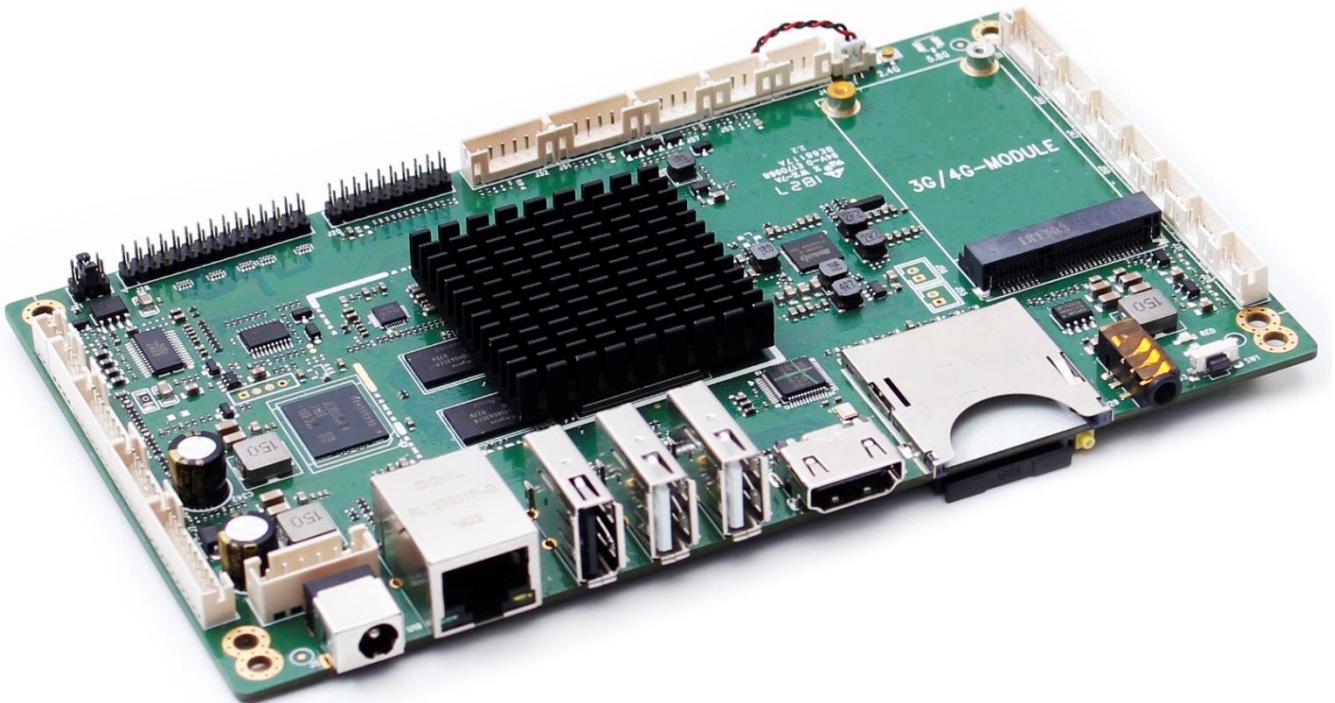


USR-ECB400A 软件设计手册

文件版本： V1.0.04



目录

USR-ECB400A 软件设计手册	1
目录	2
1. 背景	6
2. 准备工作	6
2.1. jar 包导入	6
3. API 使用说明	7
3.1. 通用 api 使用说明	7
3.2. 定时器相关使用说明	7
4. 获取系统信息	8
4.1. public String getApiVersion()	8
4.1.1. 描述：获取目前 API 的平台-版本-日期信息	8
4.2. public String getAndroidModle()	8
4.2.1. 描述：获取目前设备的型号	8
4.3. public String getAndroidVersion ()	9
4.3.1. 描述：获取目前设备的 android 系统的版本	9
4.4. public String getRunningMemory ()	9
4.4.1. 描述：获取设备的硬件内存大小容量	9
4.5. public String getInternalStorageMemory ()	9
4.5.1. 描述：获取设备的硬件内部存储大小容量	9
4.6. public String getFirmwareVersion ()	10
4.6.1. 描述：获取设备的固件 SDK 版本	10
4.7. public String getKernelVersion ()	10
4.7.1. 描述：获取设备的固件内核版本	10
4.8. public String getAndroidDisplay ()	11
4.8.1. 描述：获取设备的固件系统版本和编译日期	11
4.9. public String getFirmwareDate()	11
4.9.1. 描述：获取固件编译的时间	11
5. 网络	12
5.1. public String getEthMacAddress()	12
5.1.1. 描述：获取第一块以太网的 MAC 地址	12
5.2. public String getEthMode()	12
5.2.1. 描述：获取当前以太网的模式	12
5.3. public void setDhcpIpAddress(Context context)	13
5.3.1. 描述：设置以太网模式为动态获取	13
5.4. public void setStaticEthIPAddress(...)	13
5.4.1. 描述：设置以太网的模式为静态地址	13
5.5. public String getStaticEthIPAddress()	14
5.5.1. 描述：获取以太网的静态 IP 地址	14
5.6. public String getNetMask()	14

5.6.1. 描述: 获取以太网卡的子网掩码.....	14
5.7. public String getGateway ()	15
5.7.1. 描述: 获取当以太网卡的网关.....	15
5.8. public String getEthDns1 ().....	15
5.8.1. 描述: 获取以太网卡的 DNS1	15
5.9. public String getEthDns2().....	15
5.9.1. 描述: 获取当以太网卡的 Dns2.....	15
5.10. public int getCurrentNetType()	16
5.10.1. 描述: 获取当前网络连接的类型.....	16
5.11. public boolean getEthStatus ()	17
5.11.1. 描述: 获取设备以太网的状态	17
5.12. public void ethEnabled(boolean enable)	17
5.12.1. 描述: 开关以太网	17
6. 存储管理	18
6.1. public String getSDcardPath()	18
6.1.1. 描述: 获取外部存储 SD 卡路径.....	18
6.2. public String getUSBStoragePath(int num)	18
6.2.1. 描述: 获取外部存储 U 盘路径	18
6.3. public void unmountVolume(String path)	19
6.3.1. 描述: 卸载外部存储	19
6.4. public void mountVolume(String path)	19
6.4.1. 描述: 挂载外部存储, 如果外部存储存在的话	19
7. 硬件接口	20
7.1. public String getUartPath(String uart)	20
7.1.1. 描述: 获取串口绝对路径.....	20
7.2. public String getGPIOStatus(String io).....	20
7.2.1. 描述: 获取 IO 口输入状态.....	20
7.3. public void setGPIOStatus(String value, String io)	21
7.3.1. 描述: 设置 IO 口输出状态.....	21
8. 显示	21
8.1. public void takeScreenshot(Context context, String path)	21
8.1.1. 描述: 截取当前全屏为 jpg 格式图片并重命名到相应位置	21
8.2. public void rotateScreen(Context context, String degree)	22
8.2.1. 描述: 设置屏幕逆时针旋转 N 角度	22
8.3. public int getDisplayWidth(Context context)	22
8.3.1. 描述: 获取显示屏分辨率宽 X 像素	22
8.4. public int getDisplayHeight(Context context)	23
8.4.1. 描述: 获取显示屏分辨率高 Y 像素	23
8.5. public void hideNavBar(boolean hide)	23
8.5.1. 描述: 设置显示或者隐藏虚拟按键.....	23
8.6. public boolean getNavBarHideState().....	24
8.6.1. 描述: 获取当前虚拟按键状态	24
8.7. public void setSlideShowNavBar(boolean flag)	24

8.7.1. 描述: 禁止滑出虚拟按键.....	24
8.8. public boolean isSlideShowNavBarOpen ().....	25
8.8.1. 描述: 获取滑出导航栏状态.....	25
8.9. public void setSlideShowNotificationBar(boolean enable)	25
8.9.1. 描述: 打开或关闭滑出通知栏.....	25
8.10. public boolean isSlideShowNotificationBarOpen().....	26
8.10.1. 描述: 滑出通知栏是否打开	26
8.11. public void turnOffBacklight().....	26
8.11.1. 描述: 熄灭屏幕, 只关背光, 却不进入休眠, 软件继续运行	26
8.12. public void turnOnBacklight()	27
8.12.1. 描述: 开启背光.....	27
8.13. public void turnOnHDMI();	27
8.13.1. 描述: 打开 HDMI 输出信号	27
8.14. public void turnOffHDMI();	27
8.14.1. 描述: 关闭 HDMI 输出信号	27
9. 开关机、定时	28
9.1. public String getPowerOnMode()	28
9.1.1. 描述: 获取当前定时开关机模式	28
9.2. public void setPowerOnOff(int[] powerOnTime, int[] powerOffTime).....	28
9.2.1. 描述: 设置一组开关机时间	28
9.3. public void setPowerOnOffWithWeekly(int[] powerOnTime, int[] powerOffTime,int[] weekdays)	29
9.3.1. 描述: 设置一组周期性的开关机时间	29
9.4. public String getPowerOnTime()	29
9.4.1. 描述: 清除开关机时间	29
9.5. public String getPowerOffTime().....	30
9.5.1. 描述: 获取设置的开机时间	30
9.6. public String getLastestPowerOnTime()	30
9.6.1. 描述: 获取上一次实际的开机时间	30
9.7. public String getLastestPowerOffTime()	30
9.7.1. 描述: 获取上一次实际的关机时间	30
9.8. public void clearPowerOnOffTime()	31
9.8.1. 描述: 清除开关机时间	31
9.9. public void shutDown().....	31
9.9.1. 描述: 正常关闭系统	31
9.10. public boolean reboot (String reason)	32
9.10.1. 正常重启系统	32
10. 硬件看门狗控制相关	33
10.1. 目录结构.....	33
10.2. public static native int open()	34
10.2.1. 描述: 打开看门狗控制句柄	34
10.3. public static native void close().....	34
10.3.1. 描述: 关闭看门狗控制句柄	34
10.4. public static native int enableWatchdog()	35

10.4.1. 描述: 使能看门狗.....	35
10.5. public static native int disableWatchdog()	35
10.5.1. 描述: 禁用看门狗功能.....	35
10.6. public static native int feetDog(int time).....	36
10.6.1. 描述: 设置喂狗时间.....	36
10.7. public static native int resetMaster()	36
10.7.1. 描述: 重启系统.....	36
10.8. public static native int getWatchdogStatus()	37
10.8.1. 描述: 获取看门狗状态.....	37
10.9. public static native int setTime(int time);.....	37
10.9.1. 描述: 设置开机时间.....	37
10.10. public static native int getVersion().....	38
10.10.1. 描述: 获取看门狗的版本号	38
11. 软件安装	38
11.1. public boolean silentInstallApk(String apkPath)	38
11.1.1. 描述: 静默安装 APK 应用	38
12. 系统升级	39
12.1. public void upgradeSystem(String absolutePath)	39
12.1.1. 描述: 升级固件	39
12.2. public void rebootRecovery().....	39
12.2.1. 描述: 将重启进入 recovery 模式	39
13. 其他	40
13.1. public void setTime (int year, int month, int day, int hour, int minute).....	40
13.1.1. 描述: 设置并保存系统时间	40
13.2. public void execSuCmd (String command)	40
13.2.1. 描述: 将以 ROOT 权限运行 shell 命令.....	40
13.3. public void getAndroidLogcat(String path)	41
13.3.1. 描述: 抓取 Android 层的 LOG 并保存相应目录.....	41
13.4. public void stopAndroidLogcat()	41
13.4.1. 描述: 关闭抓取 Android 层的 LOG	41
13.5. public void getKmsgLog(String path).....	42
13.5.1. 描述: 抓取 kernel 层的 LOG 并保存相应目录	42
14. 联系方式	43
15. 免责声明	43
16. 更新历史	43

1. 背景

USR-ECB400A 软件开发包致力于降低客户的开发难度并强化原生 Android OS 的能力。例如通过一条简单的 API 调用即可实现原生 Android SDK 若干条 API 才能达到或无法达到的功能，如网络信息的获取、网络设备的控制、存储设备的挂载及卸载、GPIO 的操作、虚拟按键的显示/隐藏、全局截图、背光灯的控制、定时开关机、shell 命令的执行、Android 或 kernel LOG 的抓取、APP 看门狗的实现、应用静默安装、系统升级等功能。若对以上没有任何概念，建议安装附件中的 UsrApiRefCode.apk 以体验 USR-ECB400A 软件开发包强悍的能力，若想究其原理可以查看位于附件\源码\UsrApiRefCode 的源码。

2. 准备工作

2.1. jar 包导入

1. 将 poweronoff.jar 与 rkapi.jar 复制到【工程目录\app\libs\】下

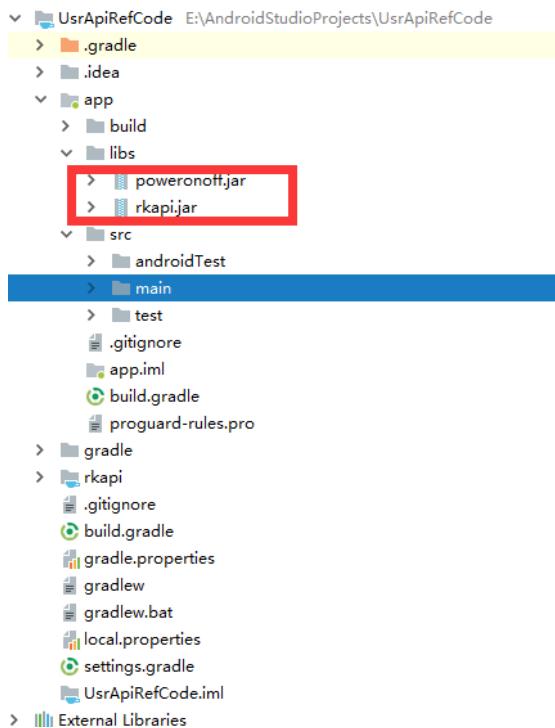


图 1 jar 包结构

2. 右键点击 libs 文件夹中的 jar 文件选择 add as Library 然后选择 Model

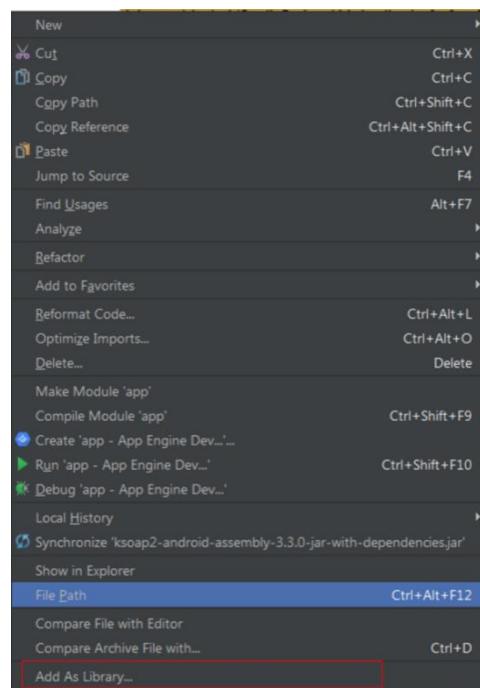


图 2 jar 包添加

3. API 使用说明

3.1. 通用 api 使用说明

声明 MyManager 对象：

```
MyManager myManager= MyManager.getInstance(this);
```

使用 API:

```
myManager.getApiVersion()
```

3.2. 定时器相关使用说明

声明 PowerOnOffManager 对象：

```
private PowerOnOffManager manager = PowerOnOffManager.getInstance(this);
```

使用 API:

```
manager.getVersion () ;
```

4. 获取系统信息

4.1. public String getApiVersion()

4.1.1. 描述：获取目前 API 的平台-版本-日期信息

参数名/返回值	类型	说明	举例
返回值	String	API 的平台 - 版本 - 日期 信息。格式如下 XXX-Vn-201YMMDD	RK3288-V1-20180311

范例：

```
String apiVersion = myManager.getApiVersion();
UsrLog.LOGD(TAG, "apiVersion = "+apiVersion);
```

输出：

```
apiVersion = V1.0-20171215
```

<特殊说明>：

无

4.2. public String getAndroidModle()

4.2.1. 描述：获取目前设备的型号

参数名/返回值	类型	说明	举例
返回值	String	设备的型号	rk3288

范例：

```
String androidModle = myManager.getAndroidModle ();
UsrLog.LOGD(TAG, "androidModle = "+androidModle);
```

输出：

```
androidModle = rk3288
```

<特殊说明>：

无

4.3. public String getAndroidVersion ()

4.3.1. 描述：获取目前设备的 android 系统的版本

参数名/返回值	类型	说明	举例
返回值	String	android 系统的版本	7.1

范例：

```
String androidVersion = myManager.getAndroidVersion();
UsrLog.LOGD(TAG, "androidVersion = "+androidVersion);
```

输出：

```
androidVersion = 22
```

4.4. public String getRunningMemory ()

4.4.1. 描述：获取设备的硬件内存大小容量

参数名/返回值	类型	说明	举例
返回值	String	硬件内存大小，以 GB 为单位	2GB

范例：

```
String runningMemory = myManager.getRunningMemory();
UsrLog.LOGD(TAG, "runningMemory = "+runningMemory);
```

输出：

```
runningMemory = 2G
```

<特殊说明>:

无

4.5. public String getInternalStorageMemory ()

4.5.1. 描述：获取设备的硬件内部存储大小容量

参数名/返回值	类型	说明	举例
返回值	String	硬件内部存储大小，以 GB 为单位	8GB

范例：

```
String internalStorageMemory = myManager.getInternalStorageMemory();
UsrLog.LOGD(TAG, "internalStorageMemory = "+internalStorageMemory);
```

输出：

internalStorageMemory = 8G

<特殊说明>:

无

4.6. public String getFirmwareVersion ()

4.6.1. 描述: 获取设备的固件 SDK 版本

参数名/返回值	类型	说明	举例
返回值	String	设备的固件 SDK 版本	v1.2rc3

范例:

```
String firmwareVersion = myManager.getFirmwareVersion();
UsrLog.LOGD(TAG, "firmwareVersion = "+firmwareVersion);
```

输出:

```
firmwareVersion = 1.0
```

<特殊说明>:

无

4.7. public String getKernelVersion ()

4.7.1. 描述: 获取设备的固件内核版本

参数名/返回值	类型	说明	举例
返回值	String	设备的固件内核版本	3.4.39

范例:

```
String kernelVersion = myManager.getKernelVersion();
UsrLog.LOGD(TAG, "kernelVersion = "+kernelVersion);
```

输出:

```
kernelVersion = 3.10.0
```

4.8. public String getAndroidDisplay ()

4.8.1. 描述：获取设备的固件系统版本和编译日期

参数名/返回值	类型	说明	举例
返回值	String	设备的固件系统版本和编译日期	eng 4.4.4 KTU84Q 20180318 test-keys

范例：

```
String androidDisplay = myManager.getAndroidDisplay();
UsrLog.LOGD(TAG, "androidDisplay = "+androidDisplay);
```

输出：

```
androidDisplay = rk3288_box-userdebug 5.1.1 LMY49F eng.edge.20180426.104757 test-keys
```

<特殊说明>:

无

4.9. public String getFirmwareDate()

4.9.1. 描述：获取固件编译的时间

参数名/返回值	类型	说明	举例
返回值	String	设备的固件编译日期	20180318

范例：

```
String firmwareDate = myManager.getFirmwareDate();
UsrLog.LOGD(TAG, "firmwareDate = "+firmwareDate);
```

输出：

```
firmwareDate = 20180426
```

<特殊说明>:

无

5. 网络

5.1. public String getEthMacAddress()

5.1.1. 描述：获取第一块以太网的 MAC 地址

参数名/返回值	类型	说明	举例
返回值	String	以太网的 MAC 地址	12:23:34:56:AA:BB

范例：

```
String MAC = myManager.getEthMacAddress();
UsrLog.LOGD(TAG, "MAC = "+MAC);
```

输出：

```
MAC = 12:23:34:56:AA:BB
```

<特殊说明>：

无

5.2. public String getEthMode()

5.2.1. 描述：获取当前以太网的模式

参数名/返回值	类型	说明	举例
返回值	String	以太网的模式	StaticIp (静态) DHCP (动态)

范例：

```
String ethernetMode = myManager.getEthMode();
UsrLog.LOGD(TAG, "ethernetMode = "+ethernetMode);
```

输出：

```
ethernetMode = DHCP
```

<特殊说明>：

- 在获取以太网静态 IP 之前，需要绑定获取以太网静态 IP 的 aidl 服务，在创建 MyManager 对象的时候绑定，方法如下

```
myManager.bindAIDLService(this);
```

- 在 activity 销毁的时候需要调用，例子如下

```
@Override
protected void onDestroy() {
    super.onDestroy();
    myManager.unBindAIDLService(this);
}
```

5.3. public void setDhcpIpAddress(Context context)

5.3.1. 描述：设置以太网模式为动态获取

参数名/返回值	类型	说明	举例
返回值	Context	上下文对象	

范例：

```
myManager.setDhcpIpAddress(this);
```

输出：

无

<特殊说明>：

无

5.4. public void setStaticEthIPAddress(...)

5.4.1. 描述：设置以太网的模式为静态地址

参数名/返回值	类型	说明	举例
IPAddr	String	IP 地址	192.168.14.223
gateWay	String	网关	192.168.14.1
mask	String	子网掩码	255.255.255.0
dns1	String	DNS1	114.114.114.114
dns2	String	DNS2	8.8.8.8
返回值	void		

范例：

```
myManager.setStaticEthIPAddress("192.168.13.22","255.255.255.0","192.168.13.1","114.114.114.114","114.114.114.114");
```

输出：

无

<特殊说明>：

➤ IP、子网掩码、网关和 DNS 都要设置为合法数据；

5.5. public String getStaticEthIPAddress()

5.5.1. 描述：获取以太网的静态 IP 地址

参数名/返回值	类型	说明	举例
返回值	String	以太网的 IP 地址	192.168.1.100

范例：

```
String staticIP = myManager.getStaticEthIPAddress();
UsrLog.LOGD(TAG, "staticIP="+staticIP );
```

输出：

```
staticIP=192.168.1.100
```

<特殊说明>：

- 该 API 必须在以太网模式为”静态获取”时使用
- 在获取以太网静态 IP 之前，需要绑定获取以太网静态 IP 的 aidl 服务，在创建 myManager 对象的时候绑定，方法如下：myManager.bindAIDLService(this);
- 在 activity 销毁的时候需要调用 myManager.unBindAIDLService(this);

5.6. public String getNetMask()

5.6.1. 描述：获取以太网卡的子网掩码

参数名/返回值	类型	说明	举例
返回值	String	以太网子网掩码	

范例：

```
String mask = myManager.getNetMask();
UsrLog.LOGD(TAG, "ethernetNetInfoShow:mask = "+mask);
```

输出：

```
ethernetNetInfoShow:mask =255.255.255.0
```

<特殊说明>：

无

5.7. public String getGateway ()

5.7.1. 描述：获取当以太网卡的网关

参数名/返回值	类型	说明	举例
返回值	String	以太网卡网关	

范例：

```
String gateway = myManager.getGateway();
UsrLog.LOGD(TAG, "ethernetNetInfoShow:gateway = "+gateway);
```

输出：

```
ethernetNetInfoShow:mask =192.168.5.1
```

<特殊说明>：

无

5.8. public String getEthDns1 ()

5.8.1. 描述：获取以太网卡的 DNS1

参数名/返回值	类型	说明	举例
返回值	String	以太网卡网的 DNS1	

范例：

```
String ethDns1 = myManager.getEthDns1();
UsrLog.LOGD(TAG, "ethernetNetInfoShow:ethDns1 = "+ethDns1);
```

输出：

```
ethernetNetInfoShow:ethDns1 =114.114.114.114
```

<特殊说明>：

无

5.9. public String getEthDns2()

5.9.1. 描述：获取当以太网卡的 Dns2

参数名/返回值	类型	说明	举例
返回值	String	以太网卡的 DNS2	

范例：

```
String ethDns2 = myManager.getEthDns2();
UsrLog.LOGD(TAG, "ethernetNetInfoShow:ethDns2 = "+ethDns2);
```

输出：

```
ethernetNetInfoShow:ethDns2 =8.8.8.8
```

<特殊说明>：

无

5.10. public int getCurrentNetType()

5.10.1. 描述：获取当前网络连接的类型

参数名/返回值	类型	说明	举例
返回值	String	网络类型。 0 表示以太网 1 表示 WIFI 2 表示移动网络 -100 表示未知网络	

范例：

```
int networkType = myManager.getCurrentNetType();
UsrLog.LOGD(TAG, integerNetworkTypeConvertString(networkType));
```

输出：

WIFI

<特殊说明>：

```
1. private String integerNetworkTypeConvertString(int networkType){
    switch (networkType){
        case NET_TYPE_ETH:{
            return "以太网";
        }
        case NET_TYPE_WIFI:{
            return "WIFI";
        }
        case NET_TYPE_MOBILE:{
            return "移动网络";
        }
        default:{
            return "未知网络";
        }
    }
}
```

5.11. public boolean getEthStatus ()

5.11.1. 描述：获取设备以太网的状态

参数名/返回值	类型	说明	举例
返回值	boolean	true: open false:close	

范例：

```
boolean ethState = myManager.getEthStatus ();  
UsrLog.LOGD(TAG, "ethState = "+ ethState);
```

输出：

```
ethState = true
```

<特殊说明>：

无

5.12. public void ethEnabled(boolean enable)

5.12.1. 描述：开关以太网

参数名/返回值	类型	说明	举例
enable	boolean	true: 使能以太网, false: 关闭以太网	
返回值	void		

范例：

```
myManager.ethEnabled(b);
```

输出：

无

<特殊说明>：

无

6. 存储管理

6.1. public String getSDcardPath()

6.1.1. 描述：获取外部存储 SD 卡路径

参数名/返回值	类型	说明	举例
返回值	String	外部存储 SD 卡绝对路径	"/mnt/external_sd"

范例：

```
String sdPath = myManager.getSDcardPath();
UsrLog.LOGD(TAG, "sdPath = "+sdPath);
```

输出：

```
sdPath = /mnt/external_sd
```

<特殊说明>：

无

6.2. public String getUSBStoragePath(int num)

6.2.1. 描述：获取外部存储 U 盘路径

参数名/返回值	类型	说明	举例
num	int	U 盘盘符，按 U 盘的接入顺序	0,1,2,3
返回值	String	外部存储 U 盘绝对路径	

范例：

```
String usbPath = myManager.getUSBStoragePath(0);
UsrLog.LOGD(TAG, "usbPath = "+usbPath);
```

输出：

```
usbPath = /mnt/usb_storage/USB_DISK2/udisk0
```

<特殊说明>：

➤ 需要提前接入 USB 存储设备

6.3. public void unmountVolume(String path)

6.3.1. 描述：卸载外部存储

参数名/返回值	类型	说明	举例
path	String	要卸载的外部存储绝对路径	/mnt/usb_storage/USB_DISK2/udisk0

范例：

```
String usbPath = myManager.unmountVolume ("/mnt/usb_storage/USB_DISK2/udisk0");
```

输出：

无

<特殊说明>：

无

6.4. public void mountVolume(String path)

6.4.1. 描述：挂载外部存储，如果外部存储存在的话

参数名/返回值	类型	说明	举例
path	String	要挂载的外部存储绝对路径	/mnt/usb_storage/USB_DISK2/udisk0

范例：

```
myManager.mountVolume("/mnt/usb_storage/USB_DISK2");
```

输出：

无

<特殊说明>：

无

7. 硬件接口

7.1. public String getUartPath(String uart)

7.1.1. 描述：获取串口绝对路径

参数名/返回值	类型	说明	举例
Uart	String	可取参数如下： TTYS0: 串口 0 TTYS1: 串口 1 TTYS2: 串口 2 TTYS3: 串口 3 TTYS4: 串口 4 TTYS5: 串口 5 ...	TTYS0
返回值	String	UART 绝对路径	/dev/ttyS0

范例：

```
String uart1 = myManager.getUartPath("TTYS1");
UsrLog.LOGD(TAG, "uart1 path = "+uart1);
```

输出：

```
uart1 path = /dev/ttyS1
```

<特殊说明>：

无

7.2. public String getGPIOStatus(String io)

7.2.1. 描述：获取 IO 口输入状态

参数名/返回值	类型	说明	举例
io	String	in1:SCL3 引脚 in2:SDA3 引脚	
返回值	String	1:IO 输入为高 0:IO 输入为低	

范例：

```
String values = myManager.getGPIOStatus("in1")
UsrLog.LOGD(TAG, " values = "+ values);
```

输出:

values=0

<特殊说明>:

无

7.3. public void setGPIOStatus(String value, String io)

7.3.1. 描述：设置 IO 口输出状态

参数名/返回值	类型	说明	举例
value	String	0: 输出低电平 1: 输出高电平	0
io	String	out1: INT out2: RST	out1
返回值	Void		

范例:

```
myManager.setGPIOStatus("1", "out1");
```

输出:

无

<特殊说明>:

无

8. 显示

8.1. public void takeScreenshot(Context context, String path)

8.1.1. 描述：截取当前全屏为 jpg 格式图片并重命名到相应位置

参数名/返回值	类型	说明	举例
context	Context	上下文对象	
path	String	保存绝对路径	"/sdcard/Pictures/ganye.jpg"

范例:

```
myManager.takeScreenshot("/sdcard/Pictures/ganye.jpg");
```

输出:

无

<特殊说明>:

➤ path 必须为可读写目录包括文件名

8.2. public void rotateScreen(Context context, String degree)

8.2.1. 描述：设置屏幕逆时针旋转 N 角度

参数名/返回值	类型	说明	举例
context	Context	上下文对象	
degree	String	只支持 0 , 90, 180, 270 四个角度。	"0" "90" "180" "270"
返回值	void		

范例：

```
myManager.rotateScreen(this, "0");
```

输出：

无

<特殊说明>：

无

8.3. public int getDisplayWidth(Context context)

8.3.1. 描述：获取显示屏分辨率宽 X 像素

参数名/返回值	类型	说明	举例
Context	Context	上下文	
返回值	int	屏幕实际宽像素	

范例：

```
int displayWidth = myManager.getDisplayWidth(this);  
UsrLog.LOGD(TAG, " displayWidth =" + dispalyeWidth);
```

输出：

displayWidth =1920

<特殊说明>：

无

8.4. public int getDisplayHeight(Context context)

8.4.1. 描述：获取显示屏分辨率高 Y 像素

参数名/返回值	类型	说明	举例
context	Context	上下文	
返回值	int	屏幕实际高像素	

范例：

```
int displayHeight = myManager.getDisplayHeight(this);  
UsrLog.LOGD(TAG, " displayHeight =" + displayHeight);
```

输出：

```
displayHeight = 1920
```

<特殊说明>：

无

8.5. public void hideNavBar(boolean hide)

8.5.1. 描述：设置显示或者隐藏虚拟按键

参数名/返回值	类型	说明	举例
hide	boolean	true: 隐藏状态栏 false: 显示状态栏	
返回值	void		

范例：

```
myManager.hideNavBar(false);
```

输出：

无

<特殊说明>：

无

8.6. public boolean getNavBarHideState()

8.6.1. 描述：获取当前虚拟按键状态

参数名/返回值	类型	说明	举例
返回值	boolean	true:隐藏， false:显示	

范例：

```
boolean isHide = myManager.getNavBarHideState()  
UsrLog.LOGD(TAG, " isHide =" + isHide);
```

输出：

```
isHide = false
```

<特殊说明>：

无

8.7. public void setSlideShowNavBar(boolean flag)

8.7.1. 描述：禁止滑出虚拟按键

参数名/返回值	类型	说明	举例
Flag	boolean	true: 禁止滑出虚拟按键 false: 允许滑出虚拟按键	

范例：

```
myManager.setSlideShowNavBar(false);
```

输出：

无

<特殊说明>：

无

8.8. public boolean isSlideShowNavBarOpen ()

8.8.1. 描述：获取滑出导航栏状态

参数名/返回值	类型	说明	举例
Flag	boolean	true: 禁止滑出虚拟按键 false: 允许滑出虚拟按键	

范例：

```
boolean isShow = myManager.isSlideShowNavBarOpen()  
UsrLog.LOGD(TAG, "isShow ="+ isShow);
```

输出：

```
isShow =false
```

<特殊说明>：

无

8.9. public void setSlideShowNotificationBar(boolean enable)

8.9.1. 描述：打开或关闭滑出通知栏

参数名/返回值	类型	说明	举例
Enable	boolean	true: 禁止滑出通知栏 false: 允许滑出通知栏	

范例：

```
myManager.setSlideShowNotificationBar(b);
```

输出：

无

<特殊说明>：

无

8.10. public boolean isSlideShowNotificationBarOpen()

8.10.1. 描述：滑出通知栏是否打开

参数名/返回值	类型	说明	举例
返回值	boolean	true: 禁止 false: 允许	

范例：

```
boolean isOpen = myManager.isSlideShowNotificationBarOpen()  
UsrLog.LOGD(TAG, "isOpen ="+ isOpen);
```

输出：

```
isOpen =false
```

<特殊说明>：

无

8.11. public void turnOffBacklight()

8.11.1. 描述：熄灭屏幕，只关背光，却不进入休眠，软件继续运行

参数名/返回值	类型	说明	举例
返回值	void		

范例：

```
myManager.turnOffBacklight();
```

输出：

无

<特殊说明>：

无

8.12. public void void turnOnBacklight()

8.12.1. 描述：开启背光

参数名/返回值	类型	说明	举例
返回值	void		

范例：

```
myManager.turnOnBacklight();
```

输出：

无

<特殊说明>：

无

8.13. public void turnOnHDMI();

8.13.1. 描述：打开 HDMI 输出信号

参数名/返回值	类型	说明	举例
返回值	void		

范例：

```
myManager.turnOnHDMI();
```

输出：

无

<特殊说明>：

无

8.14. public void turnOffHDMI();

8.14.1. 描述：关闭 HDMI 输出信号

参数名/返回值	类型	说明	举例
返回值	void		

范例：

```
myManager.turnOffHDMI();
```

输出：

无

<特殊说明>:

无

9. 开关机、定时

9.1. public String getPowerOnMode()

9.1.1. 描述：获取当前定时开关机模式

参数名/返回值	类型	说明	举例
返回值	String	返回 1 就是设置一组开关机时间, 对应的方法是 setPowerOnOff; 返回 2 就是设置周期性的开关机时间 , 对应方法是 setPowerOnOffWithWeekly;	1

范例：

```
String timerMode = manager.getPowerOnMode();
UsrLog.LOGD(TAG, " timerMode = " + timerMode);
```

输出：

```
timerMode = "1"
```

<特殊说明>:

无

9.2. public void setPowerOnOff(int[] powerOnTime, int[] powerOffTime)

9.2.1. 描述：设置一组开关机时间

参数名/返回值	类型	说明	举例
powerOnTime	Int[]	关机的时间	{2018,1,10,20,38}
powerOffTime	Int[]	开机的时间	{2018,1,10,20,48}
返回值	void	返回 1 就是设置一组开关机时间, 对应的方法是 setPowerOnOff; 返回 2 就是设置周期性的开关机时间, 对应方法是 setPowerOnOffWithWeekly;	

范例：

```
manager.setPowerOnOff({2018,1,10,20,48 }, { 2018,1,10,20,58 });
```

输出:

无

<特殊说明>:

无

9.3. **public void setPowerOnOffWithWeekly(int[] powerOnTime, int[] powerOffTime,int[] weekdays)**

9.3.1. 描述：设置一组周期性的开关机时间

参数名/返回值	类型	说明	举例
powerOnTime	Int[]	关机的时间	{8,30}
powerOffTime	Int[]	开机的时间	{8,30}
weekdays	int[]	周一到周日的工作状态	{1,1,1,1,1,1,1}
返回值	void	返回 1 就是设置一组开关机时间，对应的方法是 setPowerOnOff; 返回 2 就是设置周期性的开关机时间，对应方法是 setPowerOnOffWithWeekly;	

范例:

```
manager.setPowerOnOff({20,48}, {20,58 }, {1,1,1,1,1,1,1});
```

输出:

无

<特殊说明>:

无

9.4. **public String getPowerOnTime()**

9.4.1. 描述：清除开关机时间

参数名/返回值	类型	说明	举例
返回值	String	返回的值包含月份、日期、小时、分钟	

范例:

```
String powerOnTime = manager.getPowerOnTime();
UsrLog.LOGD(TAG, " powerOnTime = "+ powerOnTime);
```

输出:

powerOnTime = 20181102038

<特殊说明>:

无

9.5. public String getPowerOffTime()

9.5.1. 描述：获取设置的开机时间

参数名/返回值	类型	说明	举例
返回值	String	返回的值包含月份、日期、小时、分钟	

范例：

```
String powerOFFTime = manager.getPowerOffTime();
UsrLog.LOGD(TAG, " powerOFFTime = "+ powerOFFTime);
```

输出：

```
powerOFFTime = 20181102048
```

<特殊说明>：

无

9.6. public String getLastestPowerOnTime()

9.6.1. 描述：获取上一次实际的开机时间

参数名/返回值	类型	说明	举例
返回值	String	返回的值包含月份、日期、小时、分钟	

范例：

```
String lastestPowerOnTime = manager.getLastestPowerOnTime();
UsrLog.LOGD(TAG, " lastestPowerOnTime = "+ lastestPowerOnTime);
```

输出：

```
lastestPowerOnTime = getLastestPowerOnTime
```

<特殊说明>：

无

9.7. public String getLastestPowerOffTime()

9.7.1. 描述：获取上一次实际的关机时间

参数名/返回值	类型	说明	举例
返回值	String	返回的值包含月份、日期、小时、分钟	

范例：

```
String lastestPowerOffTime = manager.getLastestPowerOffTime();
```

```
UsrLog.LOGD(TAG, " lastestPowerOffTime = "+ lastestPowerOffTime);
```

输出:

```
lastestPowerOffTime = 201801110905
```

<特殊说明>:

无

9.8. public void clearPowerOnOffTime()

9.8.1. 描述: 清除开关机时间

参数名/返回值	类型	说明	举例
返回值	void		

范例:

```
manager.clearPowerOnOffTime();
```

输出:

无

<特殊说明>:

无

9.9. public void shutDown()

9.9.1. 描述: 正常关闭系统

参数名/返回值	类型	说明	举例
返回值	boolean	需安全关闭	

范例:

```
myManager.shutdown();
```

输出:

无

<特殊说明>:

无

9.10. public boolean reboot (String reason)

9.10.1. 正常重启系统

参数名/返回值	类型	说明	举例
reason	String	预留	
返回值	boolean	true:成功 false:失败	

范例：

```
myManager.reboot();
```

输出：

无

<特殊说明>：

无

10. 硬件看门狗控制相关

10.1. 目录结构

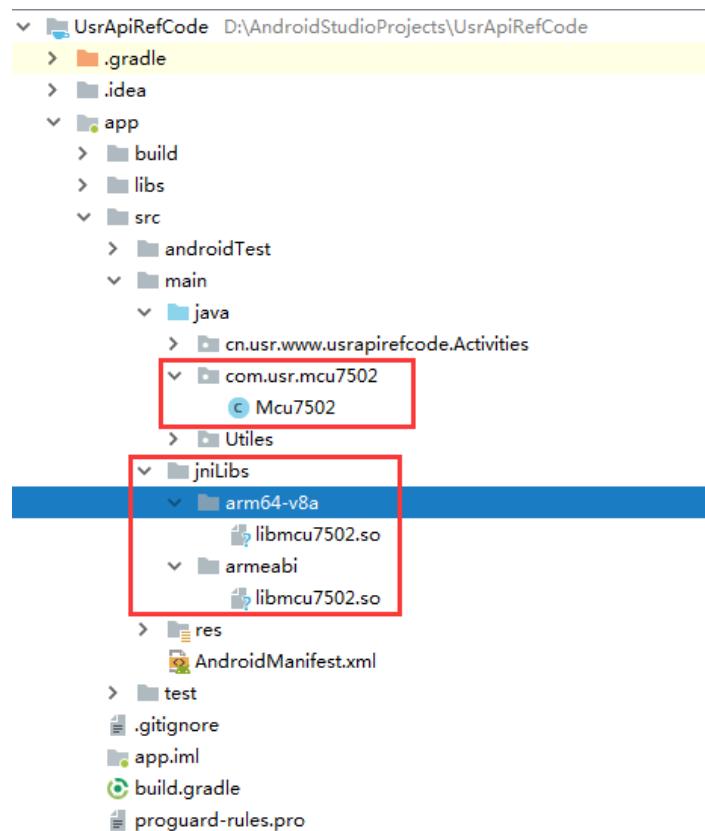


图 3 看门狗库及类添加

<说明>:

- com.usr.mcu7502 包及 jniLibs 见附件 watchdog
- 目录结构必须与示例保持一致
- 调用看门狗功能前需导入 Mcu7502, 示例如下
`import com.usr.mcu7502.Mcu7502;`

10.2. public static native int open()

10.2.1. 描述：打开看门狗控制句柄

参数名/返回值	类型	说明	举例
返回值	int	返回大于 0 的值表示打开成功	344

范例：

```
int ret = Mcu7502.open();
UsrLog.LOGD(TAG, "ret:" + ret);
```

输出：

```
ret:344
```

<特殊说明>：

- 不能重复打开，打开一次成功后，再执行此操作会返回-1；
- 需与 close 成对调用；

10.3. public static native void close()

10.3.1. 描述：关闭看门狗控制句柄

参数名/返回值	类型	说明	举例
返回值	void		

范例：

```
Mcu7502.close();
```

输出：

```
无
```

<特殊说明>：

- 需与 open 成对调用

10.4. public static native int enableWatchdog()

10.4.1. 描述：使能看门狗

参数名/返回值	类型	说明	举例
返回值	int	返回 7 表示发送指令成功	

范例：

```
int ret = Mcu7502.enableWatchdog();
UsrLog.LOGD(TAG, "ret:" + ret );
```

输出：

```
ret:7
```

<特殊说明>：

➤ 打开此功能后若不喂狗，每隔 2min 会重启

10.5. public static native int disableWatchdog()

10.5.1. 描述：禁用看门狗功能

参数名/返回值	类型	说明	举例
返回值	int		

范例：

```
int ret = Mcu7502.disableWatchdog();
UsrLog.LOGD(TAG, "ret:" + ret );
```

输出：

```
ret:7
```

<特殊说明>：

无

10.6. public static native int feetDog(int time)

10.6.1. 描述：设置喂狗时间

参数名/返回值	类型	说明	举例
time	int	喂狗时间，单位 s，必须 $\geq 10s$, $< 60s$	
返回值	ret	返回 7 表示发送指令成功	

范例：

```
int ret = Mcu7502.feetDog(10);  
UsrLog.LOGD(TAG, "ret:" + ret );
```

输出：

```
ret:7
```

<特殊说明>：

- 喂狗时间表示在该时间段内设备不会重启，所以喂狗间隔必须小于喂狗时间

10.7. public static native int resetMaster()

10.7.1. 描述：重启系统

参数名/返回值	类型	说明	举例
返回值	int	返回 7 表示发送指令成功	

范例：

```
int ret = Mcu7502.resetMaster();  
UsrLog.LOGD(TAG, "ret:" + ret );
```

输出：

```
ret:7
```

<特殊说明>：

- 该方法跟断电重启一个作用

10.8. public static native int getWatchdogStatus()

10.8.1. 描述：获取看门狗状态

参数名/返回值	类型	说明	举例
返回值	int	返回 7 表示发送指令成功	

范例：

```
int ret = Mcu7502.getWatchdogStatus();
UsrLog.LOGD(TAG, "ret:" + ret );
```

输出：

```
ret:7
```

<特殊说明>：

无

10.9. public static native int setTime(int time);

10.9.1. 描述：设置开机时间

参数名/返回值	类型	说明	举例
time	int	下一次开机距离关机的时间，以 s 为单位。大于等于 60s	
返回值	int	返回 7 表示发送指令成功	

范例：

```
int ret = Mcu7502.setTime(70);
UsrLog.LOGD(TAG, "ret:" + ret );
```

输出：

```
ret:7
```

<特殊说明>：

- 参数值得时间计算起点是关机的时候
- 例如 Mcu7502.setTime(180);
表示在关机过后 180s 会开机。若关机时间为 17:00，那么在 17:03 则开机

10.10. public static native int getVersion()

10.10.1. 描述：获取看门狗的版本号

参数名/返回值	类型	说明	举例
返回值	int	返回值为 int 型变量，第一个字节未定义，第二个字节表示版本号第前两位，第三个字节表示版本号中间两位，第四个字节表示版本号的后两位	

范例：

```
int version = Mcu7502.getVersion()
UsrLog.LOGD(TAG, "version :" + version );
```

输出：

```
version:1180188
```

<特殊说明>：

- 该方法跟断电重启一个作用

11. 软件安装

11.1. public boolean silentInstallApk(String apkPath)

11.1.1. 描述：静默安装 APK 应用

参数名/返回值	类型	说明	举例
path	String	待安装 APK 的绝对路径	"/sdcard/debug.apk"
返回值	boolean	true:成功 false: 失败	

范例：

```
boolean isInstalled = myManager.silentInstallApk("/sdcard/debug.apk");
UsrLog.LOGD(TAG, " isInstalled = "+ isInstalled);
```

输出：

```
isInstalled = true
```

<特殊说明>：

- APK 文件必须存在并且完整可用；

12. 系统升级

12.1. public void upgradeSystem(String absolutePath)

12.1.1. 描述：升级固件

参数名/返回值	类型	说明	举例
absolutePath	String	固件存放的绝对路径	/sdcard/update.img
返回值	void		

范例：

```
manager.upgradeSystem("/sdcard/update.img");
```

输出：

无

<特殊说明>：

- 函数将会自动重启系统进行 update.img 升级。
- 保证 img 文件存在，完整可用。
- 文件名一定为 update.img。

12.2. public void rebootRecovery()

12.2.1. 描述：将重启进入 recovery 模式

参数名/返回值	类型	说明	举例
返回值	void		

范例：

```
manager.rebootRecovery();
```

输出：

无

<特殊说明>：

无

13. 其他

13.1. public void setTime (int year, int month, int day, int hour, int minute)

13.1.1. 描述：设置并保存系统时间

参数名/返回值	类型	说明	举例
year	int	年	
month	int	月	
day	int	日	
hour	int	(24 小时制) 小时	
minute	int	(24 小时制) 分钟	
返回值	void		

范例：

```
myManager.setTime(2018,  
                  5,  
                  6,  
                  7,  
                  25);
```

输出：

无

<特殊说明>：

- 给出的年月日参数必须可用；

13.2. public void execSuCmd (String command)

13.2.1. 描述：将以 ROOT 权限运行 shell 命令

参数名/返回值	类型	说明	举例
command	String	Shell 命令	“ls”
返回值	void		

范例：

```
myManager.execSuCmd("reboot");输出：
```

<特殊说明>：

- 执行 reboot 命令后，板卡将直接重启；

13.3. public void getAndroidLogcat(String path)

13.3.1. 描述：抓取 Android 层的 LOG 并保存相应目录

参数名/返回值	类型	说明	举例
Path	String	Log 保存绝对路径	/sdcard/android-log.txt
返回值	Void		

范例：

```
myManager.getAndroidLogcat("/scard/android-log.txt");
```

输出：

无

<特殊说明>：

- 注意在 AndroidManifest.xml 中添加存储相关的权限，例子如下；
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"></uses-permission>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-permission>
- 开启 Android log 获取后，注意不再需要时调用 stopAndroidLogcat() 停止，否则将不断的占用内部存储；

13.4. public void stopAndroidLogcat()

13.4.1. 描述：关闭抓取 Android 层的 LOG

参数名/返回值	类型	说明	举例
返回值	Void		

范例：

```
myManager.stopAndroidLogcat();
```

输出：

无

<特殊说明>：

无

13.5. public void getKmsgLog(String path)

13.5.1. 描述：抓取 kernel 层的 LOG 并保存相应目录

参数名/返回值	类型	说明	举例
folderPath	String	Log 保存绝对路径	“/storage/usbcards1/”
返回值	Void		

范例：

```
myManager.getKmsgLog("/sdcard/kernel-log.txt");
```

输出：

无

<特殊说明>：

无

14. 联系方式

公 司：济南有人物联网技术有限公司

地 址：山东省济南市高新区新泺大街 1166 号奥盛大厦 1 号楼 11 层

网 址：<http://www.usr.cn>

客户支持中心：<http://h.usr.cn>

邮 箱：sales@usr.cn

电 话：4000-255-652 或者 0531-88826739

有人定位：万物互联使能者 工业物联网通讯专家

公司文化：有人在认真做事！

有人使命：连接价值 价值连接

价值观：天道酬勤 厚德载物 共同成长 积极感恩

有人愿景：成为工业物联网领域的生态型企业

产品理念：简单 可靠 价格合理

15. 免责声明

本文档提供有关 USR-ECB400A 产品的信息，本文档未授予任何知识产权的许可，并未以明示或暗示，或以禁止发言或其它方式授予任何知识产权许可。除在其产品的销售条款和条件声明的责任之外，我公司概不承担任何其它责任。并且，我公司对本产品的销售和/或使用不作任何明示或暗示的担保，包括对产品的特定用途适用性，适销性或对任何专利权，版权或其它知识产权的侵权责任等均不作担保。本公司可能随时对产品规格及产品描述做出修改，恕不另行通知。

16. 更新历史

2018-07-16 版本 V1.0.0 创立。

2018-08-15 版本 V1.0.01 修订。

2018-08-20 版本 V1.0.02 修订。

2018-08-29 版本 V1.0.03 修订。

2018-09-14 版本 V1.0.04 修订。