Chipsee

Industrial PC

# Android 6.0 OS on iMX6Q User Manual

For iMX6Q Products

Content can change at anytime, check documentation website for latest information.
**www.chipsee.com**

# Contents


Android 6.0 OS 4

1. Preparation 6

1.1. Hardware Requirements 6

1.2. Software Requirements 6

2. Getting Started and Tests 7

2.1. Boot Switch Configuration 7

2.2. Prepare Manufacturing Tool and Image 7

2.3. Downloading Images 8

2.3.1. Downloading Images by using MFGTool 8

2.3.1.1. Configuring MFGTool 8

2.3.1.2. Copy Image To Android Directory 9

2.3.1.3. Using MFGTool 9

2.3.2. Downloading Images by using the TF card 13

2.4. Booting Android OS And Test(Using 7inch as example) 15

2.4.1. SD Test 16

2.4.2. USB Flash Disk Test 17

2.4.3. Network Test 18

2.4.4. Sound Card Test 19

2.4.5. Video Test 20

2.4.6. HDMI Test 21

2.4.7. WIFI Test 21

2.4.8. ADB Test 22

2.4.9. Touch Screen Test 26

2.4.10. Serial Test 27

2.4.11. GPIO 30

2.4.11.1. Modify Logo 31

3. Android 6.0 system debug in Windows 34

3.1. View Android 6.0 system via the serial port 34

3.2. Adb connect via USB OTG 36

3.2.1. Use ADB command to install user APP 37

3.2.2. Use ADB command to uninstall user APP 38
3.2.3. Use ADB command to uninstall default APP 39
3.2.4. Use ADB command to uninstall default APP 40
3.3. Adb connect via internet 41
4. Android App Development 43
4.1. Example — Develop a `HelloWorld` Program 43
5. Disclaimer 48
6. Technical Support 48

# Android 6.0 OS

**Android 6.0 OS User Manual**

This manual is used to provide users with a fast guide of Chipsee Industrial Computers (Abbreviated as IPC) about Android 6.0 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 6.0 OS via serial, USB OTG and Internet.

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

**SUPPORTED BOARDS:**

> *CS10600F070 CS10768F097 CS12800F101 CS10768F121 CS10768F121-U CS10768F150 CS12102F170 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
- CS10768F097
- CS12800F101
- CS10768F121
- CS10768F121-U
- CS10768F150

- CS12102F170
- CS14900F190
- CS19108F215

## System Features

| Feature | Comment |
| --- | --- |
| System | Android 6.0 |

# 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: CS10768F121, CS10768F121-U, CS10768F150, CS12102F170, CS14900F190, and CS19108F215 requires a 15V to 36V power adapter.
- These products: CS10768F097 and CS12800F101 product needs a 12V to 36V power adapter.
- The CS10600F070 product needs a 6V 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
- TF Card (at least 4GB) and card reader

## Software Requirements

- Android 6.0 OS Prebuilt Files Package (from the link above)
- `Android Studio 2.3.3 for Windows`
- Android USB driver (for Windows)
- MFGTools_Kernel3.14.52

# Getting Started and Tests

> **Note**
>
> Throughout this section, the user can use both the pre-built Android 6.0 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 three modes, namely

- TF Card
- eMMC Boot
- Download

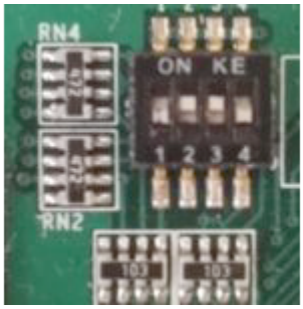



Figure 173: Boot Mode Setup

| SW Mode | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| **TF Card** | 1 | 0 | 0 | 0 |
| **eMMC** | 1 | 1 | 0 | 1 |
| **Download** | 0 | 1 | 1 | 0 |

Table 44 Boot Configuration Selection

## Prepare Manufacturing Tool and Image

The manufacturing tool, referred to as MFGTools, 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** | emmc-flash/emmc/boot-imx6q.img |
| **U-boot Image** | emmc-flash/emmc/u-boot-imx6q.imx |

| MFGTool | Windows download tool |
|---|---|
| **Recovery Image** | emmc-flash/emmc/recovery-imx6q.img |
| **Android File System** | emmc-flash/emmc/system.img |
| **Android Logo** | emmc-flash/emmc/xxx.bmp |
| **Industrial Computer** | One |
| **USB OTG Cable** | One |
| **12V-2A power adapter** | One |

## Downloading Images

Chipsee IPC supports booting from an integrated eMMC or an external TF Card (also known as the micro SD card). Booting from the external TF Card allows flashing the system OS.

### Downloading Images by using MFGTool

Chipsee IPC supports booting from an integrated eMMC.

CONFIGURING MFGTOOL

To configure MFGTool, follow these steps:

- Unzip `Mfgtools_Kernel3.14.52_V1.0.rar` file.
- Open the extracted folder `Mfgtools_Kernel3.14.52_V1.0` and edit `cfg.ini` file.

Figure 174: *Extracted folder content*

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

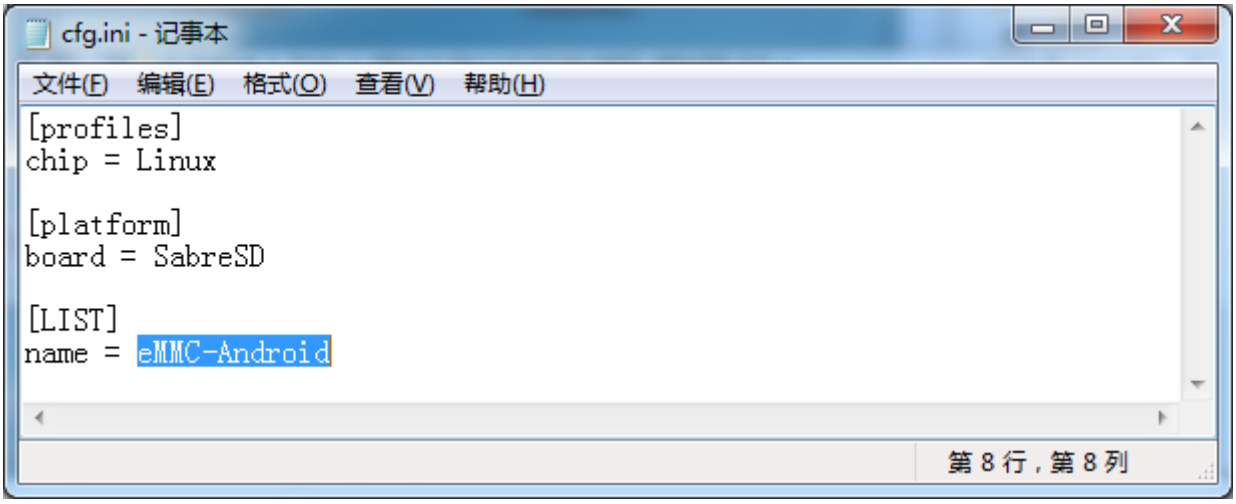



Figure 175: *Cfg.ini file*

### COPY IMAGE TO ANDROID DIRECTORY

Follow these steps to copy image to Android directory:

- Unzip `prebuilt-imxv1-csXXXXXfXXXvX-android6-emmc-YYYYMMDD.tar.gz` file. The extracted folder will contain these files: `boot-imx6q.img`, `recovery-imx6q.img,`, `system.img`, and `u-boot-imx6q.imx`. The logo file, `android6_xxx.bmp`, is located in the *emmc-flash/emmc* directory.
- Copy the files listed above from the extracted folder to *Mfgtools_Kernel3.14.52_V1.0\Profiles\Linux\OS Firmware\files\android* directory.

Figure 176: *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.**

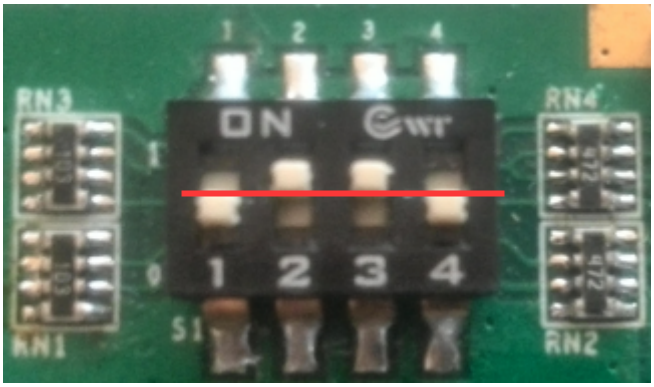



Figure 177: *Boot Switch Config*

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.**

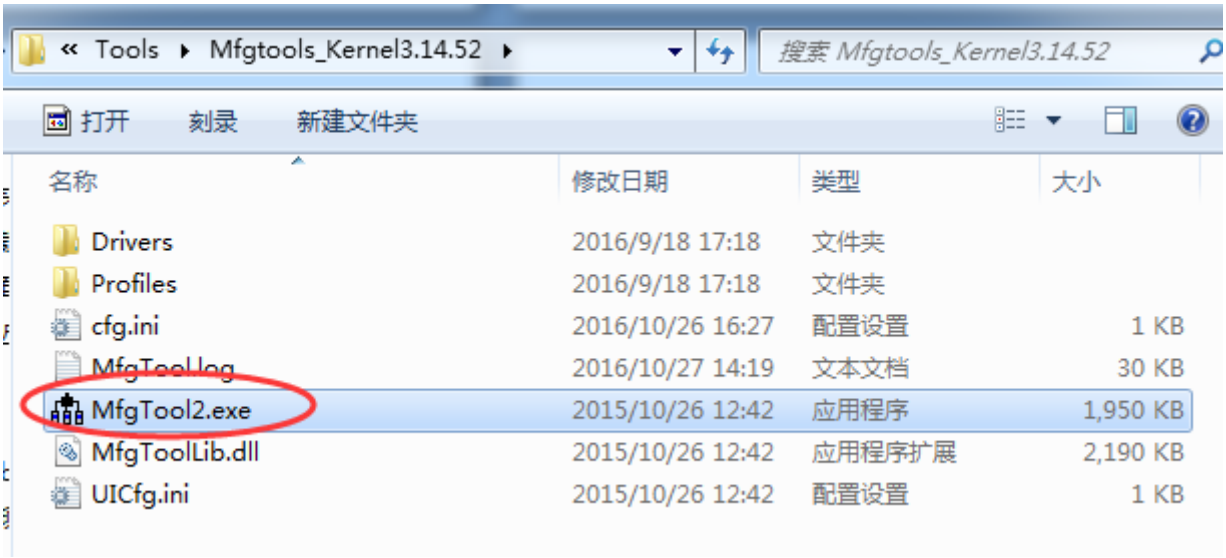



Figure 178: *Run* **MfgTools2.exe** *file*

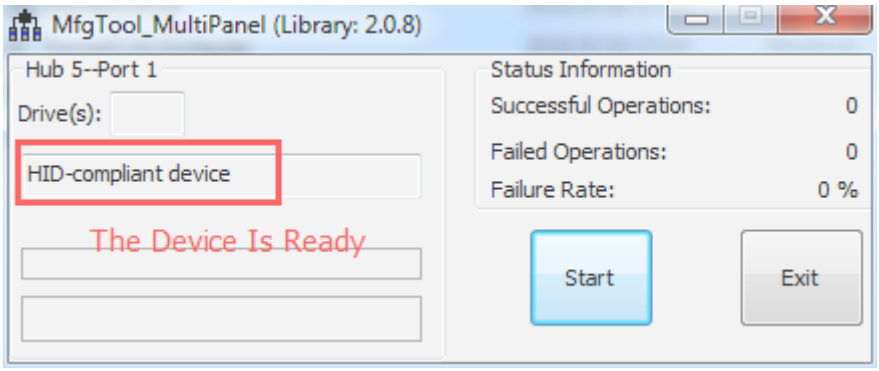



Figure 179: *Prepare to start*

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

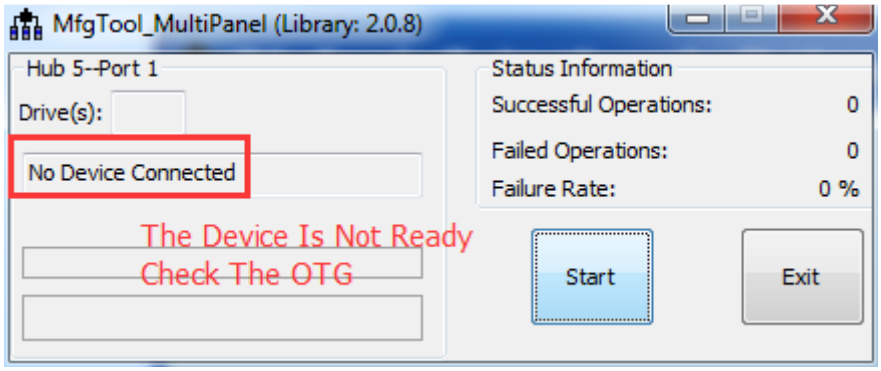



Figure 180: *The USB-OTG cable is not connected correctly.*

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

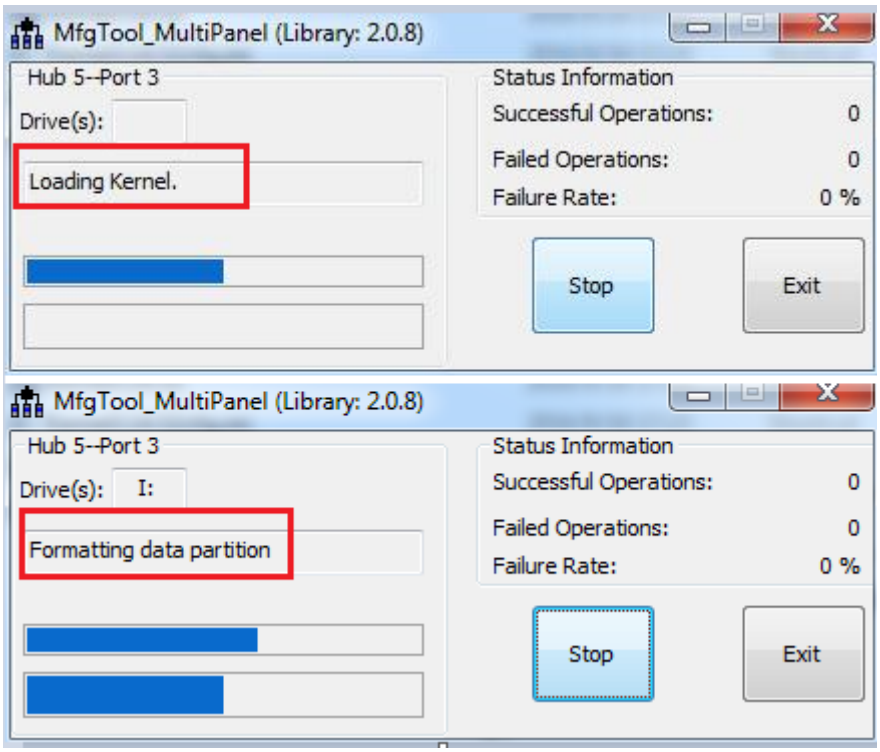
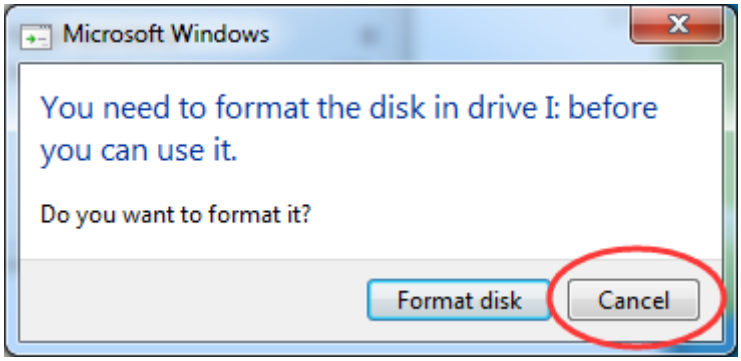



Figure 181: *Loading Kernel and Formatting rootfs partition*

**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.**

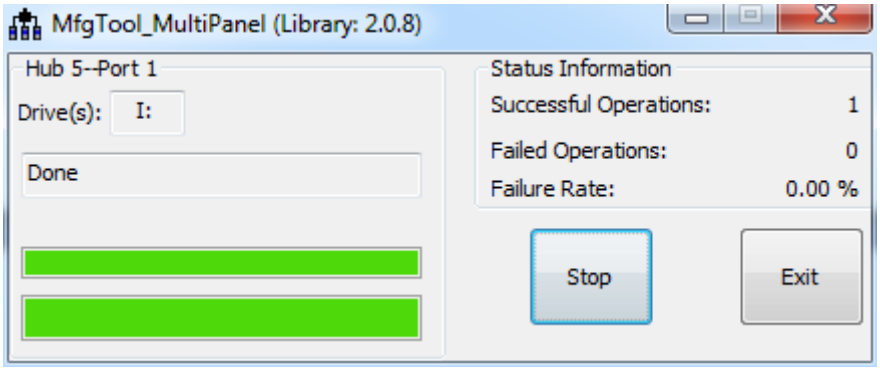



Figure 182: *Download Image is finished*

### Downloading Images by using the TF card

Follow the steps below to download images onto the eMMC by using the TF Card:

1. Copy the Prebuilt Files Package to a Linux environment (such as Ubuntu 14.04).
2. Insert the SD card into your computer. If you are using virtual machines, please ensure the SD card is mounted to the Linux operating system.
3. **Confirm the SD card mount point, `/dev/sdX` (e.g., `/dev/sdc` or `/dev/sdb`, be sure to use the right one). In a Linux system, you can use the command below to find out what *X* is.**

   ```
   $ sudo fdisk –l
   ```

4. Copy the `prebuilt-imxv1-csXXXXXfXXXvX-android6-emmc-YYYYMMDD.tar.gz` to somewhere(such as $HOME) on the Ubuntu PC.
5. **Extract the `prebuilt-imxv1-csXXXXXfXXXvX-android6-emmc-YYYYMMDD.tar.gz`**

   ```
   $ tar -xzvf prebuilt-imxv1-csXXXXXfXXXvX-android6-emmc-YYYYMMDD.tar.gz
   ```

6. **Go to the folder**

   ```
   $ cd prebuilt-imxv1-csXXXXXfXXXvX-android6-emmc-YYYYMMDD
   ```

7. **Use the following command to flash the Android 6.0 OS to the SD card**

   ```
   $ sudo ./mksdcard.sh --device /dev/sd<?>
   ```

   > **Note**
   >
   > • sd<?> means the SD card mount point, (e.g., `/dev/sdc` or `/dev/sdb`) in Ubuntu system.
   >
   > • The recommended SD card should be Sandisk Class4 level SD card or above.

8. The bootable SD Card is now ready. Power OFF the industrial PC and insert the SD Card.
9. Set the switch S1 to TF card boot mode. (refer to Boot Switch Configuration above)
10. Connect the industrial PC to PC via COM1. Power ON the IPC.
11. After 20 minutes, if the LED on industrial PC stays lit, flashing is completed. Using COM1, you can also find this message **>>>>>>> eMMC Flashing Completed <<<<<<<** which indicates that the system image was downloaded correctly to the eMMC.

12. Power OFF and set the switch S1 to eMMC boot mode. (refer to Boot Switch Configuration above)

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

The first time you start Android 6.0 OS on the industrial PC will take a little time. But after the first time, Android 6.0 OS will start quickly. It is a successful start if you see the Android 6.0 OS desktop such as the one shown in the figure below:

Figure 183: *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. The IPC will mounts the SD Card to */mnt/media_rw/* and */storage/* directory.

```
root@sabresd_6dq:/ # mmc1: host does not support reading read-only switch. assuming write-enable.
mmc1: new high speed SDHC card at address 59b4
mmcblk1: mmc1:59b4 SS08G 7.40 GiB
 mmcblk1: p1 p2
FAT-fs (mmcblk1p1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.
```

Figure 184: *Serial Message*

### USB Flash Disk Test

The USB Flash Disk is like the SD Card. The IPC mounts the USB Flash Disk to */mnt/media_rw/* and */storage/* directory.

```
usb 1-1.2: new high-speed USB device number 4 using ci_hdrc
usb 1-1.2: New USB device found, idVendor=05e3, idProduct=0736
usb 1-1.2: New USB device strings: Mfr=3, Product=4, SerialNumber=2
usb 1-1.2: Product: USB Storage
usb 1-1.2: Manufacturer: Generic
usb 1-1.2: SerialNumber: 000000000272
usb-storage 1-1.2:1.0: USB Mass Storage device detected
scsi1 : usb-storage 1-1.2:1.0
scsi 1:0:0:0: Direct-Access     Generic  STORAGE DEVICE   0272 PQ: 0 ANSI: 0
sd 1:0:0:0: [sda] 15529984 512-byte logical blocks: (7.95 GB/7.40 GiB)
sd 1:0:0:0: [sda] Write Protect is off
sd 1:0:0:0: [sda] No Caching mode page found
sd 1:0:0:0: [sda] Assuming drive cache: write through
sd 1:0:0:0: [sda] No Caching mode page found
sd 1:0:0:0: [sda] Assuming drive cache: write through
 sda: sda1 sda2
sd 1:0:0:0: [sda] No Caching mode page found
sd 1:0:0:0: [sda] Assuming drive cache: write through
sd 1:0:0:0: [sda] Attached SCSI removable disk
```

Figure 185: *USB flash disk test*

### Network Test

The network uses DHCP to get IP Addresses. You can use the **ethernet app** to set a static IP, to check the obtained IP from the router, and to set the proxy.

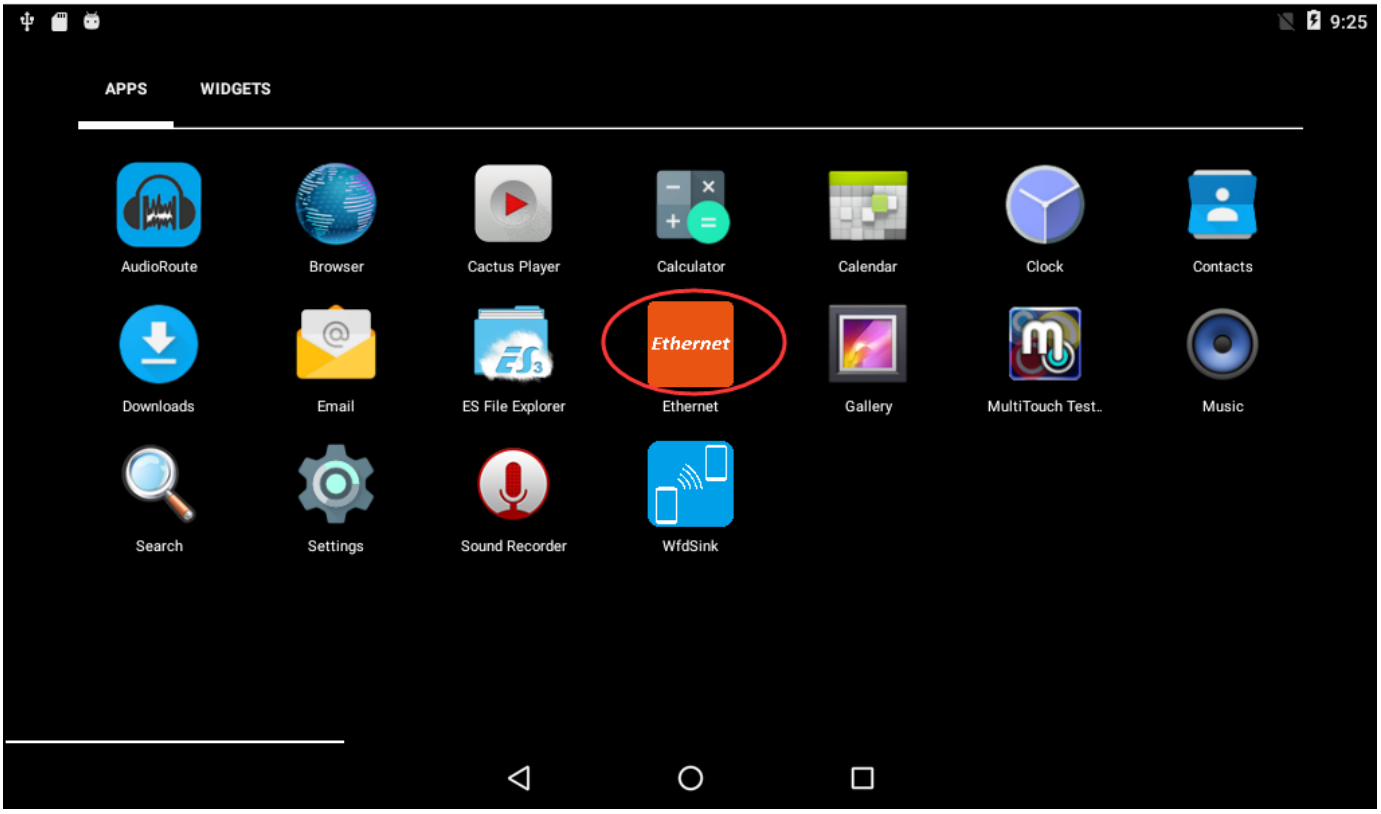
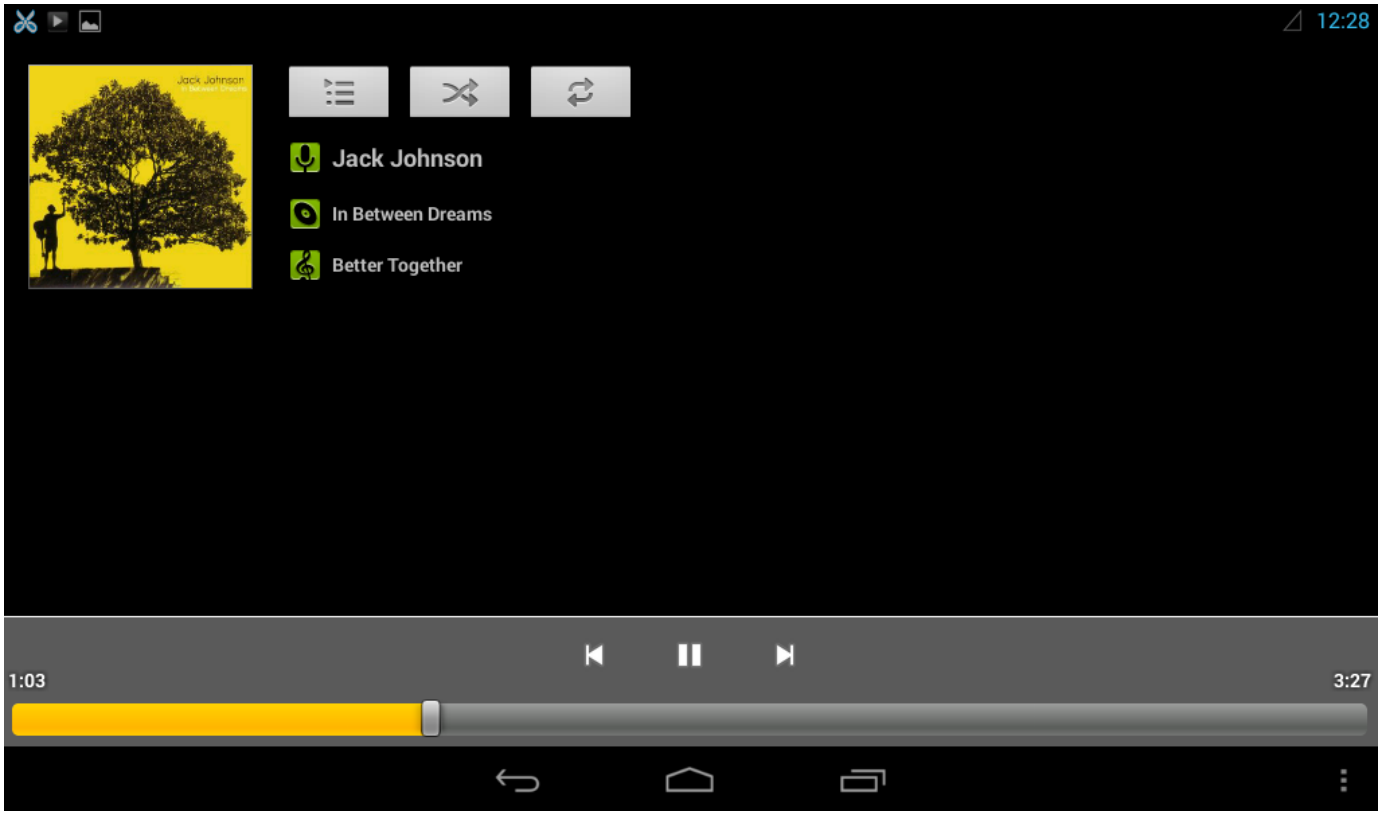



Figure 186: *Ethernet App*

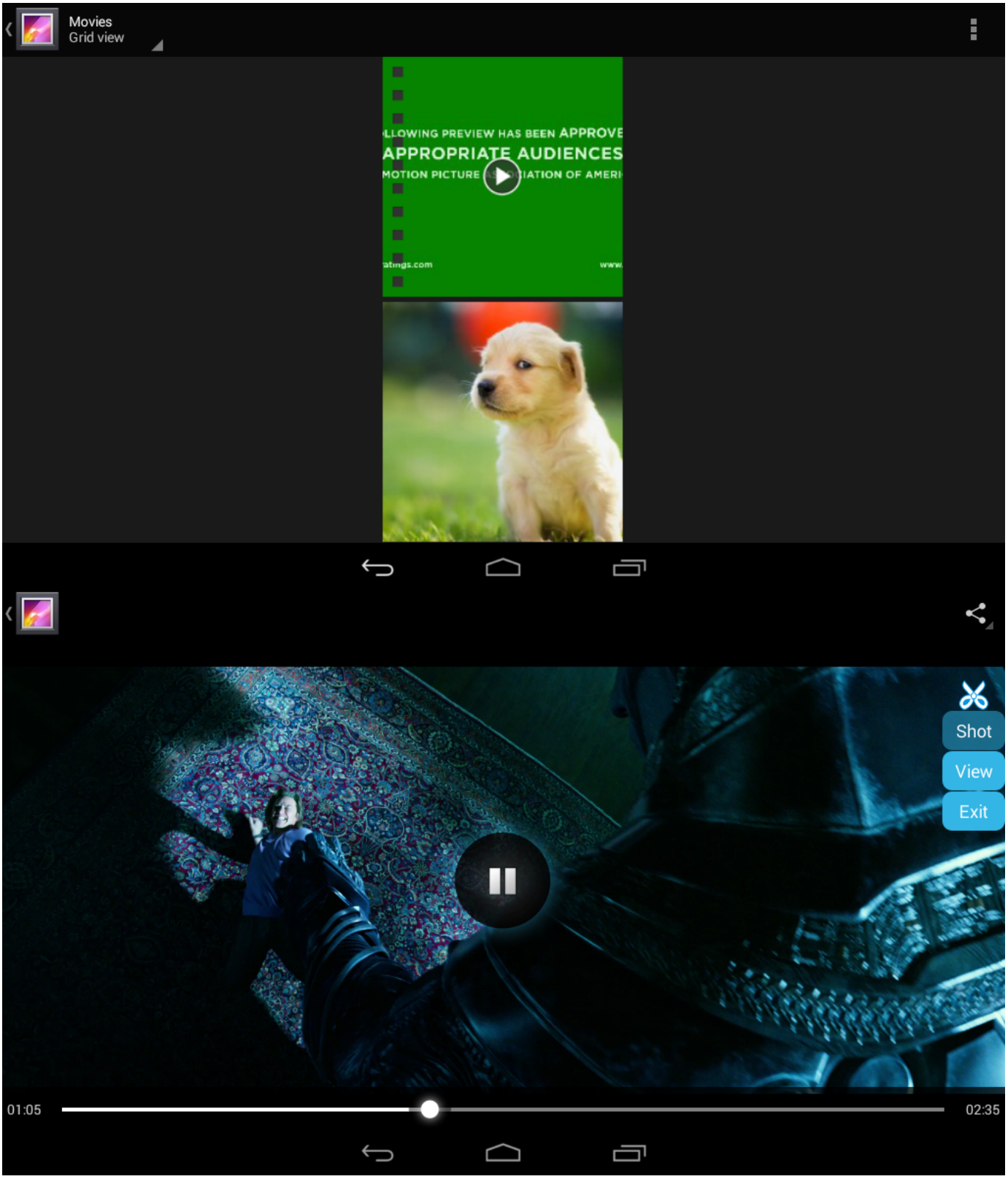
### Sound Card Test

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

Android 6.0 OS enables USB Debug by default.
You just need to insert the OTG cable into the IPC, and allow 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.**

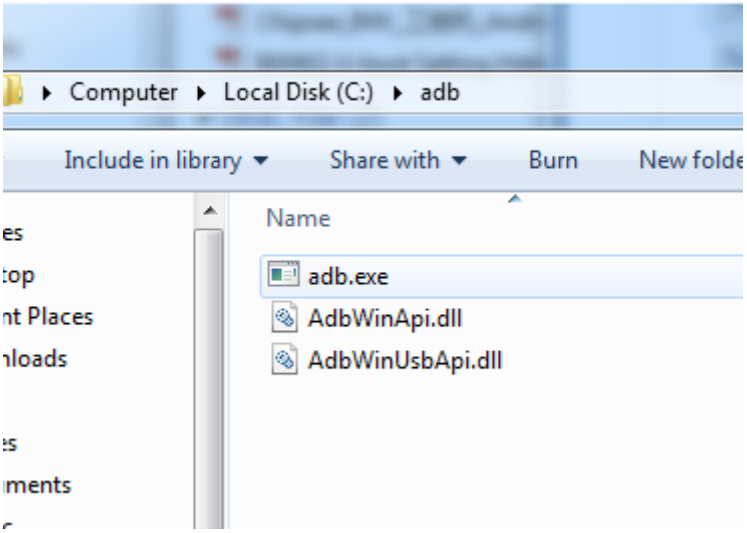
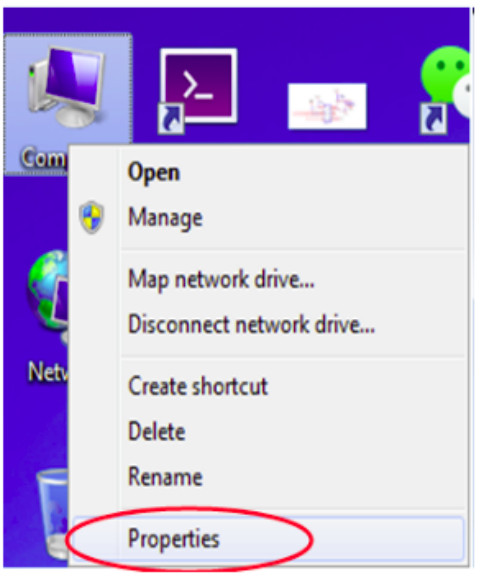



Figure 187: *Unzip adb.rar to c:*

- **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.**



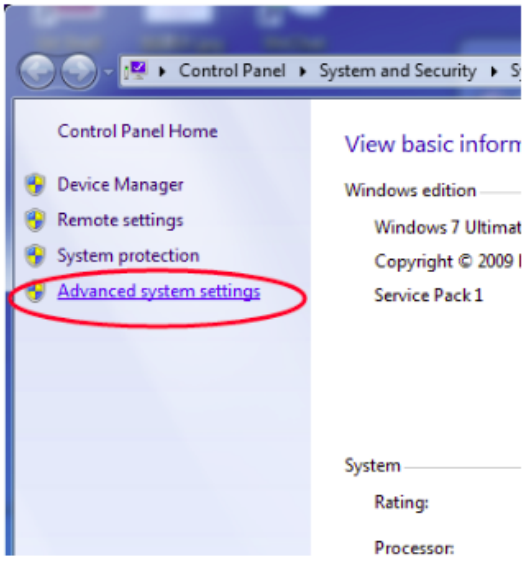




Figure 188: *Open Advance system settings*

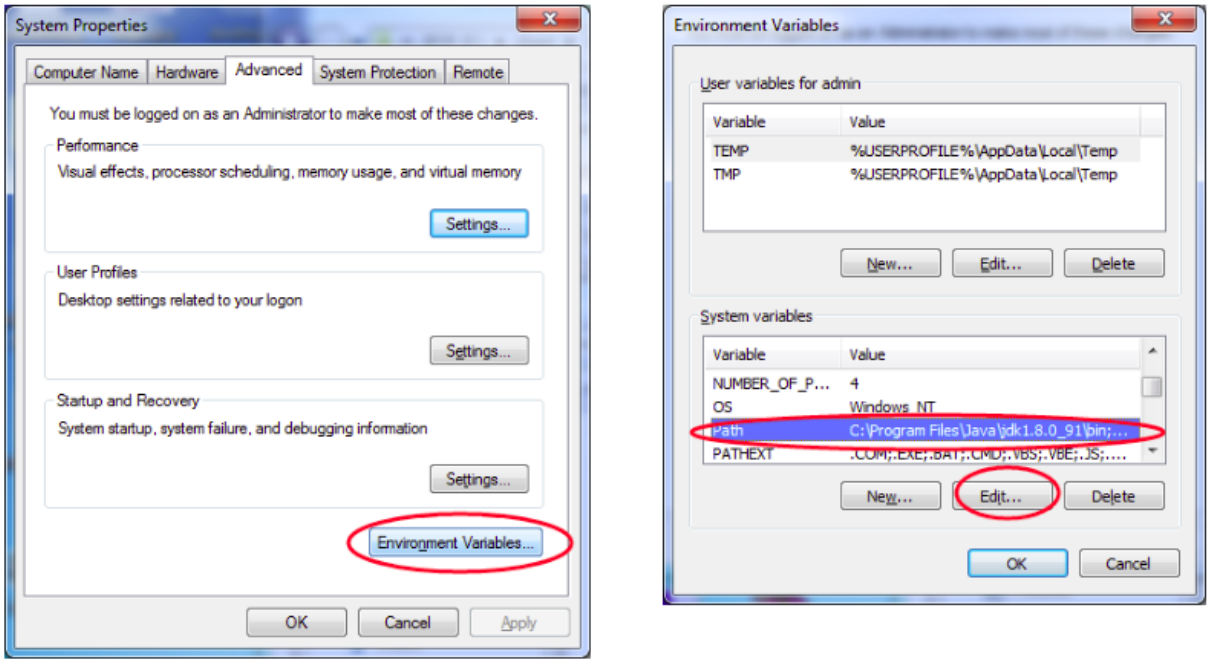



Figure 189: *Open and edit the **Path* system variable*

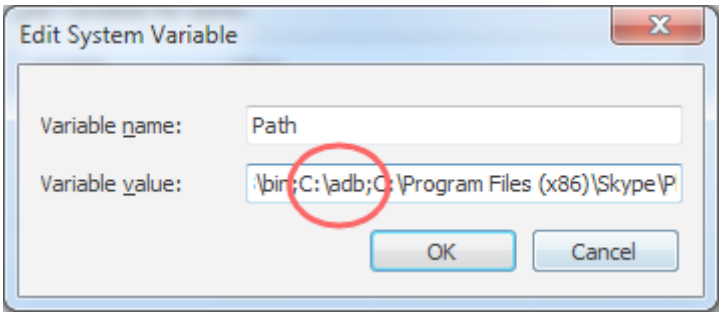



Figure 190: *Add path of the ADB directory to the* **Path** *system variable*

- **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.**

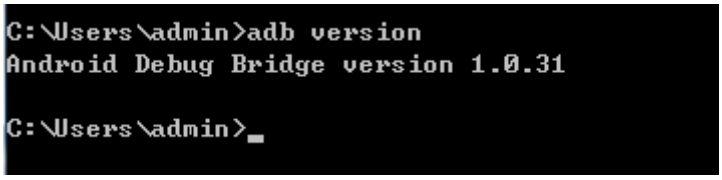



Figure 191: *ADB tool is working*

- **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.**

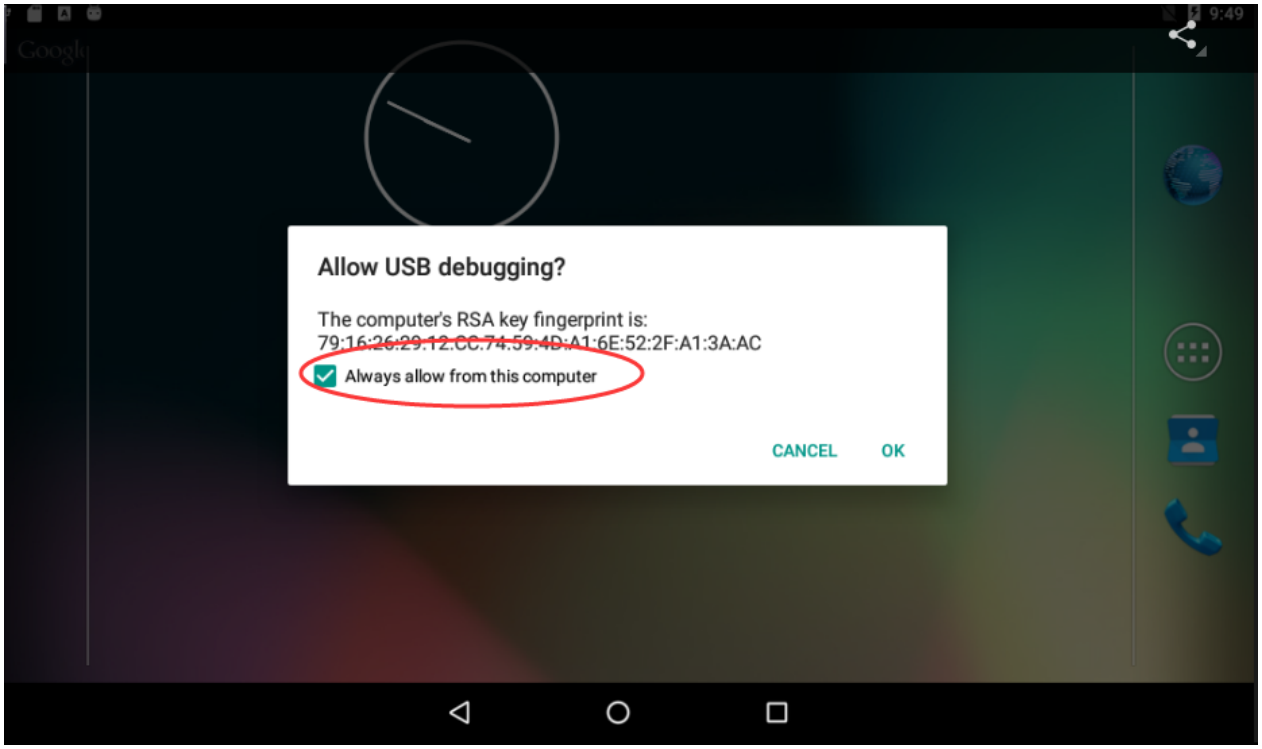



Figure 192: *Enable USB debugging*

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 193: *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>
```

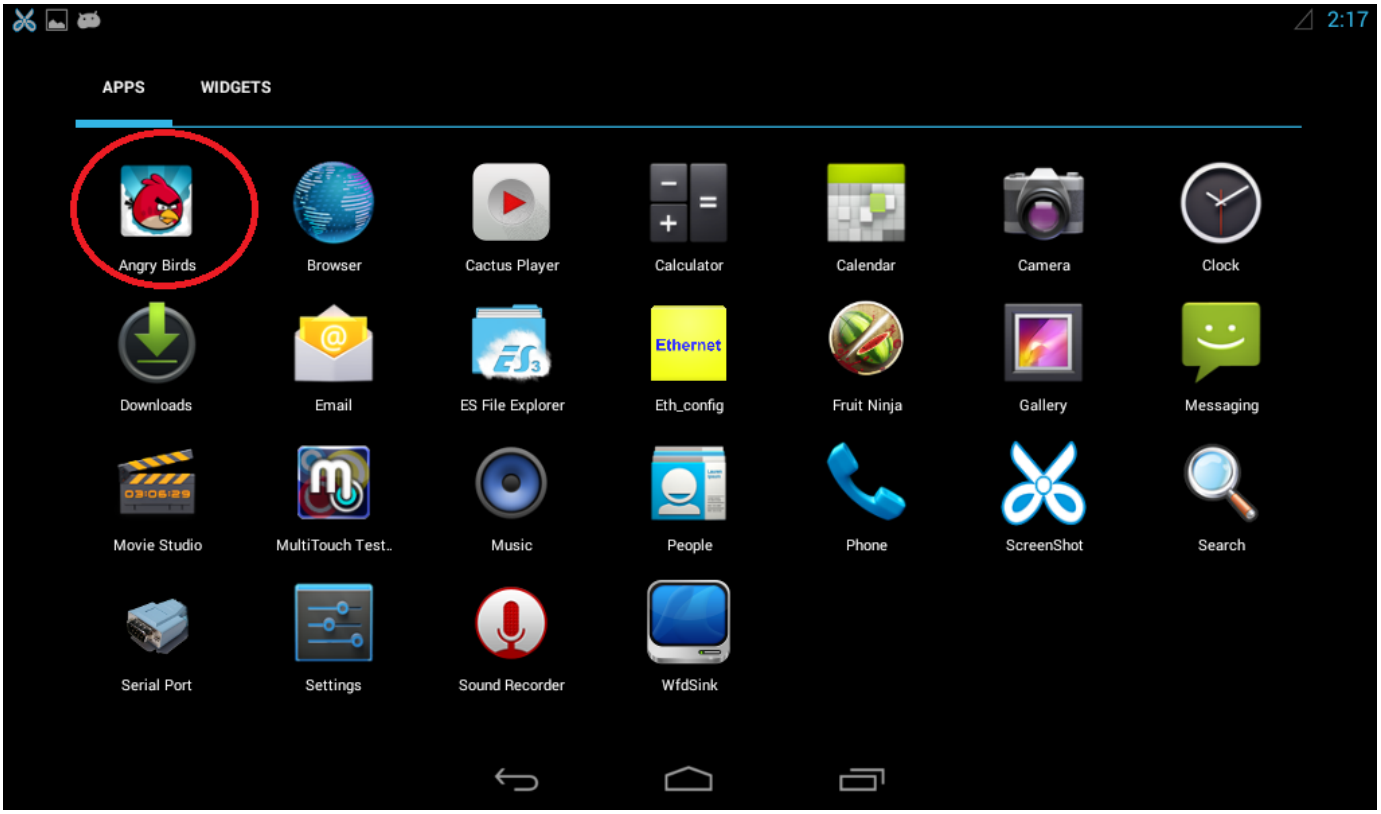



Figure 194: *Install android app*

### 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.

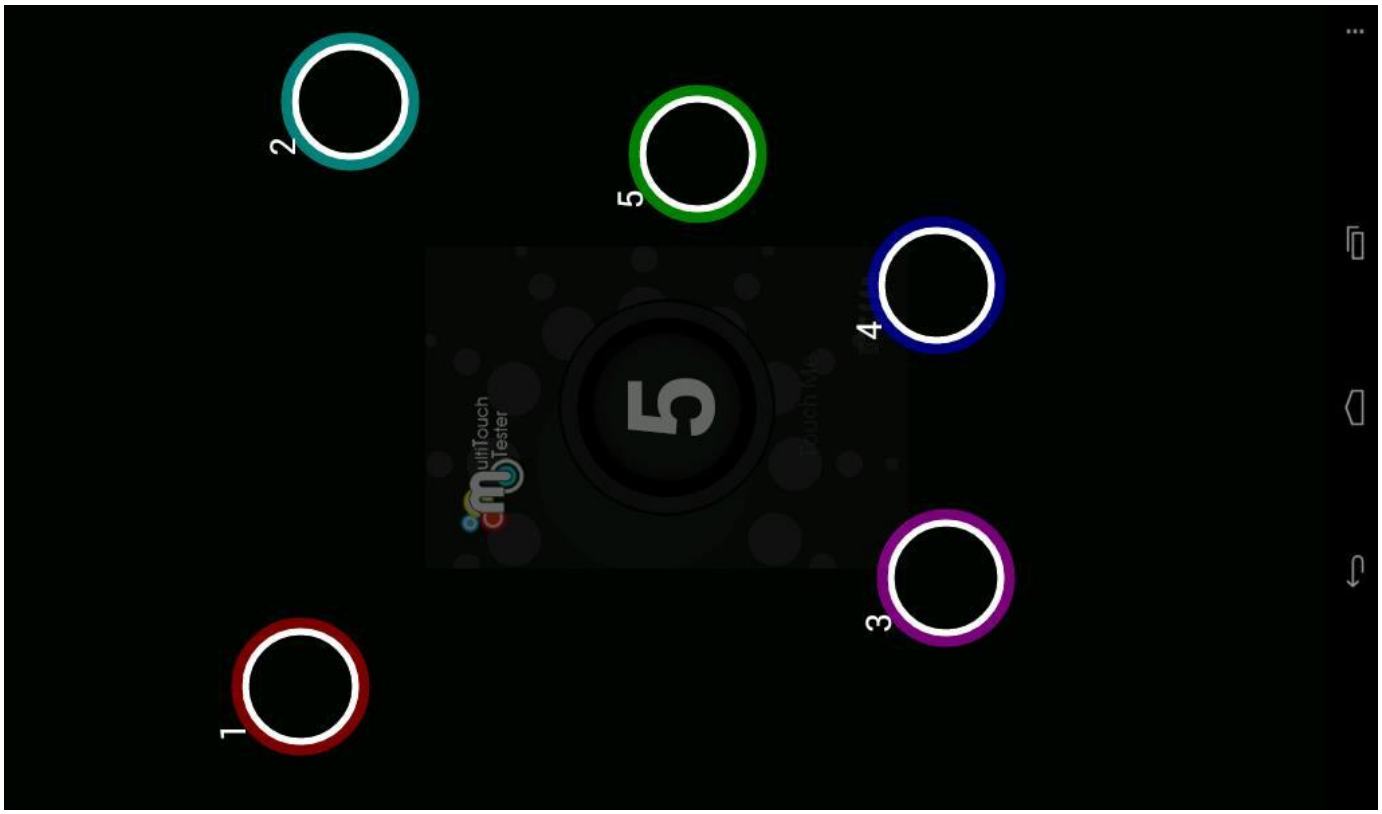



Figure 195: Touch screen test (Capacitive touch)

### 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 |

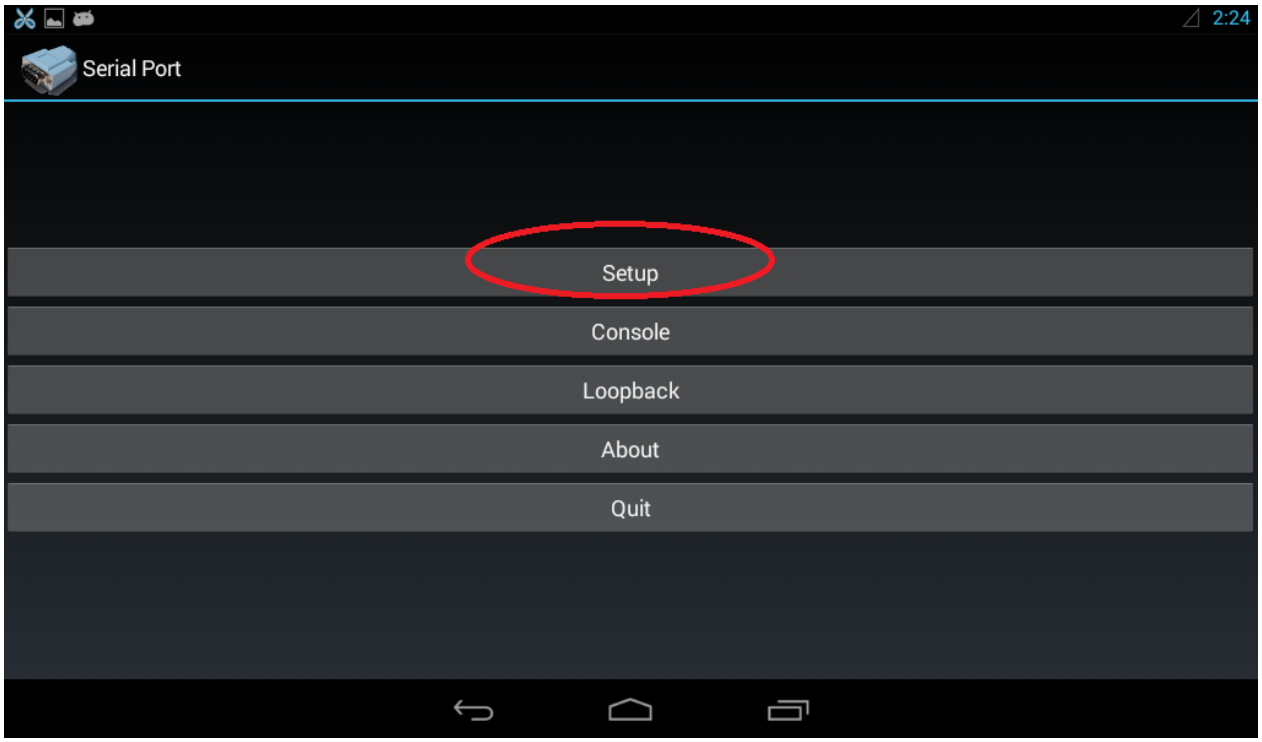
Table 45 Serial Ports Nodes on the System

**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.**



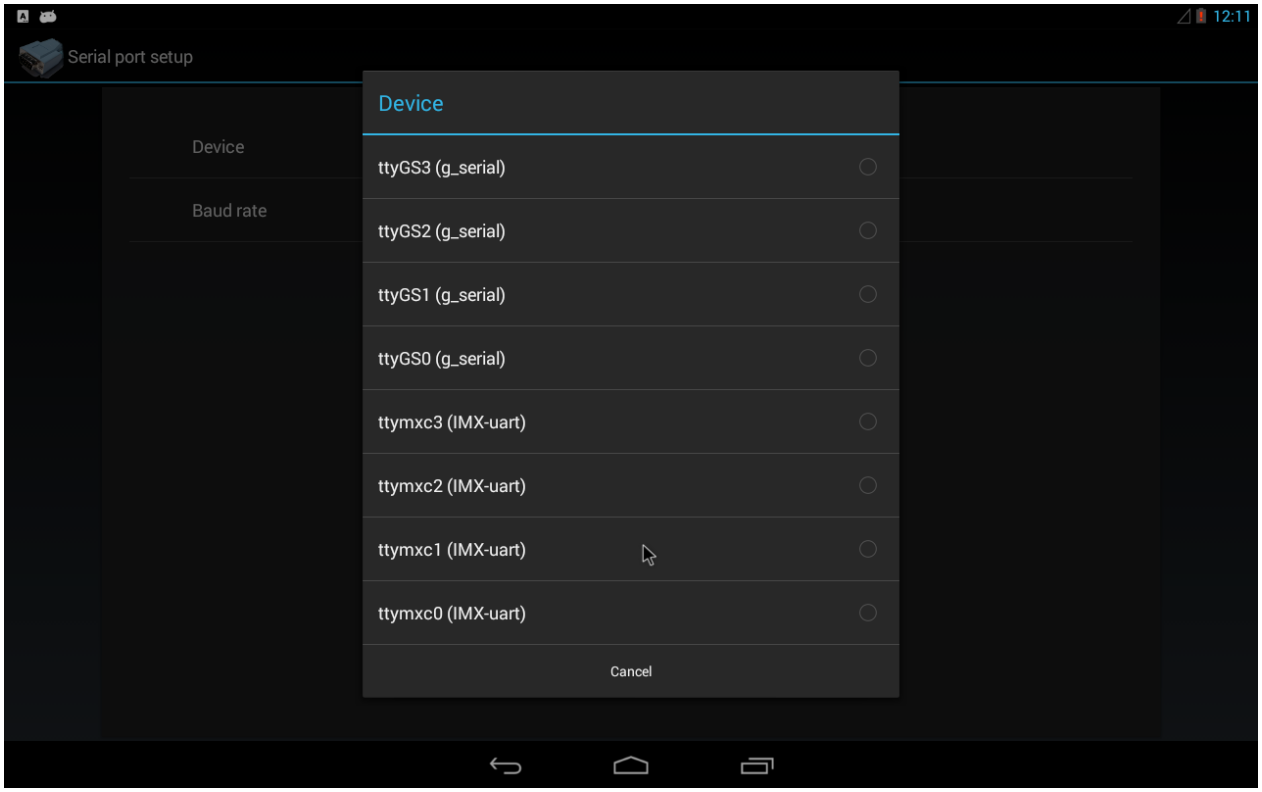



Figure 196: *Serial settings*

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

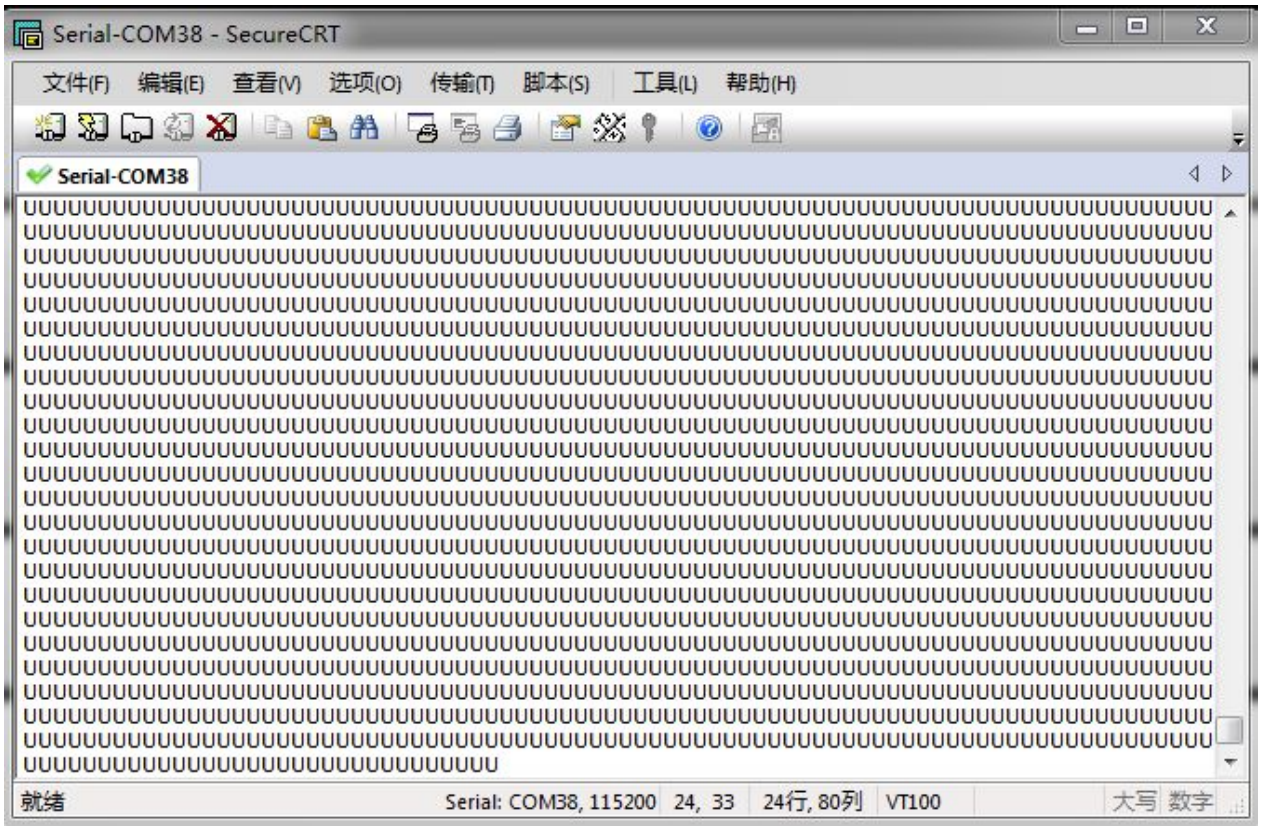



Figure 197: *Serial send test*

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

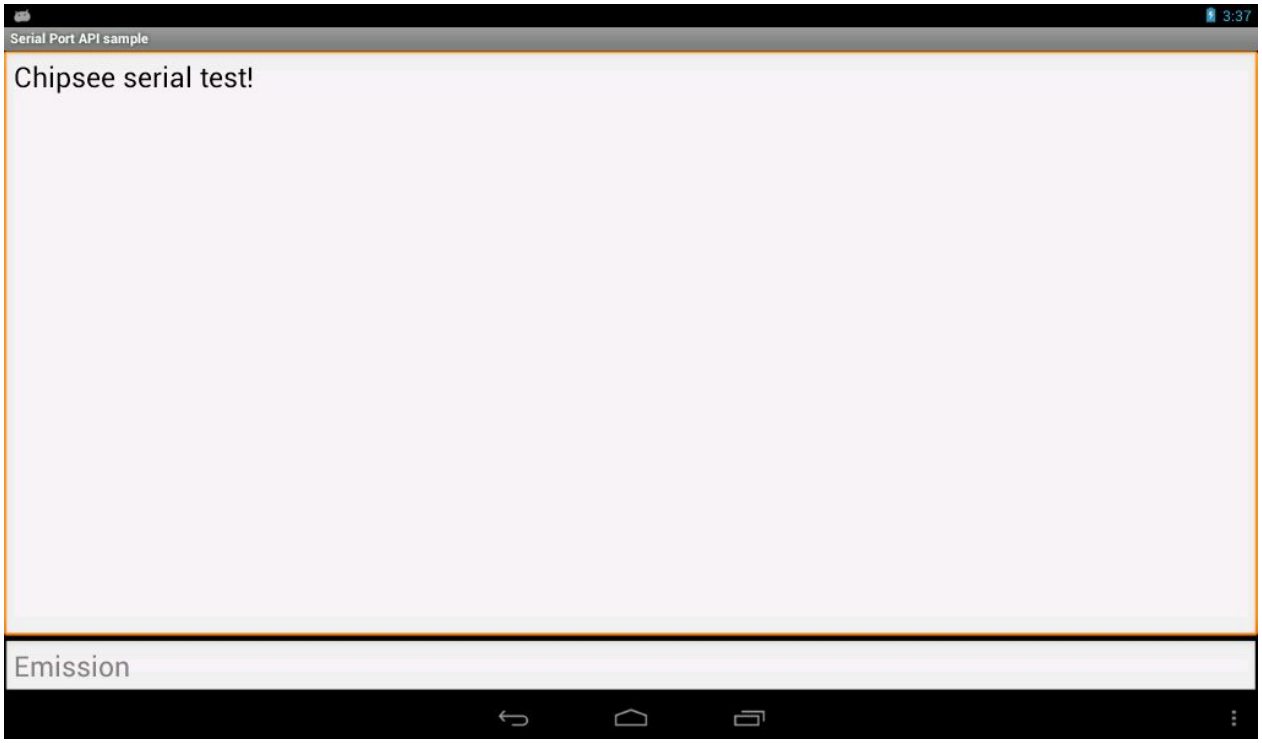



Figure 198: *Serial receive test*

### 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.

For the **CS10600F070** 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 CS10600F070 to know the position of the GPIO Connector. Refer to the table below for the available GPIO nodes on system.

> **Note**
>
> The V1 and V2 of CS10600F070 GPIO is not same.

You can use **GPIODemo** app to test the GPIO.

Figure 199: *GPIODemo app*

MODIFY LOGO

This system supports changing the logo by yourself. There are two ways:

- Replace the logo file in prebuilt images packages, and download images.
- Change the logo without downloading images.

> **Note**
>
> Logo file is one 32bpp, format is bmp.

### Method 1 - Downloading images

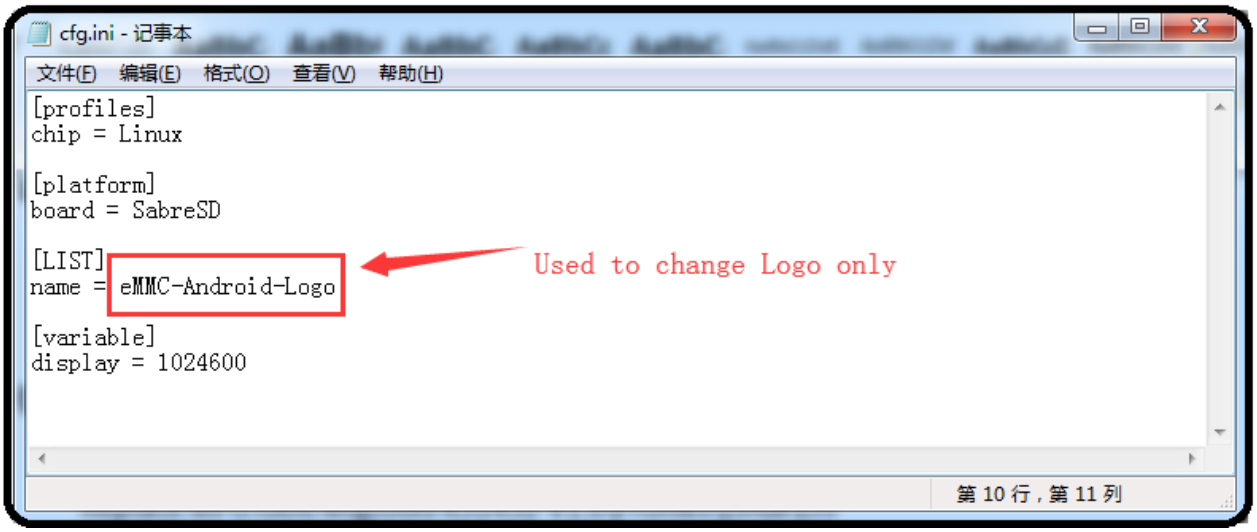
Replace the *prebuilt-xxx/emmc-flash/emmc/logo.bmp* and reference Prepare Manufacturing Tool and Image and Downloading Images by using MFGTool to flash the image.

### Method 2 - Don't Download Images

We will use **MFGTools** and the **Logoflasher** apps to change the logo.

### Use MFGTools to Change LOGO

- Replace the `logo.bmp` file in *Mfgtools-K31452-V1.0\Profiles\Linux\OS Firmware\files\ubuntu* with your customised logo file.
- **Open and edit the** *Mfgtools-K31452-V1.0\cfg.ini* **file and set the** `name` **variable to** `eMMC-Android-Logo` **as shown below.**




| PIN Number | GPIO Number | Devices File | Direction |
|---|---|---|---|
| 3 | gpio106 | /dev/chipsee-gpio7 | IN |
| 4 | gpio30 | /dev/chipsee-gpio3 | OUT |

| PIN Number | GPIO Number | Devices File | Direction |
|---|---|---|---|
| 6 | gpio95 | /dev/chipsee-gpio6 | IN |
| 7 | gpio87 | /dev/chipsee-gpio4 | OUT |
| 8 | gpio29 | /dev/chipsee-gpio1 | OUT |
| 9 | gpio28 | /dev/chipsee-gpio2 | OUT |
| 11 | gpio94 | /dev/chipsee-gpio5 | IN |
| 12 | gpio130 | /dev/chipsee-gpio8 | IN |

Table 46 CS12800F101 P18

| PIN Number | GPIO Number | Devices File | Direction |
|---|---|---|---|
| 21 | gpio106 | /dev/chipsee-gpio7 | IN |
| 22 | gpio29 | /dev/chipsee-gpio1 | OUT |
| 23 | gpio30 | /dev/chipsee-gpio3 | OUT |
| 24 | gpio28 | /dev/chipsee-gpio2 | OUT |
| 27 | gpio95 | /dev/chipsee-gpio6 | IN |
| 28 | gpio94 | /dev/chipsee-gpio5 | IN |
| 29 | gpio87 | /dev/chipsee-gpio4 | OUT |
| 30 | gpio130 | /dev/chipsee-gpio8 | IN |

Table 47 CS10600F070V1 P21

| PIN Number | GPIO Number | Devices File | Direction |
|---|---|---|---|
| 21 | gpio29 | /dev/chipsee-gpio1 | OUT |
| 22 | gpio106 | /dev/chipsee-gpio7 | IN |
| 23 | gpio28 | /dev/chipsee-gpio2 | OUT |
| 24 | gpio30 | /dev/chipsee-gpio3 | OUT |
| 27 | gpio130 | /dev/chipsee-gpio8 | IN |
| 28 | gpio87 | /dev/chipsee-gpio4 | OUT |
| 29 | gpio94 | /dev/chipsee-gpio5 | OUT |
| 30 | gpio95 | /dev/chipsee-gpio6 | IN |

Table 48 CS10600F070V2 P21

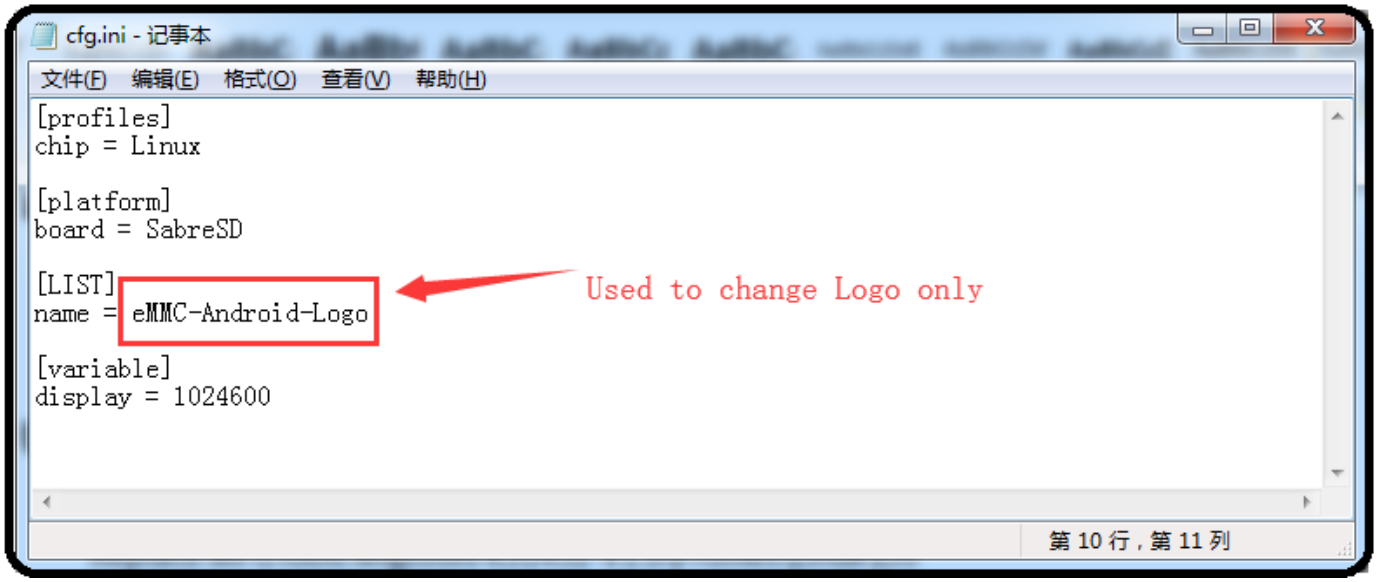



Figure 200: *Logo Modify with MFGTool*

### Use Logoflasher to Change Logo

You can get the Logoflasher file and use these tools to make one bootable TF card. Follow the steps below to change logo

- **Use the following commands to make bootable TF card.**

```
$ sudo  tar  zxvf  prebuilt-imx6qdl-bootfile-update-xxx.tar.gz
$ sudo  cd  prebuilt-imx6qdl-bootfile-update-xxx
$ sudo  ./mksdcard.sh  --device  /dev/sdX --display 1024600     //
resolution
```

- Put your custom logo file in the first partition `boot-flash` directory on the TF Card.
- Set boot mode to **TF card**. You can reference Boot Switch Configuration.
- Power ON the IPC. If you see this message, **>>>>>>> eMMC Flashing Completed <<<<<<<**, you are done:

# Android 6.0 system debug in Windows

In this section, we will discover how to view the Android 6.0 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 6.0 system via the serial port

Install the **SecureCRT** or **Putty** software on a Windows 7 PC to view the Android 6.0 system via the serial ports.
Follow these steps to view Android 6.0 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.**

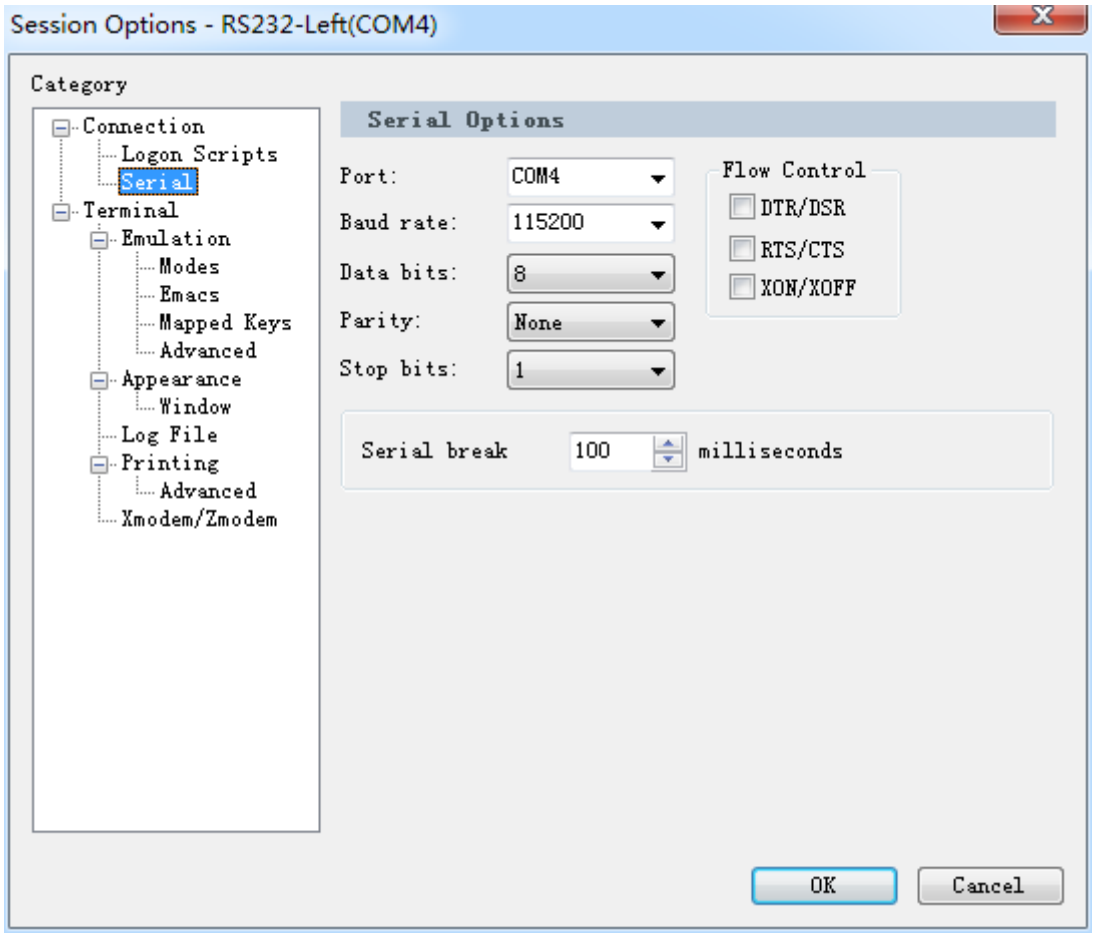



Figure 201: *SecureCRT configuration*

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

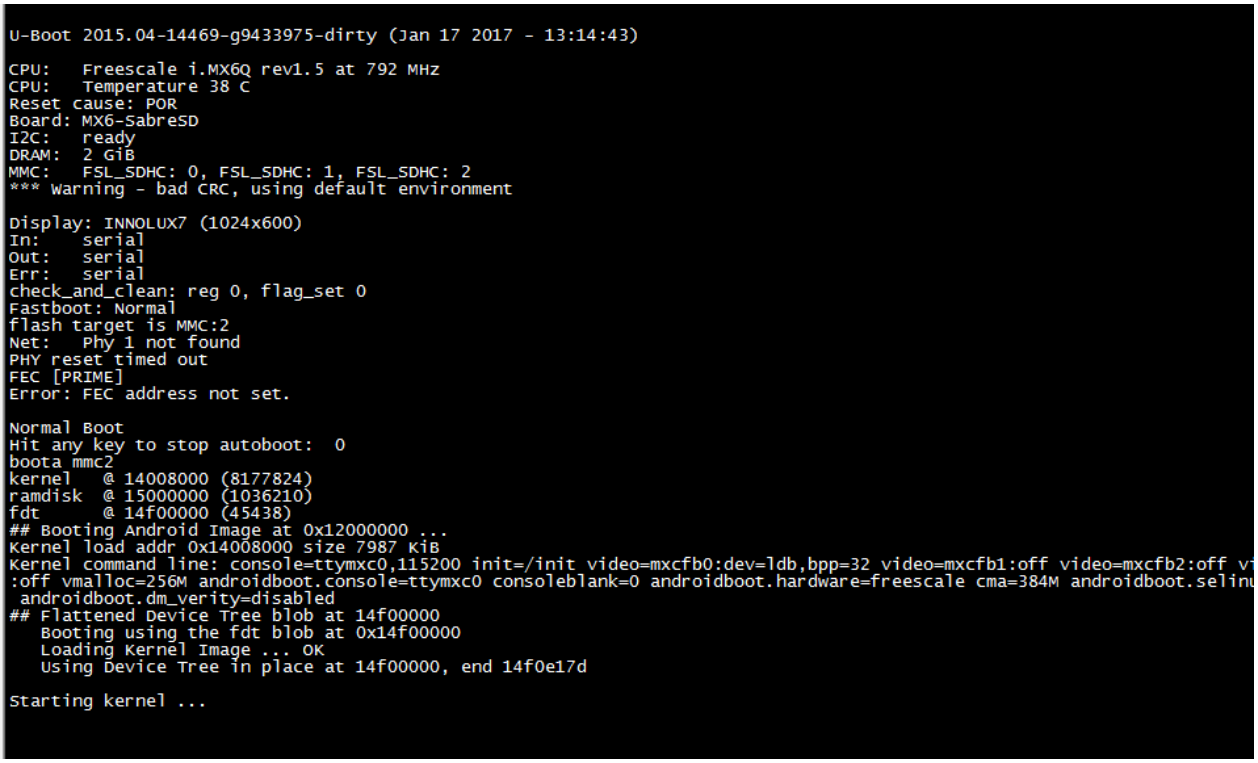



Figure 202: *Serial output information*

- 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 203: *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
```

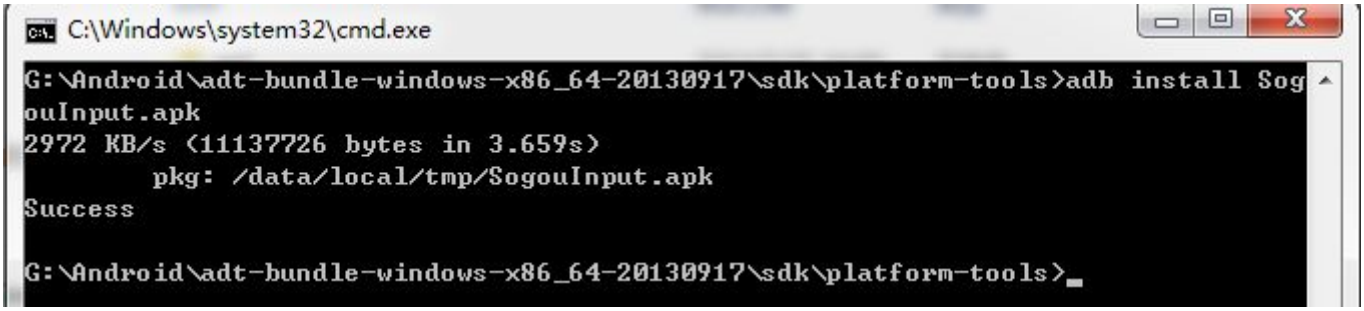



Figure 204: Install App

### 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.

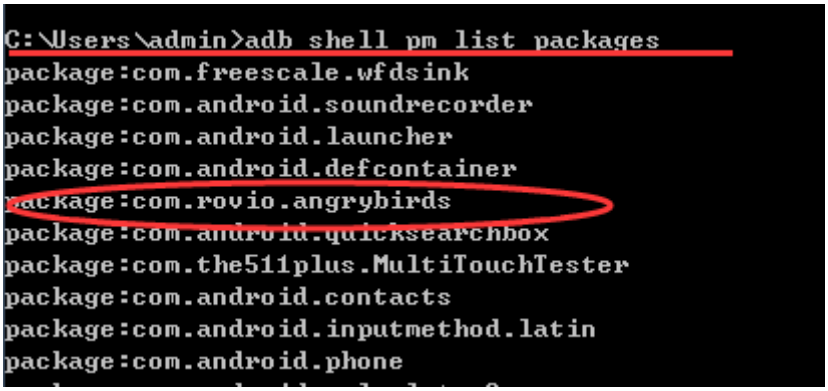



Figure 205: *Uninstall user app*

- 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
```

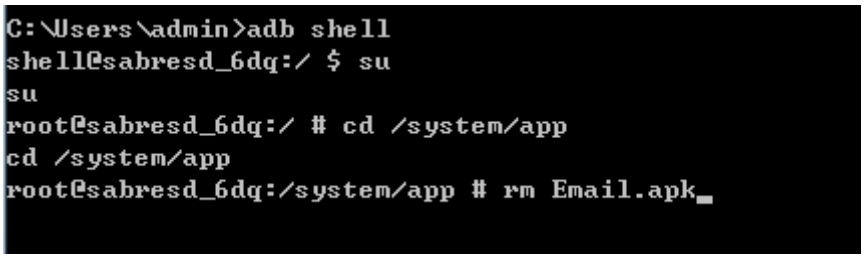



Figure 206: *Uninstall default app*

### 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 Android Studio on Windows. We assume that the USB is OTG model and the driver is already installed. (See Adb connect via USB OTG)

## Example — Develop a `HelloWorld` Program

1. **Start a new Android Studio project**

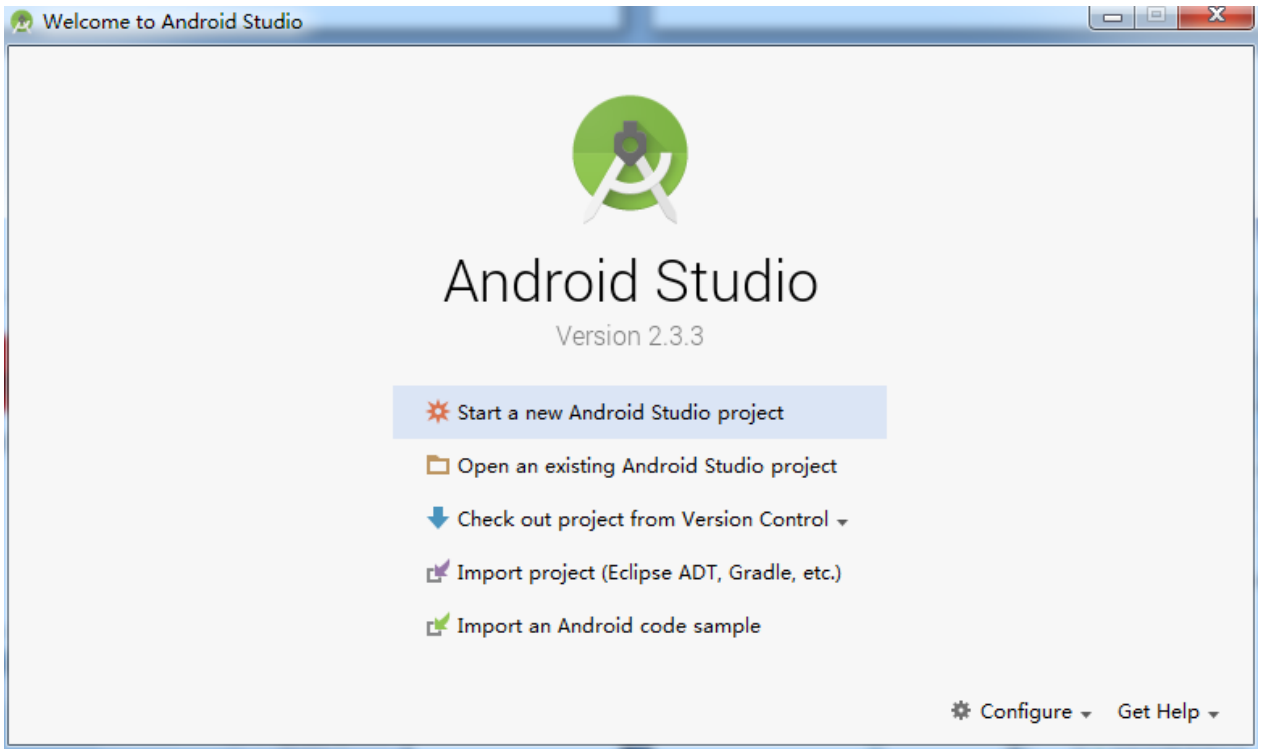



Figure 207: *New Project*

2. **Configure your new project**

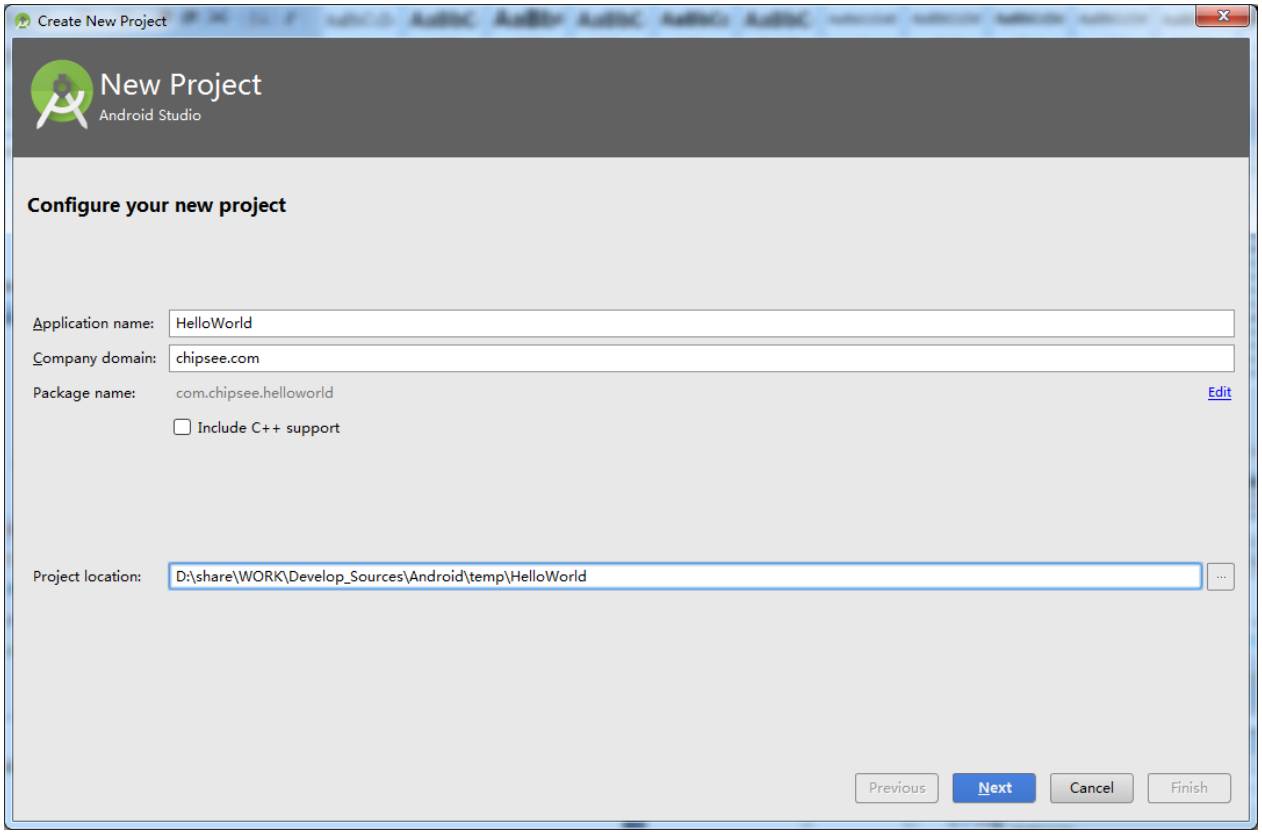



Figure 208: *Project Configuration*

### 3. Select the form factors your application will run on

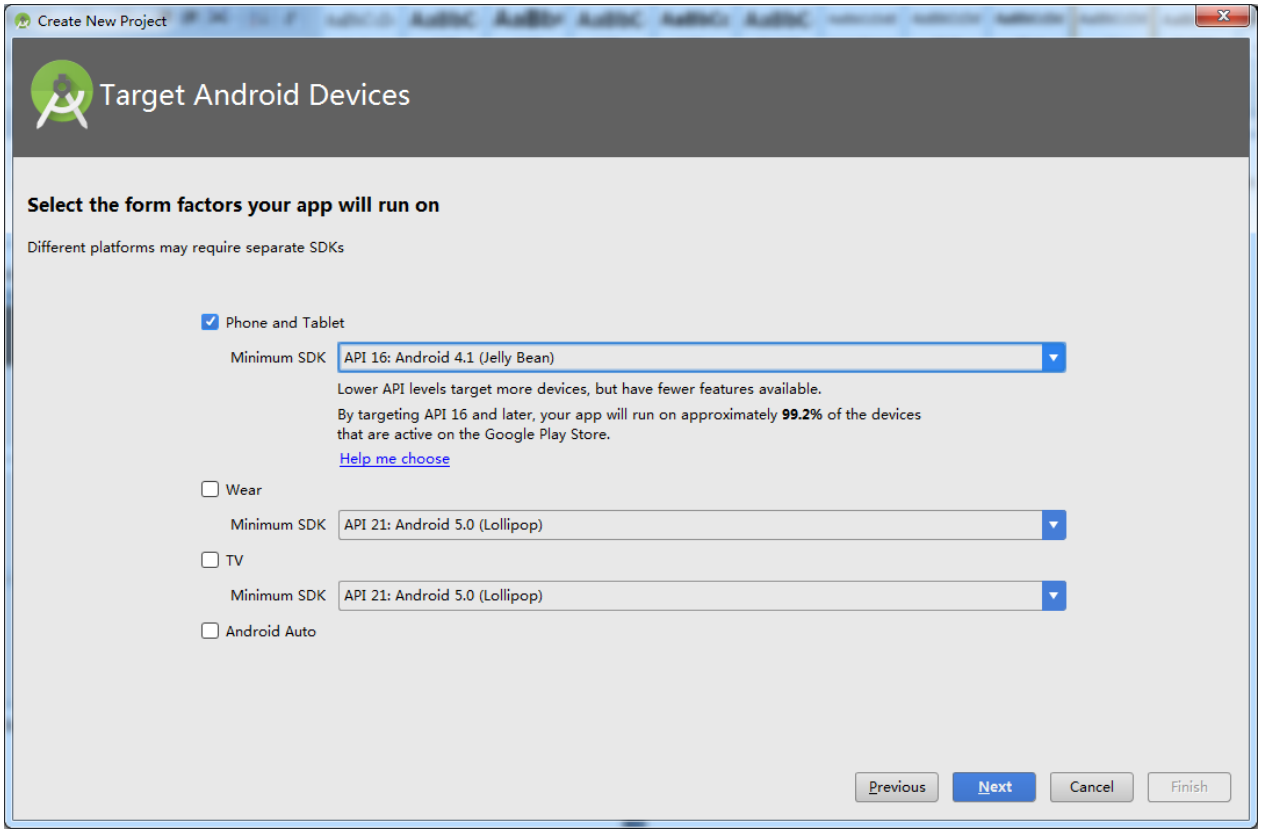



Figure 209: *App form factor*

### 4. Select one Empty Activity

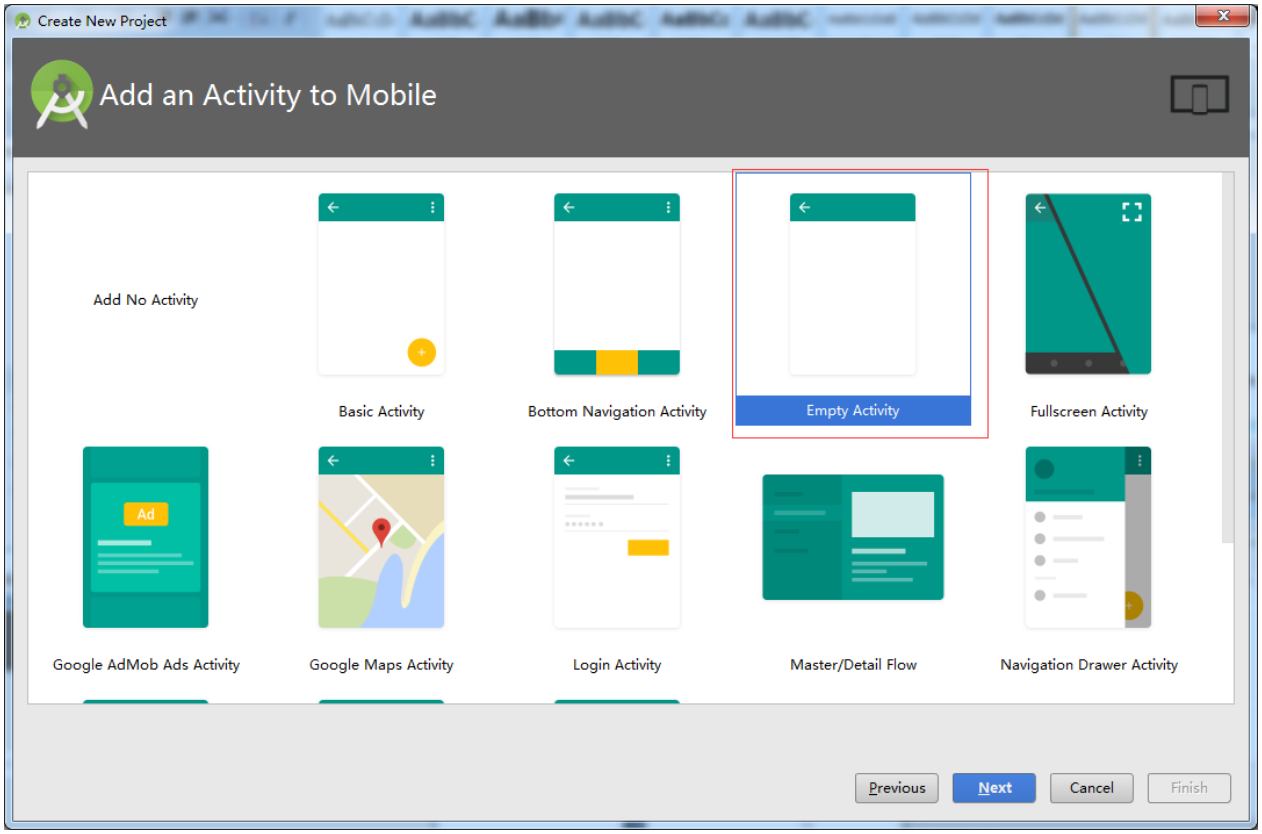



Figure 210: *Add Activity*

## 5. **Customize the Activity**

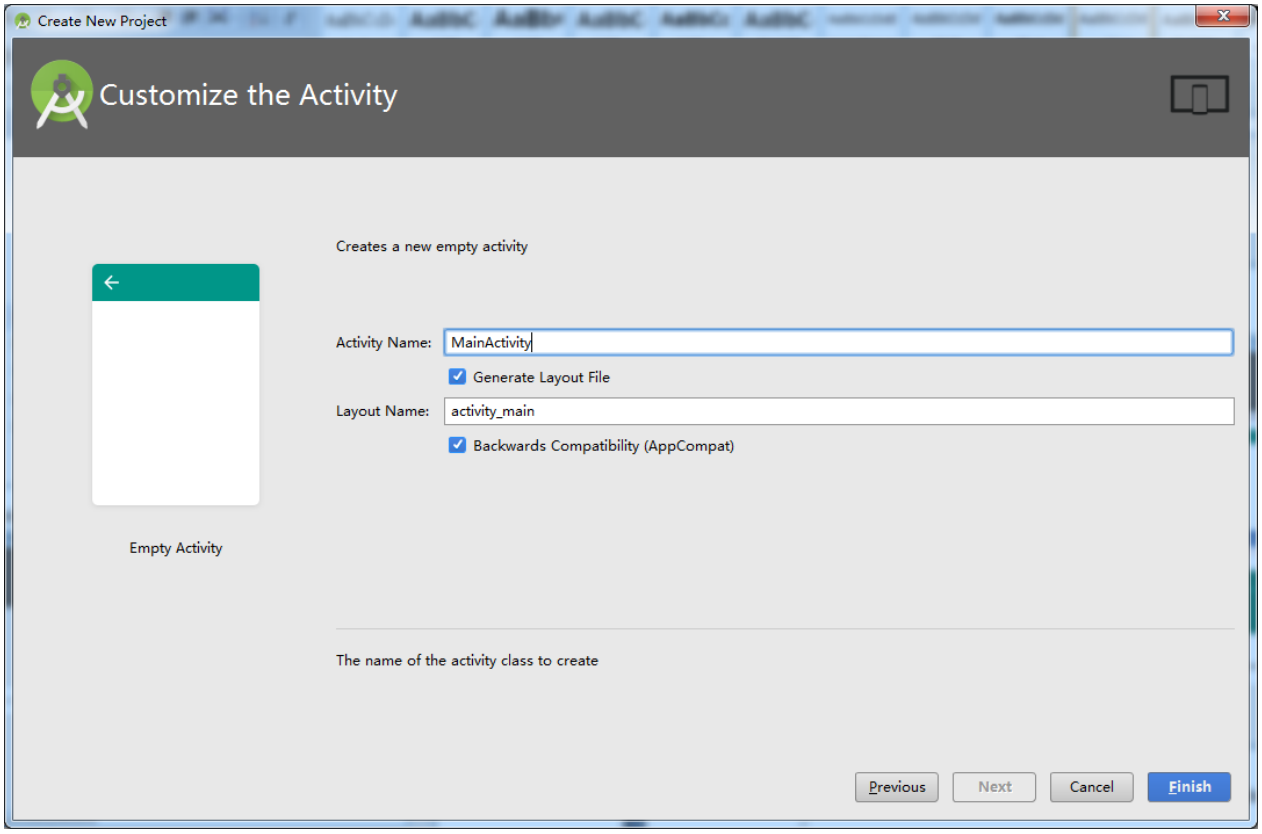



Figure 211: *Customize Activity*

## 6. **Develop the App**

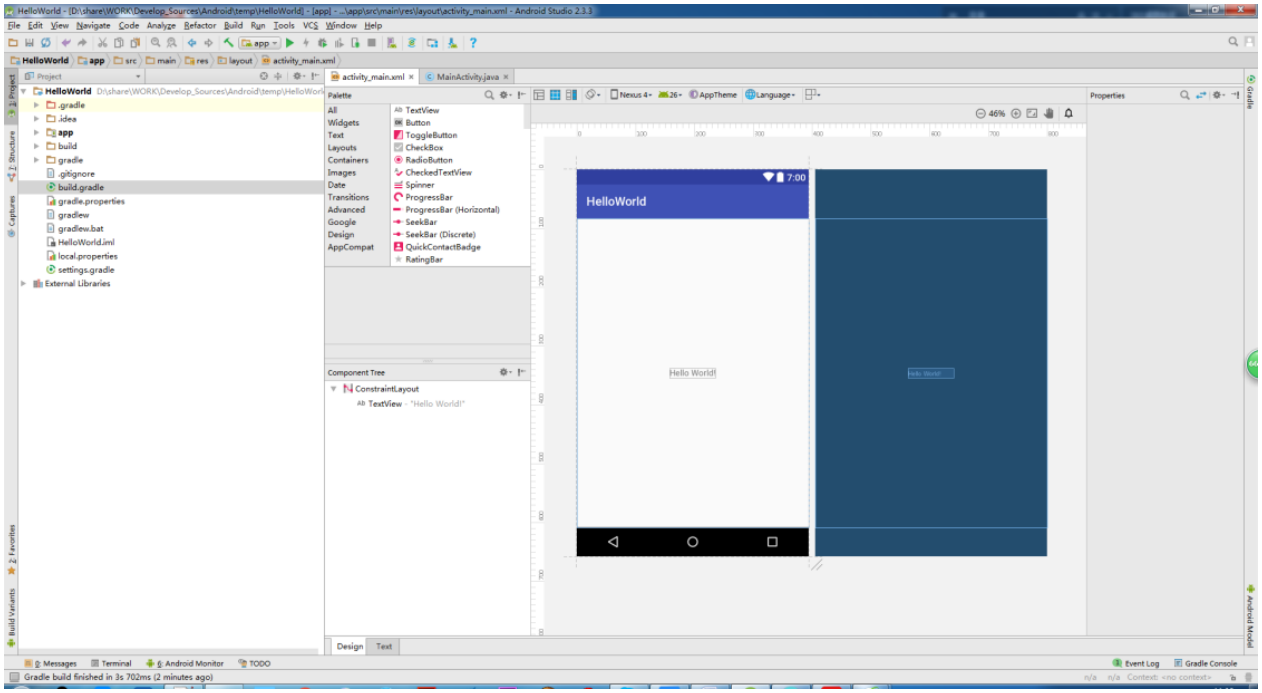



Figure 212: *App Development Interface*

7. **Run app on target IPC**

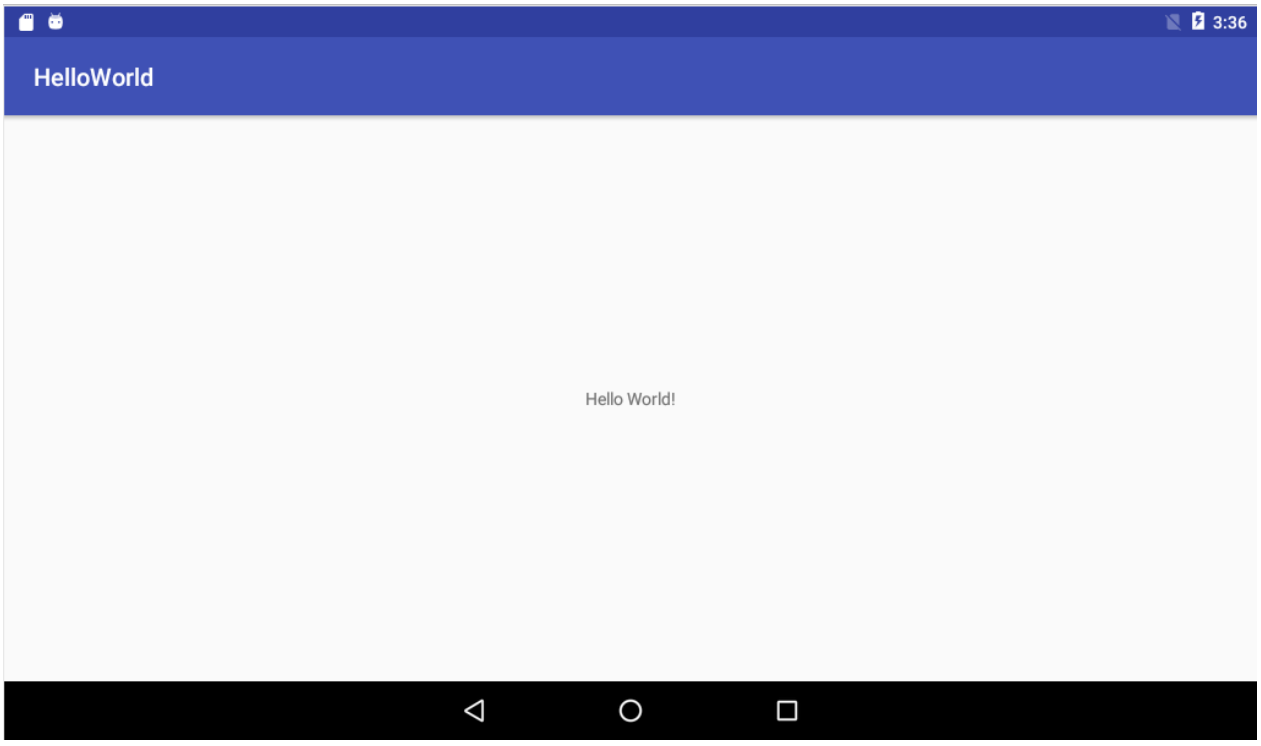



Figure 213: *HelloWorld Program*

> **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.