



Professional. Creaditable. Successful

Android LCD Screen Development Guide RK3566



Professional. Creaditable. Successful

1. Hardware Introduction.....	4
2. Firmware Modification	4
A. Tools/Resource	4
B. Overview of the Tool Interface.....	5
C. Procedure for Replacing Boot Logo.....	7
D. Boot Animation Replacement Process	8
E. Pre-installed APK Installation Process.....	10
F. Error Analysis and Resolution.....	11
3. Firmware Update and Burning.....	12
A. SD Card Flashing (SDDiskTool)	12
B. USB Mass Production Flashing (FactoryTool).....	14
C. USB Developer Flashing Tool (RKDevTool)	20
4. Setting Up and Using the Android Studio Development Environment	32
A. Tools/Resource	32
B. Software Installation	33
C. Creating a New Project	37
5. Using DWIN Android Screen Serial Port Tools	41
A. DWIN Android Tools Package Instruction	41



Professional. Creaditable. Successful

- B. Importing Dwin Test Project Source Code into Android Studio.....42
- C. Using Dwin Java Serial Libraries to Create a New Project44
- D. Example of using common functions of DWIN Android screen48
- 6. adb Installation and Usage.....54
 - A. Installation on PC:54
 - B. Debugging Using Android Studio56



1. Hardware Introduction

Refer to the related model datasheet.

2. Firmware Modification

A. Tools/Resource

- i. In the Android resource directory, the contents of the FWFactoryTool_RK3566_v2.3 compressed package are as shown in the figure below.

名称	修改日期	类型	大小
Bin	2023/5/6 11:46	文件夹	
FileContexts	2023/5/6 11:46	文件夹	
imageformats	2023/5/6 14:01	文件夹	
Language	2023/12/18 13:30	文件夹	
Licenses	2023/5/6 14:41	文件夹	
platforms	2023/5/6 14:01	文件夹	
Rockchip	2023/5/6 11:51	文件夹	
config.ini	2023/5/6 15:07	配置设置	1 KB
FWFactoryTool.exe	2023/12/18 13:29	应用程序	439 KB
icudt51.dll	2013/4/23 17:50	应用程序扩展	21,794 KB
icuin51.dll	2013/4/23 17:49	应用程序扩展	1,759 KB
icuuc51.dll	2013/4/23 17:49	应用程序扩展	1,305 KB
libEGL.dll	2013/7/2 20:59	应用程序扩展	42 KB
libGLESv2.dll	2013/7/2 20:59	应用程序扩展	682 KB
msvcp110.dll	2012/11/6 2:20	应用程序扩展	523 KB
msvcr110.dll	2022/6/9 9:38	应用程序扩展	855 KB
Qt5Core.dll	2022/6/8 13:56	应用程序扩展	3,774 KB
Qt5Gui.dll	2013/7/2 21:05	应用程序扩展	2,981 KB
Qt5Widgets.dll	2013/7/2 21:08	应用程序扩展	4,197 KB

Log Folder: This folder contains log files generated during the tool's runtime.

Temp Folder: This folder contains files extracted from the unpacked firmware update.img. The subsequent



Professional. Creaditable. Successful

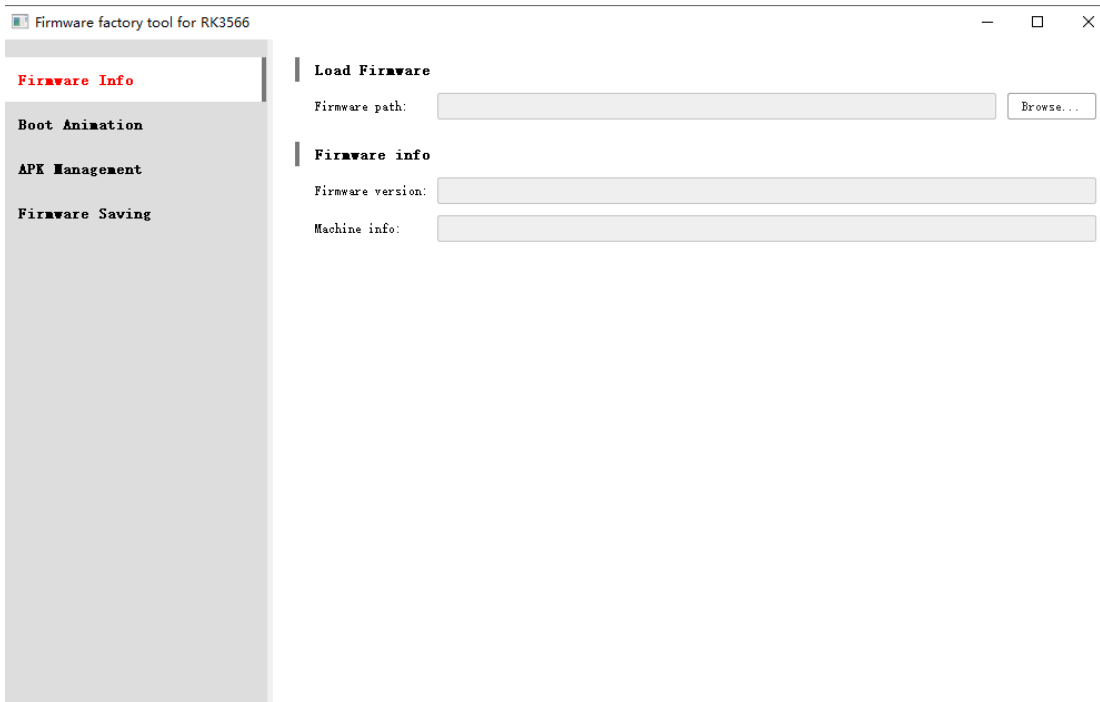
operations of the tool involve editing the files in the Temp folder and then packaging them to generate a modified update.img file.

FWFactoryTool.exe: This is the application for firmware modification.

- ii. update.img: Original firmware. Please request from the R&D team according to the model and manufacturing date to obtain this firmware. All the following operations are performed based on this firmware.

B. Overview of the Tool Interface

- i. Startup Interface



The image above shows the startup interface of the tool. Click the "Browse" button in the upper right corner



Professional. Creaditable. Successful

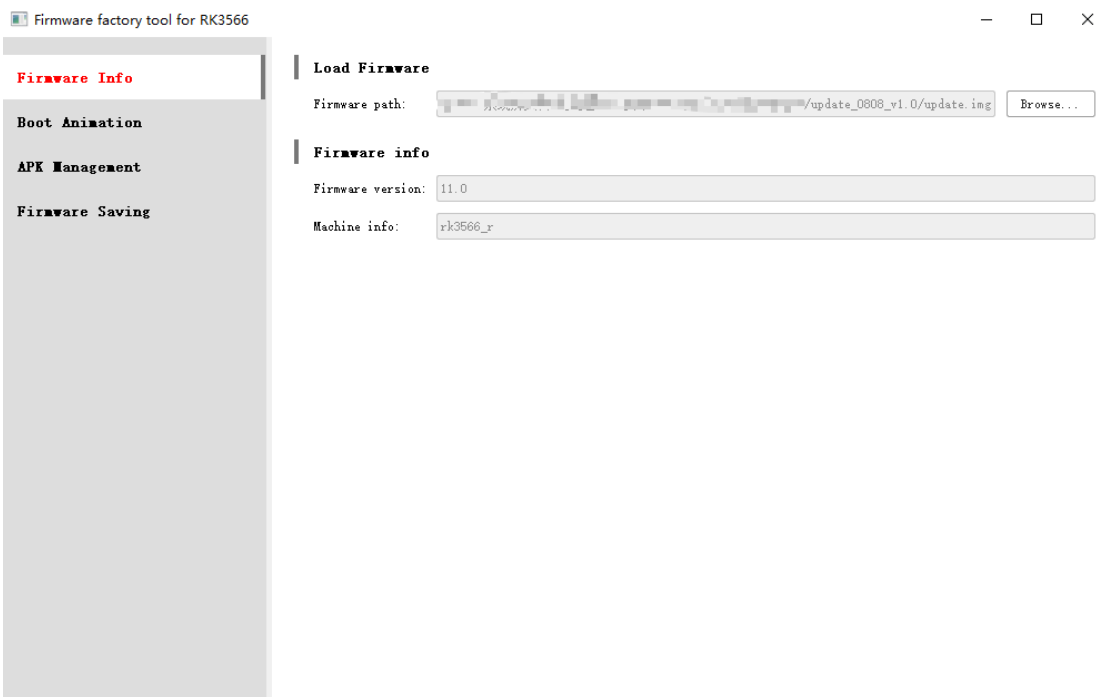
to select the path of the update.img file. This firmware is intended for device burning and serves as the base for user modifications.

While loading, the tool begins to unpack the firmware, extracting the corresponding images to the Temp folder in the tool's root directory.

Depending on the size of the firmware and the configuration of the computer, the unpacking time may vary. Generally, it takes about 5 minutes to complete the unpacking process.

It's important to note that since the tool invokes various third-party programs, some of which might not fully support directories with special characters, this could affect the unpacking process of the images. Therefore, it is recommended to place the entire firmware tool in a relatively simple path that **does not include Chinese characters** (for example, D:\FWFactoryTool\) and then run the firmware modification tool program.

ii. Firmware loaded successfully Interface Successfully





Professional. Creaditable. Successful

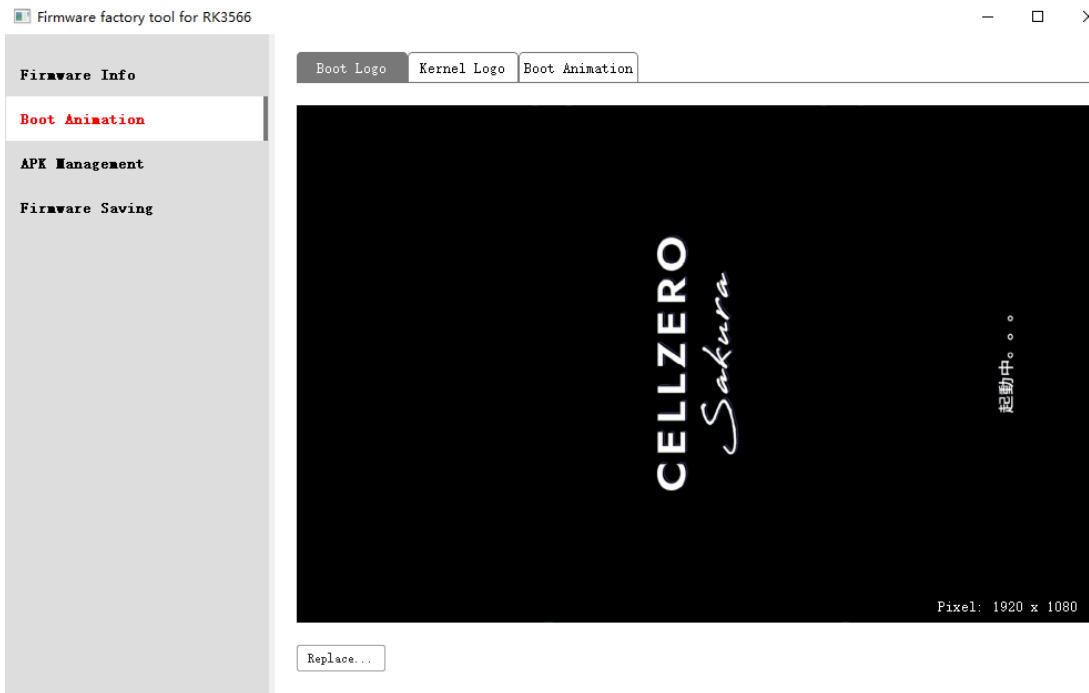
Above is the interface after successfully extracting the update.img firmware.

Once the extraction is done, if you look in the Temp folder in the tool's root directory, you'll see the contents of the extracted firmware. This is when you can start using the tool to make the changes you want.

After you're done making your changes, on the right side, there's a " Firmware Saving " option. Click it, and then press the save button. This way, you'll save the firmware with your modifications. The time this takes can vary based on how big the firmware is and how fast your computer is. Generally, it might take around 5 to 10 minutes to create the new modified firmware.

c. Procedure for Replacing Boot Logo

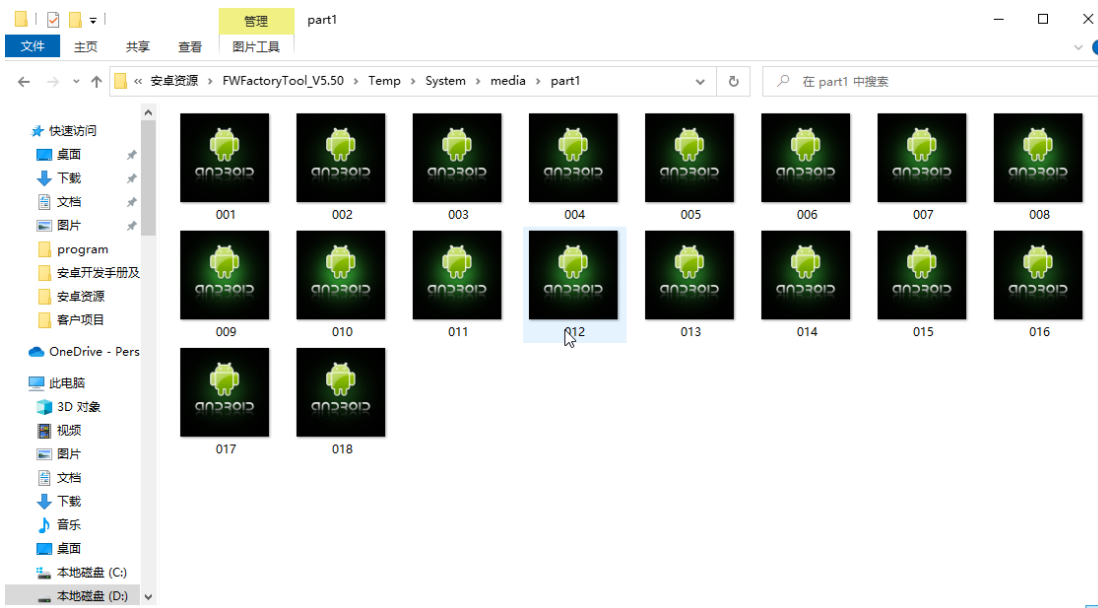
Prepare an 8-bit BMP image for replacement in advance. It's recommended to convert it to the required format before use. You can do this by visiting the following link: <https://online-converting.com/image/convert2bmp/> Click on [Boot Animation] and select either the boot logo or kernel logo. Click [Replace...], then choose an image that meets the requirements (adjust the image as needed when switching from portrait to landscape mode). Finally, click Save to complete the replacement process.



D. Boot Animation Replacement Process

i. Creating Boot Animation:

To start, you'll need to determine the size and number of images required according to your equipment and needs and create an image set similar to the following one.



Then, place these images into the "part1" folder and create a "desc.txt" file according to your needs.

```
l1080 1920 7  
p 0 0 part1
```

In the example image, "1080" and "1920" represent the animation image resolution, and "7" denotes the frames per second for playback.

The line "p 0 (repeat) 0 (empty command) part1" indicates that the images within the "part1" folder will loop continuously. After these two lines of instructions, need to enter the Enter Key to change the line.

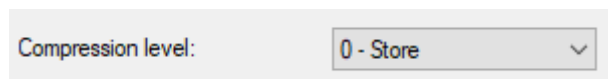
Don't need to enter anything, just save the desc file.

Compress the "part1" folder and "desc.txt" file into a "bootanimation.zip" file to complete the animation creation.

There is no need to put the desc and picture folders into the upper-level folder for compression. The correct example is as follows:



select 0-store when compressing. When the startup animation is too large, you can choose to handle the picture collection size.



ii. Boot Animation Replacement:

Click on [Boot Animation], select the boot animation, and then click [Replace...]. Choose the previously created compressed package to replace the animation.



Professional. Creaditable. Successful

E. Pre-installed APK Installation Process

Choose [APK Management], then select either the System APK, pre-installed APK, or preinstall_del APK. Next, select a specific item and right-click to perform the add or delete operation.

Firmware Info

Boot Animation

APK Management

Firmware Saving

System APK | Preinstall APK | Preinstall_del APK

	File name	File size	File date
1	BasicDreams.apk	52 K	2023-08-08
2	Bluetooth.apk	7776 K	2023-08-08
3	BluetoothMidiService.apk	24 K	2023-08-08
4	BookmarkProvider.apk	24 K	2023-08-08
5	Camera2.apk	5093 K	2023-08-08
6	CaptivePortalLogin.apk	474 K	2023-08-08
7	CarrierDefaultApp.apk	140 K	2023-08-08
8	CertInstaller.apk	5093 K	2023-08-08
9	CompanionDeviceManager.apk		2023-08-08
10	CtsShimPrebuilt.apk		2023-08-08
11	EasterEgg.apk	4357 K	2023-08-08
12	ExtShared.apk	12 K	2023-08-08
13	HTMLViewer.apk	24 K	2023-08-08
14	KeyChain.apk	140 K	2023-08-08
15	LiveWallpapersPicker.apk	2815 K	2023-08-08
16	NfcNci.apk	3499 K	2023-08-08
17	PacProcessor.apk	16 K	2023-08-08
18	PinZhengLauncher.apk	9145 K	2023-08-08
19	PrintRecommendationService.apk	118 K	2023-08-08
20	PrintSpooler.apk	1039 K	2023-08-08



Professional. Creaditable. Successful

F. Error Analysis and Resolution

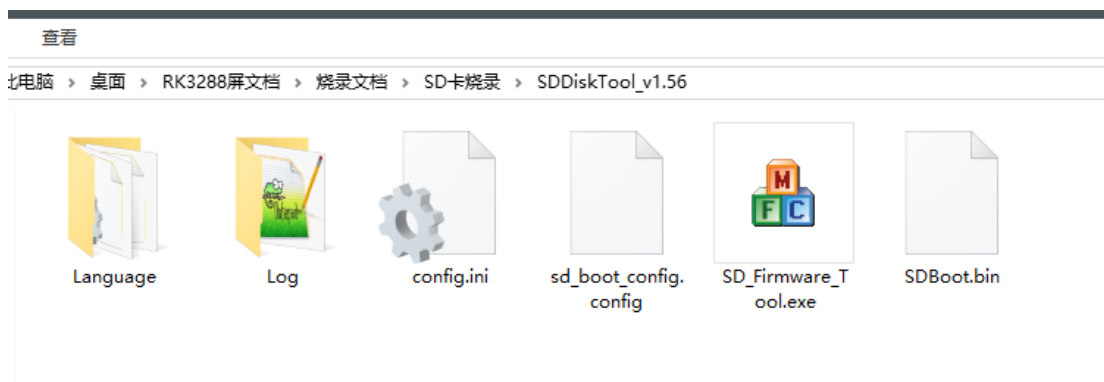
- i. When attempting to unpack a firmware within an update.img file that is currently in use by another tool, an error in unpacking the firmware may occur.
- ii. In Android 7.1, for RK3288, pre-installed APKs are placed in the vendor partition. However, in Android 8.0/8.1, they are placed in the oem partition. It is essential to ensure that the corresponding partition's "parameter.txt" contains sufficient available space. Failure to do so could result in issues during the firmware loading process, such as inability to load firmware from the oem or vendor partition due to insufficient space.
- iii. If errors arise during the flashing process of a generated update.img, consider updating the flashing tool to a newer version before attempting again.

3. Firmware Update and Burning

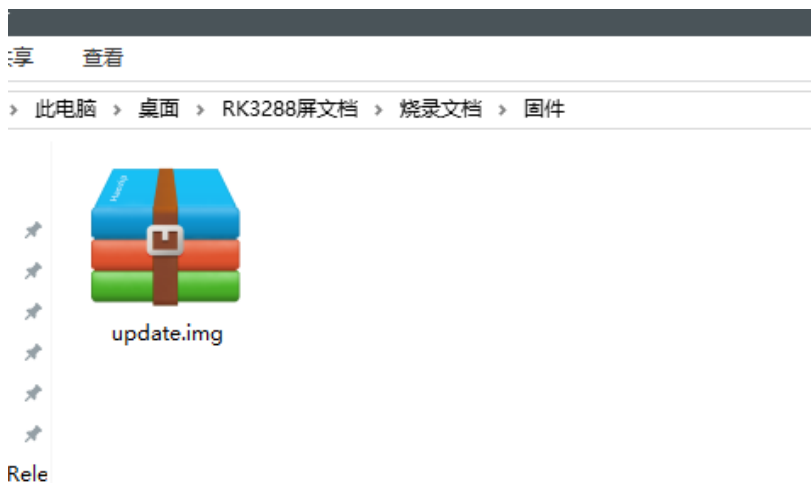
A. SD Card Flashing (SDDiskTool)

i. Tools/Resource

1.1.1 SDDiskTool - SD Card flashing software



1.1.2 Firmware for upgrading (same as used for USB flashing)

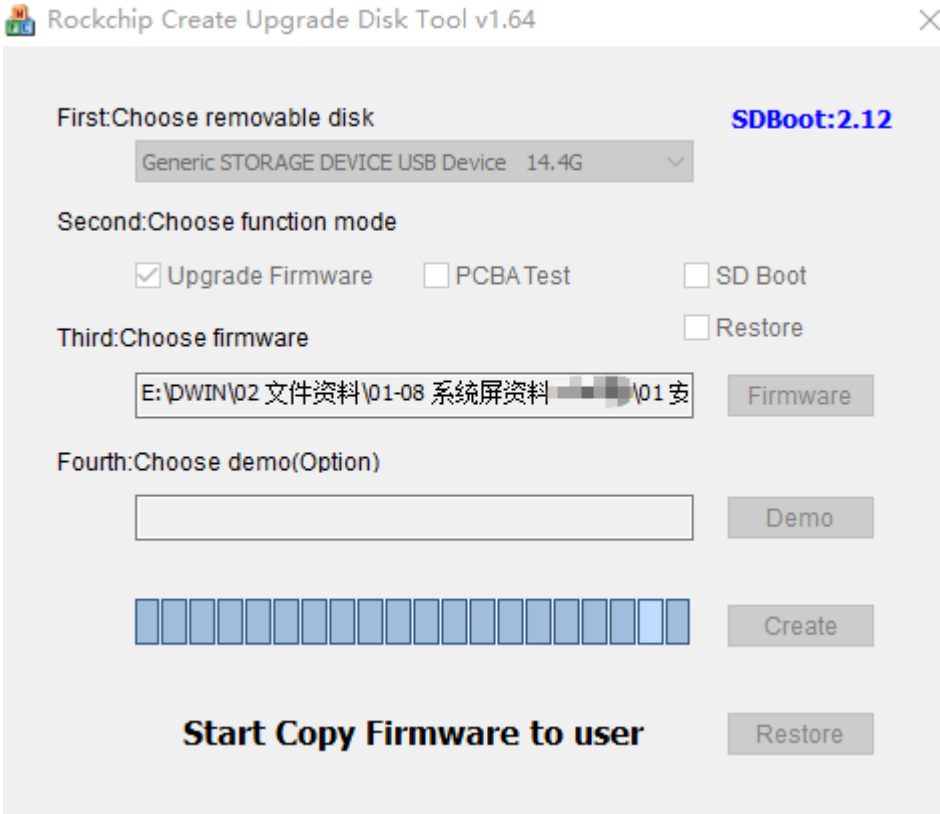




Professional. Creaditable. Successful

ii. SD Card Burning Procedure

Open the folder containing the flashing tool and double-click on "SD_Firmware_Tool.exe."



Insert the SD card into a card reader and connect it to the computer. Follow these steps:

At the first step, choose your SD card's drive letter.

At the second step, select [Upgrade Firmware].

Click [Firmware] on the right side to choose the firmware. Locate the .img file of the firmware you want to burning.

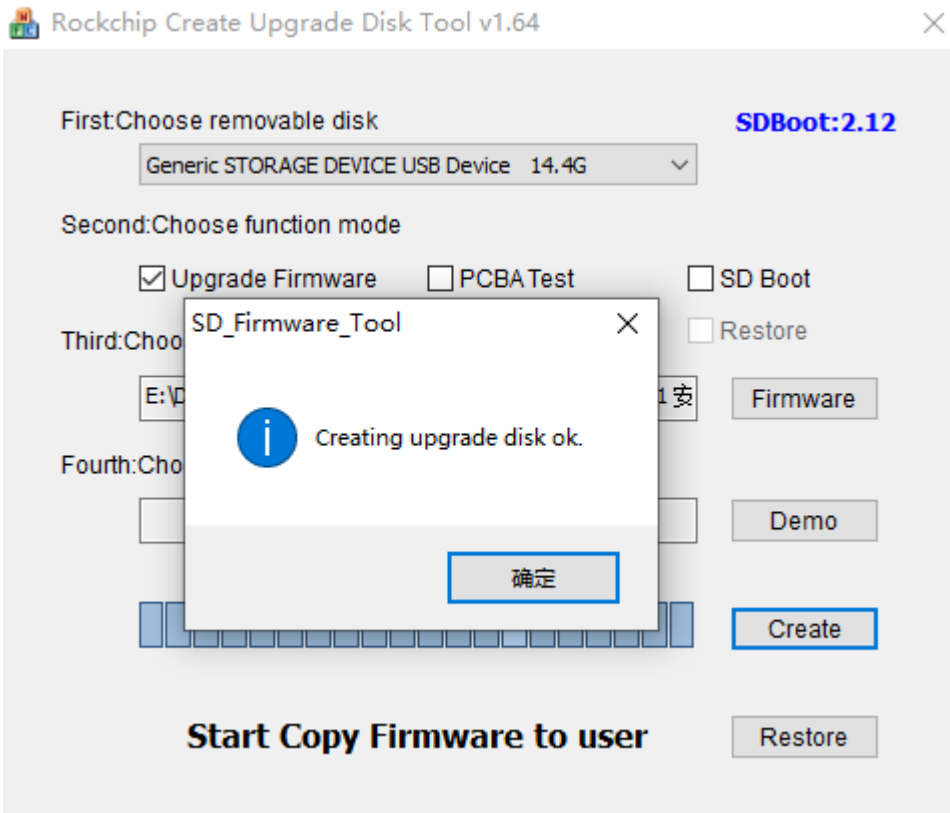
Finally, click on " Create."

Note: Creating the burning card will format the SD card, erasing all existing files. Please make sure to back up your data before proceeding.

Upon successful creation, you will receive a prompt: " creating upgrade disk ok."



Professional. Creaditable. Successful



Note: If the process fails, it's recommended to try the above steps again with a different SD card.

iii. Burning Process

1.1.1 With the device powered off, insert the SD card into the device.

1.1.2 Power on the device to initiate the flashing process.

1.1.3 After the flashing process is complete, you will receive a prompt to remove the SD card. Simply eject the SD card at this point.

B. USB Mass Production Flashing (FactoryTool)

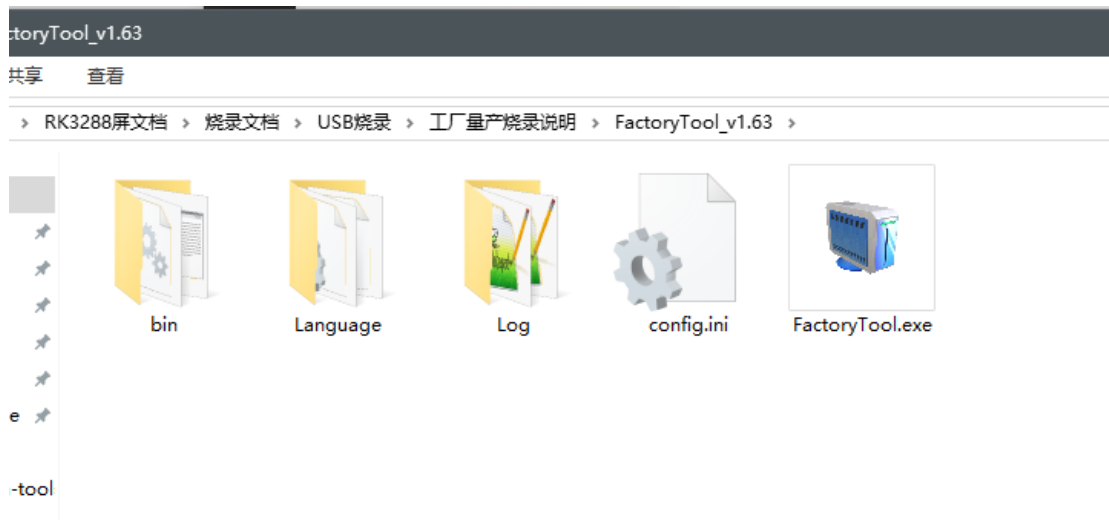
If this is the first time you're using the computer for flashing, you might need to install drivers. Please refer

Websites: www.dwin-global.com

to the "USB Driver Installation Instructions" in the USB driver directory.

i. Tools/Resource

FactoryTool - USB mass production flashing software



Firmware for upgrading (same as used for SD card flashing)

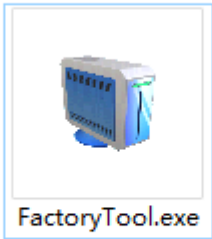
ii. USB Mass Production Burning Procedure

Open the FactoryTool.exe application for USB mass production burning.

actoryTool_v1.63



config.ini



FactoryTool.exe

Once the software is open, choose the firmware you wish to burn.

The screenshot shows the FactoryTool v1.72.2 interface. The 'Firmware' and 'Run' buttons are highlighted with a red box. The 'Device List' table is as follows:

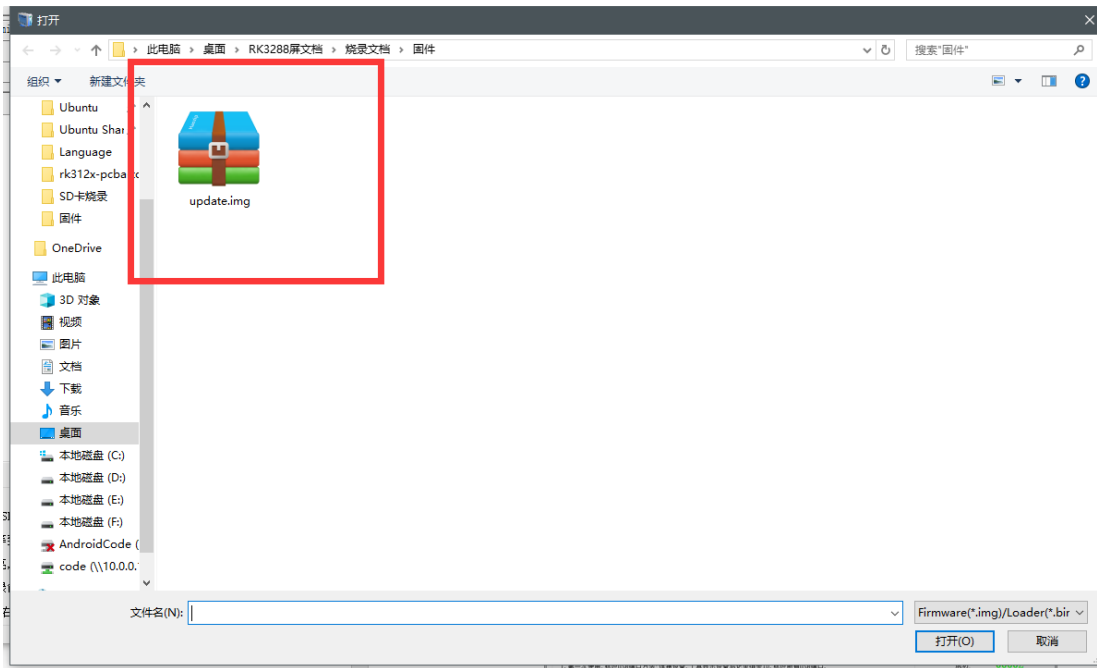
ID	Fail	Device List	Device Type	ID	Upgrade Prompt	ID	Success
		RootHub20				25	0:43
		Port[1]					
		Port[2]	Hub	2			
		Port[1]					
		Port[2]					
		Port[3]					
		Port[4]					
		Port[3]					
		Port[4]					
		Port[5]					
		Port[6]	Hub	6			
		Port[1]					
		Port[2]					
		Port[3]					
		Port[4]					
		Port[7]					
		Port[8]					

The 'Wizard' section contains the following instructions:

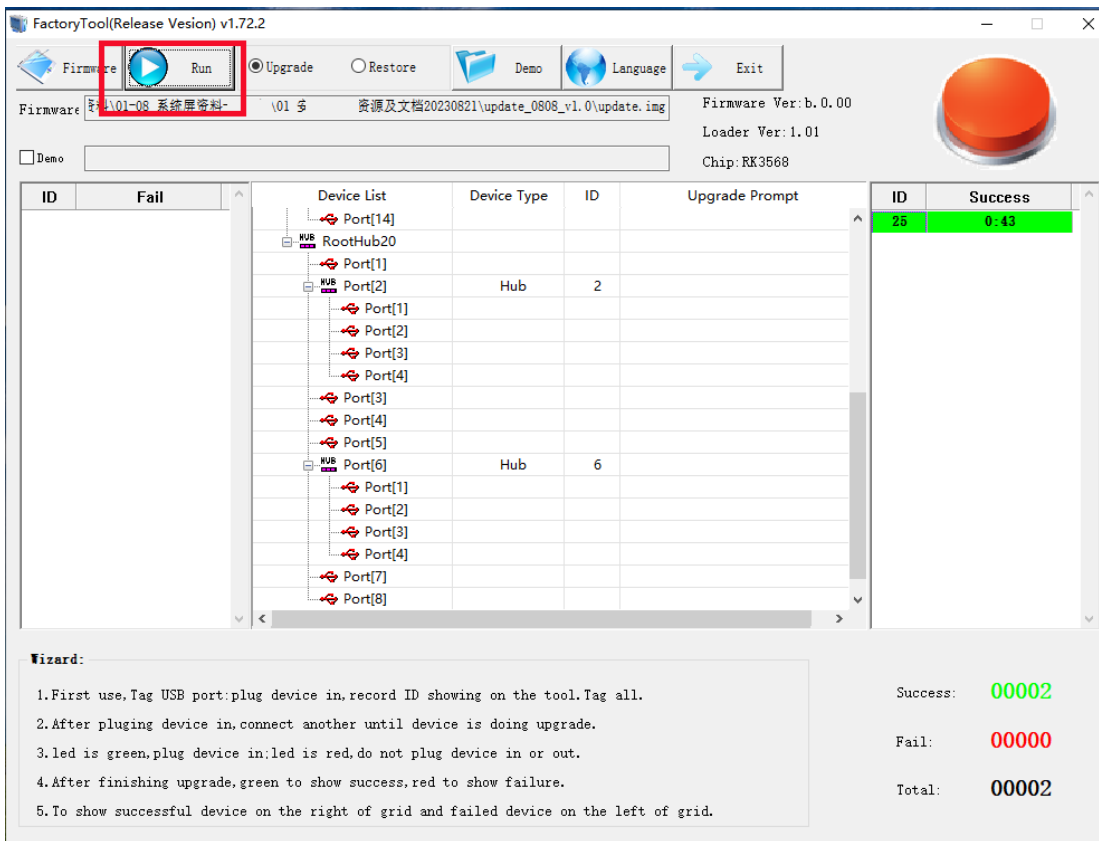
1. First use, Tag USB port: plug device in, record ID showing on the tool. Tag all.
2. After plugging device in, connect another until device is doing upgrade.
3. led is green, plug device in; led is red, do not plug device in or out.
4. After finishing upgrade, green to show success, red to show failure.
5. To show successful device on the right of grid and failed device on the left of grid.

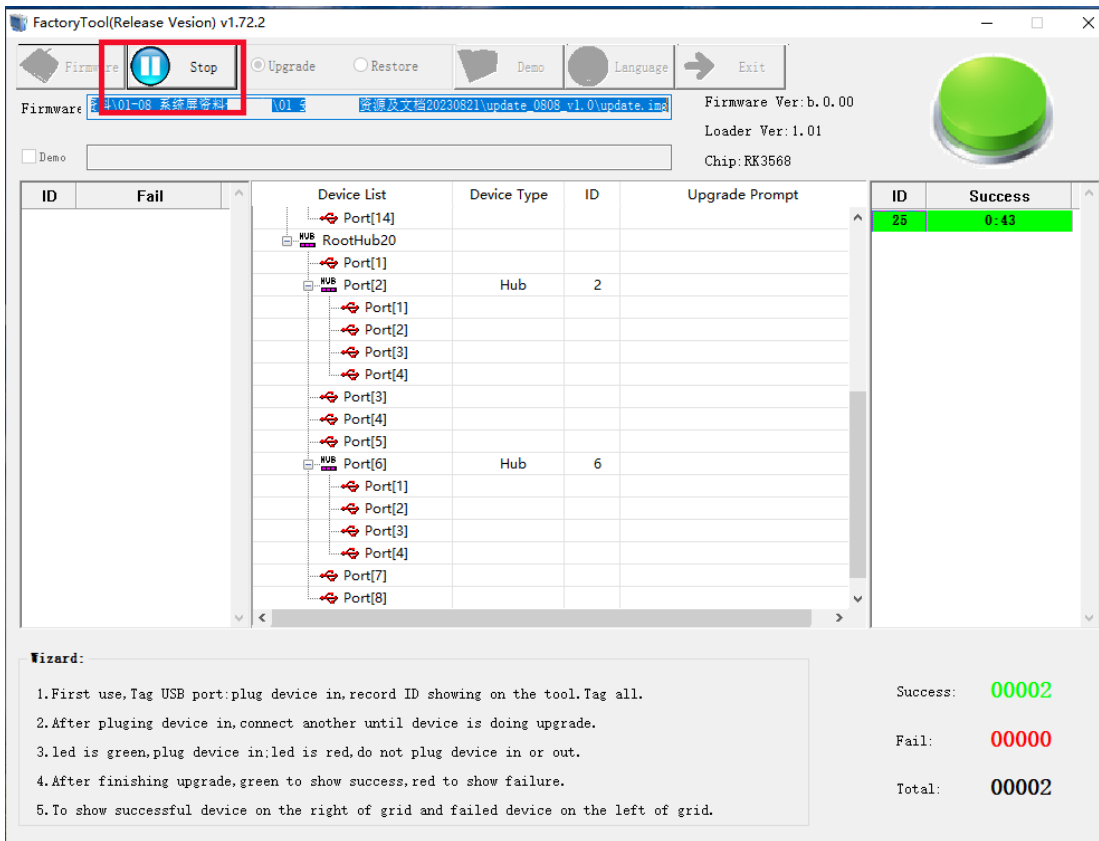
Summary statistics:

- Success: 00002
- Fail: 00000
- Total: 00002



Once you have made your selection, click on "Run."

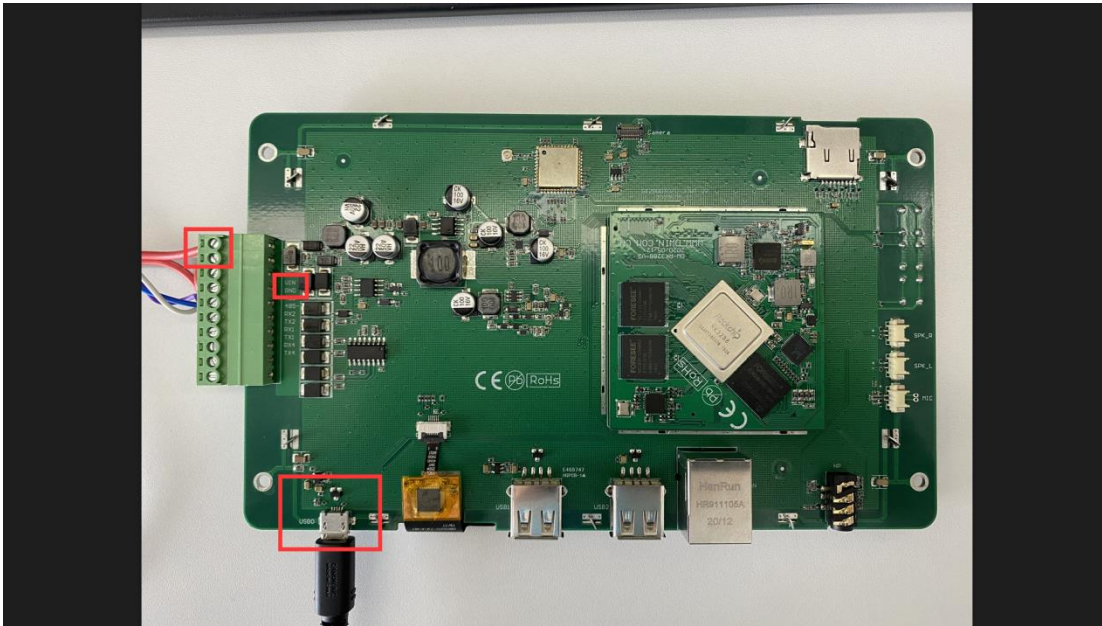




At this point, burning has been initiated.

In a power-off state, press and hold the MASROM key on the Android screen, connect the device to your PC using a USB cable and then access the power supply.





When you connect the device to your computer as outlined in step 3, the burning process will start automatically. The software will display a list of devices.

FactoryTool(Release Vesion) v1.72.2

Firmware: 4.01-08 系統屏密料 \01.3 密源及文档20230821\update_0808_v1.0\update.img Firmware Ver: b. 0.00
 Loader Ver: 1.01
 Chip: RK3568

ID	Fail	Device List	Device Type	ID	Upgrade Prompt	ID	Success
		Port[14]					
		RootHub20					
		Port[1]					
		Port[2]	Hub	2			
		Port[1]					
		Port[2]					
		Port[3]	Loader	25	Uploading firmware(2%).....		
		Port[4]					
		Port[3]					
		Port[4]					
		Port[5]					
		Port[6]	Hub	6			
		Port[1]					
		Port[2]					
		Port[3]					
		Port[4]					
		Port[7]					
		Port[8]					

Wizard:

- 1.First use,Tag USB port:plug device in,record ID showing on the tool.Tag all.
- 2.After plugging device in,connect another until device is doing upgrade.
- 3.led is green,plug device in;led is red,do not plug device in or out.
- 4.After finishing upgrade,green to show success,red to show failure.
- 5.To show successful device on the right of grid and failed device on the left of grid.

Success: 00002
 Fail: 00000
 Total: 00002



Professional. Creaditable. Successful

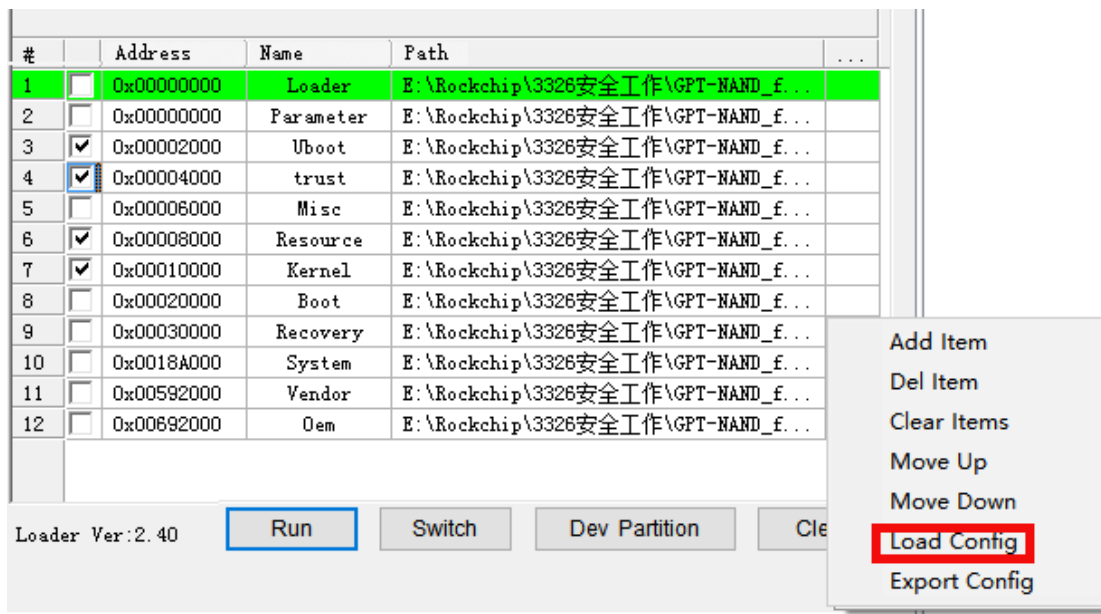
Each port corresponds to a USB port on your computer. Once the burning process is completed, the device will power on automatically. You can check the bottom right corner for a statistical summary of successful or failed burnings.

C. USB Developer Flashing Tool (RKDevTool)

Note: If this is the first time you're using the computer for flashing, you might need to install drivers. Please refer to the "USB Driver Installation Instructions" in the USB driver directory.

Key Features:

i. Import Partition Configuration



Often, the partition configuration provided with the new tool differs slightly from the project's configuration. Users can right-click in the partition configuration area and choose "Import Configuration" to load a previously saved project partition configuration file. To load a specific configuration file upon tool startup,

modify the DEFAULT_IMAGE_CONFIG entry in the Config.ini file.

```

config.ini - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
LangPath=Language\
Lang1File=Chinese.ini
Lang1FontName=宋体
Lang1FontSize=9
Lang2File=English.ini
Lang2FontName=Arial
Lang2FontSize=9
[System]
#Msc设备VendorID和ProductID, 值十六进制表示, 例:MSC_VID=0x0BB4, MSC_PID=C
MSC_VID=0x0bb4
MSC_PID=0x0c02
#指定是否支持全速usb设备, 当SUPPORTLOWUSB=TRUE时, 增加全速usb设备支持, 默
备
SUPPORTLOWUSB=true
MSC_TIMEOUT=30
ROCKUSB_TIMEOUT=30
#指定启动时加载的镜像配置文件, 默认为config.cfg
DEFAULT_IMAGE_CONFIG=config.cfg
#下载Image镜像后是否进行设备重启
RESET_AFTER_DOWNLOAD=true
#当设置FW_NOT_CHECK=TRUE时, 固件加载时不进行完整性校验
FW_NOT_CHECK=true

```

ii. Export Partition Configuration

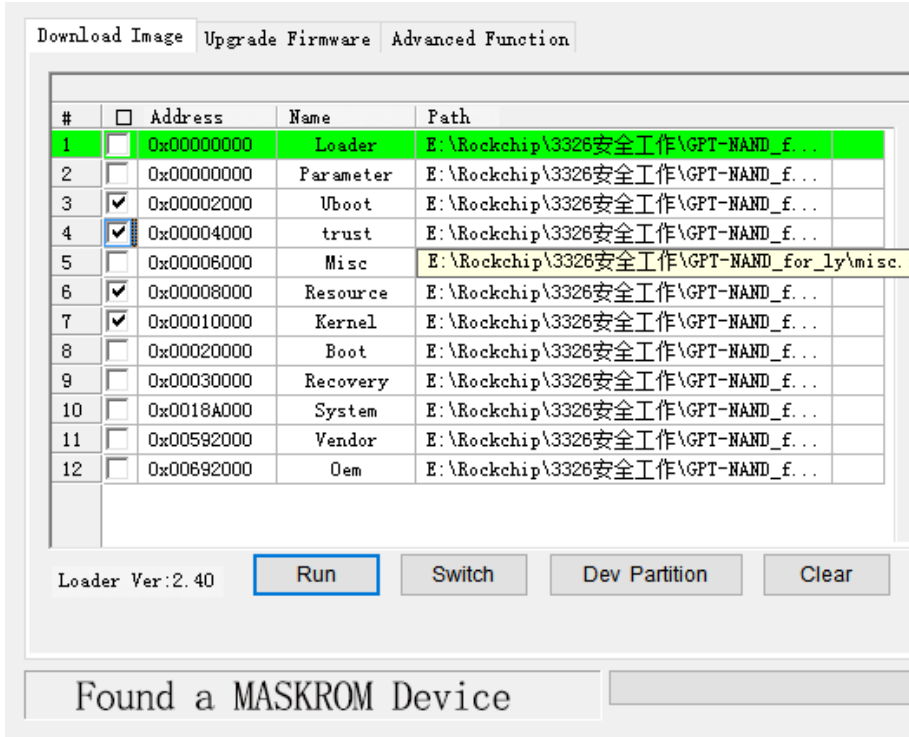
#	Address	Name	Path
1	<input type="checkbox"/> 0x00000000	Loader	E:\Rockchip\3326安全工作\GPT-NAND_f...
2	<input type="checkbox"/> 0x00000000	Parameter	E:\Rockchip\3326安全工作\GPT-NAND_f...
3	<input checked="" type="checkbox"/> 0x00002000	Uboot	E:\Rockchip\3326安全工作\GPT-NAND_f...
4	<input checked="" type="checkbox"/> 0x00004000	trust	E:\Rockchip\3326安全工作\GPT-NAND_f...
5	<input type="checkbox"/> 0x00006000	Misc	E:\Rockchip\3326安全工作\GPT-NAND_f...
6	<input checked="" type="checkbox"/> 0x00008000	Resource	E:\Rockchip\3326安全工作\GPT-NAND_f...
7	<input checked="" type="checkbox"/> 0x00010000	Kernel	E:\Rockchip\3326安全工作\GPT-NAND_f...
8	<input type="checkbox"/> 0x00020000	Boot	E:\Rockchip\3326安全工作\GPT-NAND_f...
9	<input type="checkbox"/> 0x00030000	Recovery	E:\Rockchip\3326安全工作\GPT-NAND_f...
10	<input type="checkbox"/> 0x0018A000	System	E:\Rockchip\3326安全工作\GPT-NAND_f...
11	<input type="checkbox"/> 0x00592000	Vendor	E:\Rockchip\3326安全工作\GPT-NAND_f...
12	<input type="checkbox"/> 0x00692000	Oem	E:\Rockchip\3326安全工作\GPT-NAND_f...

Loader Ver: 2.40 Run Switch Dev Partition Cle

- Add Item
- Del Item
- Clear Items
- Move Up
- Move Down
- Load Config
- Export Config

After configuring partitions, users can save their partition setup by exporting it. Click "Export Configuration" in the blank area of the partition configuration zone. Specify a file name and directory for saving. If you replace the config.cfg file in the tool directory, the saved configuration will load when the tool starts.

iii. Burning One or Multiple Partition Images



Steps:

- Connect the device to the PC in Loader or Maskrom mode. In Maskrom mode, select the "Loader" download option.
- Check the desired flashing items. Confirm that the addresses before the flashing items are correct. If an address is zero, load partition information from the "Parameter" partition table.
- Click "Execute."

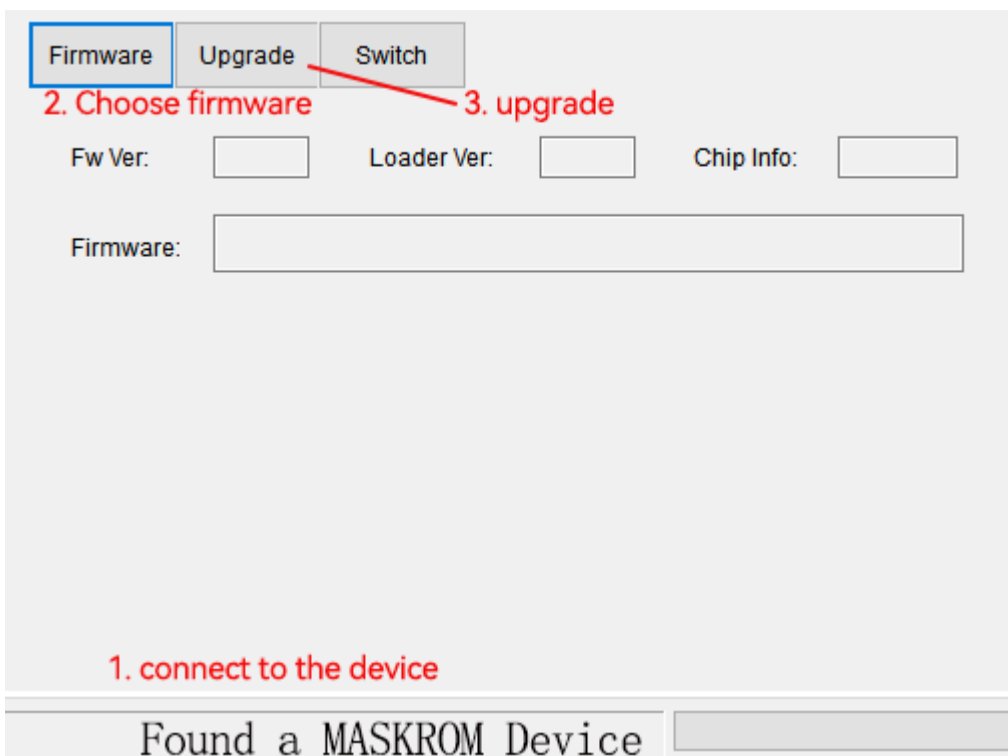
iv. Switching

When Found a MASKROM Device appears, execute "Switch" to enter Loader or Maskrom mode.

v. Device Partition Table

To use the device's current partition table for flashing, click " Dev Partition". At this point, the partition table is read from the device side. Partition information will be parsed and loaded.

vi. Burning update.img



Note:

Flashing update.img is only possible in loader and maskrom modes; other modes require switching first.

The firmware can be an update.img firmware or a loader file.

vii. Erase Flash

Executing the "Erase Flash" function erases all blocks of the Flash, including the system blocks before the



Professional. Creaditable. Successful

firmware area. If repeated firmware upgrades fail, try performing "Erase Flash" before upgrading again.

viii. Unpack update.img

The screenshot shows the 'Advanced Function' tab of the DWIN tool. It features two rows of controls: 'Boot' and 'Firmware'. Each row has a text input field, a browse button (three dots), and an action button. The 'Unpack' button for the 'Firmware' row is highlighted with a red rectangular border.

Unpacked files will be saved in the "output" directory under the tool's location.

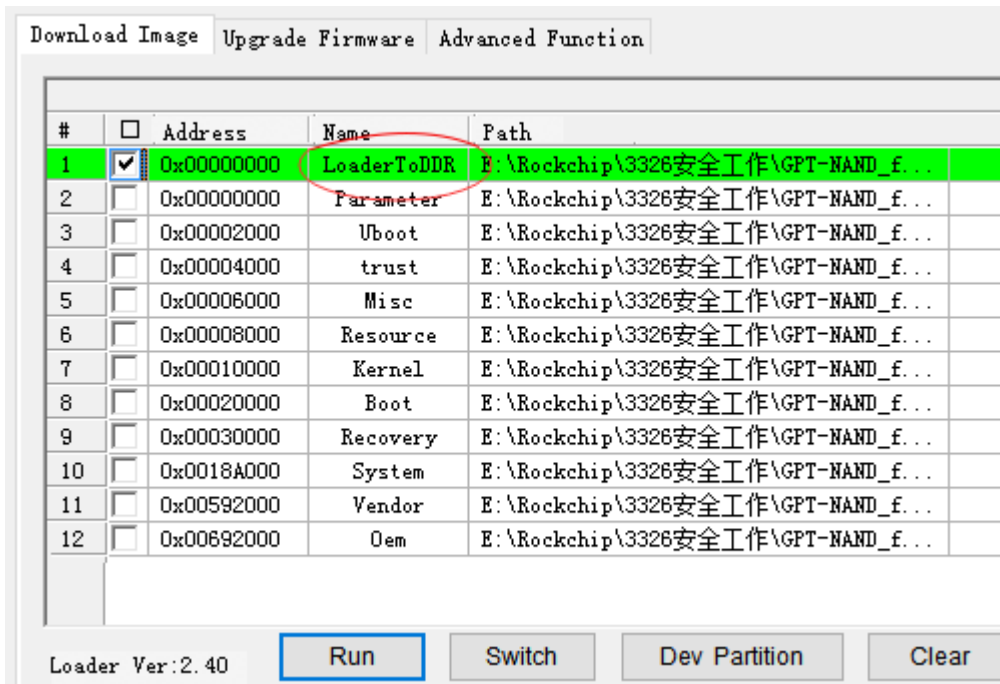
ix. Download Boot

Downloading the boot initiates DDR initialization and loads the usbplug code into DDR through the loader.

You can perform this by the following two ways.

The screenshot shows the 'Download Image' tab of the DWIN tool. The 'Boot' field contains the file path 'ry_git\windows\androidTool\bin\RK322HBoot.bin', which is highlighted in blue. The 'Download' button is highlighted with a red rectangular border. The 'Firmware' field is empty.

double-clicking "Loader" and selecting "LoaderToDDR" from the dropdown list.



X. Download GPT

A GPT partition can be in the format of a parameter file or a partition_table.img file. If it's a parameter file, note:

The parameter file should have the attribute "TYPE:GPT."

```

: FIRMWARE_VER:7.1
: MACHINE_MODEL:RK3126c
: MACHINE_ID:007
: MANUFACTURER:rk3126c
: MAGIC: 0x5041524B
: ATAG: 0x00200800
: MACHINE: 3126c
: CHECK_MASK: 0x80
: PWR_HLD: 0,0,A,0,1
: TYPE: GPT
: CMDLINE:mtddparts=rk29xxnand:0x00002000@0x00004000(uboot),0x00002000@0x0000

```

```

6000 (atf), 0x00038000@0x00008000 (boot:bootable), -@0x0040000 (rootfs:grow)

```

The last partition should use the "grow" attribute to allocate any remaining space to that partition.

#	<input type="checkbox"/>	Address	Name	Path
1	<input checked="" type="checkbox"/>	0x00000000	LoaderToDDR	E:\Rockchip\3326安全工作\GPT-NAND_f...
2	<input type="checkbox"/>	0x00000000	Parameter	E:\Rockchip\3326安全工作\GPT-NAND_for_ly\gpt.txt
3	<input type="checkbox"/>	0x00004000	Uboot	E:\Rockchip\3326安全工作\GPT-NAND_f...
4	<input type="checkbox"/>	0x00006000	trust	E:\Rockchip\3326安全工作\GPT-NAND_f...
5	<input type="checkbox"/>	0x00008000	Misc	E:\Rockchip\3326安全工作\GPT-NAND_f...
6	<input type="checkbox"/>	0x0000A000	Resource	E:\Rockchip\3326安全工作\GPT-NAND_f...
7	<input type="checkbox"/>	0x00012000	Kernel	E:\Rockchip\3326安全工作\GPT-NAND_f...
8	<input type="checkbox"/>	0x00022000	Boot	E:\Rockchip\3326安全工作\GPT-NAND_f...
9	<input type="checkbox"/>	0x00032000	Recovery	E:\Rockchip\3326安全工作\GPT-NAND_f...
10	<input type="checkbox"/>	0x0018C000	System	E:\Rockchip\3326安全工作\GPT-NAND_f...
11	<input type="checkbox"/>	0x00594000	Vendor	E:\Rockchip\3326安全工作\GPT-NAND_f...
12	<input type="checkbox"/>	0x00614000	Oem	E:\Rockchip\3326安全工作\GPT-NAND_f...

Loader: 2. 40 **Run** Switch Dev Partition Clear

xi. Read Device Extended Functions

Advanced Function

Download

Unpack

ReadChipInfo **ReadCapability**

Go Maskrom Switch Storage

ExportComLog t Selected Stora

EraseAll

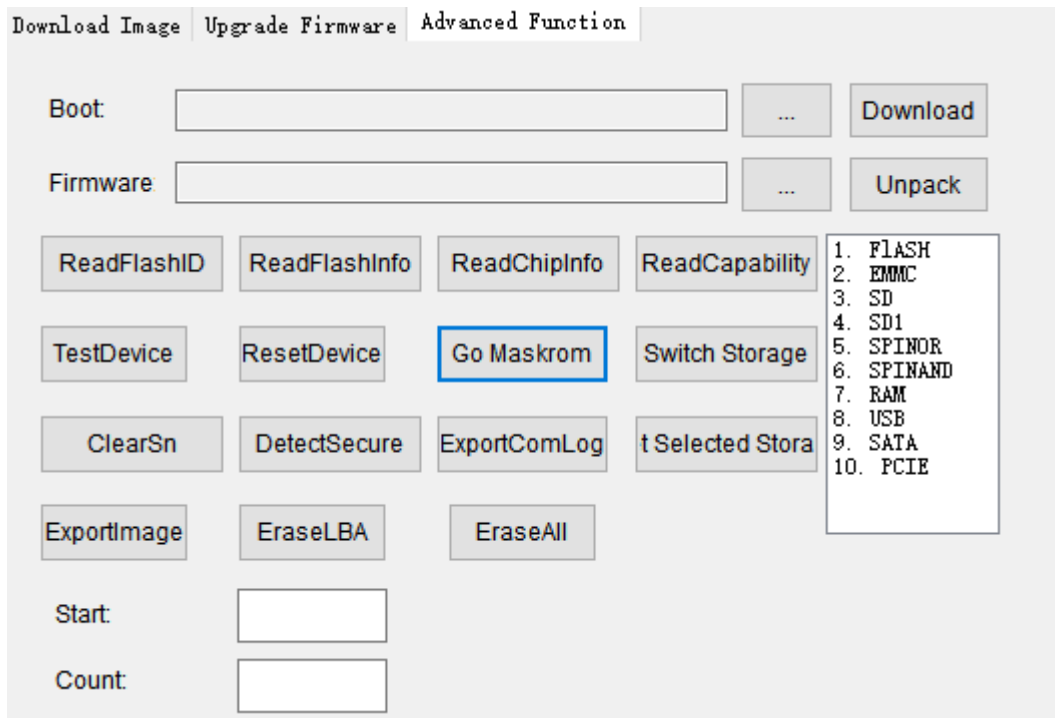
```

获取Capability开始
Capability: 03 00 00 00 00 00 00 00
DirectLBA: Enable
获取Capability成功
          
```

1. FLASH
2. EMMC
3. SD
4. SD1
5. SPINOR
6. SPINAND
7. RAM
8. USB
9. SATA
10. PCIE

For GPT burning, the "DirectLBA" in the extended functions must be enabled; otherwise, the tool will prompt that the "current device doesn't support GPT."

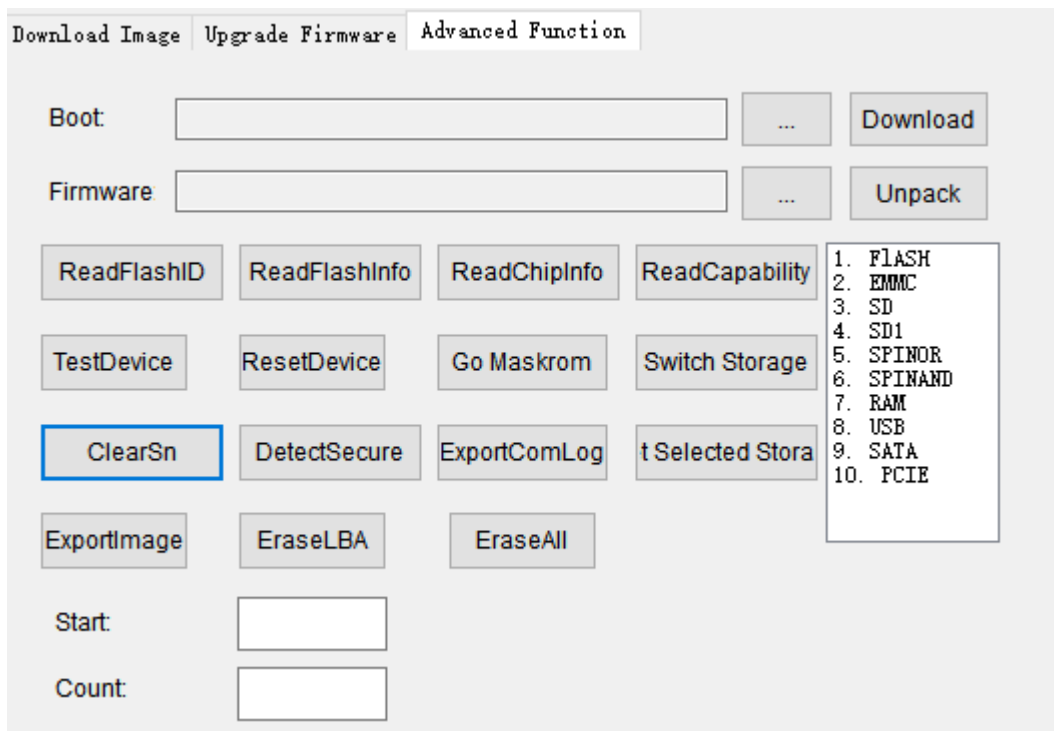
xii. Enter Maskrom



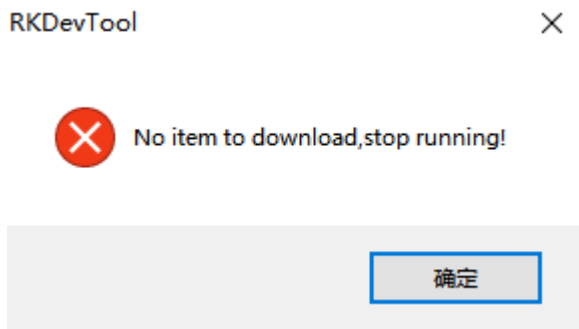
When the device is in Loader mode, you can use this function to switch the device to Maskrom mode.

xiii. Clear Serial Number

If a serial number was written using a unified dynamic library tool, it can be cleared using this function.



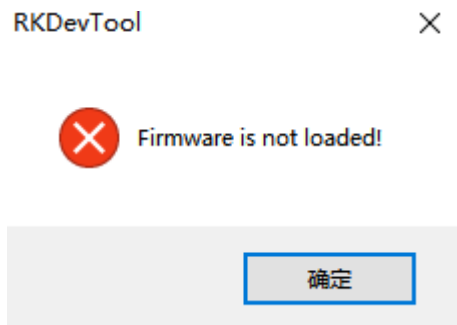
Common Issues:



i. Missing Download Items

As shown in the image above, it indicates that the download item for the kernel partition doesn't exist.

ii. Failed to Load Firmware



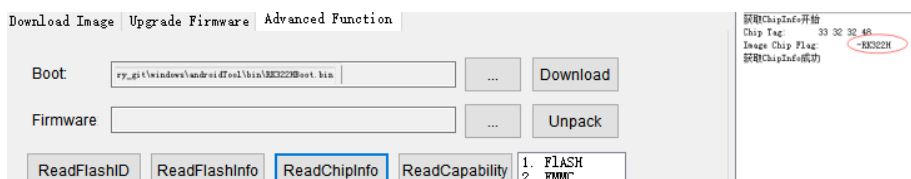
Check the following issues:

- Whether the firmware is being used by other programs.
- Incorrect firmware format.
- Corrupted firmware.

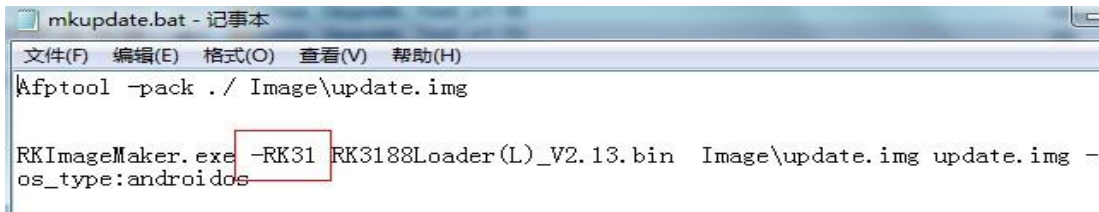
iii. Chip Verification Failed

This indicates that the loader or firmware doesn't match the device.

Use the "Read Chip Information" function to confirm the chip identifier.



Make sure that the chip identifier is set the same as what's displayed in "Image Chip Flag" when packaging firmware or loader.



```
mkupdate.bat - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
Afptool -pack ./ Image\update.img

RKImageMaker.exe -RK31 RK3188Loader(L)_V2.13.bin Image\update.img update.img -
os_type:androidos
```

iv. Failed to Download Boot

Check if DDR is properly soldered.

Ensure that the loader matches the device.

v. Failed to Download Firmware or Partition Image

Update Rockusb driver or the burning tool.

Use a shielded USB cable and connect to rear USB ports.

Check if the flash is loose, damaged, or unsupported.

vi. Failed to Verify Firmware or Partition Image

Confirm that the sizes of partitions in the parameter file can accommodate the corresponding image files.

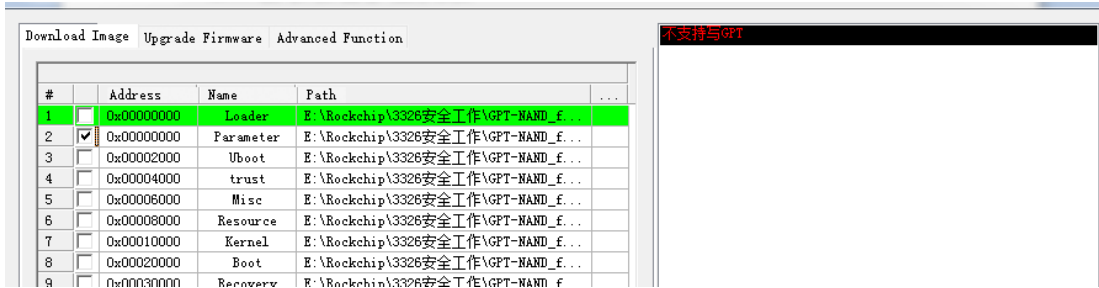
If there are issues with Flash software mapping, try erasing the flash first.

Check the stability of the DDR device.



Professional. Creaditable. Successful

vii. GPT Writing Not Supported



Use the "Read Capability" function to confirm if the current device supports DirectLBA functionality.

Writing GPT requires DirectLBA functionality to be enabled.

Important Notes:

- When using the tool in a non-Chinese operating system, ensure that the tool's directory path contains only English characters.
- On Windows 7 and Vista systems, right-click and run the program with administrator privileges.
- After modifying the Config.ini configuration file, you need to restart the tool for the changes to take effect.

4. Setting Up and Using the Android Studio Development Environment

A. Tools/Resource



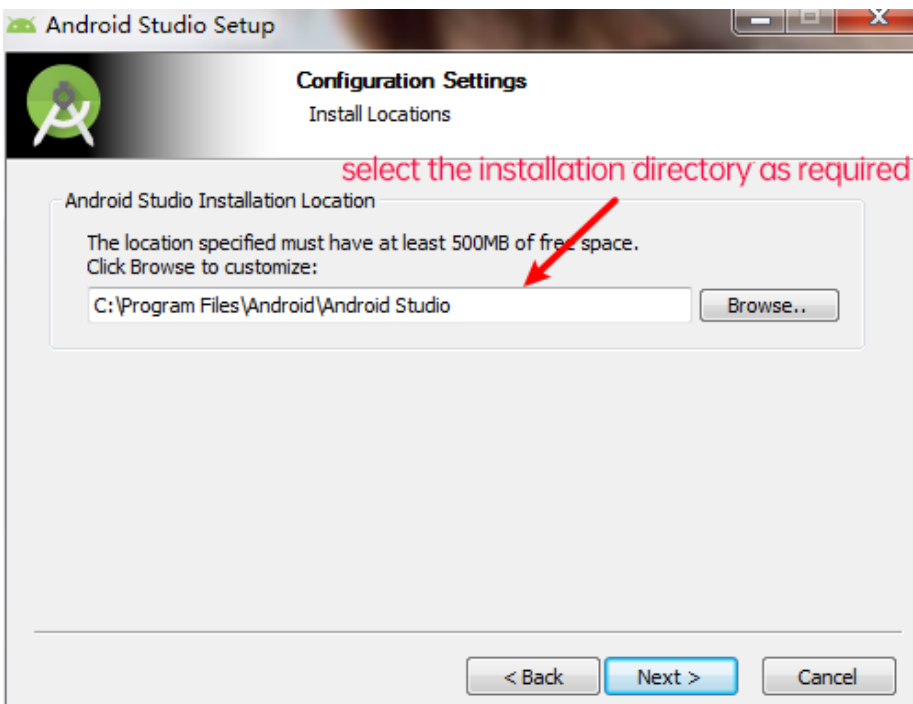
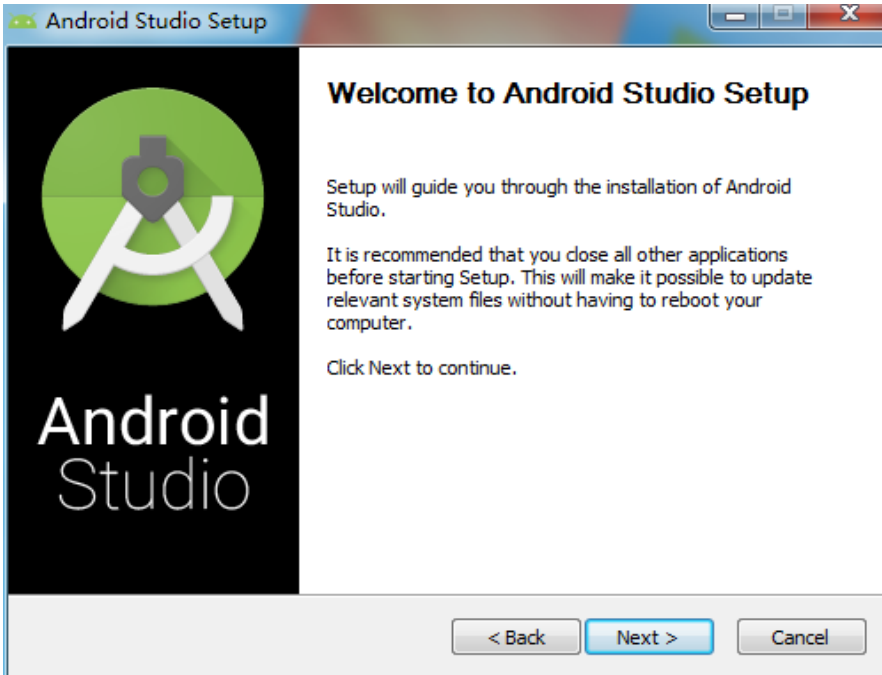
The installation programs can be downloaded from the following link:

<https://developer.android.google.cn/studio/archive>

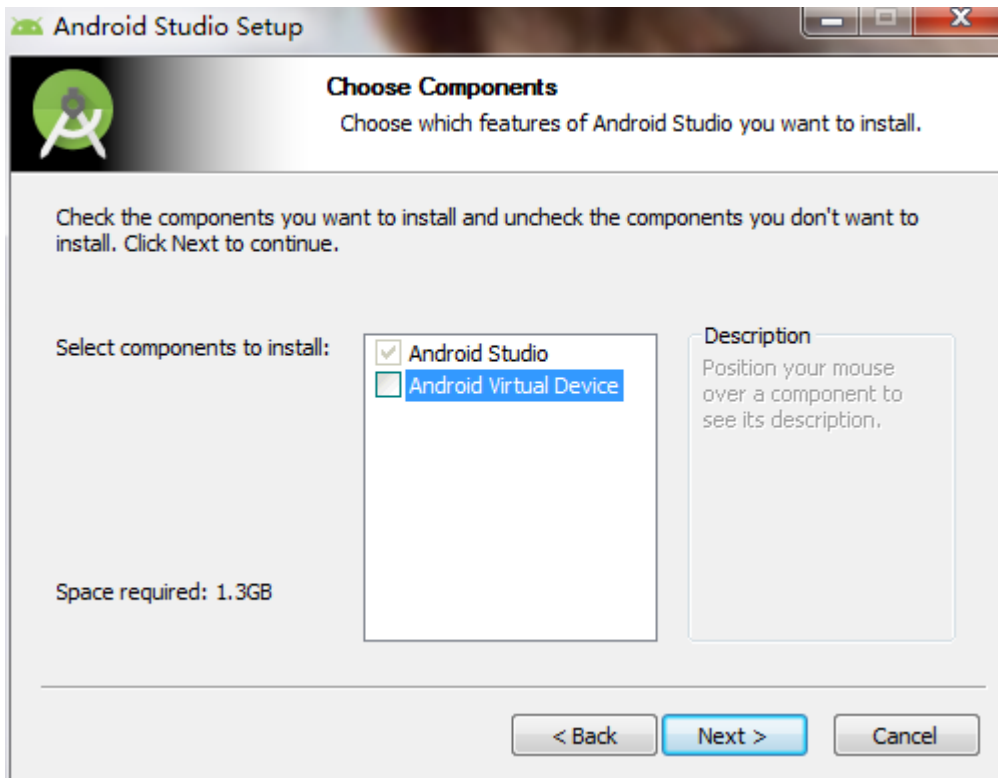
(Please download Android Studio 3.6 or later versions)

B. Software Installation

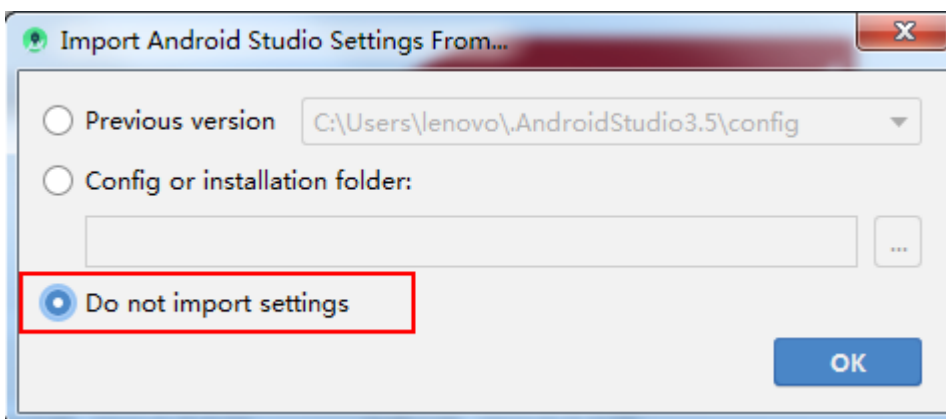
- i. Double-click the installation package to start the installation.



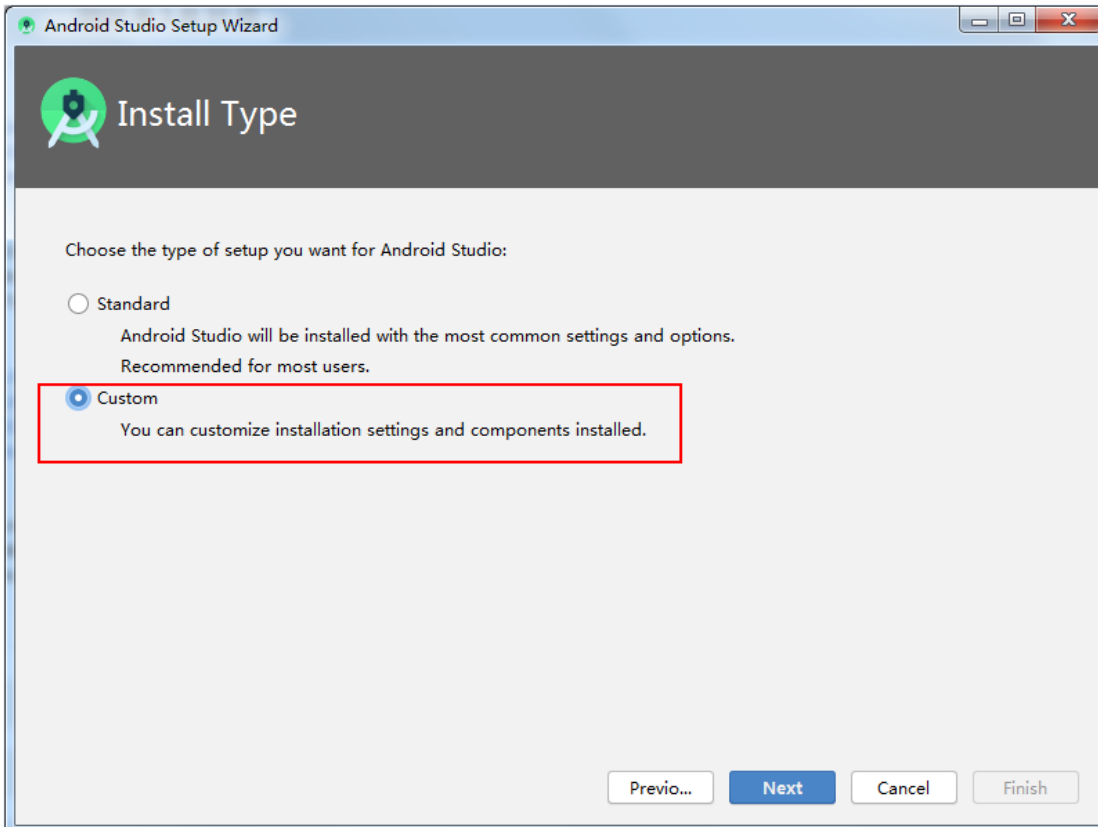
Choose whether to install AVD (Android Virtual Device) based on your needs.



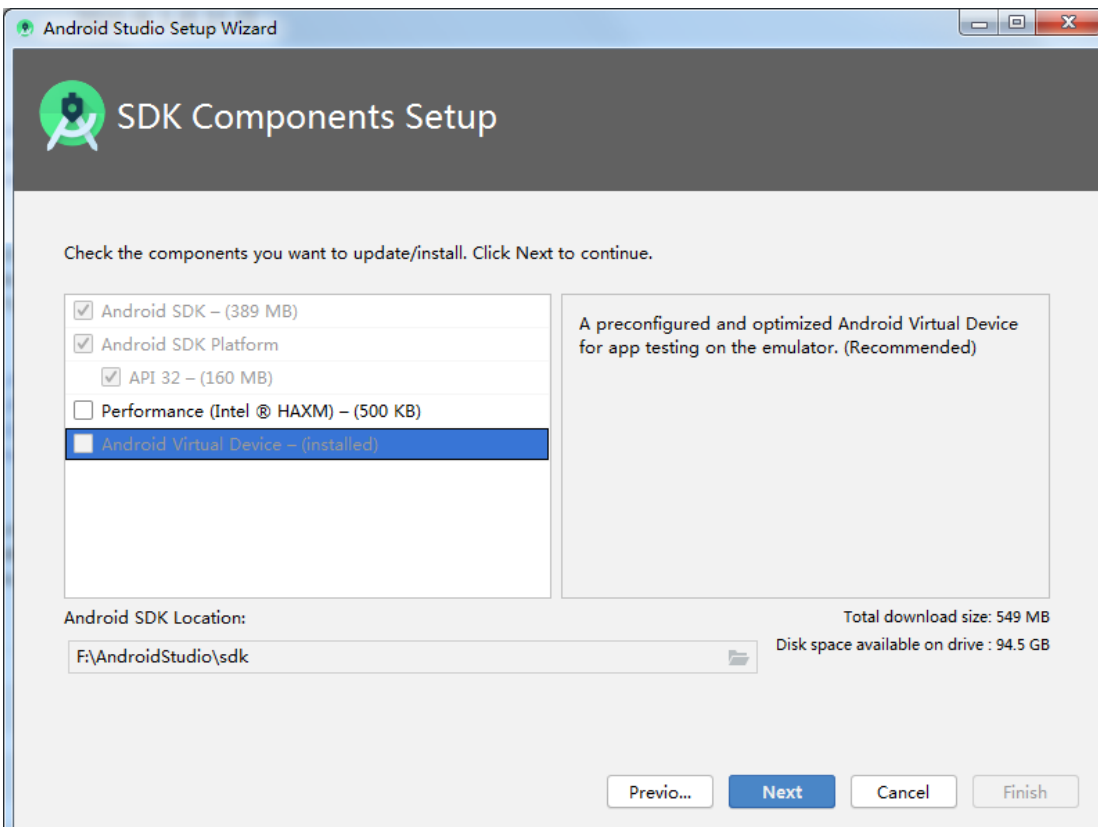
For the first-time use of Android Studio, it's recommended not to import settings.



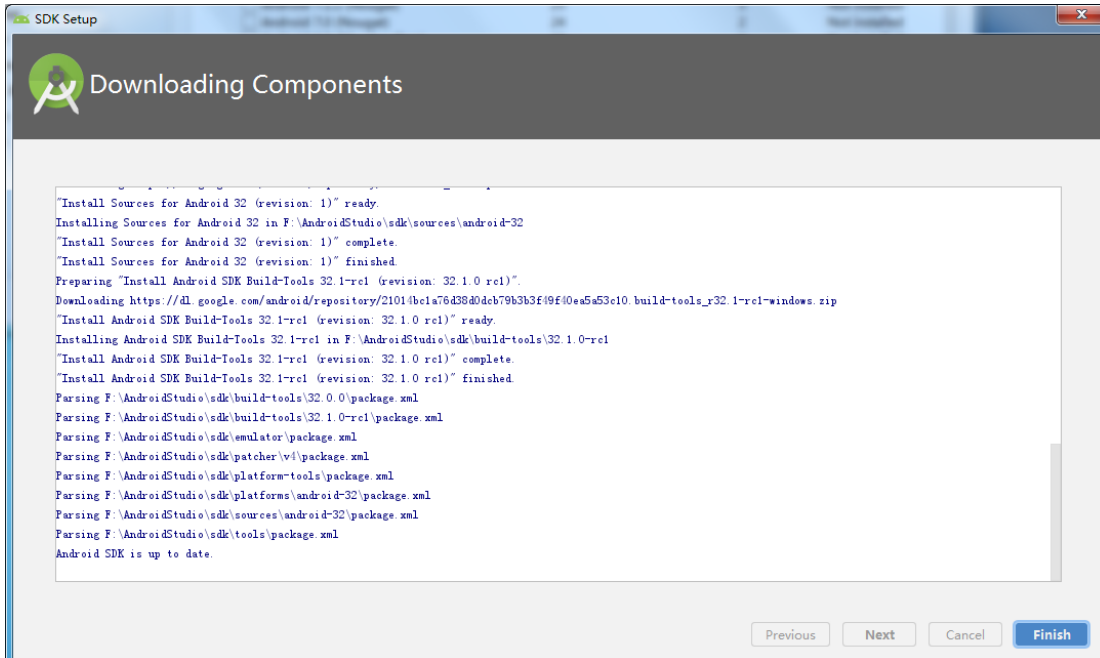
Select the installation type. Choose custom installation.



Select the location for SDK installation.



Wait for the installation to complete.



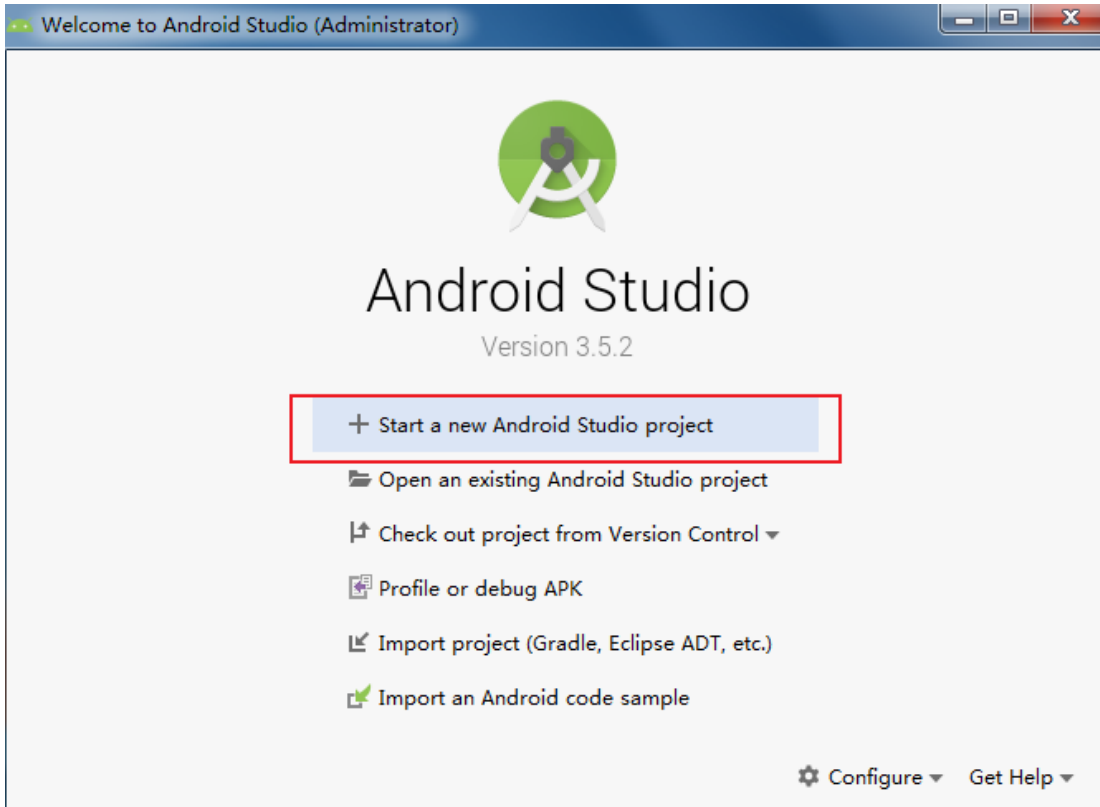
ii. Open the installed Android Studio and create a new project.

The first time you open it, you might encounter an error as shown below. Choose [Cancel] to close it.

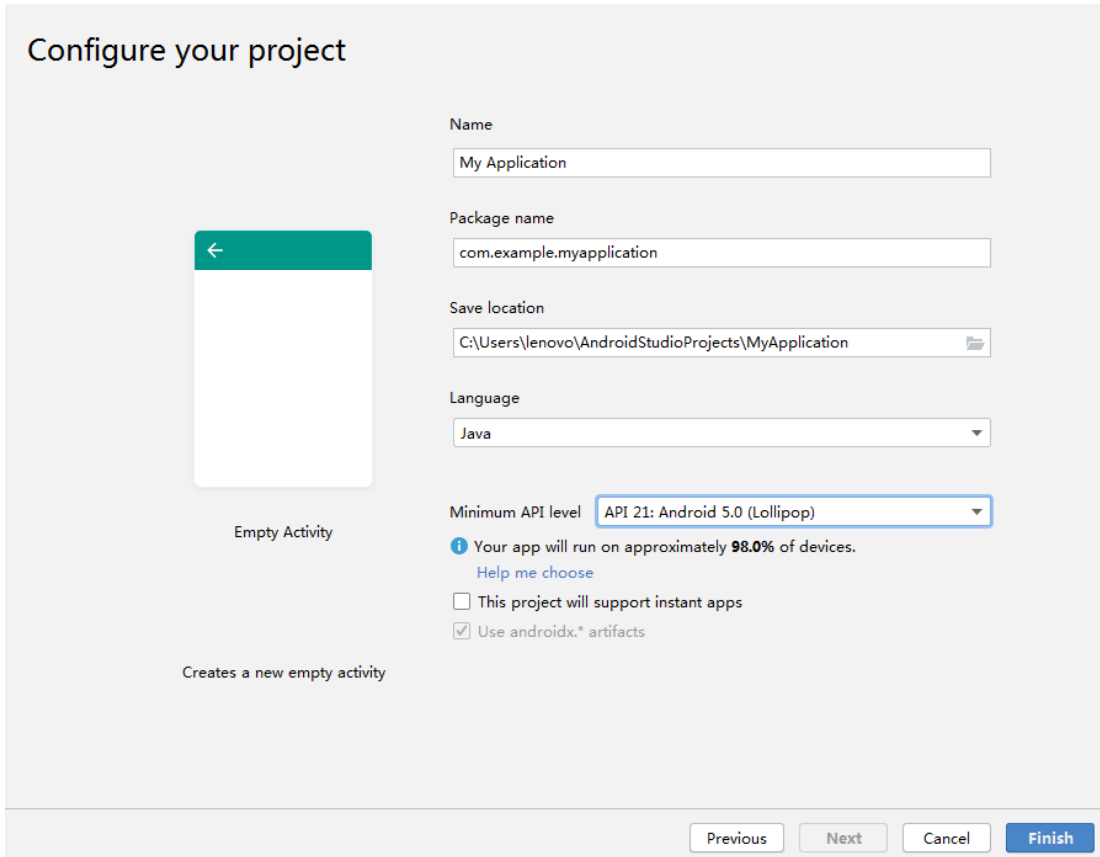


For the first use, Android Studio will perform downloads and automatic environment configuration. Just wait for the process to finish.

C. Creating a New Project



- i. Choose "Start a new Android Studio project."
- ii. Fill in the project name, package name, set the storage location, and select the programming language. In this case, we'll use Java and set the minimum compatible Android version to 5.0.



Configure your project

Name: My Application

Package name: com.example.myapplication

Save location: C:\Users\lenovo\AndroidStudioProjects\MyApplication

Language: Java

Minimum API level: API 21: Android 5.0 (Lollipop)

Your app will run on approximately 98.0% of devices.
[Help me choose](#)

This project will support instant apps

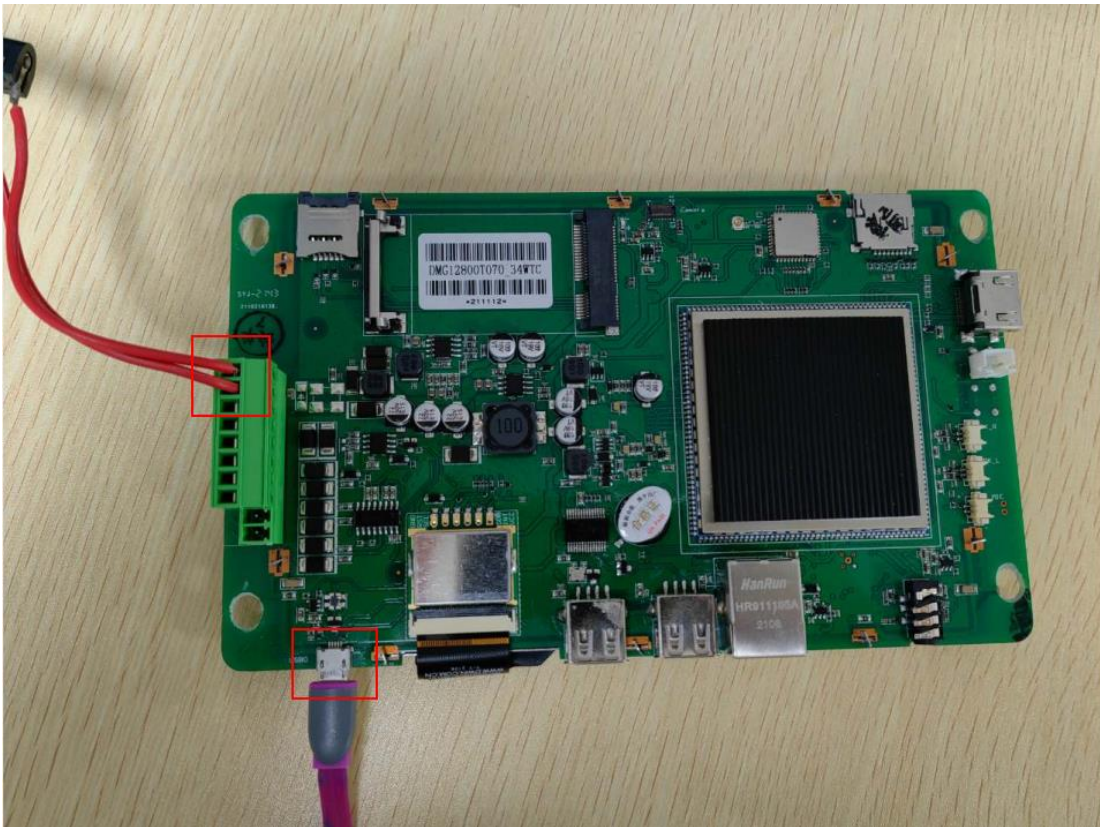
Use android.* artifacts

Creates a new empty activity

Previous Next Cancel Finish

- iii. Build and Run Your First Project

Make sure to connect your Android device to the PC before selecting debug and run.

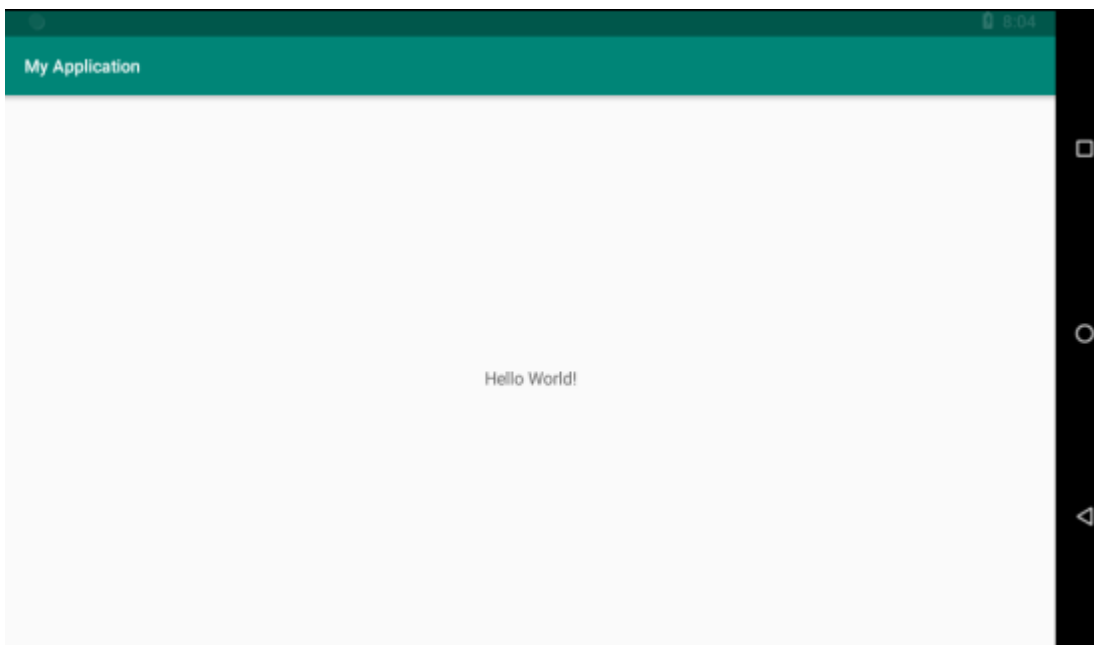
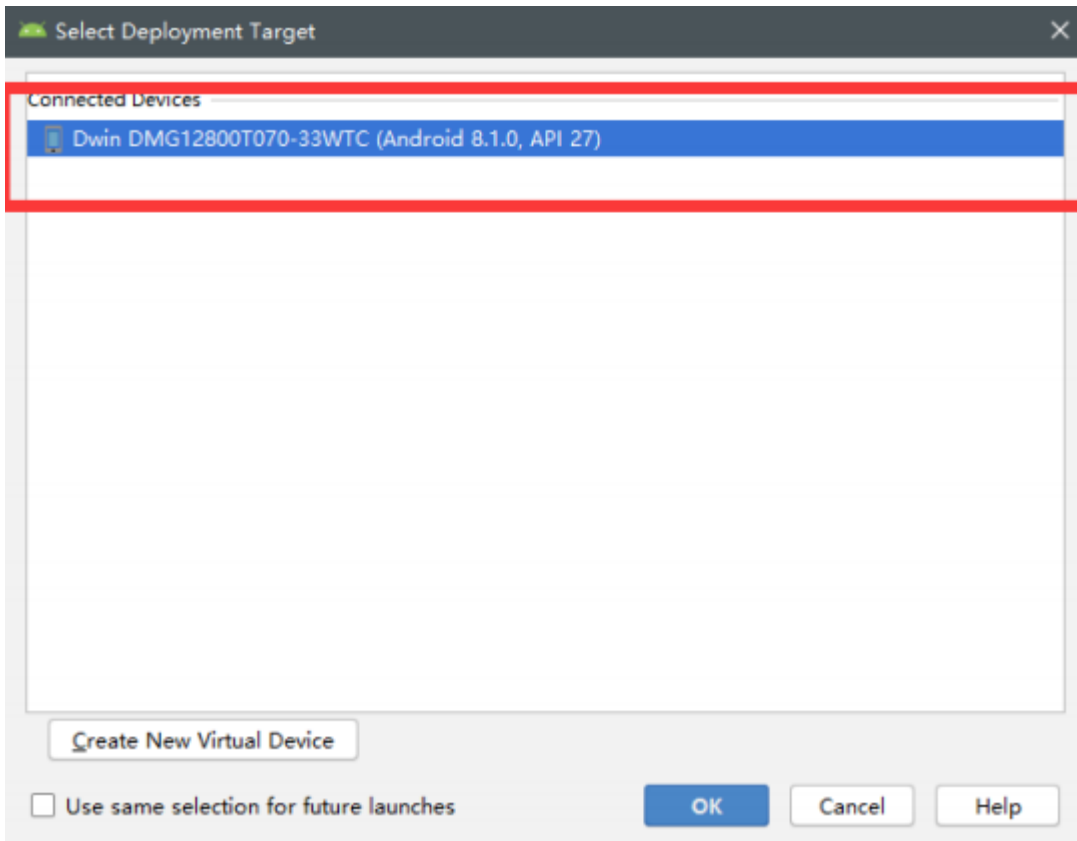


As shown, connect the Android device to the computer via USB and power on the device. (Different Android devices might have varying layouts; the image below is for reference only.)

Click on the DWIN screen in the designated area, then click "Run." The example program will run as shown below.



Professional. Creaditable. Successful



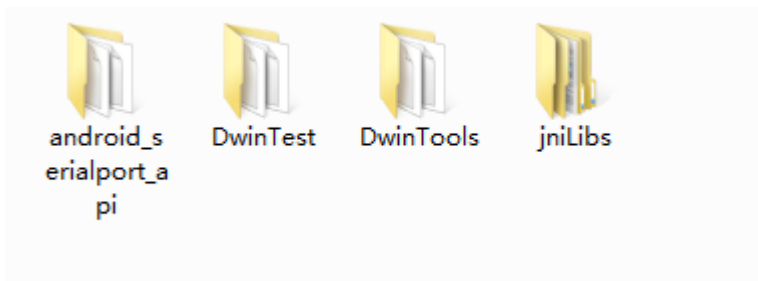


Professional. Creaditable. Successful

5. Using DWIN Android Screen Serial Port Tools

A. DWIN Android Tools Package Instruction

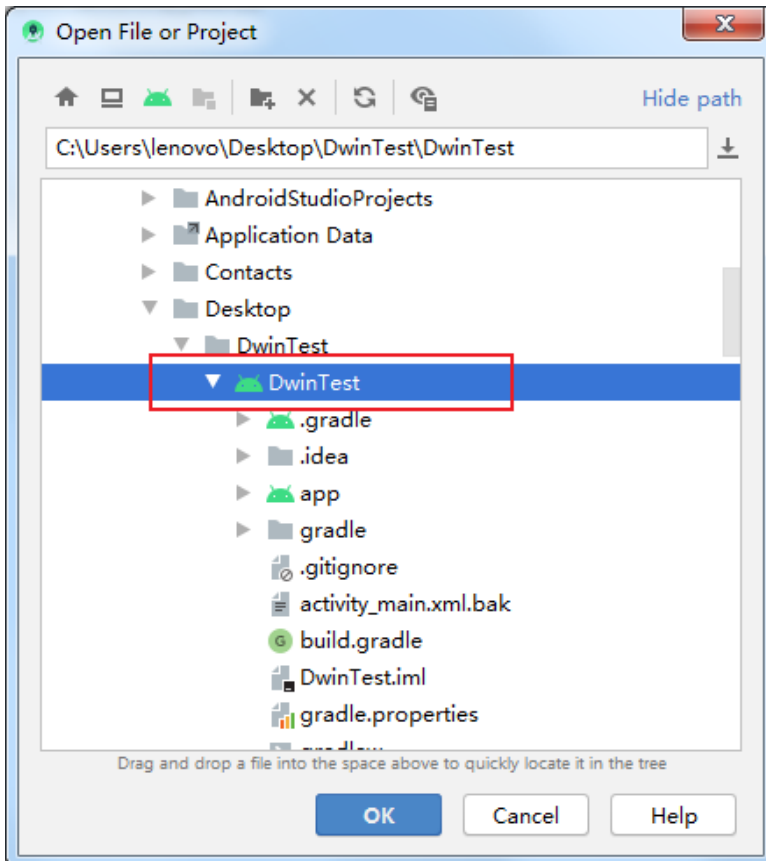
DWIN provides an Android testing software source code and a Java library file for serial communication. The provided files include:



- jniLibs: Pre-packaged shared object libraries (so files)
- android_serialport_api: Library utility classes for calling
- DwinTools: Complete project source code for Dwin tools
- DwinTest: Dwin test software

B. Importing Dwin Test Project Source Code into Android Studio

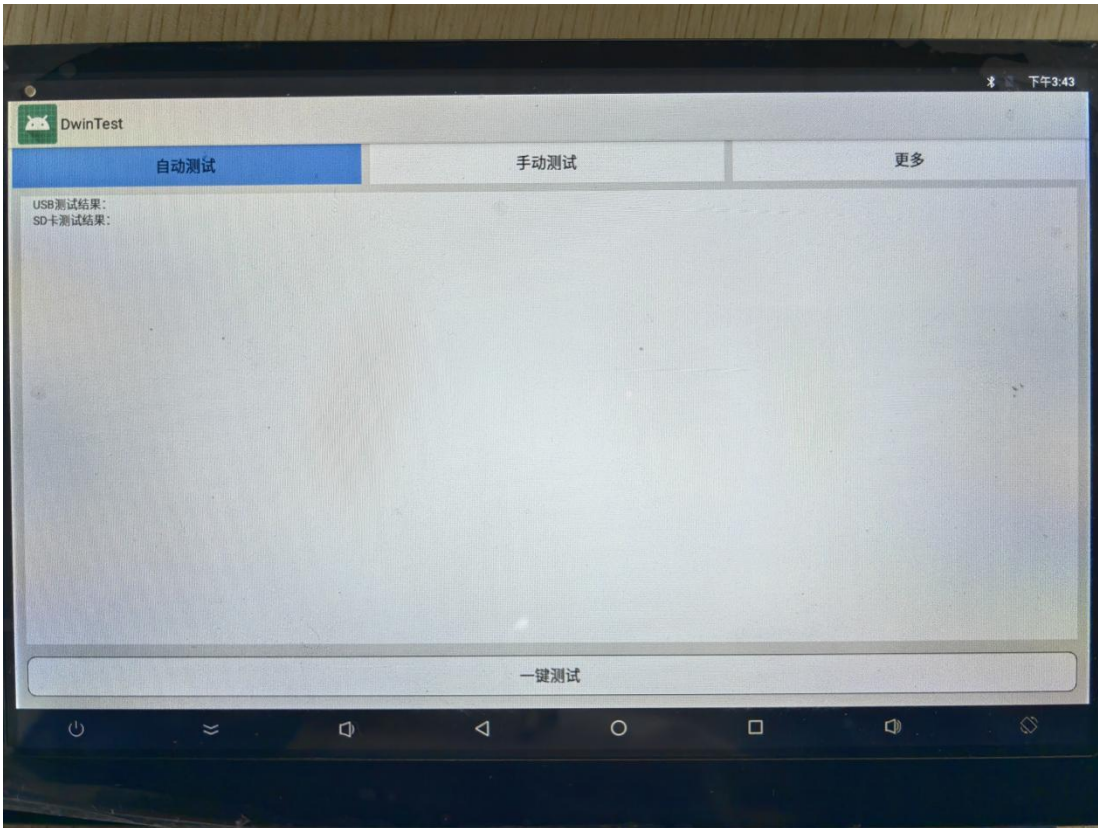
- i. In Android Studio, click on "File" in the top left corner, then click "Open." Choose the directory of the source code.





Professional. Creaditable. Successful

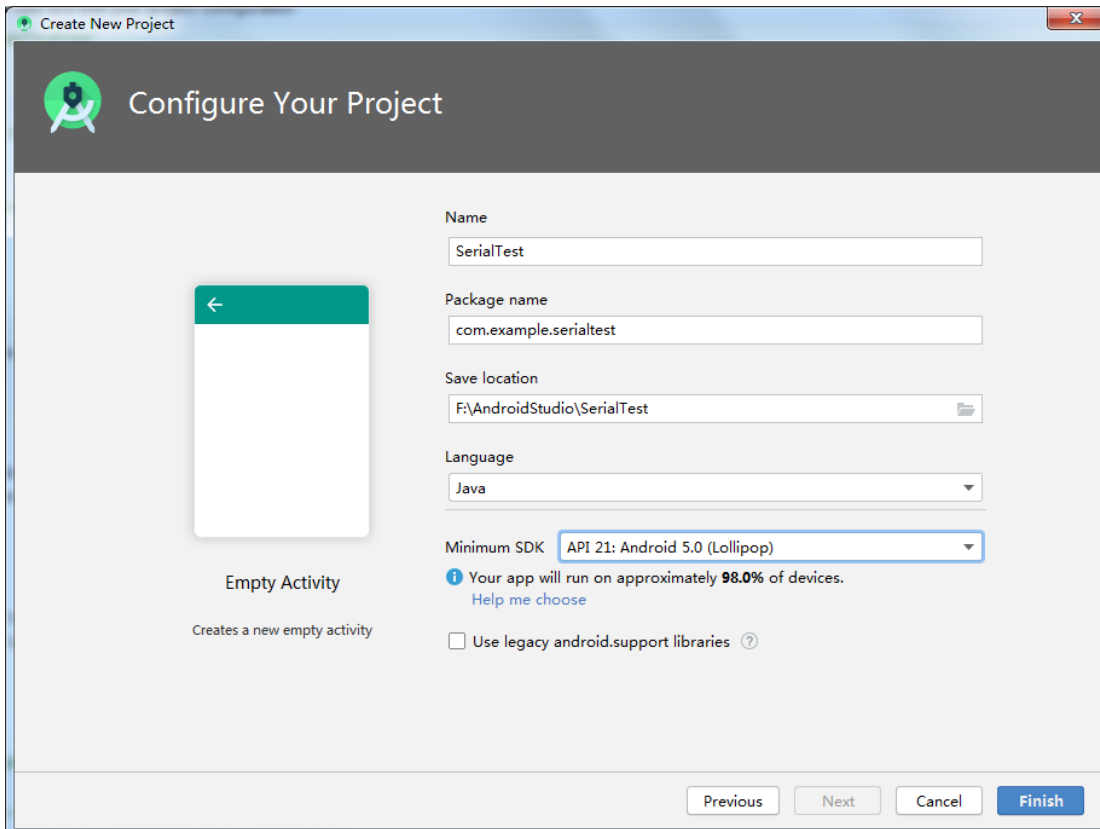
- ii. Power up the motherboard and connect the Android screen to the PC using a USB cable.
- iii. Select the appropriate Android screen, click "Run," and download DwinTest onto the Android screen.



You can click [More] -> [Instructions] to see how to use the software for basic testing.

C. Using Dwin Java Serial Libraries to Create a New Project

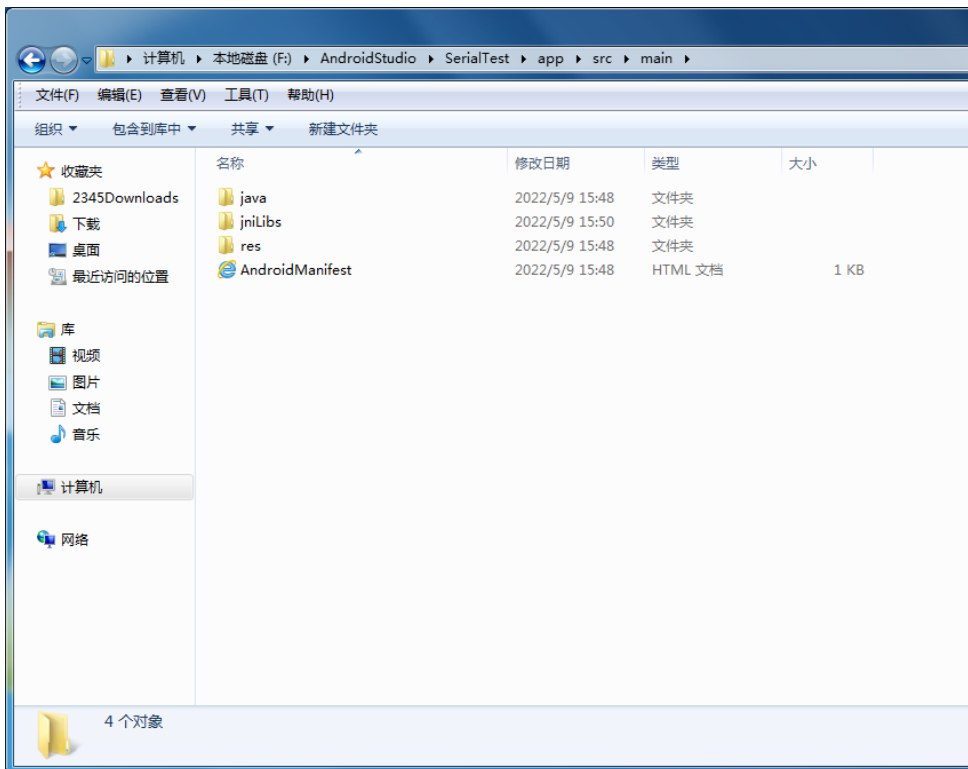
- i. Create a new project. In this example, the project is named "SerialTest," saved in the path "F:\AndroidStudio\SerialTest," and the programming language is Java.

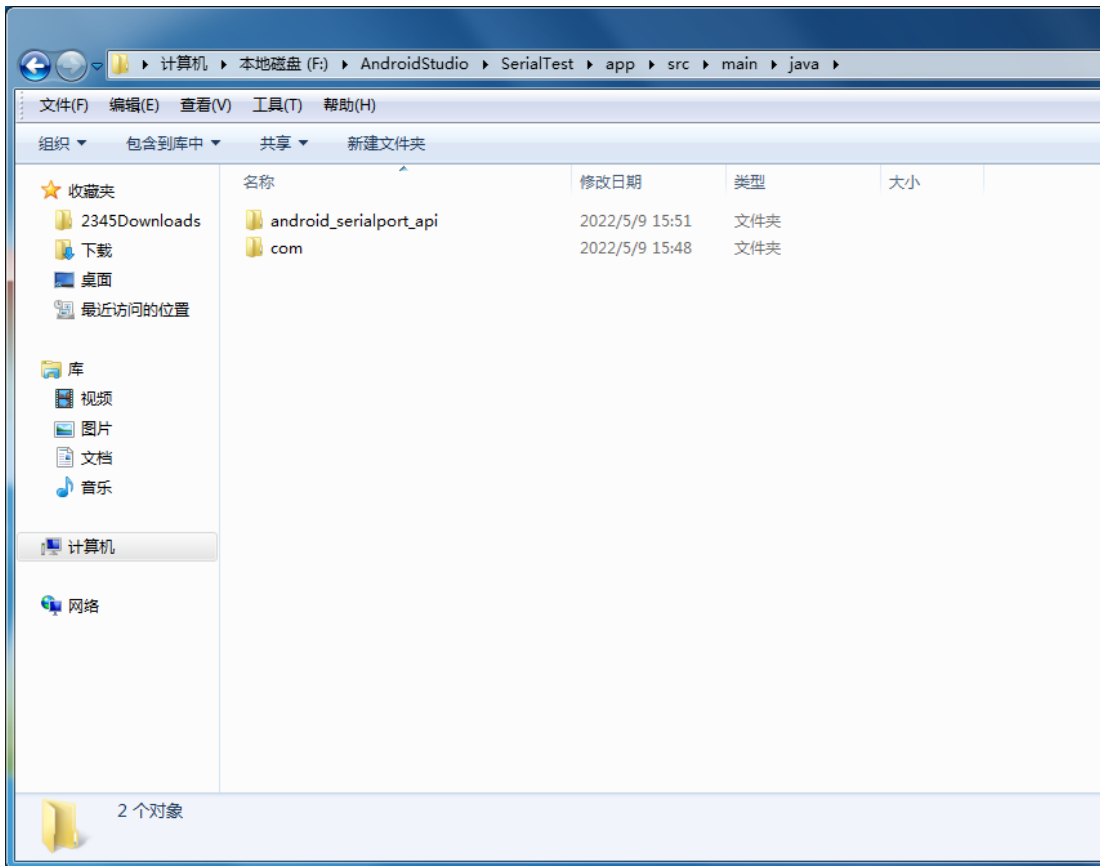


ii. Copy the "jniLibs" from the Dwin Android tools package to "F:\AndroidStudio\SerialTest\app\src\main.

" Also, copy the "android_serialport_api" folder to "F:\AndroidStudio\SerialTest\app\src\main\java." Ens

ure that the directoy is named "android_serialport_api" and remains unchanged.





iii. After completing the steps, you can operate the serial port. Specific operations can be referenced in the activities within the DwinTools project source code at "DwinTools\app\src\main\java\com\dwin\DwinTools\serial." Modify and use these according to your own project.

-Application.java: It integrates `android.app.Application` and is primarily used to generate SharedPreferences. The configuration of baud rate and serial port devices is completed using SharedPreferences. Generally, no modification is required.

-SerialPortActivity.java: This class is used to create input/output streams and threads for serial port



Professional. Creaditable. Successful

communication. Users can inherit this class in their own activities to perform read/write operations on the serial port.

-SerialPortPreferences.java: This class scans available serial port devices like ttyS* and writes their configurations to SharedPreferences. Users can modify this class based on their own app's requirements.

-ConsoleActivity.java: It is integrated from `SerialPortActivity` and is used for character-based serial communication.

-HexConsoleActivity.java: It is integrated from `SerialPortActivity` and is used for hexadecimal-based serial communication.

-SerialMainMenu.java: This is the main menu activity that facilitates navigation to specific activities.

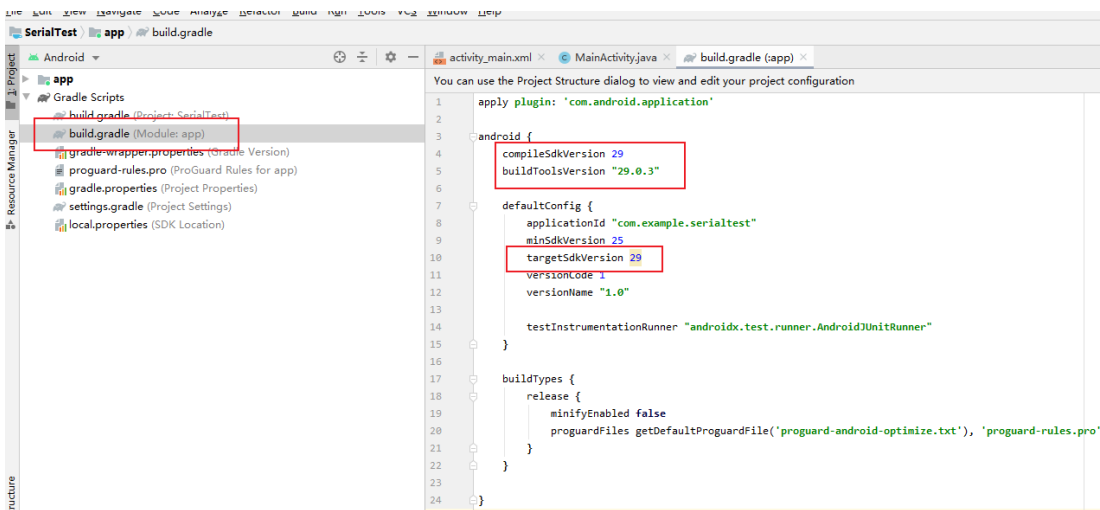
-Sending01010101Activity.java: This activity constantly sends the sequence "0101" to test the serial waveform.

-LoopbackActivity.java: This activity is used to test whether serial communication for loopback is functioning correctly.

-HexHelper.java: This is a series of utility classes used for converting between characters and hexadecimal representations.

Before using these classes, make sure to update the configuration in the `build.gradle(Module:app)` file as shown in the image.

compileSdkVersion 29
buildToolsVersion "29.0.3"
targetSdkVersion 29



D. Example of using common functions of DWIN Android screen

Most of the special functions of the DWIN Android screen are operated in the form of sending broadcasts

- i. Dynamic restart

Just send broadcast android.intent.action.dwin_reboot

Intent sr = new

Intent("android.intent.action.dwin_reboot");

sendBroadcast(sr);



Professional. Creaditable. Successful

ii. Dynamic return and the implementation of returning home

Return to send broadcast android.intent.action.dwin_input_back_key

```
Intent sr = new Intent("android.intent.action.dwin_input_back_key");
sendBroadcast(sr);
home send boardcast android.intent.action.dwin_input_home_key
Intent sr = new Intent("android.intent.action.dwin_input_home_key");
sendBroadcast(sr);
```

iii. Read chip_id

Use the getChipId() method in the tool class ChipUtil in the DwinTools source code to return, and the source code has been given.

```
case R.id.btn_get_chipid:
    //获取Chip ID
    String chipId = ChipUtil.getInstance().getChipId();
    mTextViewChipId.setText("Chip ID : " + chipId);
    break;
```

iv. Rotate the screen through the app

Use the form of sending broadcast to operate. The code is in the RotationActivity class in DwinTools. After the sending is successful, the system will restart and modify the screen rotation.

@Override

```
public void onClick(View v) {  
    switch (v.getId()) {  
        case R.id.btn_rotation_0:  
            setRotation("0");  
            break;  
        case R.id.btn_rotation_90:  
            setRotation("90");  
            break;  
        case R.id.btn_rotation_180:  
            setRotation("180");  
            break;  
        case R.id.btn_rotation_270:  
            setRotation("270");  
            break;  
    }  
}
```

```
/**  
 * 通过发送广播, 系统则会进行屏幕旋转 The system rotates the screen by  
 * sending Broadcast  
 * @param value 旋转的值 the value of rotation  
 */  
private void setRotation(String value) {  
    Log.v(TAG, msg: "Dwin test sendBroadcast!");  
    Intent sr = new Intent(DWIN_ROTATION);  
    sr.putExtra(name: "message", value);  
    sendBroadcast(sr);  
}
```

V. Permanently Hiding the Navigation Bar via App

To permanently hide the navigation bar using the app, you can utilize broadcast messages. The relevant code can be found in the `NavigationBarActivity` class within the `DwinTools` project.

When you send the broadcast successfully, the system will reboot, and the navigation bar status will be



Professional. Creaditable. Successful

modified accordingly.

```
send 0 to show navigation bar, send 1 to hide navigation bar
//发送"0"则显示导航栏, 发送"1"则隐藏导航栏
public static final String DWIN_NAVIGATION_BAR_SHOW_VALUE = "0";
public static final String DWIN_NAVIGATION_BAR_HIDE_VALUE = "1";

/**
 * 通过发送广播, 系统则会进行相应的隐藏\显示导航栏 The system hides/shows navigation bar
 * @param value 是否显示导航栏的值 by sending Broadcast
 * wether to display the value of the navigation bar
 */
private void setNavigationBar(String value){
    Log.v(TAG, msg: "Dwin test sendBroadcast!");
    Intent sr = new Intent(DWIN_NAVIGATION_BAR);
    sr.putExtra(name: "message", value);
    sendBroadcast(sr);
}
```

vi. Setting up Auto Start on Boot

There are generally two ways to achieve auto-start on boot: one is by receiving the boot broadcast and starting the app, and the other is by setting your APK as the launcher. In the former method, the app starts after booting into the system desktop, while in the latter method, the app bypasses the system desktop and starts directly.

- Enter the program after booting the android interface

When Android starts, it broadcasts a system message with the content `ACTION_BOOT_COMPLETED`, represented as `Android.intent.action.BOOT_COMPLETED`. By capturing this message in your app, you can trigger the app to start. This is typically done by implementing a BroadcastReceiver.

1.1.1 Interface Activity, MainActivity .java document

```
public class MainActivity extends Activity {
```

Websites: www.dwin-global.com



Professional. Creditable. Successful

```
/** Called when the activity is first created. */  
@Override  
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    // without title  
    requestWindowFeature(Window.FEATURE_NO_TITLE);  
    // full screen  
    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,  
        WindowManager.LayoutParams.FLAG_FULLSCREEN);  
    setContentView(R.layout.activity_main);  
}
```

This code is very simple. When the Activity starts, the TextView will be displayed, use it to display the words you want to display.

The typeface displayed.

1.1.2 Receive broadcast messages

```
public class BootBroadcastReceiver extends BroadcastReceiver {  
    static final String action_boot = "android.intent.action.BOOT_COMPLETED";  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        if (intent.getAction().equals(action_boot)) {  
            Intent ootStartIntent = new Intent(context, MainActivity.class);  
            ootStartIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);  
            context.startActivity(ootStartIntent);  
        }  
    }  
}
```

This class inherits from BroadcastReceiver. In the overridden method onReceive, check whether the received Intent conforms to BOOT_COMPLETED. If so, start the MainActivity.

1.1.3 Configuration file

(1)AndroidManifest.xml :

```
<?xml version="1.0" encoding="utf-8" ?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.ajie.bootstartdemo"  
    android:versionCode="1"  
    android:versionName="1.0" >  
    <uses-sdk  
        android:minSdkVersion="8"
```

Websites: www.dwin-global.com



Professional. Creditable. Successful

```
android:targetSdkVersion= "17" />
<application
android:allowBackup= "true"
android:icon= "@drawable/ic_launcher"
android:label= "@string/app_name"
android:theme= "@style/AppTheme" >
<activity
android:name= "com.dwin.bootstartdemo.MainActivity"
android:label= "@string/app_name" >
<intent-filter>
<action android:name= "android.intent.action.MAIN" />
<category android:name= "android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
<receiver android:name= "com.dwin.bootstartdemo.BootBroadcastReceiver" >
<intent-filter>
<action android:name= "android.intent.action.BOOT_COMPLETED" />
<category android:name= "android.intent.category.HOME" />
</intent-filter>
</receiver>
</application>
<uses-permission android:name= "android.permission.RECEIVE_BOOT_COMPLETED" >
</uses-permission>
</manifest>
```

Please note the part highlighted in red. This section registers a receiver with the system. The sub-node intent-filter indicates that it is set to receive the android.intent.action.BOOT_COMPLETED message.

Additionally, the android.permission.RECEIVE_BOOT_COMPLETED permission must be configured.

After completion, compile the apk package and install it on the Android screen. Shut down and restart, and the page displayed by the MainActivity will be displayed. **DWIN Android broadcast receiving is operated by the default interface of the Android system, and uses the standard Android API function library. For specific usage methods, please refer to the Android official website API Guides.**

- Boot to user-selected android interface or application

To achieve boot self-starting is mainly to modify the application program to have the authority of launcher.



Professional. Creditable. Successful

Replace the <intent-filter> of the first started activity in the configuration file

(1)AndroidManifest.xml :

```
<?xml version= "1.0" encoding= "utf-8" ?>
<manifest xmlns:android= "http://schemas.android.com/apk/res/android"
package= "com.ajie.bootstartdemo"
android:versionCode= "1"
android:versionName= "1.0" >
<uses-sdk
android:minSdkVersion= "8"
android:targetSdkVersion= "17" />
<application
android:allowBackup= "true"
android:icon= "@drawable/ic_launcher"
android:label= "@string/app_name"
android:theme= "@style/AppTheme" >
<activity
android:name= "com.dwin.bootstartdemo.MainActivity"
android:label= "@string/app_name" >
<intent-filter>
<action android:name= "android.intent.action.MAIN" />
<category android:name= "android.intent.category.HOME" />
<category android:name= "android.intent.category.DEFAULT" />
<category android:name= "android.intent.category.MONKEY" />
</intent-filter>
</activity>
</application>
</manifest>
```

After running the program, click on the Home button. A Launcher selection box will pop up. Choose your developed APK and click "Always."

6. adb Installation and Usage

A. Installation on PC:

Download the platform-tools adb version tool.

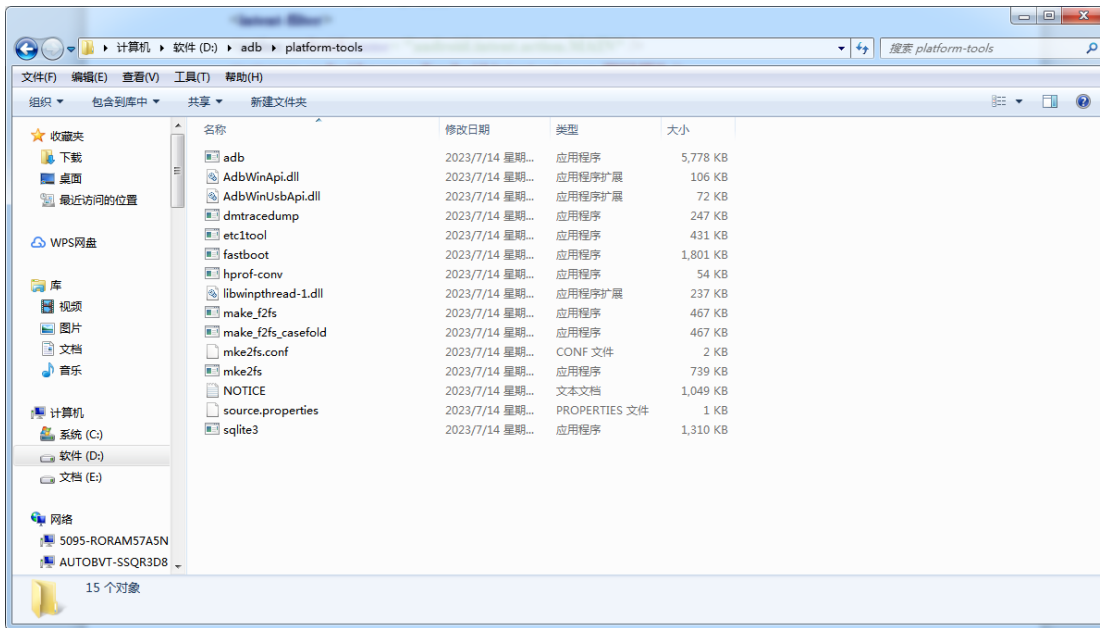
Websites: www.dwin-global.com



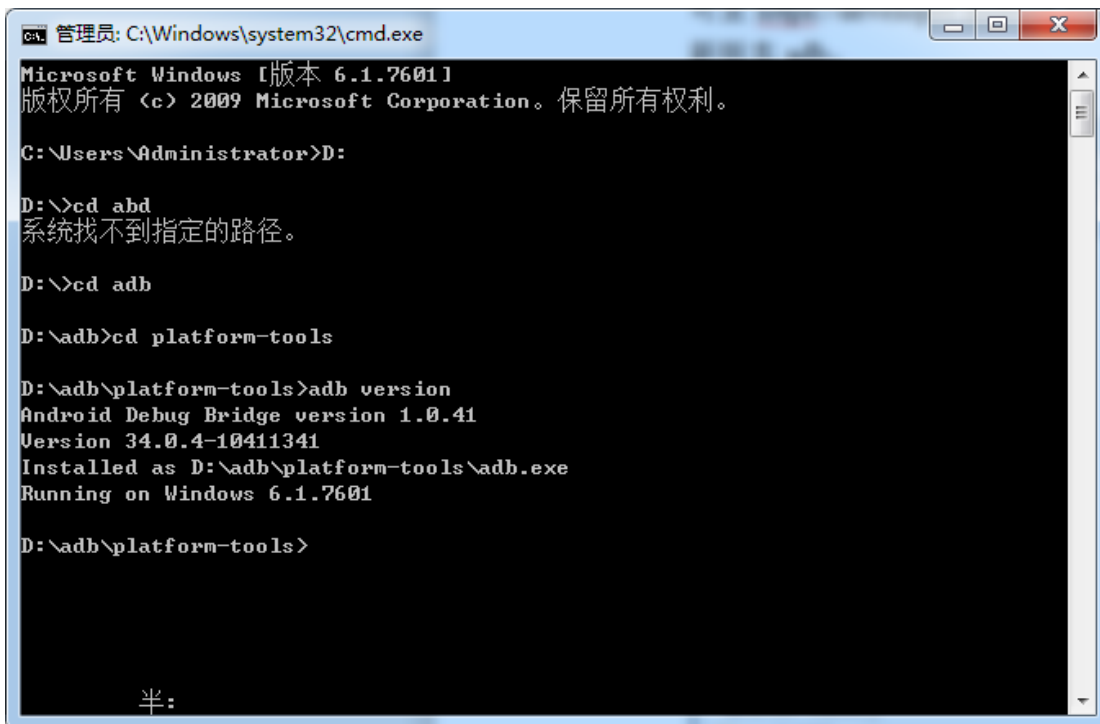
Professional. Creaditable. Successful

You can obtain the latest version of adb from this website: <https://developer.android.google.cn/studio/releases/platform-tools?hl=zh-cn>.

Once downloaded, extract it to your local directory.



Then, open the Run dialog (Win+R), type "cmd," and press Enter. Use commands to perform operations. Input "adb version" to check your adb version and confirm if the installation was successful.



```
ca. 管理员: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>D:

D:\>cd abd
系统找不到指定的路径。

D:\>cd adb

D:\adb>cd platform-tools

D:\adb\platform-tools>adb version
Android Debug Bridge version 1.0.41
Version 34.0.4-10411341
Installed as D:\adb\platform-tools\adb.exe
Running on Windows 6.1.7601

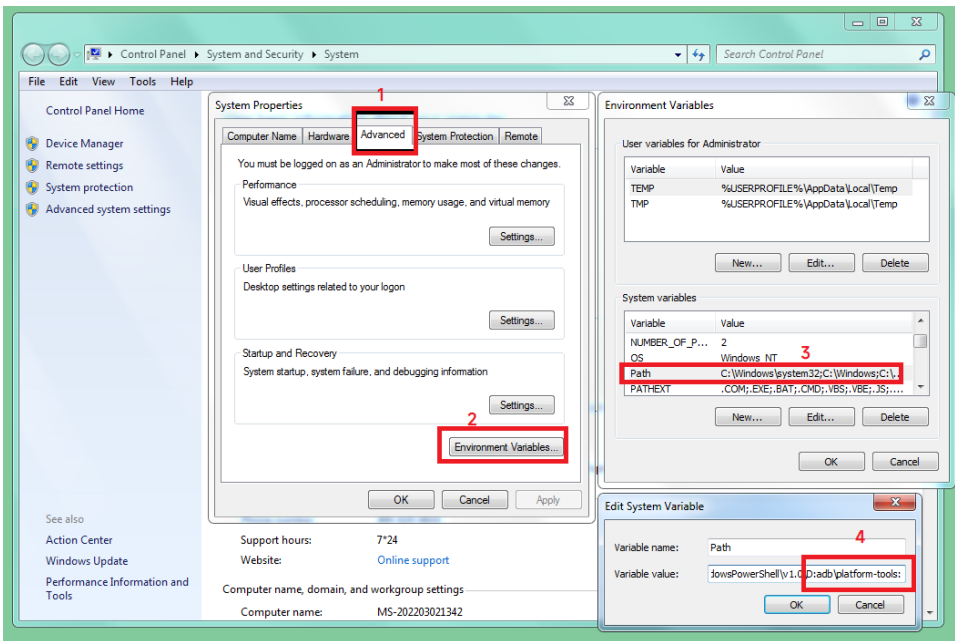
D:\adb\platform-tools>
```

B. Debugging Using Android Studio

Add Environment Variable:

- Open [My Computer] → Click on [Properties] → Select [Advanced system settings] → Click on [Environment Variables].

- Edit the system variable "Path" and Add the adb installation path after the variable value as shown in the image below:



This will enable you to use the adb command for debugging within the Android Studio Terminal.