Android Studio 应用开发简介

英创公司

2017年8月

2017 年 6 月, 英创公司在 ESM6802+10.1"LCD 的硬件平台上成功运行 Android 7.1.1 原生操作系统。2017 年 8 月, ESM6802 Android 版可以支持 GPIO、串口、以太网、U 盘。 后续将进一步完善对 CAN、SPI、I2C、WIFI、4G 等接口的支持。

Google 公司专门为开发 Android 应用程序提供了集成开发环境 Android Studio,涵盖 了所有 Android 应用开发相关的功能。Android 应用程序主要使用 Java 语言编写,要用到 开发工具集 SDK (Software Development Kit,提供 java 编译工具、Android 系统 API 等, 可以直接在 Android Studio 中下载)。当应用程序中要直接访问硬件,或者需要提高运行效 率时,需要将访问硬件、复杂逻辑部分使用 C/C++实现。要在 Android Studio 中开发编译 C/C++代码,需要用到工具集 NDK (Native Development Kit,提供 C/C++编译工具、API、 打包工具等,可直接在 Android Studio 中下载)。使用 NDK 可以将 C/C++源码编译成动态 链接库,供 Java 调用。由于 Java 语言要调用 C/C++函数需要用到 JNI(Java Native Interface) 技术,这就要求使用 NDK 开发 C/C++时, C/C++源码要符合 JNI 规范要求。



图 1、ESM6802 Android 版本工作分布图

1 / 40

为方便 Android 用户专注于 Android 应用层(Java 语言)的开发, 英创公司对所支持的硬件接口均提供了符合 JNI 规范的 C/C++动态链接库, 用户只需要加载英创的动态链接库, 就可以在纯 Java 语言环境中调用动态链接库中的函数, 达到访问硬件资源的目的。如图 1 所示, 用户的工作只是应用层的 java 程序, 英创己完成了其他部分工作。这篇文章会介绍 Android Studio 的环境搭建, 并以 Step2_SerialPort 为例, 来介绍使用 Android Studio 开发、调试、安装应用到 ESM6802 的过程。英创对已支持的硬件接口提供对应的应用程序示例, 包括 C/C++部分的 so 文件, 供用户参考。

Android 应用开发更详细的资料请参考开发者网站学习: https://developer.android.google.cn/index.html。

一、安装 android studio 开发环境

- 1、安装 android studio
- 2、下载 sdk、ndk 工具集



图 2、Android Studio Welcome 界面

打开 Android Studio,在 Configure 下拉菜单中选择 SDK Manager,如下图

Appearance & Behavior > System Settings > Android SDK Manager for the Android SDK and Tools used by Android Studio Android SDK Location: D:\Android\sdk Edit SDK Platforms SDK Tools SDK Platforms SDK Update Sites Each Android SDK Platform package includes the Android platform and sources pertaining to an Allevel by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components. Name API Level Revision Android API 26 26 2 Android 7.1.1 (Nougat) 25 3 Android 7.0 (Neurat) 24 2	PI Not Not Not
Manager for the Android SDK and Tools used by Android Studio Android SDK Location: D:\Android\sdk Edit SDK Platforms SDK Tools SDK Update Sites Edit Each Android SDK Platform package includes the Android platform and sources pertaining to an Allevel by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components. Name API Level Revision Android API 26 26 2 Android 7.1.1 (Nougat) 25 3 Android 7.0 (Newast) 24 2	PI Not Upo Not Not
Android SDK Location: D:\Android\sdk Edit SDK Platforms SDK Tools SDK Update Sites Each Android SDK Platform package includes the Android platform and sources pertaining to an Allevel by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components. Name API Level Revision Android API 26 26 2 Android 7.1.1 (Nougat) 25 3 Android 7.0 (Neurat) 24 2	PI Not Upo Not Not
SDK Platforms SDK Tools SDK Update Sites Each Android SDK Platform package includes the Android platform and sources pertaining to an Allevel by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components. Name API Level Revision Android API 26 26 2 Android 7.1.1 (Nougat) 25 3 Android 7.0 (Nougat) 24 2	PI Not Upo Not Not
Each Android SDK Platform package includes the Android platform and sources pertaining to an A level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components. Name API Level Revision Android API 26 26 2 Android 7.1.1 (Nougat) 25 3	PI Not Upo Not Not
Name API Level Revision Android API 26 26 2 Android 7.1.1 (Nougat) 25 3 Android 7.0 (Nougat) 24 2	Not Upo Not Not
□ Android API 26 26 2 ☑ Android 7.1.1 (Nougat) 25 3 □ Android 7.0 (Nougat) 24 3	Not Upo Not Not
Android 7.1.1 (Nougat) 25 3	Upa Not Not Not
Android 7.0 (Nougat) 24 2	Not Not Not
	Not Not
Android 6.0 (Marshmallow) 23 3	Not
	1404
🔊 Default Settings	x
Appearance & Behavior > System Settings > Android SDK Manager for the Android SDK and Tools used by Android Studio	
Appearance Android SDK Location: D:\Android\sdk Edit	
Menus and Toolbars SDK Jongs SDK Tools SDK Undate Siter	
System Settings Below are the available SDK developer tools. Once installed, Android Studio will automatically check for updates. Check "show package details" to display available versions of an SDK Tool.	
HTTP Proxy Name Version Status	
Updates OfUL Devering tools Installed	
Usage Statistics CMake Installed	
Android SDK	
Notifications	
Ouick Lists Z Android Emulator 26.1.2 Installed	
Path Variables V Android SDK Platform-Tools 26.0.0 Installed	
Keymap Documentation for Android SDK 1 Not installed	
Google Play APK Expansion library 1 Not installed	
Plugine Google Hay Linensing Library 1 Not installed	
Build Execution Deployment Google Play services 43 Not installed	
Table Coogle USB Driver 11 Not installed	
Instant Apps Development SDK 1.0.0 Not installed	
☐ Intel x86 Emulator Accelerator (HAXM installer) 6.0.6 Not installed	
Support Repository	
Constraintly optimized and a second s	
Sover for ConstraintLayout Instanted	tails
OK Cancel Apply He	lp

图 3 SDK Manager 界面

选择好 Android SDK 的保存路径,并在 SDK Platforms 页面勾选 Android 7.1.1(Nougat) 选择 SDK 版本;在 SDK Tools 页面如上图所示勾选 NDK, CMake, LLDB 等,点击 OK, 下载完成即可(不需要国外代理)。

二、新建项目 Step2_SerialPort

点击图 2 中的 Start a new Android Studio project,弹出新建工程的导航界面,输入应 用名字(首字母大写), Company domain 是生成 java 包的目录,此处最终生成 com/emtronix/ccl/step2_serialport 目录,此目录用来存放应用程序的 Java 源代码,点击 next:

Create New Project	Project ^{tudio}
<u>A</u> pplication name: <u>C</u> ompany domain:	Step2_SerialPort ccl.emtronix.com
Package name:	com.emtronix.ccl.step2_serialport Edit
Project location:	D:\01.public\AndroidStudioProjects\Step2_SerialPort

💮 Create New Project		<u> </u>
Reference Targe	t Android Devices	
Select the form	factors your app will run on	
Different platforms m	ay require separate SDKs	
Phone and Table		
Minimum SDK	API 25: Android 7.1.1 (Nougat)	
	API 19: Android 4.4 (KitKat)]
	API 20: Android 4.4W (KitKat Wear)	
	API 21: Android 5.0 (Lollipop)	
	API 22: Android 5.1 (Lollipop)	
Wear	API 23: Android 6.0 (Marshmallow)	
	API 24: Android 7.0 (Nougat)	
Minimum SDK	API 26: Android	
□ TV		1
Minimum SDK	API 21: Android 5.0 (Lollipop)	
Android Auto		1
	Previous Next Cancel Finish]

勾选 Phone and Tablet,选择 Minimum SDK 版本,小于此版本的 Android 系统可能不能运行该程序。由于这里只安装了 7.1.1 的 sdk,所以选择 Android 7.1.1,点击 Next。

Create New Project	y to Mobile		
Add No Activity	Easic Activity	Bottom Navigation Activity	← Empty Activity
Fullscreen Activity	Google AdMob Ads Activity	Google Maps Activity	Cogin Activity
	, Lineary	Previous	Next Cancel Finish

Android Studio 预设了常用的 activity 模板供开发者选择,这里选择 Empty Activity,点击 Next。

👳 Create New Proj	ect			×
Real Cus	tomize tl	he Activity		
÷		Creates a new e	mpty activity	
		Activity Name:	MainActivity	
			✔ Generate Layout File	
		Layout Name:	activity_main	
			Backwards Compatibility (AppCompat)	
En	npty Activity			
		The name of the	activity class to create	
			Previous Next Cancel	Finish

点击 Finish 就进入了项目的开发界面。

进入开发界面后点击下图所示位置,选择 Project 视图,可以更方便观察整个项目的文件结构。

æ.	tep2_SerialPort - [D:\01.public\AndroidStudio	Projects\Step2_SerialPort] - [app]	\app\src\main\java\com\em
<u>F</u> ile	<u>E</u> dit <u>V</u> iew <u>N</u> avigate <u>C</u> ode Analy <u>z</u> e <u>R</u> e	factor <u>B</u> uild R <u>u</u> n <u>T</u> ools VC <u>S</u>	<u>W</u> indow <u>H</u> elp
	₩ Ø 🗸 🔶 🛣 🖞 🖉 🖉	🔹 💠 🔨 🔂 app 💌 🕨 🧍 🕯	🖡 🖡 🖬 🖳 🧕 🖬
	Step2_SerialPort	🖞 🖿 java 🔪 🖿 com 🔪 🖬 emtronix	🔹 💼 ccl 🕽 💼 step2_serialport
t	👾 Android 🧬 Project Files 🛛 📌 Problems	Project	🧕 activity_main.xml × 🕓
, jo	🔻 🛅 app	🗆 Packages	
÷.	manifests	□ Scratches	
~	🔻 🛅 java	🗹 Android	1 package com
2	🔻 🛅 com.emtronix.ccl.step2_serialpoi	n 🗆 Project Files	2
đ	🕒 MainActivity	🗆 Problems	3 +import
ŝ	Image: Com.emtronix.ccl.step2_serialpoint	Production	5
1	Image: The second se	n 🗆 Tests	
•	res	Local Unit Tests	6 🔯 public clas
res	Oradle Scripts	\Box Android Instrumented Tests	7
aptu			8 @Overrid
10			9 👩 🖯 protect



点击 File>>Project Structure 设置 SDK、JDK、NDK 的路径:

💮 Project Structure	X
+ -	SDK Location
SDK Location Project Developer Servic Ads Authentication Notifications Modules app	Android SDK location: The directory where the Android SDK is located. This location will be used for new projects, and for existing projects that do not have a local.properties file with a sdk.dir property. D:\Android\sdk D:\Android\sdk Image: the start of the start
	OK Cancel

如果没有安装 JDK,可以勾选 Use embedded JDK,使用 Android Studio 内置的 jre 环境。

三、封装串口类

通过以上步骤, Android Studio 已经搭建起了应用开发的基本环境,新建了 Step2_SerialPort项目。为了方便管理,可以把所有串口的操作封装成一个串口类。

1、设计封装串口的 java 类

在 java\com.emtronix.ccl.step2_serialport 目录下新建一个串口类 SerialPort,在这个 类中加载 C/C++动态链接库,并提供公共接口,方便在 MainActivity 中调用。SerialPort 类 中最终都是通过 native 函数调用到 C/C++库,从而访问硬件。

🔻 🛄 app		-	
🕨 🗖 .externalNativeBuild		4	# Sets the minimum version of CMa
🕨 🗖 build		5	
🗖 libs		6	cmake_minimum_required(VERSION 3.
🔻 🗖 src		7	
androidTest			
T main		8	# Creates and names a library, se
▼ □ cpp		9	# or SHARED, and provides the rel
anative-lib.cpp		10	# You can define multiple librari
🔻 🛅 java		11	# Gradle automatically packages s
com.emtronix.ccl.ste	en? serialnort		
	New		🕨 🕻 Java Class
	N/ -		Android resource file
🕨 📴 res	Ж Cu <u>t</u>		Ctrl+X Android resource me
Android Manifest.xml	Copy		Ctrl+C Android resource directory
			E ette

🙍 Create New	Class			×
<u>N</u> ame:	SerialPort			
<u>K</u> ind:	Class			
<u>S</u> uperclass:				
Interface(s):				
<u>P</u> ackage:	com.emtronix.ccl.step2	2_serialport		
<u>V</u> isibility:	• P <u>u</u> blic	◯ Package P <u>r</u> ivate		
<u>M</u> odifiers:	O N <u>o</u> ne	○ <u>A</u> bstract	◯ <u>F</u> inal	
Show Se	lect Overrides <u>D</u> ialog			
		ок	Cancel	lelp

点击 ok 后,开始编辑 SerialPort.java 文件,主要对外提供 public 接口,而这些接口如 果需要访问硬件资源,都需要调用 native 声明的函数, native 声明的函数是 C/C++库提供 的。SerialPort 类中使用 System.loadLibrary()函数加载对应的 so 库。 static {
 System.loadLibrary("SerialPort"); // 调用jni生成的库
}
private native int native_openPort(String dev, String baudrate, String databits, String stopbits
private native int native_closePort(int fd);
private native int native_readPort(int fd, byte[]buf, int len, int milsec);
private native int native_writePort(int fd, byte[] buf, int len);

这四个 native 函数参数介绍如下图:

```
/w函数名 : native openPort,打开串口设备
*输入参数 : String dev -- 设备名 "/dev/ttyS1"~"/dev/ttyS6"
         String baudrate -- 波特率, "9600", "115200"等
*
         String databits -- 数据位, "7"或者"8"
2/2
         String stopbits -- 停止位,"1"或者"2"
*
          String parity — 校验位, "N"、"O"、"E"
2kt
* 返回参数: int --- 文件描述符, 大于0, Success; 否则, 失败
*/
private native int native_openPort(String dev, String baudrate, String databits, :
/*
*函数名 : 关闭串口
* 输入参数: int fd -- 串口设备描述符
* 返回参数: int , 成功返回0, 失败返回-1
*/
private native int native_closePort(int fd);
```

```
/*函数名 : native_readPort,从串口读取数据,没有数据时阻塞,直到超时返回0;有数据时,立即返回
*输入参数: int fd -- 串口设备描述符
        byte[] buf -- 缓存数组,读取到的数据将存于其中
        int len -- 数组长度
 *
        int milsec — 超时时间
 2/0
*返回参数: int -- 大于0, 读取数据长度
             等于0,超时
2/0
 冰
              -1, 表示失败
*/
private native int native_readPort(int fd, byte[]buf, int len, int milsec);
/*函数名 : native_writePort, 向串口发送数据
*输入参数: int fd -- 串口设备描述符
       bvte[] buf -- 发送数据所在数组
 26
        int len -- 数组长度
*返回参数: int -- 大于0, 成功发送数据的长度
              -1,表示失败
*
 */
private native int native_writePort(int fd, byte[] buf, int len);
```

以上 native 函数,英创已经使用 C/C++实现,并编译成了 libSerialPort.so 库文件,用 户只需要将 so 库添加到项目中即可(见下一小节)。

在 SerialPort.java 这个类中,将文件描述符 fd 作为自己的私有变量,再提供 public 函数:

```
public int OpenPort(String dev, String baudrate, String databits, String stopbits, String parity)
{...}
public int ClosePort()
{...}
public int ReadPort(byte[]buf, int len, int milsec) {...}
```

public int Read of (byte[]bui, int len, int milsec) (...

```
public int WritePort(byte[] buf, int len)
{...}
```

每个 public 函数内部调用对应的 native 函数,并对读写计数,即完成了串口类的封装。封装好的串口类可以在 MainActivity 中实例化,然后调用这些 public 函数。

编辑完 SerialPort 类后,如果应用中还需要其他 java 类,像上一样新建 java 类,只有 需要调用 C/C++库的类才需要 native 函数。本例中又添加了查找串口类 SerialPortFinder 和常用工具类 ToolsFunc (自己常用的通用函数)。

2、添加动态链接库

在 app\src\main 目录下新建文件夹 jniLibs\armeabi-v7a,将英创提供的 libSerialPort.so 文件拷贝到这个目录下。

▼ □ app	52
🗖 libs	53
🔻 🛅 src	54 publi
androidTest	55 - {
▼ C New	Android resource file
Link C++ Project with Gradle	File {
Line Copy Path Ctrl+Shift+C	S C++ Class }
I gitig Copy as Plain Text I app. Copy Reference Ctrl+Alt+Shift+(Ctrl + Alt+Shift+(C C/C++ Header File
	- Vector Asset
🕭 New Directory	
Enter new directory name: jniLibs\armeabi-v7a	
 app build libs src androidTest main java 	
 com.emtronix.ccl jniLibs armeabi-v7a 	l.step2_serialport
libSerialPort.	so
▶ 📑 res	
AndroidManifest.xm	l

四、编写 java 应用程序

经过以上步骤之后,SerialPort 类可以通过 JNI 接口访问 C/C++动态库,已经封装好了 对底层的访问,就可以在纯 java 语言环境中开发应用程序了。此时的开发过程,跟不直接 访问硬件的是一样的,在 MainActivity 中实例化一个 SerialPort 类,调用其公共接口就可以 完成对硬件资源的访问了。

1、设计界面



在 app\src\main\res 下有个 layout 文件夹, android 应用的界面布局文件都在这里, 此 应用只有一个界面 activity_main.xml, 默认只有一个文本显示 "Hello World!"。对 Android 界面的设计可以通过 xml 语言修改这个文件,也可以通过可视化编辑窗口拖动相应部件到 Android 模拟窗口。使用 xml 语言修改能更精细的控制界面,具体见源码;使用可视化编辑 窗口能够比较直观方便的实现界面的初步编辑。这里简单介绍下可视化编辑步骤。

a)、新建一个 1024*600 的 Android 虚拟设备

点击编辑窗口中 activity_main.xml 左下角的 Design 图标进入可视化窗口,默认选择的 设备是 Nexus 4,点击下拉窗口,选择 Add Device Definiton...。进入到了 Virtual Device Configuration,点击 Create Virtual Device。



9	Android Virtual Device Manager
	Your Virtual Devices
	Virtual devices allow you to test your application without having to own the physical devices.
	To prioritize which devices to test your application on, visit the <u>Android Dashboards</u> , where you can get up-to-date information on which devices are active in the Android and Google Play ecosystem.

	evice definition					
	Q.		<i>c</i> '			D Pixel C
Category IV	Pixel C	Play Store	9.94"	2560x1800	Density xhdpi	
Near	Nexus 9		8.86"	2048x1536	xhdpi	2560px
hone	Nexus 7 (2012)		7.0"	800x1280	tvdpi	9 94" 1800pv
ablet	Nexus 7		7.02"	1200x1920	xhdpi	
	Nexus 10		10.05"	2560x1600	xhdpi	Size: vlarne
	7" WSVGA (esm6802)		7.0"	1024x600	mdpi	Ratio: notiong Density xhdpi
	7" WSVGA (Tablet)		7.0"	600x1024	mdpi	
	10.1" WXGA (Tablet)		10.1"	800x1280	mdpi	
	7" WSVGA (Esmobol2) 7" WSVGA (Tablet) 10.1" WXGA (Tablet)		7.0" 10.1"	600x1024 800x1280	mapi mdpi mdpi	Densily: xhdpi

进入 Select Hardware 后选择 Tablet,点击左下角的 New Hardware Profile,设置虚 拟设备的参数,如下所示,主要是分辨率、屏幕大小要与实际设备相同。

Hardware Profile Co	nfiguration	×
Confi Android S	igure Hardware Profile ^{studio} hardware profile	
Device Name	esm6802	□ esm6802
Device Type	Phone/Tablet 🔻	
Screen	Screen size: 10.1 inch Resolution: 1024 x 600 px Round	1024px
Memory	RAM: 1024 MB 💙	
Input	Has Hardware Buttons (Back/Home/Menu) Has Hardware Keyboard Navigation Style: None	Size: xlarge Ratio: long Density: ldpi
Supported device states	☐ Portrait ✓ Landscape	
Cameras	 Back-facing camera Front-facing camera 	
Sensors	 Accelerometer Gyroscope GPS Proximity Sensor 	
Default Skin	No Skin 🔽 How do I create a custom hardware skin?	Enables gyroscope support in emulator.
		Previous Next <u>Cancel</u>

点击 Finish 后回到 Select Hardware,选中 esm6802,点击 Next。进入 System Image 页面,选择 download android 7.1.1 的 image,点击 download。下载完成后,在下载页面 点击 Finish,回到 System Image 页面点击 next。

Virtual Device C	Configuration lect Hardware oid Studio					
Choose a de	vice definition	Dlay Store	Circo	Paralutian	Danrity	□ esm6802
TV	esm6802	Play Store	10.1"	1024x600	ldpi	
Wear	Pixel C		9.94"	2560x1800	xhdpi	1024px
Phone	Nexus 9		8.86"	2048x1536	xhdpi	10.1" 600px
Tablet	Nexus 7 (2012)		7.0"	800x1280	tvdpi	
	Nexus 7		7.02"	1200x1920	xhdpi	Size: xlarge
	Nexus 10		10.05"	2560x1600	xhdpi	Ratio: long Density: Idpi
	7" WSVGA (esm6802)		7.0"	1024x600	mdpi	
	7" WSVGA (Tablet)		7.0"	600x1024	mdpi	
	10.1" WXGA (Tablet)		10.1"	800x1280	mdpi	
New Hardware	e Profile Import Hardy	vare Profiles			Ø	Edit Device
					Previ	ious Next Cancel Finish Help

Virtual Device Configuration System Im Android Studio elect a system image	age			
Recommended x86 Images Release Name	Other Images	ABI	Target	Nougat
0 Download	26	x86	Android 8.0 (Google APIs)	
Nougat Download	25	x86	Android 7.1.1 (Google APIs)	API Level
Nougat Download	24	x86	Android 7.0 (Google APIs)	25
Marshmallow Download	23	x86	Android 6.0 (Google APIs)	Android
Lollipop Download	22	x86	Android 5.1 (Google APIs)	7.1.1
				System Image X86
				We recommend these images because they run fastest and support Google APIs. Questions on API level?
A system image must be sel	ected to continue.		25 Previ	ous Next <u>C</u> ancel Finish Help

确认配置后,点击 Finish。这里并不需要运行虚拟设备,关闭 Your Virtual Devices 窗口。

🕭 Virtual Device Configuration	×
Android Virtual Device (AVD)	
AVD Name esm6802 API 25	Default Orientation
Esm6802 10.1 1024x600 ldpi Chang	e
Nougat Android 7.1.1 x86 Chang	you can also rotate the device screen.
Startup orientation Portrait	
Emulated Graphics: Automatic 🔻	
Device Frame 🗌 Enable Device Frame	
Show Advanced Settings	
	Previous Next Cancel Finish Help

Type Name Pla	ay Store Resolution 1024 × 6	API 25	Target Android	CPU/ABI	Size on Disk	Actions
esm6802	1024 × 6	25	Android			richerto
				x86	650 MB	> 2

b) 可视化编辑界面

回到 activity_main.xml 的编辑页面,再次点击设备下拉菜单,就可以选择刚才建立的 esm6802 了。

🗌 Nexus 4 🕶 🚈 25 🗸 🔘 AppTheme 🛛 🜐 Languag
📮 3.7, 480 × 800, hdpi (Nexus One)
4.0, 480 × 800, hdpi (Nexus S)
4.7, 720 × 1280, xhdpi (Galaxy Nexus)
✓ 4.7, 768 × 1280, xhdpi (Nexus 4)
5.0, 1080 × 1920, xxhdpi (Nexus 5)
📮 5.0, 1080 × 1920, xxhdpi (Pixel)
5.2, 1080 × 1920, 420dpi (Nexus 5X)
5.5, 1440 × 2560, 560dpi (Pixel XL)
5.7, 1440 × 2560, 560dpi (Nexus 6P)
6.0, 1440 × 2560, 560dpi (Nexus 6)
7.0, 800 × 1280, tvdpi (Nexus 7 2012)
7.0, 1200 × 1920, xhdpi (Nexus 7)
8.9, 2048 × 1536, xhdpi (Nexus 9)
9.9, 2560 × 1800, xhdpi (Pixel C)
10.1, 2560 × 1600, xhdpi (Nexus 10)
🔷 320 × 290, tvdpi (Round Chin)
280 × 280, hdpi (Square)
320 × 320, hdpi (Round)
🖵 1080p, 1920 × 1080, xhdpi (TV)
720p, 1280 × 720, tvdpi (TV)
AVD: esm6802API_25
Generic Phones and Tablets
Add Device Definition

点击左上角的 show design 图标,就可以得到如下所示的大图显示。

Show Design va × 🝺 com_emtronix_ccl_step2_serialport_SerialPort.h ×	📴 esm6802GPIO.cpp × 💿 app × 📄 CMakeLists.txt × 🕓 SerialPort.java × 🕓 T	bc
· I [←] 🗎 🔢 🛇 · 🔲 esm6802 · 🗯 25 · ◎ AppTheme ⊕Language ·	· []]•	
≥w Ø Ø ‡× +* 8 #∃• ≡• ∃•	⊝ 85% ⊕ ⊑	1
*But100	200 300 400 500	
Box -		
edTe	▼ ▮ 7:00	-
r Step2_SerialPort		
rssb rssB		1
ar		
ar (L) Cont		
Bar		
8		
ext		
	Halls World	
	Heilo Wolld:	
8		
·- 8←		
xtView		
		Ĺ

这时就可以拖动左边窗口中需要的组件到虚拟设备,并在右边 property 窗口设置具体

参数了。

activity_main	n.xml × C MainActivity.java	×	com_	emtronix	_ccl_step2_s	erialport	t_SerialPort.h	× 🔤 esm6802	GPIO.cpp ×	💽 app 🛛	CMakeLists.txt >	C SerialPort.java ×	C ToolsFun	c.java ×	C Sei	ialPortFinder.java	×
Palette	Q, ∲- +	E		0-	esm6802 -	25 -	C AppTheme	🜐 Language 🕶						Prope	rties	Q, 🚅 🔅-	÷₿
All Widgets Text Layouts	Ab TextView K Button ToggleButton CheckBox	ø	6	; +* : ;	B ‡8 - =	• <u>I</u> •	00	200		30	0	⊖ 77% (400		ID	-	button	
Containers Images Date Transitions Advanced Google Design AppCompat	 RadioButton CheckedTextView Spinner ProgressBar ProgressBar (Horizonta SeekBar (Discrete) QuickContatBadge 			Step2_	SerialPort		OFF						▼ 8 200			» × «	
	KatingBar Switch Space Plain Text	- 8											•	layou layou	t_widtł t_heig.	ap_content =	
	Password Password (Numeric) E-mail Phone				витто 1× 8	N			Hello World	1				Butte style back	on ground	buttonStyle 🔻	
	Postal Address Multiline Text Time	- 8												back state	groun ListAni.		
_	Date Number	-												eleva visibi onCli	tion lity ck	none	
	um													Text	View		-
Component Tree	\$- +	1						4	0					text		Button	
Ab came	intLayout nle text (TextView) - "Hello Wo												11	≁ te:	đ		
✓ chec	kBox - "CheckBox"													conte	ntDes.		
🗾 togg	leButton - "ToggleButton"													⊳ te:	tApp	.Widget.Button	
≓ spin	ner													Favo	rite At	tributes	-
oK butt	on - "Button"													visibi	lity	none	
																	-

不管是可视化拖动设计的界面,还是使用 xml 语法编写的界面,最后都可以在 design 页面看到设计的最终样子。Step2_SerialPort 的界面如下图所示。

口设置	接收区:	I,	的能区设置	
4口号: 2特率: 9600 2据位: 7	•	r F	 HEX显示 HEX发送 □动发送周期(ms): 1000 	
▶止位: 1	•	5	使提功能区	
短句: None	•		清除发送区	清除接收区
			计数清零	关闭显示
打开串口 关闭串口			自动发送	手动发送
态提示				
隐: aiting for setting serial!	发送区:			
X:0 TX:0	input what you want to send			

2、编写 java 代码,实现应用的业务逻辑

Android 系统与用户交互的界面通过 Activity 加载呈现给用户,在 Activity 中实现对界面操作的监听,并完成相应的业务逻辑。onCreate 函数就是 Activity 创建时候回调函数,一般 在这里找到 xml 界面中的元部件,设置元部件的初始参数,并设置事件监听。

@Override protected void onCreate(Bundle savedInstanceState) { super. onCreate(savedInstanceState); setContentView(R.layout.activity_main); init_create(); init_findView(); 2 init_config(); init_ui_state(); setOnlistener(); }

需要注意的是,应用程序的主线程一般又称为 UI 线程。UI 线程 5s 得不到响应,就会 产生 ANR (Application Not Responding)错误。所以,一般 UI 线程主要是监听界面操作的,

一切耗时阻塞的操作都要放在其他线程中实现,通过 handler 发送消息给 UI 线程更新 UI,这样才能使应用运行流畅稳定。因此,在 MainActivity 中分别声明了串口接收线程和串口发送线程:

```
private RecvThread mRecvThread ;
private SendThread mSendThread;
```

serialPort 是前面封装的 SerialPort.java 类的实例化对象,当调用 serialPort.OpenPort()后, 实例化一个 RecvThread 对象并启动后,就可以使用 ReadPort 接收串口消息了。

```
case R.id.bt_open:
if(!mIsSerialOpen) {
    serialPort.OpenPort(m_dev, m_baudrate, m_databits, m_stopbit, m_parity);
    mIsSerialOpen = true;
```

//启动接收线程

```
mRecvThread = new RecvThread(mHandler);
mRecvThread.start();
```

RecvThread 类继承 Thread,在构造函数中传入主线程的 handler,当读取到数据的时候,就通过此 handler 发送消息给主线程,完成 UI 刷新。

```
public RecvThread( Handler handler)
ſ
    this.mHandler = handler;
   mBufSize = 2048:
    mBuf = new byte[mBufSize];
   mTimeUs = 300000; //300ms
}
@Override
public void run() {
    while (!isInterrupted()) {
        int nbytes = serialPort.ReadPort(mBuf, mBufSize, mTimeUs);
        if(nbytes > 0) {
           //发送msg给主线程
           Message msg = new Message();
           msg.what = ToolsFunc. MSG_UART_READ_DATA;
           msg. obj = new String(mBuf, 0, nbytes);
           mHandler.sendMessage(msg);
        }
    }
   Log. i(TAG, "recv thread exit");
}
```

java 部分的具体实现,请参考源码。Android 应用开发内容很多,更详细的资料请参考 开发者网站学习:https://developer.android.google.cn/index.html 。

3、编译项目

编辑好 java 部分代码后,就可以使用 Ctrl+F9 或者 Build>>Make Project 编译项目了。 然后在 Message 窗口可以看到编译结果。

Messag	es Gradle Build
■ 3 × 4 + 1 1 ?	 Gradle tasks [:app:generateDebugSources, :app:mockableAndroidJar, :ap BUILD SUCCESSFUL Total time: 1.2 secs 0 errors 0 warnings See complete output in console
🧐 ТОВ	O 🏺 <u>6</u> : Android Monitor 📃 <u>0</u> : Messages 🗵 Terminal

五、安装应用到 ESM6802 进行调试

1、运行应用

编译通过后可以将应用部署到 ESM6802 上运行。通过 usb 线连接 PC 机和 ESM6802, 点击工具栏中的 Run app 按键,在弹出的对话框中选择已连接的 ESM6802,点击 OK,就 会把 Step2_SerialPort 安装到 ESM6802 中,并运行起来。



😨 Select Deployment Target	×
Connected Devices	
Freescale ESM6802-MX6DL (Android 7.1.1, API 25)	
Available Virtual Devices	
🔳 esm6802 API 25	
<u>Create New Virtual Device</u>	Don't see your device?
Use same selection for future launches	ОК Cancel

此时点击左下角的 Android Monitor 可以查看的应用程序的运行过程中,程序打印的 log

信息。

And	roid Me	onitor
	📮 Fr	reescale ESM6802-MX6DL Android 7.1.1, API 25 🔽 com.emtronix.ccl.step2_serialport (1851) 🔽
Ō	∎≊ lo	lgcat Monitors →*
	-	08-10 05:23:39.851 1851-1851/com.emtronix.ccl.step2_serialport D/SerialPortFinder: 149/dev
Q.		08-10 05:23:39.851 1851-1851/com.emtronix.ccl.step2_serialport D/SerialPortFinder: Found new device: /dev/tty52
		08-10-05:23:39.851-1851-1851/com.emtronix.ccl.step2_serialport_D/SerialPortFinder: Found_new_device: /dev/ttyS3
8	+	08-10 05:23:39.851 1851-1851/com.emtronix.ccl.step2_serialport D/SerialPortFinder: Found new device: /dev/ttyS4
O		08-10 05:23:39.852 1851-1851/com.emtronix.ccl.step2_serialport D/SerialPortFinder: Found new device: /dev/ttyS1
2	-	08-10 05:23:39.852 1851-1851/com.emtronix.ccl.step2_serialport D/SerialPortFinder: Found new device: /dev/tty56
1	<u>9</u>	08-10 05:23:39.852 1851-1851/com.emtronix.ccl.step2_serialport D/SerialPortFinder: Found new device: /dev/tty55
	1	08-10 05:23:39.852 1851-1851/com.emtronix.ccl.step2_serialport D/SerialPortFinder: Found new device: /dev/tty50
		08-10-05:23:39.853-1851-1851/com.emtronix.ccl.step2_serialport_I/SerialPortFinder:_console_name_ttyS0
	G	08-10 05:23:39.855 1851-1851/com.emtronix.ccl.step2_serialport I/TestSerialPort: /dev/ttyS1
	-∰E	08-10 05:23:39.855 1851-1851/com.emtronix.ccl.step2_serialport I/TestSerialPort: /dev/ttyS2
	-	08-10 05:23:39.855 1851-1851/com.emtronix.ccl.step2_serialport I/TestSerialPort: /dev/ttyS3
	1	08-10 05:23:39.855 1851-1851/com.emtronix.ccl.step2_serialport I/TestSerialPort: /dev/ttyS4
		08-10 05:23:39.855 1851-1851/com.emtronix.ccl.step2_serialport I/TestSerialPort: /dev/ttyS5
		08-10 05:23:39.855 1851-1851/com.emtronix.ccl.step2_serialport I/TestSerialPort: /dev/ttyS6
		08-10 05:23:39.973 1851-1851/com.emtronix.ccl.step2_serialport I/imx6.gralloc: open gpu gralloc module!
		08-10 05:23:40.241 1851-1874/com.emtronix.ccl.step2_serialport I/OpenGLRenderer: Initialized EGL, version 1.4
		08-10 05:23:40 241 1851-1874/com emtronix col sten2 serialnort D/OnenGERenderer: Swan hebavior 2
- 🌬 4	: Run	🐏 TODO 🛛 🖷 💁 Chdroid Monitor 🛛 🧱 🖸: Messages 💷 Terminal

2、调试应用

如果,在运行过程中出现问题,一般可以通过日志信息查看错误原因,如果需要单步调试可以 Ctrl+F12 或者点击工具栏的停止按钮,停止运行应用。



然后 Shift+F9 或者点击调试按钮,进入调试模式,在弹出的对话框中选择已连接设备, 点击 OK。



这里在点击"打开串口"的程序里设置断点,左键点击所在行的最左边就可以设置断点。 然后点击 ESM6802 上的"打开串口按键",就会运行到断点处停止,等待用户调试。



用户可以使用红框处的按键单步调试,也可以直接按快捷键。

textine.xm	184	case R. 10. DT_open:	
V 🗖 layout	185 🥪	if(!mIsSerialOpen) {	
o activity_main.xml	186	serialPort.OpenPort	(m_dev, m_baudrate,
actionbar menu.xml	187	mIsSerialOpen = tr	ue;
mipmap-hdpi	188	updateSerialConfig(m	IsSerialOpen);
🕨 🖻 mipmap-mdpi	189	<pre>tv_recv. setText("")</pre>	;
E mipmap-xhdpi	190	mTxCount = 0	
Debug: La app app-java			
		1 -	
Frames		→* Variables	
"main"@4,370 in group "main": RUNNING	▼ ↑ ↓	This = {MainActivity\$BtOnClickListener@	4850} I support v7 widget AppComps
 onClick:185, MainActivity\$BtOnClickListener (com.emtronix.ccl.step2_ porformClick:5627, View (android view) 	_serialport)	► Image: A state of the sta	independent mindgetakepeeninge
run:22429. View PerformClick (android.view)		► 🐼 m_parity = "None"	
handleCallback:751, Handler (android.os)		• 6/ m_baudrate = "115200"	
dispatchMessage:95, Handler (android.os)		w misSerialOpen = faise	
k loop:154, Looper (android.os)		► & m_stopbit = "1"	
invoke:-1. Method (iava.lang.reflect)		▶ ŵ⁄ m_dev = "/dev/ttyS1"	
× run:886, ZygoteInit\$MethodAndArgsCaller (com.android.internal.os)			
? main:776, ZygoteInit (com.android.internal.os)			
+9 Restart Activity			
🔻 Step Over		F8	
Force Step Over		Alt+Shift+F8	
🔰 Step Into		F7	
Force Step Into		Alt+Shift+F/	
💁 Smart Step Into		Shift+F7	
		cL16 - F0	
step Ou <u>t</u>		Shitt+F8	
[™] (Run to <u>C</u> ursor		Alt+F9	
Y Force Run to Cursc	or	Ctrl+Alt+F9	
🚡 Drop <u>F</u> rame			
Pause Program			
Resume Program		F9	

调试完成后,与退出运行模式一样,点击停止按钮退出调试。

六、开机自启动用户的应用程序

用户开发好应用,有时需要应用在每一次开机的时候自启动。Android 系统在启动完成的时候,会发生 android.intent.action.BOOT_COMPLETED 的广播,接收到此广播的应用就可以自己启动。所以,用户可以在应用中添加一个广播接收器,监听此广播,以开机启动此应用。

1、新建 BootBroadcastReceiver 类

在 com.emtronix.ccl.step2_serialport 包中新建 BootBroadcastReceiver 类,继承广播

监听类 BroadcastReceiver。重写 onReveive 函数,在此函数中判断 Intent 是否是 BOOT_COMPLETE,如果为真就启动 MainActivity。

```
@Override
public void onReceive(Context context, Intent intent) {
    // TODO: This method is called when the BroadcastReceiver is receiving
    // an Intent broadcast.
    boolean bootCompleted;
    String action = intent.getAction();
    bootCompleted = Intent.ACTION_BOOT_COMPLETED.equals(action);
    if(bootCompleted) {
        Intent mainActivityIntent = new Intent(context, MainActivity.class);
        mainActivityIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        context.startActivity(mainActivityIntent);
    }
```

```
2、向系统注册广播
```

BroadcastReceiver 属于系统级监听,需要向系统注册。这里可以编辑 app\src\main 目 录下的 AndroidManifest.xml 文件,在 application 标签中注册 receiver,并添加接收此广播 的权限

<uses-permission</pre>

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

就可以了,如下图所示。android:priority="999"可以增加此应用的优先级,可以优先接收到 广播消息,更早启动。



系统为防止恶意程序的自启动,第三方安装的应用软件在最开始都是 stop 状态,这种 状态下是不能自启动的。当用户自己手动点击,运行过一次后,系统认为此应用已经过用户 确认安全,之后开机就能自动运行了。

七、应用程序生成 apk

应用程序调试完毕,没有问题后会发布 release 版本的安装包。android 应用程序安装 包的后缀是.apk。发布后的 apk 文件通过 U 盘或者网络下载到 esm6802 后,点击即可完成 安装。

1、生成签名

每一个应用都需要一个签名表明应用发布者的身份,只有同一个签名才能覆盖此应用。 以此可以防止其他发布者以相同的应用名覆盖原有应用,确保应用的升级等操作只能由同一 个发布者完成。

首先,点击 Build>>Generate Signed APK...,在弹出的对话框中选择 Create new。



💮 Generate Signed AP	
Key store path:	
	Create new Choose existing
Key store <u>p</u> assword:	
K <u>e</u> y alias:	
Key pass <u>w</u> ord:	
<u>R</u> emember passw	rords
	Previous Next Cancel Help

		<u></u>	
167			
168	default:		
169	break;		🕭 Choose keystore file 📃
New Key Store		X	Save as *.jks
Key store path:			🗌 😤 🖻 🗟 🔀 🗙 💋 🗐 🛛 Hide pat
Password	Confirm:		D:\Android\signedKey
			D3.download
Кеу		[a]í	▶ 🗅 04.主板手册
Alias:	kev0		D5.Virtual Machine
<u> </u>			D6.works
Pa <u>s</u> sword:	<u>C</u> onfirm:		107.kernel
Validity (years	25 🔺		D8.tools
Tanand Ocara		6	D9.ebook
Certificate			10.Program Files
<u>First</u> and Las	t Name:	ew	▶ 1 360驱动大师目录
Organization	al Unit:		🔻 🗖 Android
			Android Studio
Organization			AndroidProject
City or <u>L</u> ocal	ity:		▶ 🗖 sdk
Charles and Dara	·		🗖 signedKey 🛛 2
State or Prov	/ince:		apache-tomcat-6.0.16
Country Cod	e (<u>X</u> X):		Drag and drop a file into the space above to quickly locate it in the tree
			File name: emtronix_ccl_release.jks 3 jks v
	ок	Cancel	
			4 Cancel Help

💮 New Key Store					×			
Key store path:	D:\Andro	oid\signedKey\em	tronix_ccl_rele	ase.jks				
<u>P</u> assword:	•••••		Co <u>n</u> firm:					
Кеу	Key							
<u>A</u> lias:	cclre	ease						
Pa <u>s</u> sword:	•••••	•	<u>C</u> onfirm:	•••••				
<u>V</u> alidity (years): 25	•						
Certificate								
First and Last	t Name:	ccl						
Organization	al Unit:	ccl						
O <u>rg</u> anization	:	emtronix						
City or <u>L</u> ocali	ty:	chengdu						
S <u>t</u> ate or Prov	ince:							
Country Code	e (<u>X</u> X):	ZH						
				ОК	Cancel			

在 Key store path 中点击最右边的...,在弹出的对话框中选择要存放签名的路径,并在 File name 中输入发布者想要的文件名字,点击 OK。回到 New Key Store 页面,设置 Store 的密码,以及 key 相关的密码信息等。

2、发布签名应用

回到 Generate Signed APK 页面,点击 Next。在 Build Type 中选择 release 版, Signature Versions 勾选 V1,点击 Finish。

💮 Generate Signed API	< X
Key store path:	D:\Android\signedKey\emtronix_ccl_release.jks
	Create new Choose existing
Key store <u>p</u> assword:	•••••
K <u>e</u> y alias:	cclrelease
Key pass <u>w</u> ord:	•••••
<u>R</u> emember passw	rords
	Previous Next Cancel Help

💮 Generate Signed APK	×
Note: Proguard settings are	specified using the Project Structure Dialog
APK <u>D</u> estination Folder:	D:\Android\AndroidProject\Step2_SerialPort\app
<u>B</u> uild Type:	release 🔻
<u>F</u> lavors:	
	No product flavors defined
<u>S</u> ignature Versions:	✓ V1 (Jar Signature)
	Previous Finish Cancel Help

34 / 40

最后,会在 app 目录下生成 app-release.apk 文件。



在 app 目录下的 build.gradle 文件中可以指定发布 apk 文件的名字,在 android 节点中添加

如下代码,重新生成 signed apk 即可。

 Li unp 		
🖿 libs	32	applicationVariants.all { variant ->
Image: Second	33	variant.outputs.each { output ->
.gitignore	34	def outputFile = output outputFile
🗋 app.iml		
📀 build.gradle	35	def fileName
CMakeLists.txt	36	<pre>if (outputFile != null && outputFile.name.endsWith('.apk')) {</pre>
proguard-rules.pro	37	<pre>if (variant.buildType.name.equals('release')) {</pre>
🏙 step2_serialport-release-1.0.apk	38	fileName = "step2_serialport-release-\$ {defaultConfig.versionName}.apk"
🕨 🗖 build	39	<pre>} else if (variant, buildType, name, equals('debug')) {</pre>
🕨 🗖 gradle	10	
gitignore	40	filename - step2_serialport-debug-) (defaultConfig. VersionName). apk
🖲 build.gradle	41	}
📊 gradle.properties	42	output.outputFile = new File(outputFile.parent, <mark>fileName</mark>)
🖹 gradlew	43	}
🗐 gradlew.bat	44	
📓 local.properties	45	}
	10	

八、应用程序的安装与卸载

1、安装应用程序

a、通过U盘

将最终生成了 step2_serialport-release-1.0.apk,将此文件通过 U 盘拷贝到 esm6802 上,通过 AnExplorer 找到此文件,点击即可完成安装。

N							⇔i⊠ 🛿 7:44
	SanDisk						₹ 1
STOR	AGE	^ 🔼	step2_serialport-release-1.0.a	pk		1.45 MB	3:22 PM
A	Home		apache-tomcat-6 () 16 zin			84.62 MB	Jul 31
	Internal Storage	0				011021110	:
ψ	SanDisk	0	test_gpio	14 fil	es		Jul 26
	Phone	0	6802	2 file	ŝ		Jul 18
# NETW	Root	^ O	System Volume Information	2 file	5		Jun 14
	Transfer to PC /storage/emulated/0	C	uboot-imx-android.tar.bz2			19.14 MB	Jun 14
->←	Connections		Sdk	9 file	6		Jun 7
APPS							:
			0				
N							«»؛ 🖹 🖪 7:44
N	SanDisk		20 anial Darat				(२) ▲ 7:44
N STORA	SanDisk GE	tart Test	tSerialPort			1.45 MB	<00 1 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2
N STORA	SanDisk IGE Home	Do you wan	tSerialPort at to install this application? It ess.	does not require	any	1.45 MB	↔ <u>א</u> א 7:44 = : 3:22 PM
	SanDisk IGE Home Internal Storage	Do you wan special acce	tSerialPort at to install this application? If ess.	: does not require	e any	1.45 MB 84.62 MB	↔ 🖹 🖬 7:44 〒 : 3:22 PM : Jul 31 :
STORA	SanDisk GE Home Internal Storage SanDisk	Do you wan special acco	tSerialPort It to install this application? If ess.	: does not require	e any	1.45 MB 84.62 MB	(가) 입 7:44 코 1 코 1 3:22 PM 1 Jul 31 1 Jul 26 1
STORA	SanDisk GE Home Internal Storage SanDisk Phone	Do you wan special acco	tSerialPort It to install this application? It ess.	does not require	e any	1.45 MB 84.62 MB	↔ ₩ 0 7:44
STORA A U U U U U U	SanDisk GE Home Internal Storage SanDisk Phone Root	Do you wan special acce	tSerialPort at to install this application? If ess.	: does not require	any	1.45 MB 84.62 MB	↔ ₩ 0 7:44
STORA STORA	SanDisk GE Home Internal Storage SanDisk Phone Root	Do you wan special acco	tSerialPort at to install this application? It ess.	: does not require	any	*** 1.45 MB 84.62 MB	↔ ₩ 0 7:44
STORA STORA	SanDisk GE Home Internal Storage SanDisk Phone Root JRK Transfer to PC /storage/emulated/0	Do you wan special acco	tSerialPort at to install this application? It ess.	: does not require	any	1.45 MB 84.62 MB	↔ ₩ 0 7:44
N STORA	SanDisk GE Home Internal Storage SanDisk Phone Root SRK Transfer to PC /storage/emulated/0 Connections	Do you wan special acco	tSerialPort at to install this application? If ess.	does not require	e any	1.45 MB 84.62 MB 19.14 MB	•>> •> •> •
N STORA	SanDisk GE Home Internal Storage SanDisk Phone Root JRK Transfer to PC /storage/emulated/0 Connections	Test Do you wan special acco	tSerialPort at to install this application? It ess.	does not require	any	1.45 MB 84.62 MB 19.14 MB	•>> •> •> • • • •> •> • • • •> •> •> • • •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •> •>

b、通过 USB 线,直接访问 esm6802 内部存储器

在屏幕顶端向下滑动,在弹出的对话框中点击 USB charging this device



在弹出的对话框中选择 Transfer files,这样在 PC 上就可以直接访问 esm6802 的内部 存储器了。把 step2_serialport-release-1.0.apk 拷贝到 esm6802 的 Download 目录,打开 AnExplorer,点击 Internal Storage,进入 Download 目录,点击 step2_serialport-release-1.0.apk,即可进行安装。



N									«≫j 🖉 🦻	7:51
	Download	-					٩		Ŧ	÷
STORA	GE	^		step2_serialport-re	elease-1.0.apk		1.4	5 MB	7:50 A	ам :
f	Home									·
=	Internal Storage									
Ŷ	SanDisk									
	Phone									
#	Root									
NETWO	DRK	^								
	Transfer to PC /storage/emulated/0									
→←	Connections								4	
APPS		^								
			<	1	0					

通过 apk 文件安装的应用,安装后,处于 stop 状态,如果需要开机启动,请手动打开 一次,激活此应用,才能确保下次开机自启动。

2、卸载应用程序

如果用户需要卸载应用程序,可以有两种方法。

比较快捷的方法是在 home 页面,长按应用程序图标,会回到桌面,此时将图标拖动到 左上角的回收站,会弹出卸载对话框,点击 OK 即可自动完成应用卸载。



另一种方法是在 home 界面点击 Settings,选择 Apps 管理项,在里边找到要卸载的应用,点击进去,在 App info 中点击 UNINSTALL,同样弹出卸载对话框,点击 ok 即可完成 卸载。

Ν				÷>با🗷 💈 8:04
Setti	ngs			م
Ð	Display Adaptive brightness is OFF			
٠	Notifications All apps allowed to send			
	Sound Ring volume at 71%			
۲	Apps 23 apps installed			
	Storage 2.31 GB of 3.64 GB used			
	Battery 100% - Charging on AC			
	Memory			
	\triangleleft	0		
N				«≫با ∑ ∮ 8:05
÷	App info			
UART	TestSerialPort version 1.0			
	UNINSTALL		FORCE	STOP
Stora 4.68 M	ge IB used in Internal storage			
Data No da	usage ta used			
Perm No pe	issions missions requested			
Notifi	cations			
Open	by default			
	5	0		