



| | |
|---------------------|---|
| Product Name | GAOTek Embedded Industrial Linux Android Development Boards |
| Product SKU | GAOTek-IDK-164 |
| Product URL | https://gaotek.com/product/gaotek-embedded-industrial-linux-android-development-boards/ |

Contact us: sales@gaotek.com

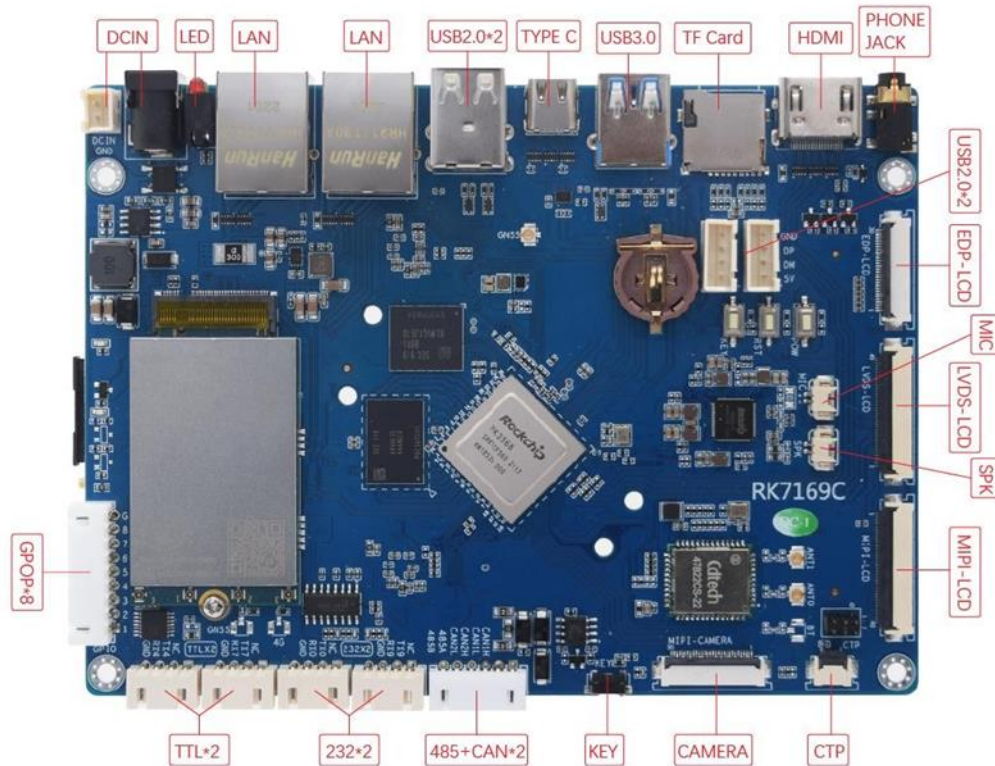


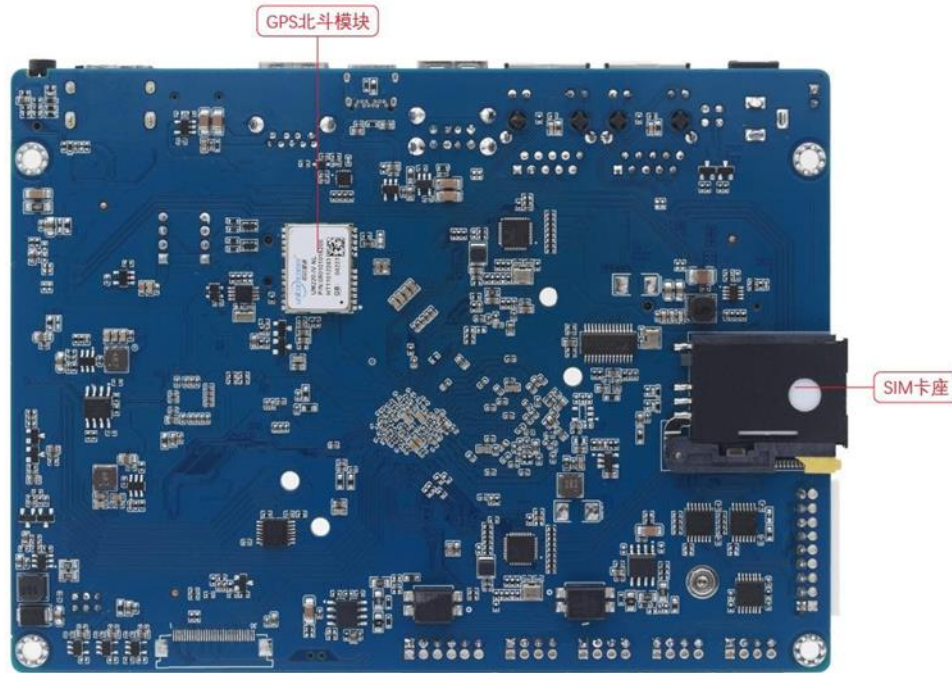
Contents

| | |
|--|----|
| 1. Interface introduction..... | 3 |
| 1.1. Serial port..... | 4 |
| 1.2. USB..... | 4 |
| 1.3. GPIO..... | 4 |
| 1.4. CAN..... | 4 |
| 1.5. Ethernet..... | 5 |
| 2. Software development..... | 5 |
| 2.1. Root permission | 5 |
| 2.2. Serial port sample program | 5 |
| 2.3. GPIO port application development..... | 8 |
| 2.4. CAN Application Development..... | 10 |
| 2.5. Boot-up | 11 |
| 2.5.1. Receiving power-on broadcasts..... | 11 |
| 2.5.2. Set as desktop program..... | 12 |
| 2.6. Modify system signature..... | 12 |
| 2.7. APK encryption development..... | 13 |
| 2.8. Hide and show the navigation bar..... | 13 |
| 2.9. Hide and show status bar..... | 14 |
| 2.10. Software shutdown and restart..... | 14 |
| 2.11. Modify boot animation..... | 15 |
| 2.12. Application Setting System Time | 16 |
| 2.13. Disable status bar drop-down for full-screen applications..... | 16 |
| 2.14. APK installation in the background..... | 17 |
| 2.15. CAMERA Application Development | 17 |
| 2.16. Set the HDMI output resolution..... | 18 |
| 2.17. Watchdog Development..... | 18 |
| 3.1. USB Device Driver Installation | 18 |
| 3.2. Connected to the computer is not recognized..... | 19 |
| 3.3. Problems with serial port development..... | 19 |
| 3.4. The program pops up an unresponsive prompt box | 19 |
| 3.5. How to set the default horizontal and vertical display of the system..... | 19 |

GAOTek Embedded Industrial Linux Android Development Boards

1. Interface introduction





1.1. Serial port

The product provides 5 serial ports, 2 TTL, corresponding to the device paths `/dev/ttyS4` and `/dev/ttyS7` in the system, 2 RS232, corresponding to the device paths `/dev/ttyS0` and `/dev/ttyS9` in the system, and 1 RS485, corresponding to the device path `/dev/ttyS3` in the system.

1.2. USB

The product provides six USB ports, one of which is USBOTG, connected through a USB TYPE-C header, and the other four are USBHOST, of which four are USB 2.0, two are connected through a USB header, two are connected through a 4pin socket, and one is USB3.0.

1.3. GPIO

Provides 8 GPIO ports with configurable inputs and outputs, default outputs, and GPIO numbers 8, 15, 33, 32, 36, 40, 41, 42.

The product provides 2 LEDs, 1 red, controlled through GPIO port 60, and 1 green, controlled through GPIO port 96.

1.4. CAN

The product provides 2 CANs, each with a 120-ohm termination resistor, which is controlled by a dip switch to access or not

1.5. Ethernet

The product provides two wired networks with two RJ45 connectors, which can be used individually or both at the same time.

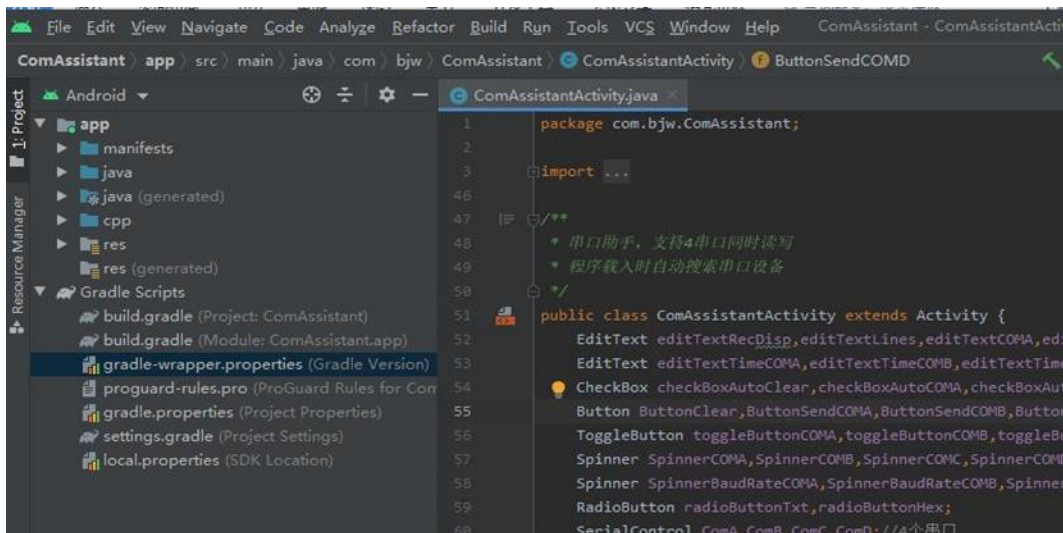
2. Software development

2.1. Root permission

The system has been granted root access by default.

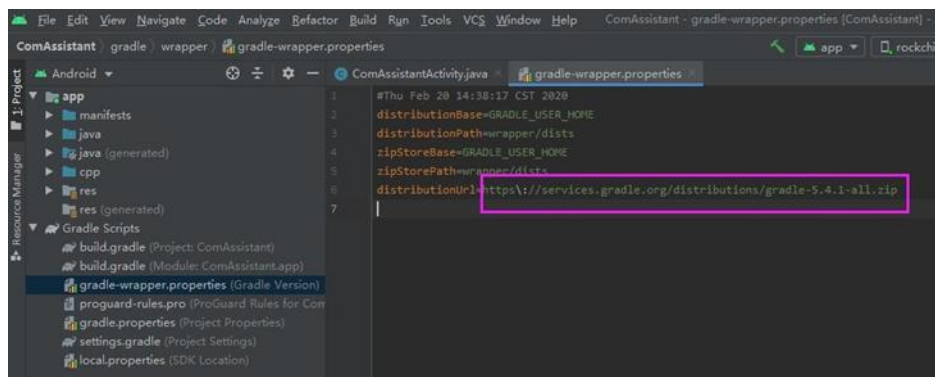
2.2. Serial port sample program

Use Android Studio to import the serial port sample program. Firstly, select the serial port example and open it. After opening, the project will be presented as the following directory:

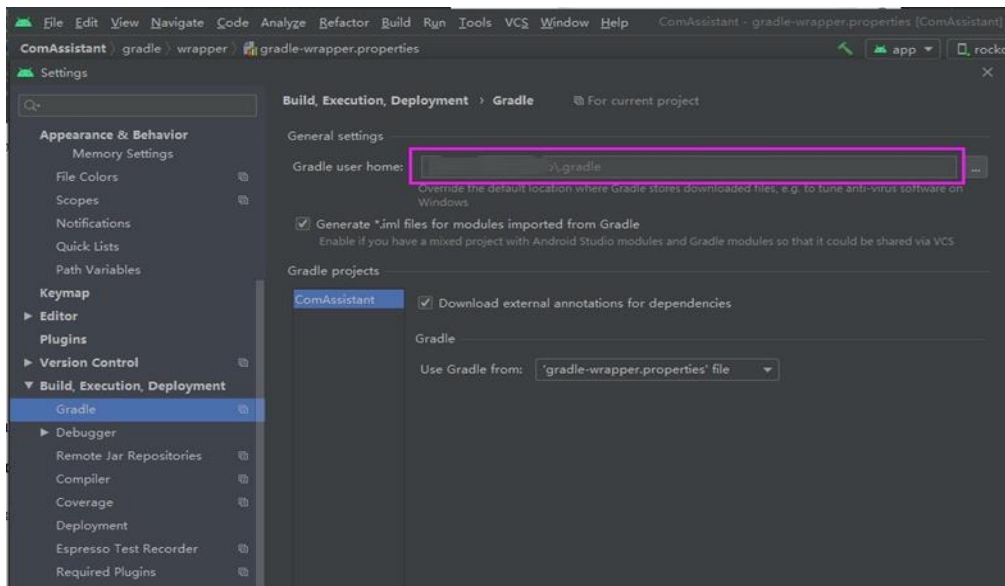


After successful import, you can update/adapt the corresponding Gradle version in the following ways

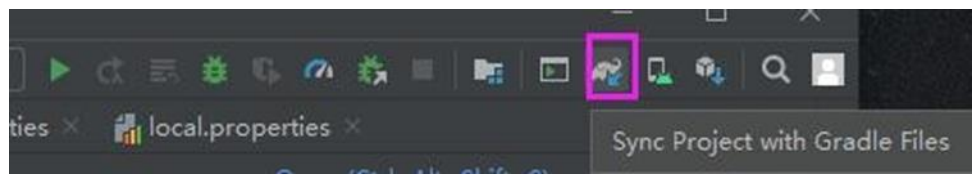
- Modify the value of distribution URL in the gradle-wrapper. Properties of the configuration file. As shown:



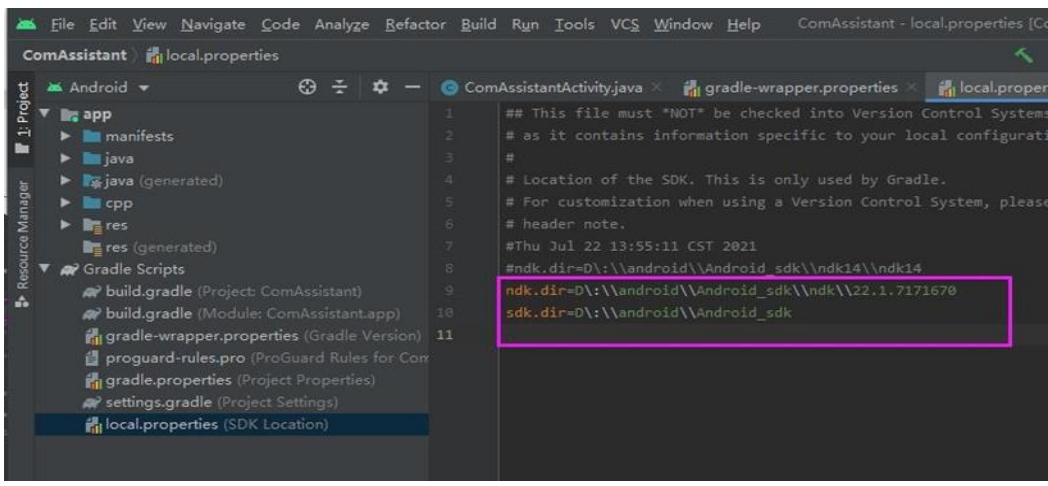
- Modify the value of distribution URL in the gradle-wrapper.properties of the configuration file. As shown:



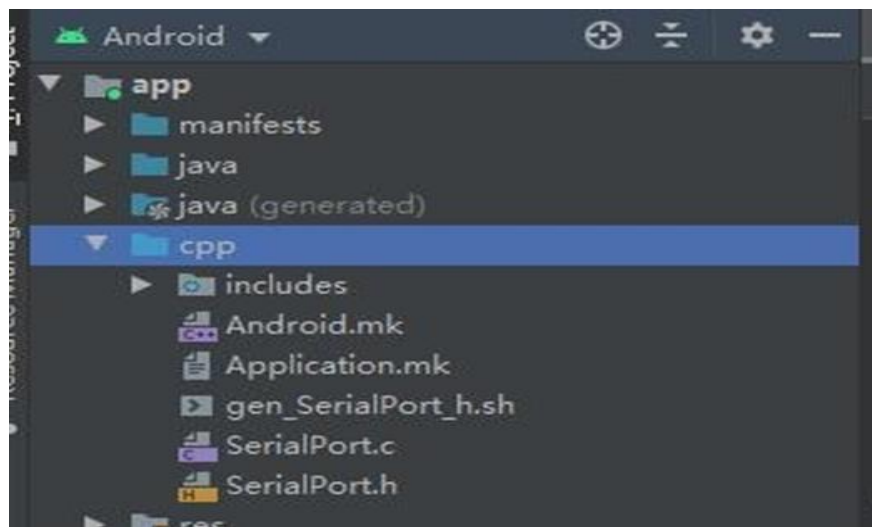
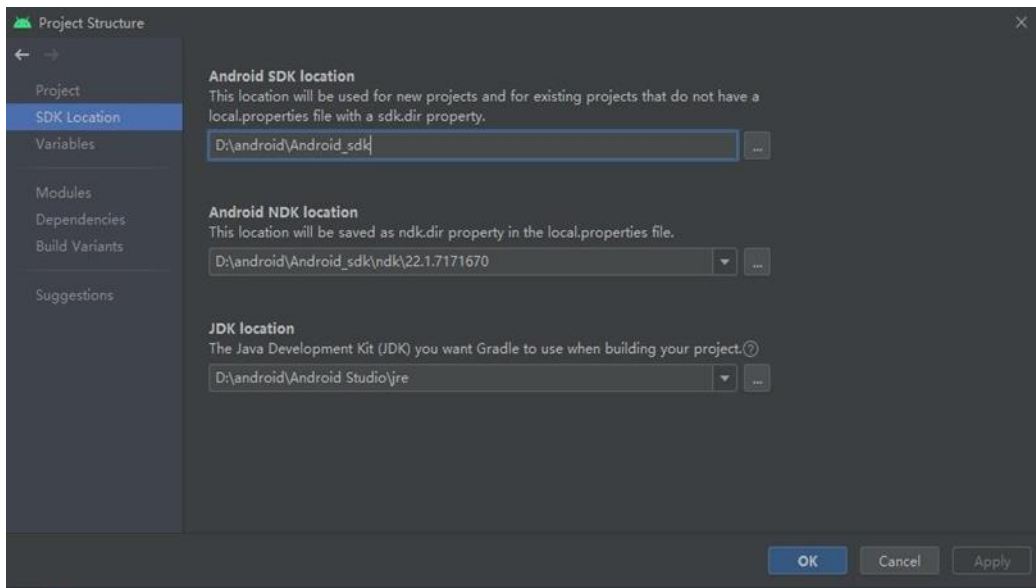
Once set up, the Gradle configuration can be synchronized by clicking on the following button



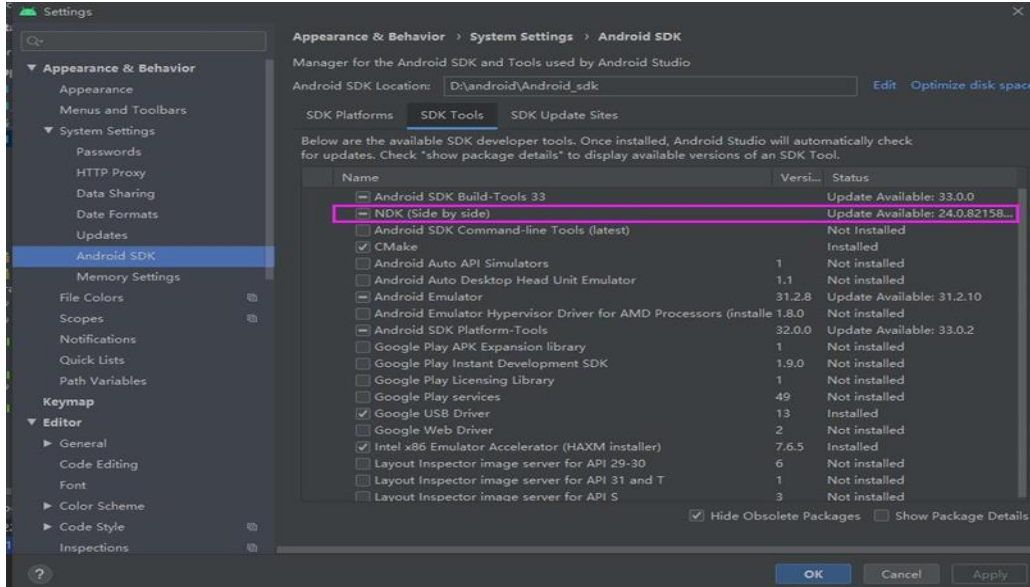
The SDK and NDK of Android Studio need to be configured to run the sample program, and the SDK & NDK of the project can be set to the locally installed path through the configuration file, as shown in the figure.



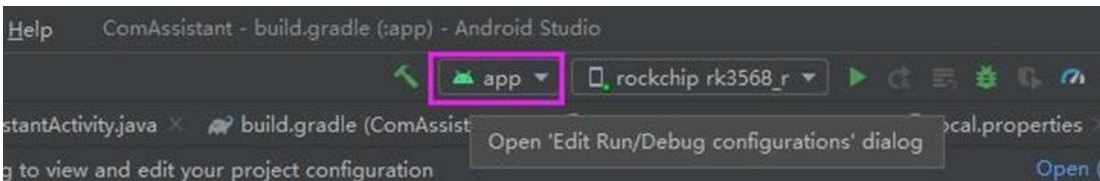
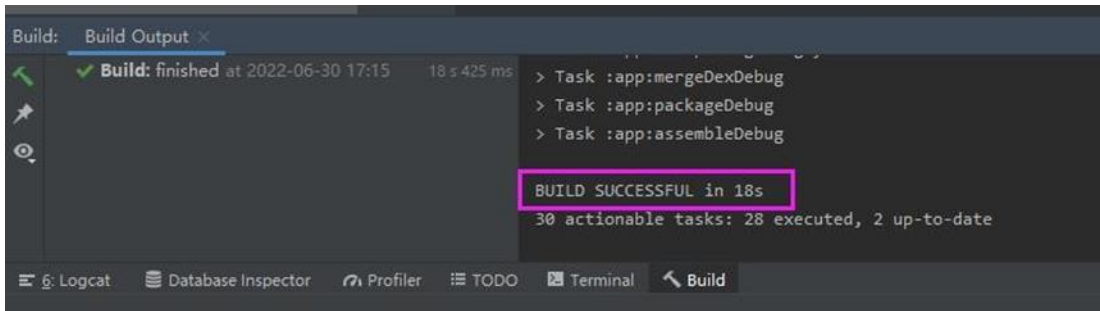
Alternatively, the SDK & NDK import can be set up via File-Project Structure at



If you don't have NDK installed, you can install it directly in Android Studio through Settings page.



After finish configuring SDK and NDK, rebuild the project and wait for the following status in the top middle of Android Studio to run the serial port example successfully.



2.3. GPIO port application development

The product provides C/C++ and JAVA layer GPIO port operation interface, reference development examples, you can configure the specified GPIO port as input or output state, set the output level high or low. GPIO reference development examples are provided in C language and JAVA demo program.

GPIO port and device number index correspondence

| Corresponding pins | GPIO Number |
|--------------------|-------------|
| IO1 | 8 |
| IO2 | 15 |
| IO3 | 33 |
| IO4 | 32 |
| IO5 | 36 |
| IO6 | 40 |
| IO7 | 41 |
| IO8 | 42 |

- **C/C++ layer interface**

Under the /sys/class/gpio folder, a folder will be generated for each GPIO port, for example, the folder corresponding to IO1 is gpio8, the folder corresponding to IO2 is gpio15, and so on for other GPIO ports. The following figure shows.

```
rk3568_r:/sys/class/gpio $ ls
ls
export  gpio33  gpio41  gpio8      gpiochip128  gpiochip64
gpio15  gpio36  gpio42  gpio95     gpiochip32   gpiochip96
gpio32  gpio40  gpio60  gpiochip0  gpiochip511  unexport

rk3568_r:/sys/class/gpio/gpio8 $ ls
ls
active_low  device  direction  edge  power  subsystem  uevent  value
```

You can set the input and output status of the corresponding GPIO port by writing "in" or "out" to the direction file in the GPIO node directory. If the GPIO port is set to output, then write 0 or 1 to the value file to control the GPIO port output low or high.

- Java layer interface

Add the GPIO development library `gpio-dev.jar` to the client project. The library provides the `writeGpioDirection` interface to set the input and output state of the GPIO port, and the `writeGpioValue/ readGpioValue` interface to set and read the high and low levels of the GPIO port, refer

to the sample program and the following code:

```
import com.cosofteck.Gpio;
Gpio.writeGpioDirection(8,1);
Gpio.writeGpioValue(8,1);
```

2.4. CAN Application Development

The product provides JAVA layer CAN development interface, refer to the development example, you can realize the configuration, switch and read/write function of CAN interface. Add the CAN development library `candev.jar` to the customer project, set the bit rate, open the CAN interface, and start reading and writing.

Reference can be made to the sample program and the following code. import `com.cosofteck.Can;`

```
import com.cosofteck.CanFrame;
//Open CAN0
Can.SetBitrate(250000);
Can.Start();
//Sending data
Can.SendInit(29, 3, 1);
byte testData[] = {1,3,5,7,2,4,6,8,2,3};
Can.SendData(testData, 10, false, false);
Can.SendClose();
//Receiving data, receiving data is generally processed through threads
Can.ReadInit(29, 3, 1);
CanFrame canFrame = Can.ReadData();
Can.ReadClose
```

```
//Open CAN1
Can.SetBtrateOne(250000);
Can.StartOne
//Sending data
Can.SendInitOne (29, 3, 1);
byte testData[] = {1,3,5,7,2,4,6,8,2,3};
Can.SendDataOne (testData, 10, false, false);
Can.SendCloseOne
//Receiving data, receiving data is generally processedthroughthreads
Can.ReadInitOne (29, 3, 1);
CanFrame canFrame = Can.ReadDataOne
Can.ReadCloseOne ();
```

2.5. Boot-up

There are generally two ways to boot up: one receiving the boot.

broadcast and then launching, and one setting your APK to Laucher, the former will boot into the system desktop and then launch the APK, the latter will boot up and skip the system desktop and go directly to the APK.

2.5.1. Receiving power-on broadcasts

- When the system starts up, it will send out a system broadcast with the content ACTION_BOOT_COMPLETED, which is represented by the string constant Android.intent.action.BOOT_COMPLETED. then start it. Refer to the self-start example program. Receive power-on broadcast BootBroadcastReceiver.java

```
@Override
```

```
public void onReceive(Context context, Intent intent)
```

```
// TODO Auto-generated method stub
```

```
if (intent.getAction().equals(action_boot))
```

```
Intent bootStartIntent = new Intent(context, MainActivity.class);
```

```
bootStartIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
```

```
context.startActivity(bootStartIntent);
```

This class inherits from Broadcast Receiver, and in the override method on Receive, detects whether the received Intent meets BOOT_COMPLETED, and if so, starts the Activity, Main Activity. □ Configuration file AndroidManifest.xml

Adding permissions to the configuration file: <uses-permission

android:name="android.permission.RECEIVE_BOOT_COMPLETED" />

Add registration in the application node receiver: <receiver android:name=".BootBroadcastReceiver">

www.cosofteck.com 16

<intent-filter>

<action

android:name="android.intent.action.BOOT_COMPLETED"/></intent-filter>

</receiver>

After compiling the APK package and installing it into the system, you have to start it manually for the first time, and then it will boot up after shutting down and rebooting.

2.5.2. Set as desktop program

Setting up the desktop application is mainly a matter of modifying the configuration of the application by adding to the <intent-filter>of the first activity launched by the application :

<category android:name="android.intent.category.HOME" /><category android:name="android.intent.category.DEFAULT"/>Compile the APK package, install it on your system, click theHomebutton, or restart the system, a desktop selection box will popup, select your APK, and click "Always".

2.6. Modify system signature

- Rename the apk file to .zip format. Open it with zip softwareanddelete the CERT.SF and CERT.RSA files in the META-INF directory.
- Rename the deleted file back to apk and copy it to thefolder

f:/androidsource (create this file randomly, it is recommendednot

to have Chinese). Put platform.pk8, platform.x509.pem, signapk.jar in the same folder as the signature toolkit.

- Open the command line, cd to the androidsource directory, execute the command line java -jar signapk.jarplatform.x509.pem platform.pk8 SetSystemTime.apk test.apk, as shown :

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>f:

F:\>cd androidsource

F:\androidsource>java -jar signapk.jar platform.x509.pem platform.pk8 SetSystemI
ine.apk test.apk

```

- The apk in front of the above command is the file to be signed, and the test.apk behind it is the output file, the name can be written freely

2.7. APK encryption development

The RK7169C provides a hardware encryption interface for

application development so that the encrypted application can only run on the specified hardware platform.

- The project adds rkdevlib.jar.
- Add a reference to the program header: `import com.cosofteck.rkdevlib.RkDevApi;`
- Function Description

Testing ID in `CheckIdPermission()`

Return 0, the verification failed; return 1, the verification passed.

- Sample Code

```

RkDevApi rkDevApi = new RkDevApi();
if(rkDevApi.CheckIdPermission() == 0)
//check error
else if(rkDevApi.CheckIdPermission() == 1)
//check ok

```

2.8. Hide and show the navigation bar

The product system defines broadcasts to hide and show the navigation bar, and sends the corresponding broadcast when needed to hide or show the navigation bar. At the same time, the display section of

the system settings provides a setting item to hide the navigation bar, you can manually hide or show the navigation bar.

- hide the navigation bar

```
Intent intent=new Intent();
```

```
intent.setAction("ACTION_SHOW_NAVBAR");  
intent.putExtra("cmd","hide");  
sendBroadcast(intent,null);
```

- Show navigation bar

```
Intent intent=new Intent();  
intent.setAction("ACTION_SHOW_NAVBAR");  
intent.putExtra("cmd","show");  
sendBroadcast(intent,null);
```

2.9. Hide and show status bar

The product system defines broadcasts to hide and show the status bar, and sends the corresponding broadcast when needed to hide or show the status bar. At the same time, the display section of the system settings provides a setting item to hide the status bar, you can manually hide or show the status bar.

- hide status bar

```
Intent intent=new Intent();  
intent.setAction("ACTION_SHOW_STATUSBAR");  
intent.putExtra("cmd","hide");  
sendBroadcast(intent,null);
```

- Show status bar

```
Intent intent=new Intent();  
intent.setAction("ACTION_SHOW_STATUSBAR");  
intent.putExtra("cmd","show");  
sendBroadcast(intent,null);
```

2.10. Software shutdown and restart

The product system is customized with broadcasts for shutdown and reboot, and users can implement software shutdown or software reboot in the code by sending the corresponding broadcast when needed.

- software shutdown

```
Intent intent=new Intent();  
intent.setAction("ACTION_RK_SHUTDOWN");  
sendBroadcast(intent,null);
```

- Software reboot

Intent intent=new Intent

```
intent.setAction("ACTION_RK_REBOOT");
```

```
sendBroadcast(intent,null);
```

2.11. Modify boot animation

Please refer to the bootanimation.zip archive, unzip the file and you can see the structure in the boot animation folder. Replace the files in the boot animation part0 or part1 directory with the images you want to animate, and set the desc.txt text as needed, using the zip archive example as an example.

```
800 1280 20
```

```
p 1 0 part0
```

```
p 0 0 part1
```

- 800 1280 is the resolution and 20 is the picture playback framerate.
- p 1 0 part0, P is the meaning of PLAY play, 1, 0 two numbers refer

to the number of cycles and stage interval time respectively, part0 is the name of the folder, this means play the pictures in the part0 folder once.

- p 0 0 part1, means loop the images in the part1 folder. part0 and part1, or only part0 part. P 0 0 part0, then it means loop the pictures in part0 folder.

Repackage the created bootanimation folder into a bootanimation.zip package. When compressing, take care to directly select part0, part1 and desc.txt, or just part0 and desc.txt for compression, do not have a second level folder, select the zip file format as ZIP and the compression method as store, as follows.



Then copy the compressed ZIP file to the root directory of the U disk, rename it to rk_boot.zip, insert the U disk into the USB HOST port of the RK7169C, the system will prompt that it is being updated, and reboot after the update is complete.

2.12. Application Setting System Time

The product system is customized with a broadcast to set the system time. Users can implement setting the system time in their own code by sending the corresponding broadcast when needed. The sample code is to set the system time to 19:28:32 on May26, 2016. Intent intent=new Intent();

```
intent.setAction("ACTION_UPDATE_TIME");
```

```
intent.putExtra("cmd","2016-05-26-19-28-32");
```

```
sendBroadcast(intent,null);
```

2.13. Disable status bar drop-down for full-screen applications

The product system defines to disable the status bar drop-down broadcast when the full screen state is applied, and you can set whether to disable the status bar drop-down by sending the corresponding broadcast when needed. At the same time, in the display section of the system settings, there is a setting item to disable the status bar.

drop-down, so you can manually select whether to disable the status bar

drop-down or not.

- disables the status bar drop-down


```
Intent intent=new Intent
intent.setAction("ACTION_STATUSBAR_DROPDOWN");
intent.putExtra("cmd","hide");
sendBroadcast(intent,null);
```

- Restore the status bar drop-down

```
Intent intent=new Intent
intent.setAction("ACTION_STATUSBAR_DROPDOWN");
intent.putExtra("cmd","show");
sendBroadcast(intent,null);
```

2.14. APK installation in the background

The product system defines a broadcast to install the APK in the background, the application can install its own APK in the background by sending the corresponding broadcast when needed. the parameter path is set to the absolute path of the apk file to be installed, for example :

```
Intent intent=new Intent
intent.setAction("ACTION_UPDATE_START");
intent.putExtra("path", "/mnt/sdcard/Download/app-debug.apk");
sendBroadcast(intent,null);
```

After the APK installation starts, the application can determine that the APK installation is complete by receiving the system broadcast ACTION_PACKAGE_ADDED or ACTION_PACKAGE_REPLACED.

2.15. CAMERA Application Development

The product uses the standard Camera interface to select different

cameras by selecting different Camera devices. When connecting multiple cameras at the same time, if some cameras do not work or work abnormally, you can give the USB camera an external independent power supply. Here is the sample code for the common camera API: //Get the number of cameras

```
//Open the specified camera device. When there are multiple camera devices, the device serial number starts from 0. Camera mCamera0 = Camera.open(0);
```

```
Camera mCamera1 = Camera.open(1);
```

```
... //Configuration camera parameters
```

```
Parameters parameters = mCamera0.getParameters
```

```
mCamera.setParameters(parameters);

//Launch Preview

mCamera0.startPreview
```

2.16. Set the HDMI output resolution

Users can set the display output resolution by the following methods. Enter System Settings -> Display -> Other -> Resolution Settings, the system will automatically obtain a list of HDMI screen supported resolution information, select the required resolution can be.

2.17. Watchdog Development

The product provides a JAVA layer of hardware watchdog operation interface, refer to the development example, you can turn on and off the watchdog, you can set and read the watchdog wait time, and "feed the dog" function. The customer adds the watchdog development library.

watchdogdev.jar to the project. The library provides WatchdogOpen, Watchdog Close, Watchdog Set Timeout, Watchdog Get Time out and Watchdog Keep Alive interfaces to the hardware watchdog for this sample program and the following code are available:

```
import com.cosofteck.Watchdog;

Watchdog.Watchdog Open

Watchdog.Watchdog Set Timeout(10);

new Thread

public void run

while(is Watchdog)

Watchdog.Watchdog Keep Alive

try

sleep(5000)

catch(Exception e)

.start

Watchdog.Watchdog Close
```

3. Frequently Asked Questions

3.1. USB Device Driver Installation

USB connection needs to be inserted into the Device mode USBOTG port set, after power on the computer's device manager will

automatically refresh pop-up new devices for installation, if the device manager is not refreshed, please check the USB connection problems. If the prompt box pops up, choose to install the

driver, the driver can be searched for directly and automatically. After installation, there will be one more Android device in device manager, if it shows exclamation mark, uninstall this driver and reinsert USB to reinstall.

3.2. Connected to the computer is not recognized

- Reboot your device
- Close or restart pea pods, cell phone assistants and other software
- Restart your computer
- Refer to the previous section to check the USB Device driver.

3.3. Problems with serial port development

- The serial port corresponds to devices in the system as /dev/ttyS0, /dev/ttyS3, /dev/ttyS4, /dev/ttyS7, /dev/ttyS9.
- If the garbled code appears, check whether the baud rate and other serial port parameters are set correctly.
- appears to send and receive abnormal, you can run the serial port test tool that comes with the system, by shorting the TX and RX ports of the serial port to see if the serial port is working normally, if it is normal, check whether there is any problem with the serial port cable or USB to serial port cable and other cables or wiring.
- users develop serial port program problems, you can first refer to the serial port sample program.

3.4. The program pops up an unresponsive prompt box

Transfer some time-consuming operations from the main thread to a sub-thread to execute them.

3.5. How to set the default horizontal and vertical display of the system

Go to System Settings->Display->Other->System Orientation, set the default system display orientation, and reboot to take effect.