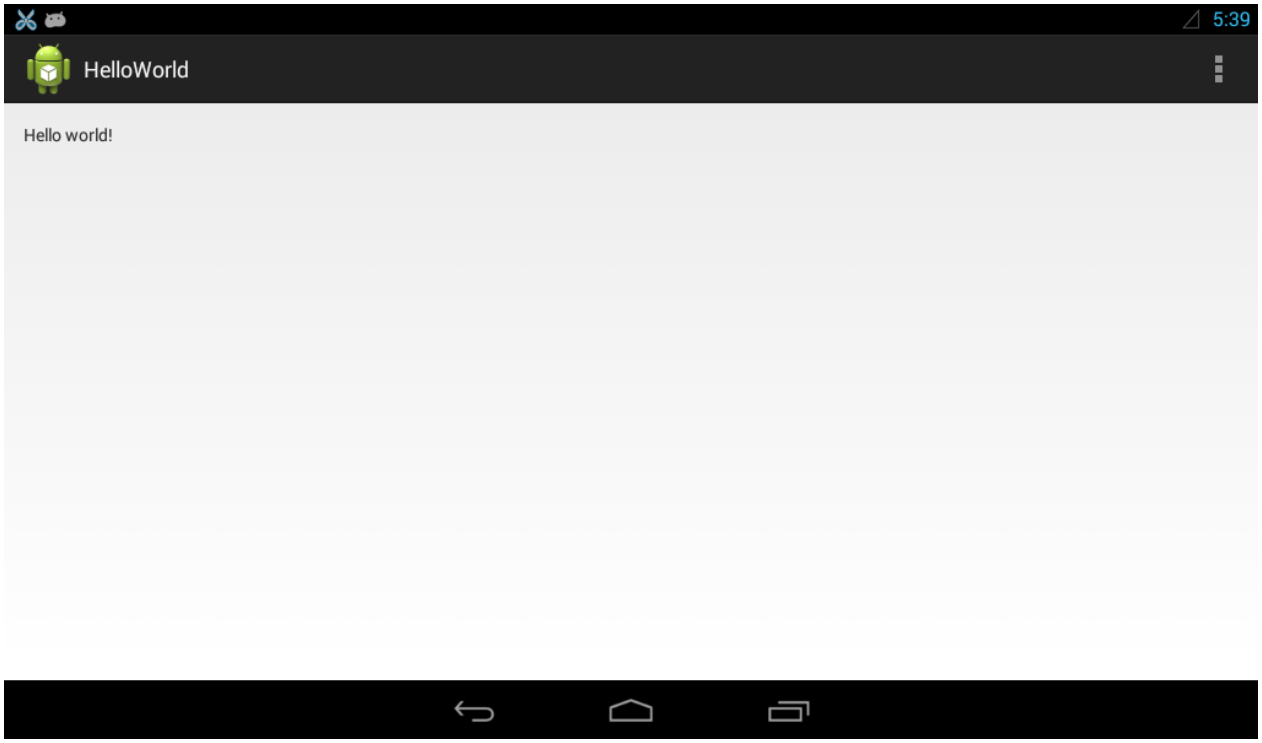


Figure 172: *HelloWorld Program*

**Note**

If the USB is not configured as an OTG model, you can copy and install the file `HelloWorld.apk` from the project folder `HelloWorld/bin/`, or install the `HelloWorld.apk` via the internet (See Adb connect via internet).

For more resources about Android development, visit these links:

> https://developer.android.com/guide/index.html https://developer.android.com/develop/index.html http://developer.android.com/support.html http://blog.apptopia.com/android-development-forums/ http://androidforums.com/application-development/

# Disclaimer

**This document is provided strictly for informational purposes. Its contents are subject to change without notice. Chipsee assumes no responsibility for any errors that may occur in this document. Furthermore, Chipsee reserves the right to alter the hardware, software, and/or specifications set forth herein at any time without prior notice and undertakes no obligation to update the information contained in this document.**

**While every effort has been made to ensure the accuracy of the information contained herein, this document is not guaranteed to be error-free. Further, it does not offer any warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document.**

**Despite our best efforts to maintain the accuracy of the information in this document, we assume no responsibility for errors or omissions, nor for damages resulting from the use of the information herein. Please note that Chipsee products are not authorized for use as critical components in life support devices or systems.**

# Technical Support

If you encounter any difficulties or have questions related to this document, we encourage you to refer to our other documentation for potential solutions. If you cannot find the solution you're looking for, feel free to contact us. Please email Chipsee Technical Support at **support@chipsee.com**, providing all relevant information. We value your queries and suggestions and are committed to providing you with the assistance you require.