Machine Translated by Google

Mainboard API Documentation

Edition	V1.7.2
Date	2023-05-22

Notice: The copyright of this document belongs to the original content company, and all rights are reserved. If the document content is modified or updated, please contact the provider to obtain the latest version.

This without further notice.

Machine Translated by Google

Modification Record

1.0.0	2017-05-12	First version of this document.
1.1.0	2017-06-15	Added serial port close interface description.
1.2.0	2017-07-04	Added instructions on how to import libraries in the Android Studio development environment.
1.3.0	2017-08-18	Added HDMIIN programming interface; added APK signature description.
1.3.1	2017-12-26	APK signature address updated http://120.78.220.29:18080/
1.3.2	2018-03-05	Corrected the typo in the text of section $3.1 on display density.
1.4.0	2018-04-09	Added calling permission description for screen rotation interface.
1.5.0	2018-07-30	Added Android Studio 3.0 integration parameter description.
1.6.0	2018-10-30	Added permission description for setSystemTime.
1.6.1	2018-11-12	Updated the signature server IP address.
1.6.2	2019-02-18	Added signature URL for Android 7.1 and above.
1.6.3	2020-09-28	Added description of AS obfuscation configuration file.
1.7.0	2021-12-15	V1.2.0.20211022 version supports no ROOT required, fixes Android 7.1 and above static IP interface DNS string Format issue; add description of scheduled power on/off broadcast interface and application startup management broadcast interface.
1.7.1	2022-03-18	Corrected the text error in the example code of daily mode timed power on/off
1.7.2	2023-05-22	Scheduled power on/off and application startup management broadcast interface adds system call instructions for Android 8.0 and above!

Machine Translated by Google

Table of contents

Machine Translated by Google

Machine Translated by Google

Machine Translated by Google

Machine Translated by Google

1 Programming Instructions

Please integrate the corresponding library files into the project and call them according to the API instructions. Note: Some APIs require root

Please consult the motherboard manufacturer to obtain the root method of the Android system. Please check the corresponding API description for specific APIs that require root permissions.

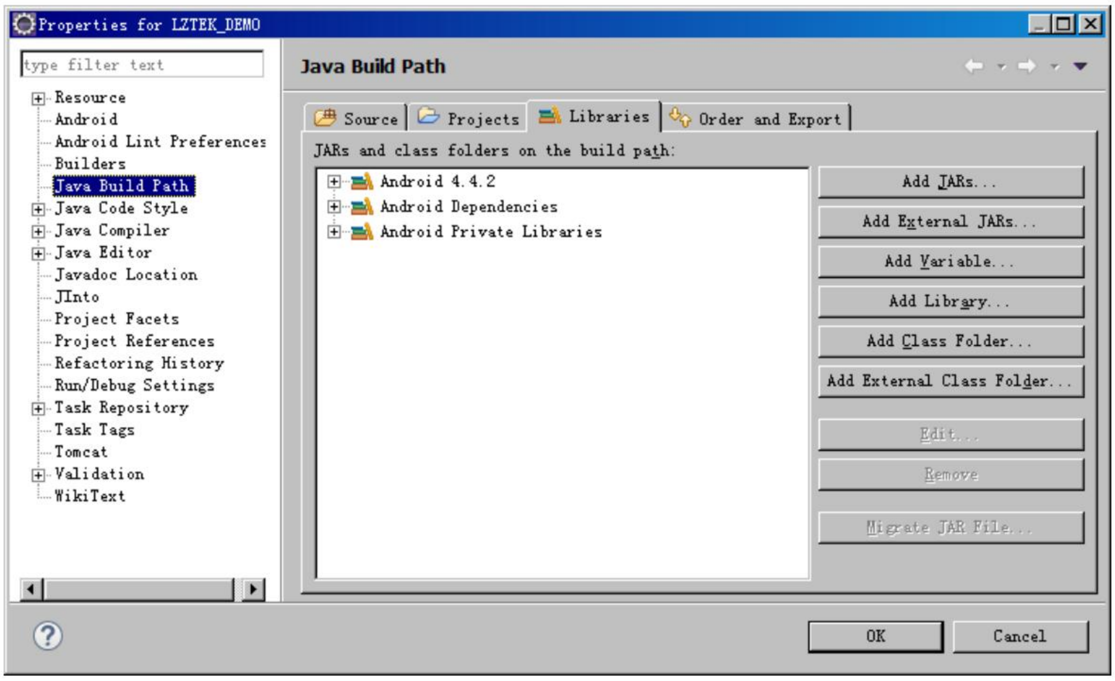
Tip: The API interface in this document can only be used on our company's Android version 20170518 or above!

1.1 Eclipse IDE integration

ÿ How to use the API in Eclipse is as follows:

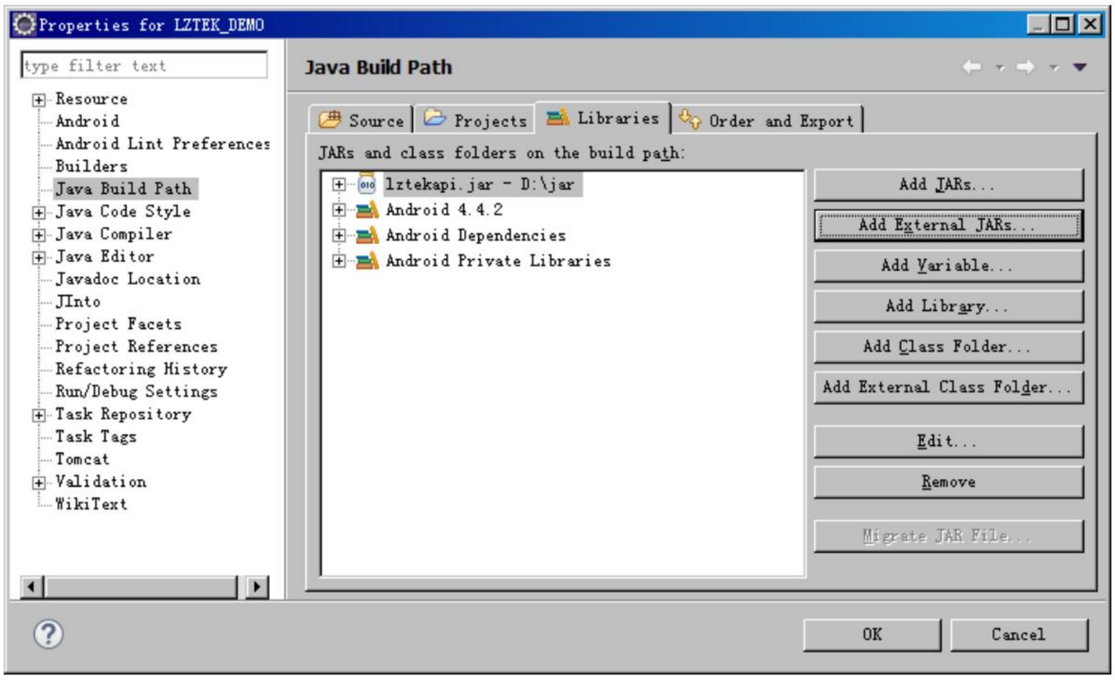
1) Open Eclipse IDE, select the project, right-click, and select Properties.

2) In the pop-up dialog box, select "Java Build Path" and select the "Libraries" tab on the right.



Machine Translated by Google

3) In Libraries, click the "Add External Jars" button on the right, and select "sdkapi.jar" in the pop-up dialog box.



JARs and class folders on the build path:
- lztekapi.jar - D:\jar
- Android 4.4.2
- Android Dependencies
- Android Private Libraries

4) Select the "Libraries" tab on the right, click the "Add External Jars" button on the right, and select the corresponding directory in the pop-up dialog box.

Add "sdkapi.jar" to the directory.

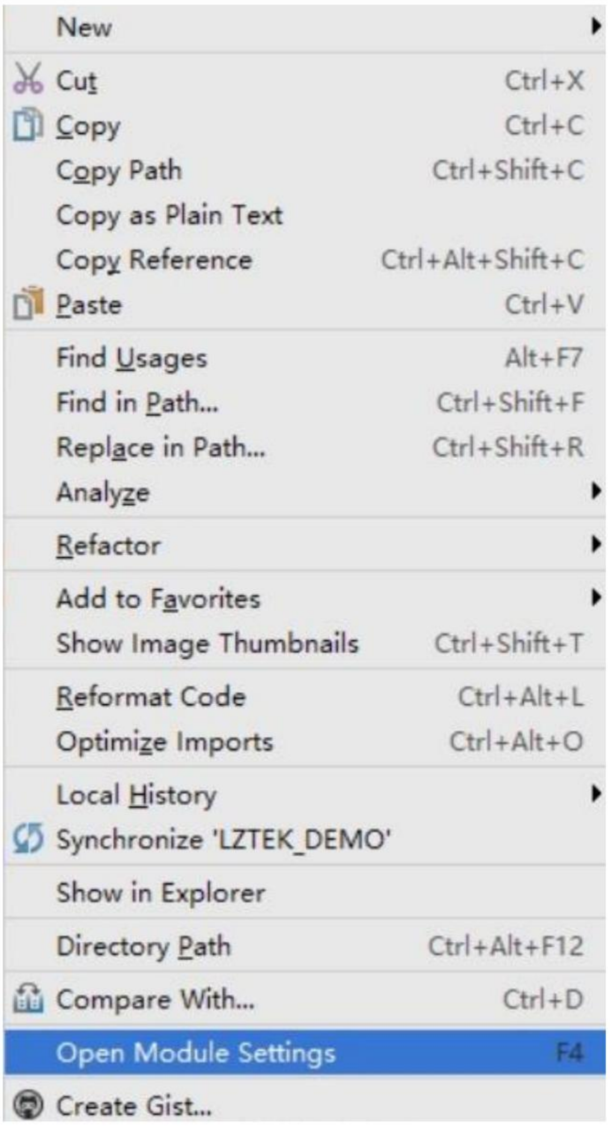
1.2 Android Studio Integration

ÿ How to use the API in Android Studio is as follows:

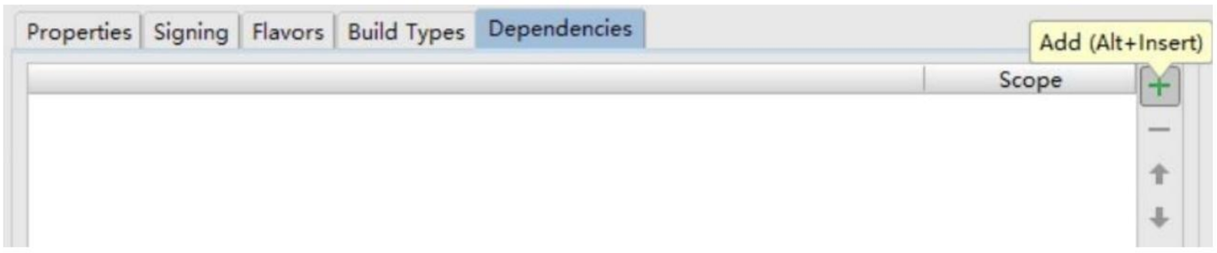
1) Put sdkapi.jar into the third-party library folder of the project module, such as app\libs.

2) Right-click the project name in the Project View to pop up the function menu and click "Open Module Settings".

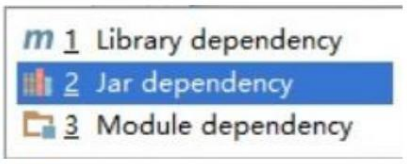
Machine Translated by Google



3) Select the Dependencies tab.

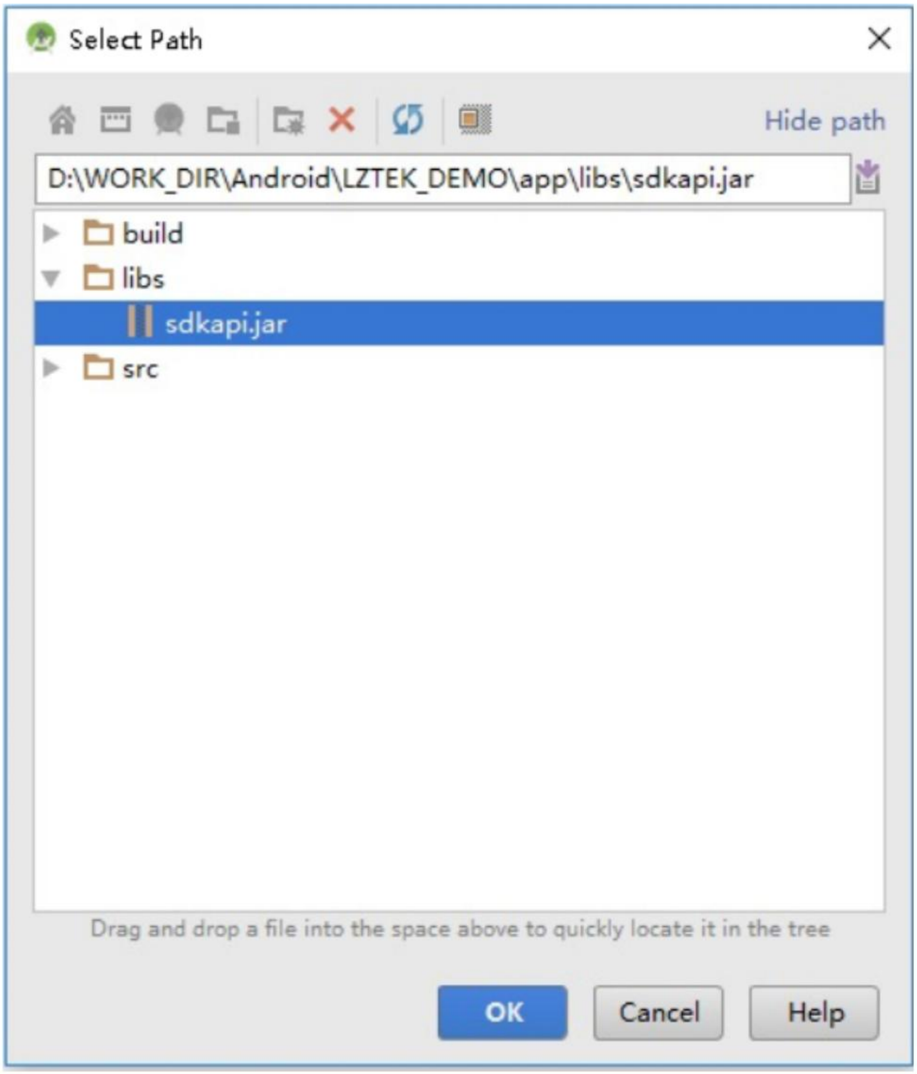


4) Click the plus sign on the right and select "2 Jar Dependency" in the pop-up menu



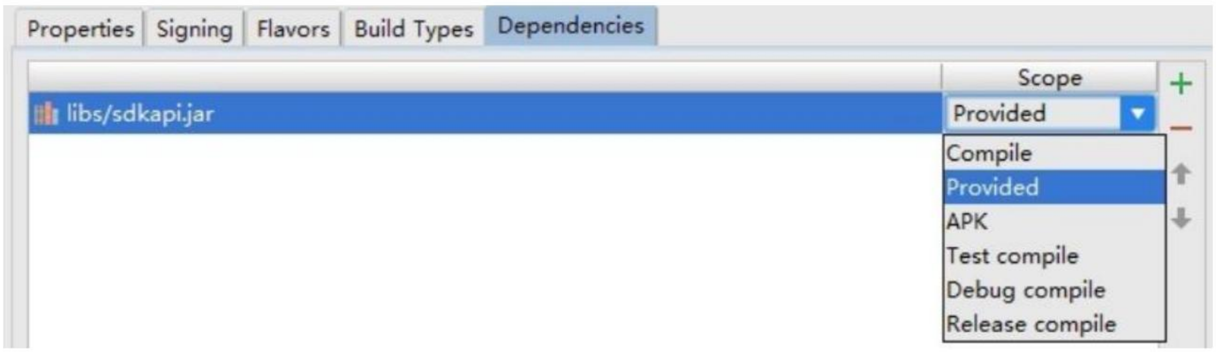
Machine Translated by Google

5) Select sdkapi.jar in the file selection dialog and click the “OK” button to return.


Select Path
Hide path
D:\WORK_DIR\Android\LZTEK_DEMO\app\libs\sdkapi.jar
build
libs
sdkapi.jar
src
Drag and drop a file into the space above to quickly locate it in the tree
OK
Cancel
Help

6) Select the sdkapi.jar row in the list box, click the drop-down box in the "Scope" column on the right, and select the "Provided" item in the drop-down box.

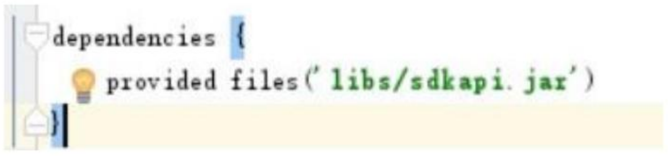
And click the "OK" button below to confirm.



Machine Translated by Google

Note 1: For versions prior to Android Studio 3.0, you can edit the project module build.gradle file and add dependency configuration

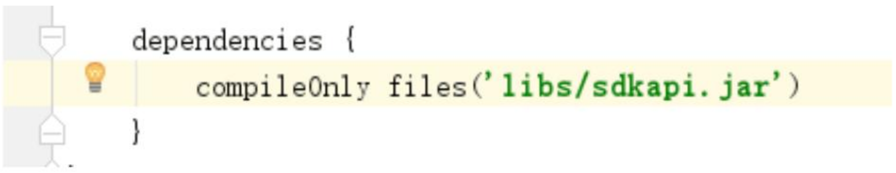
The content of this section is such as provided files('libs/sdkapi.jar').



```
dependencies {
    provided files('libs/sdkapi.jar')
}
```

Note 2: For Android Studio 3.0 and above, you can edit the project module build.gradle file and add dependency configuration

The content of the section is such as compileOnly files('libs/sdkapi.jar').



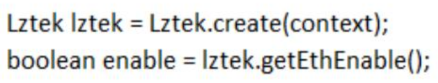
```
dependencies {
    compileOnly files('libs/sdkapi.jar')
}
```

If obfuscation is required, please add the following to the proguard-rules.pro configuration file:

```
-dontwarn com.lztek.toolkit.**
-keep class com.lztek.toolkit.** { *; }
```

1.3 API Call Method

ÿ Add sdkapi.jar to the project, where all APIs can be called through Lztek objects. The calling methods are as follows:



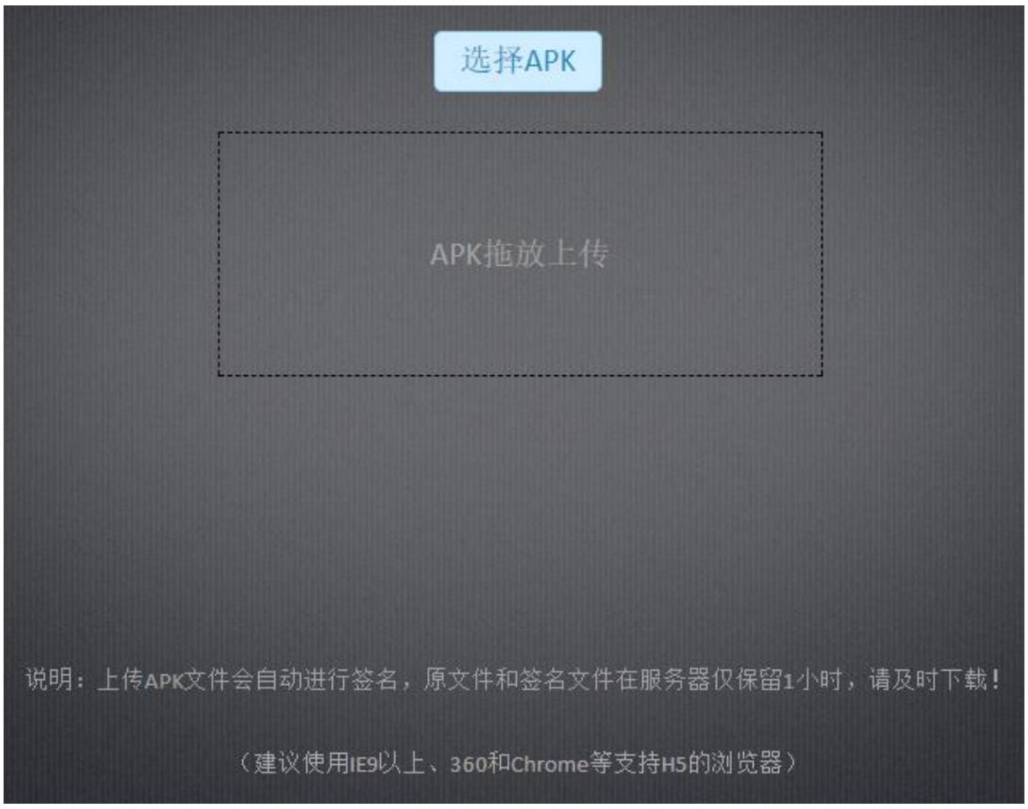
```
Lztek lztek = Lztek.create(context);
boolean enable = lztek.getEthEnable();
```

1.4 APK system signature

If the APK needs to be signed by the system, please visit our signature cloud server http://47.107.162.209:18080 (below 7.1)

Or http://47.107.162.209:19090 (7.1 and above) to upload the APK for signing and downloading.

Machine Translated by Google



说明：上传APK文件会自动进行签名，原文件和签名文件在服务器仅保留1小时，请及时下载！

（建议使用IE9以上、360和Chrome等支持H5的浏览器）

Machine Translated by Google

2 Hardware Management

This chapter describes the software programming interface for motherboard hardware resources, including wireless and network, storage space, running memory, backlight on/off, etc.

Off, system time, GPIO, serial port, hardware watchdog, shutdown/restart, scheduled power on/off, etc.

2.1 Wireless and Network

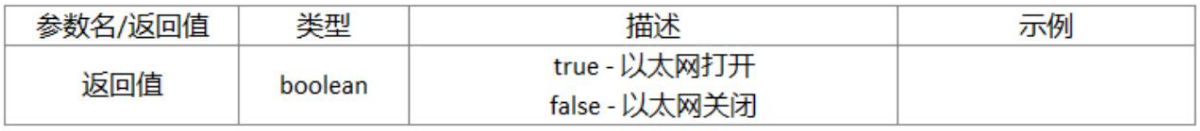
Function overview: Set Ethernet IP parameters, read Ethernet IP parameters, turn Ethernet on/off, read current network connection.

Note: When using the Ethernet settings function, the following permission statement needs to be added to the manifest file:

```
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
```

2.1.1 Read Ethernet status

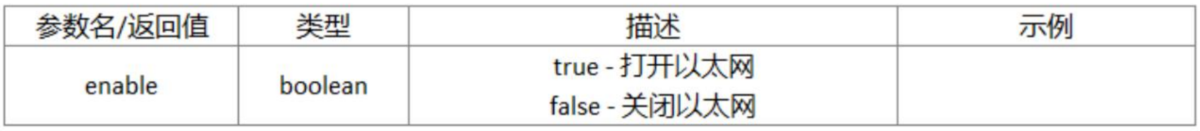
ÿ boolean getEthEnable()



参数名/返回值	类型	描述	示例
返回值	boolean	true - 以太网打开 false - 以太网关闭	

2.1.2 Turn Ethernet on/off

ÿ void setEthEnable(boolean enable)



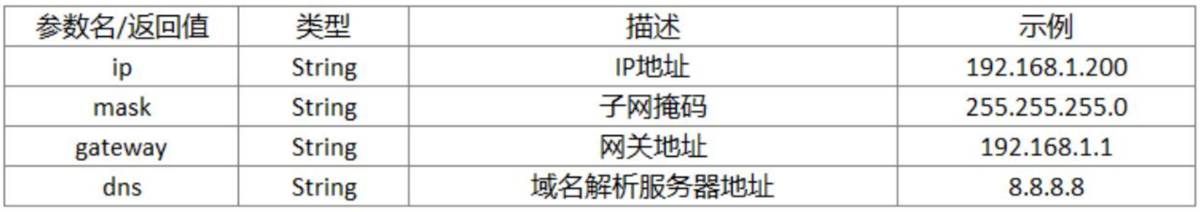
参数名/返回值	类型	描述	示例
enable	boolean	true - 打开以太网 false - 关闭以太网	

2.1.3 Setting Ethernet static address

ÿ void setEthIpAddress(String ip, String mask, String gateway, String dns)

Description: Set Ethernet static address. If Ethernet was previously set to DHCP mode, this call will turn off DHCP mode.

Machine Translated by Google



参数名/返回值	类型	描述	示例
ip	String	IP地址	192.168.1.200
mask	String	子网掩码	255.255.255.0
gateway	String	网关地址	192.168.1.1
dns	String	域名解析服务器地址	8.8.8.8

2.1.4 Setting Ethernet DHCP mode

ÿ void setEthDhcpMode()

Description: Set Ethernet to automatically read IP address mode; if you want to turn off DHCP mode, please directly call Set Ethernet

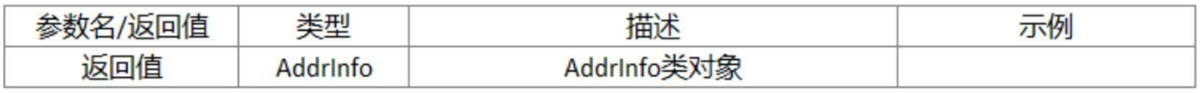
Static address API (setEthIpAddress).

2.1.5 Read Ethernet settings

ÿ AddrInfo getEthAddrInfo()

Description: Returns the network address information class object, which is defined as public class AddrInfo { }, including DHCP mode, IP address,

Subnet mask, gateway address and other information.



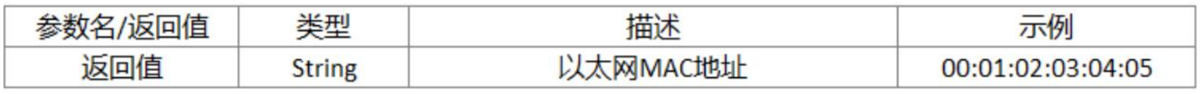
参数名/返回值	类型	描述	示例
返回值	AddrInfo	AddrInfo类对象	

2.1.6 Read Ethernet MAC address

ÿ String getEthMac()

Description: Read the Ethernet hardware address string. This address is generated by CPUID or motherboard serial number by default.

If you want to specify a special MAC address, please contact the original manufacturer's technical support interface.



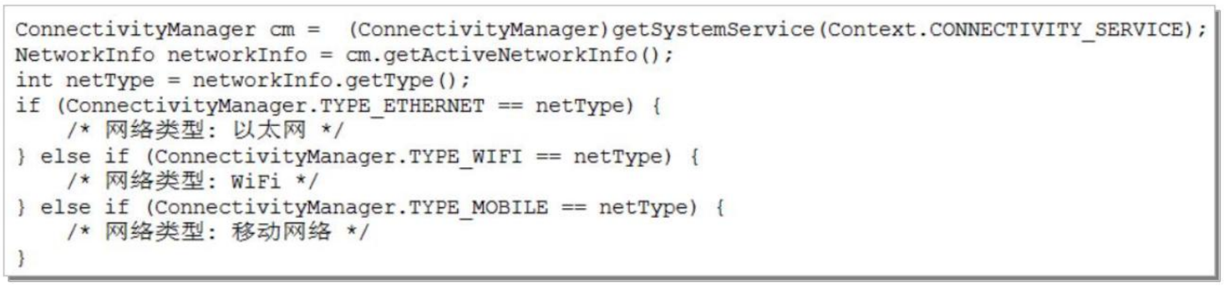
参数名/返回值	类型	描述	示例
返回值	String	以太网MAC地址	00:01:02:03:04:05

Machine Translated by Google

2.1.7 Read the current network type

Please use the getActiveNetworkInfo() method of the Android ConnectivityManager object to obtain the NetworkInfo object.

Use getType() of this object to make a judgment.



```
ConnectivityManager cm =  (ConnectivityManager)getSystemService(Context.CONNECTIVITY_SERVICE);
NetworkInfo networkInfo = cm.getActiveNetworkInfo();
int netType = networkInfo.getType();
if (ConnectivityManager.TYPE_ETHERNET == netType) {
    /* 网络类型: 以太网 */
} else if (ConnectivityManager.TYPE_WIFI == netType) {
    /* 网络类型: WiFi */
} else if (ConnectivityManager.TYPE_MOBILE == netType) {
    /* 网络类型: 移动网络 */
}
```

2.1.8 WiFi Network Operation

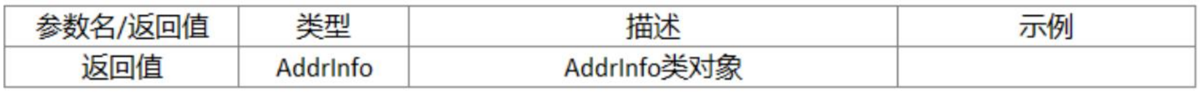
Please use the Android system standard API interface directly.

2.1.9 Reading mobile network settings

ÿ AddrInfo getMobileAddrInfo()

Description: Returns the network address information class object, which is defined as public class AddrInfo { }. Currently, only supports reading IP addresses,

Subnet mask.



参数名/返回值	类型	描述	示例
返回值	AddrInfo	AddrInfo类对象	

2.2 Storage Space

Function overview: read internal storage path, read external SD card path, read external USB disk path, read internal storage space,

Read external SD card space, read USB storage space.

2.2.1 Reading the internal storage path

ÿ String getInternalStoragePath()

Machine Translated by Google

Description: Read the mainboard internal storage path such as /mnt/embsd or /mnt/internal_sd. It is recommended to use the Android system standard method

android.os.Environment.getExternalStorageDirectory() is read to improve program compatibility.

参数名/返回值	类型	描述	示例
返回值	String	主板内置存储器路径	/mnt/internal_sd

2.2.2 Read external SD card path

ÿ String getStorageCardPath()

Description: Read the mainboard external SD card storage path such as /mnt/extsd or /mnt/external_sd.

参数名/返回值	类型	描述	示例
返回值	String	主板外部SD卡路径	/mnt/external_sd

2.2.3 Read external USB disk path

ÿ String getUsbStoragePath()

Description: Read the motherboard USB storage path such as /mnt/udisk or /mnt/usb_storage.

参数名/返回值	类型	描述	示例
返回值	String	主板外部U盘存储路径	/mnt/usb_storage

2.2.4 Reading memory capacity

It is recommended to use the system API to read the capacity of the memory, which can be read through the android.os.StatFs object. For example, read the external

The capacity of the SD card can be realized with the following code:

```
android.os.StatFs statfs = new android.os.StatFs("/mnt/extsd");

long blocSize = statfs.getBlockSizeLong();

long availableSize = statfs.getAvailableBlocksLong()*blocSize;

long totalSize = statfs.getBlockCountLong()*blocSize;
```

Machine Translated by Google

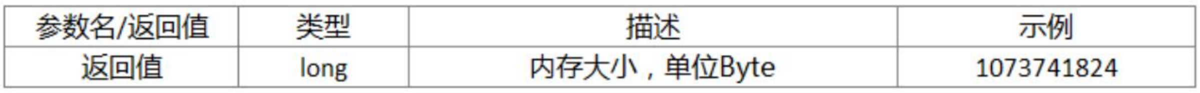
2.3 Running Memory

Function Overview: Read the system running memory size.

2.3.1 Reading system memory size

ÿ long getSystemMemory()

Description: Read the mainboard running memory capacity in bytes. For example, if it is 1GB, the return value is 1024*1024*1024.



参数名/返回值	类型	描述	示例
返回值	long	内存大小，单位Byte	1073741824

2.4 Backlight switch

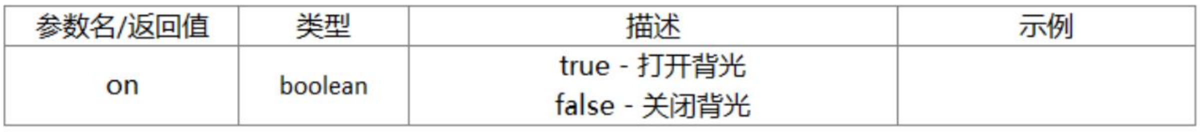
Function Overview: Turn on/off the LCD backlight.

2.4.1 Turn on/off the LCD backlight

ÿ void setLcdBackLight(boolean on)

Description: Turn on or off the backlight power of the LCD screen and turn off the display output signal; note that this interface cannot control the HDMI

Backlight for external monitor/TV.



参数名/返回值	类型	描述	示例
on	boolean	true - 打开背光 false - 关闭背光	

2.5 System Time

Function Overview: Set the Android system clock.

2.5.1 Setting the Android system clock

ÿ void setSystemTime(long milliseconds1970)

Machine Translated by Google

Description: Set the real-time clock time of the Android system. The parameter is the total number of seconds from 1970 to the set point.

Clear permissions and system signature <uses-permission android:name="android.permission.SET_TIME" />.

参数名/返回值	类型	描述	示例
milliseconds1970	long	从1970到设置点的秒数	

2.6 GPIO

Function overview: Set IO input/output, read IO high and low status, set IO high and low status.

2.6.1 Enable IO port

ÿ boolean gpioEnable(int port)

Description: Set a certain IO port to the enabled state. Only after the IO is enabled can the input or output settings be performed.

The number needs to be determined according to the hardware design, please refer to the corresponding motherboard manual. Note: This **API requires root privileges!**

参数名/返回值	类型	描述	示例
port	int	GPIO对应的端口号	180
返回值	boolean	true - 使能IO成功 false - 使能IO失败	

2.6.2 Set IO direction - input

ÿ void setGpioInputMode(int port)

Description: Set a certain IO port to input state; note that you need to enable the IO before setting.

Please check the corresponding motherboard manual. Note: This **API requires root privileges!**

参数名/返回值	类型	描述	示例
port	int	GPIO对应的端口号	180

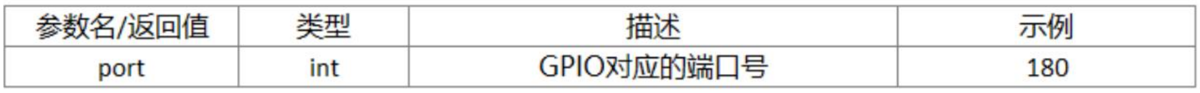
2.6.3 Set IO direction - output

ÿ void setGpioOutputMode(int port)

Machine Translated by Google

Description: Set a certain IO port to output state; note that you need to enable the IO before setting.

Please check the corresponding motherboard manual. Note: This **API requires root privileges!**



参数名/返回值	类型	描述	示例
port	int	GPIO对应的端口号	180

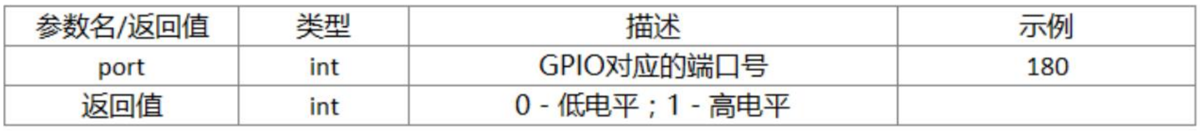
2.6.4 Read IO high and low status

ÿ int getGpioValue(int port)

Description: Read the high and low status of a certain IO port, return 0 for low level, return 1 for high level; note that you need to

Enable IO. If the IO direction is not set, this operation will automatically set the IO as input. The specific IO number needs to be determined according to the hardware.

Please check the corresponding motherboard manual for details. Note: This **API requires root privileges!**



参数名/返回值	类型	描述	示例
port	int	GPIO对应的端口号	180
返回值	int	0 - 低电平；1 - 高电平	

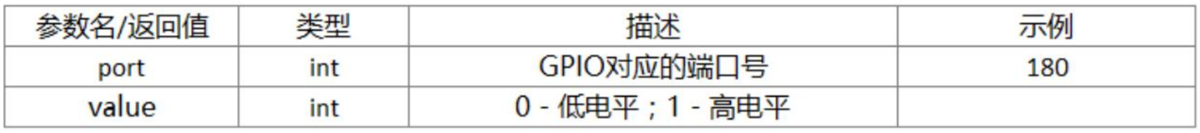
2.6.5 Setting IO high and low status

ÿ void setGpioValue(int port, int value)

Description: Set the high and low state of a certain IO port. Value = 0 represents the output low level, and value = 1 represents the output high level.

Before setting, you need to enable the IO and set it to output state. The specific IO number needs to be determined according to the hardware design. Please check the corresponding

Mainboard manual. Note: This **API requires root privileges!**



参数名/返回值	类型	描述	示例
port	int	GPIO对应的端口号	180
value	int	0 - 低电平；1 - 高电平	

2.7 Serial Port

Function overview: open serial port, close serial port, read data from serial port, write data from serial port.

Machine Translated by Google

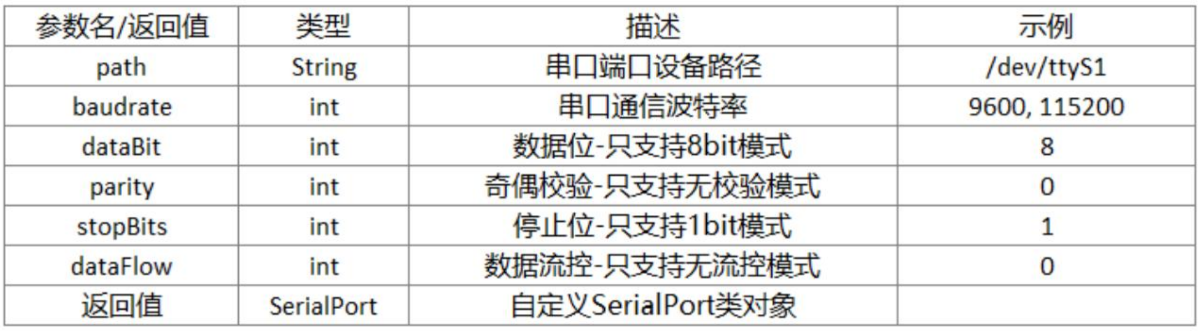
2.7.1 Open the serial port

ÿ SerialPort openSerialPort(String path, int baudrate, int dataBit, int parity, int stopBits, int dataFlow)

Description: Open the specified serial port device and return a SerialPort object. For the definition of this object, please refer to the jar package.

If it fails, it returns null. Currently, this interface only implements the path and baudrate settings. Other parameters are default values. It is recommended to use it directly.

Simplified interface for the web.



参数名/返回值	类型	描述	示例
path	String	串口端口设备路径	/dev/ttyS1
baudrate	int	串口通信波特率	9600, 115200
dataBit	int	数据位-只支持8bit模式	8
parity	int	奇偶校验-只支持无校验模式	0
stopBits	int	停止位-只支持1bit模式	1
dataFlow	int	数据流控-只支持无流控模式	0
返回值	SerialPort	自定义SerialPort类对象	

ÿ SerialPort openSerialPort(String path, int baudrate)

Description: Open the specified serial port device and return a SerialPort object. For the definition of this object, please refer to the jar package.

If it fails, it returns null. This interface only implements the path and baudrate settings, and other parameters are the system default values (data bits = 8, parity

Parity = None, Stop Bits = 8, Flow Control = None).

2.7.2 Serial port reading data

Get an InputStream interface through the getInputStream method of the SerialPort object and use the system API of this interface

Perform data read operation.

2.7.3 Serial port write data

Get an OutputStream interface through the getOutputStream method of the SerialPort object. The system using this interface

API performs data output operations.

Machine Translated by Google

2.7.4 Close the serial port

ÿ void close(SerialPort port)

Description: Close the specified serial port object. Note: When the serial port is closed, the opened stream interface will be automatically closed!

2.8 Hardware Watchdog

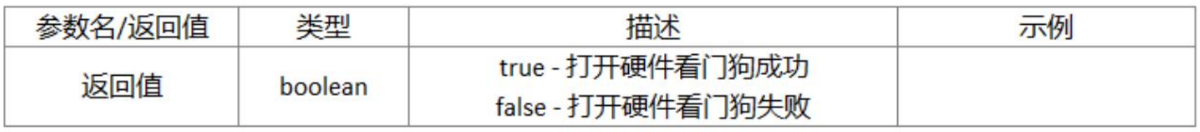
Function overview: turn on the hardware watchdog, turn off the hardware watchdog, feed the hardware watchdog.

2.8.1 Enable the watchdog

ÿ boolean watchDogEnable()

Description: Turn on the hardware watchdog. After the hardware watchdog is turned on, it needs to be fed regularly. Once the software stops feeding the dog, the watchdog timer

After the timeout, the system will automatically reboot. Note: This **API requires root privileges!**



参数名/返回值	类型	描述	示例
返回值	boolean	true - 打开硬件看门狗成功 false - 打开硬件看门狗失败	

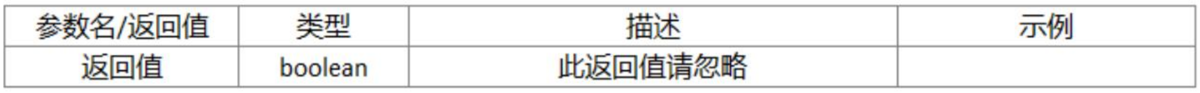
2.8.2 Watchdog Feeding

ÿ boolean watchDogFeed()

Description: Hardware watchdog timing feeding interface. Please ignore the return value of this interface, unless the watchdog device is not turned on and the dog is fed.

The operation will return **false. Note: The hardware watchdog timeout is usually 10 to 20 seconds. Please check it at least every 5 seconds in the application software.**

Perform a dog feeding operation. This API requires root permission!



参数名/返回值	类型	描述	示例
返回值	boolean	此返回值请忽略	

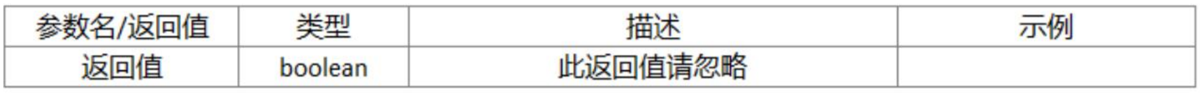
2.8.3 Disable the watchdog

ÿ boolean watchDogDisable()

Machine Translated by Google

Description: Turn off the hardware watchdog. Please ignore the return value of this interface. After the hardware watchdog is turned off, there is no need to feed the watchdog regularly. Note:

This API requires root permissions!



参数名/返回值	类型	描述	示例
返回值	boolean	此返回值请忽略	

2.9 Power on and off

Function overview: hardware shutdown, hardware restart, scheduled restart, scheduled startup.

2.9.1 Hardware shutdown

ÿ void hardShutdown()

Description: Hardware shutdown operation based on the power on/off circuit. After shutdown, the power can only be turned on by remote control, replugging the power supply, or pressing the power button.

Note: This **API requires root privileges!**

2.9.2 Software Restart

ÿ void softReboot()

Description: System hot reset restart achieved through the system reboot command. Note that software restart only makes the CPU restart from the initial instruction

Starts booting, but does not ensure a complete reset of all hardware devices and interfaces. Note: This **API requires root privileges!**

2.9.3 Hardware Restart

ÿ void hardReboot ()

Description: The system is shut down and restarted through the hardware switch circuit. This interface can ensure that the system has 12V input and 5V standby

It is a more reliable power-off restart mechanism. Because this command needs to trigger a shutdown and power-off action,

It cannot restart in real time. The time interval from power failure to power on again is tens of seconds (this time will not exceed 60 seconds at most).

Note: This API requires root permissions!

Machine Translated by Google

2.9.4 Scheduled startup

ÿ void alarmPoweron(int onSeconds)

Description: This interface will shut down the computer immediately after calling it and automatically restart it after the specified number of seconds. The minimum parameter number of seconds is 60 seconds, i.e. 1 minute.

For example, if it is 18:00 now and you want the motherboard to shut down now and start up at 08:00 the next day, the parameter should be set to

50400. Note: This **API requires root privileges!**

参数名/返回值	类型	描述	示例
onSeconds	int	从当前关机点到下次开机点的秒数	

2.10 HDMIIN

Function overview: read HDMI input status, read HDMI input resolution. Note: HDMIIN function requires hardware motherboard support.

2.10.1 Read HDMI input status

ÿ int getHdmiinStatus()

Description: Get the status of HDMI input signal. Return 1 for HDMI input signal, return 0 for no HDMI input signal.

Number.

参数名/返回值	类型	描述	示例
返回值	int	1 - HDMIIN有信号 0 - HDMIIN无信号	

Note: This API requires SDK version 1.1.0.20170818 to support; and this API is only used when the HDMIIN camera preset is not turned on.

Once the signal is detected and enters the HDMIIN preview process, it can only be called by querying the Android system property getprop

sys.hdmiin.status is used to determine the HDMIIN input signal status (return 1 for signal, return 0 for no signal).

2.10.2 Read HDMI input resolution

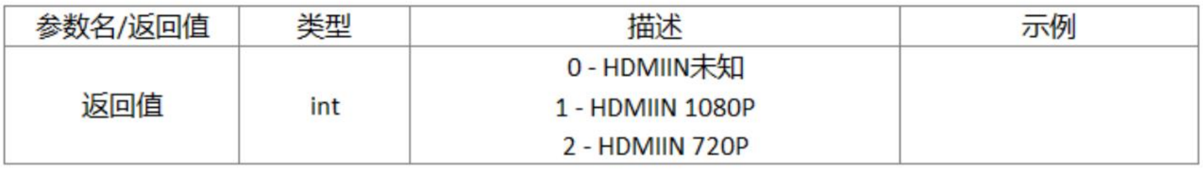
ÿ int getHdmiinResolution()

Machine Translated by Google

Description: Get the resolution of the HDMI input signal. Return 0 for unknown, return 1 for 1080P, i.e. 1920x1080.

2 represents 720P, i.e. 1280x720. Note: Currently only 1080P and 720P input resolutions are supported! And 1.1.0.20170818 is required

Version to support!



参数名/返回值	类型	描述	示例
返回值	int	0 - HDMIIN未知 1 - HDMIIN 1080P 2 - HDMIIN 720P	

Note: This API requires SDK version 1.1.0.20170818 to support; and this API is only used when the HDMIIN camera preset is not turned on.

Once the signal is detected and enters the HDMIIN preview process, it can only be called by querying the Android system property getprop

sys.hdmiin.resolution is used to determine the HDMIIN input signal format (return 1 for 1080P, return 2 for 720P).

Machine Translated by Google

3 Android Management

This chapter describes the software programming interface of the mainboard Android platform related resources, including display management, navigation bar, status bar, application management,

System information, system upgrade, Shell command execution.

3.1 Display Management

Function overview: read resolution, screenshot, screen rotation, read display density, set display density (requires restart).

3.1.1 Reading resolution

Please use the Android system API to read the resolution, such as WindowManager().getDefaultDisplay().getRealMetrics(dm)

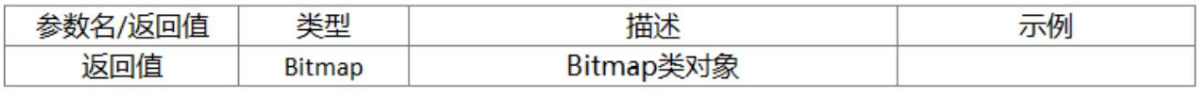
The method can get the physical resolution of the screen, and WindowManager().getDefaultDisplay().getMetrics(dm) can get

Drop the resolution of the navigation bar height.

3.1.2 Screen capture

ÿ Bitmap screenCapture()

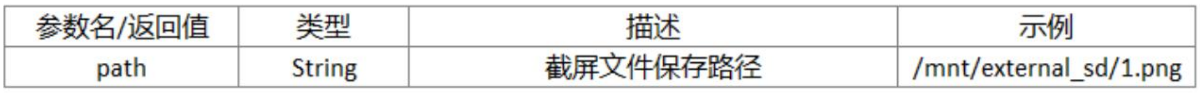
Description: Capture the screen content according to the original size of the screen and save it to a bitmap object; if the call fails, it returns null.



参数名/返回值	类型	描述	示例
返回值	Bitmap	Bitmap类对象	

ÿ void screenCapture(String path)

Description: Capture the screen content according to the original screen size and save it to the specified path file.



参数名/返回值	类型	描述	示例
path	String	截屏文件保存路径	/mnt/external_sd/1.png

Machine Translated by Google

3.1.3 Screen Rotation

Please use the getInt/putInt method of the Android system interface Settings.System to read/set the screen rotation angle property.

The name is Settings.System.USER_ROTATION. The rotation angle value parameters include Surface.ROTATION_0/90/180/270 (for example:

Settings.System.putInt(contentResolver, Settings.System.USER_ROTATION, Surface.ROTATION_270)ÿ

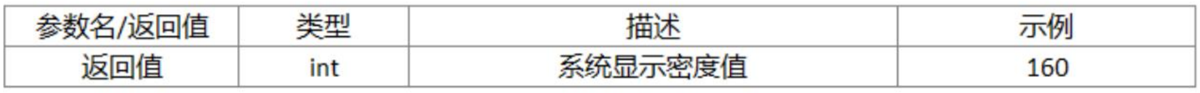
Declare permissions: <uses-permission android:name="android.permission.WRITE_SETTINGS" />

3.1.4 Reading Display Density

ÿ int getDisplayDensity()

Description: Read the Android system display density value (ro.sf.lcd_density). Common density values are 120 - ldpi, 160 - mdpi, 240

- hdpi, 320 - xhdpi, 480 - xxhdpi.



参数名/返回值	类型	描述	示例
返回值	int	系统显示密度值	160

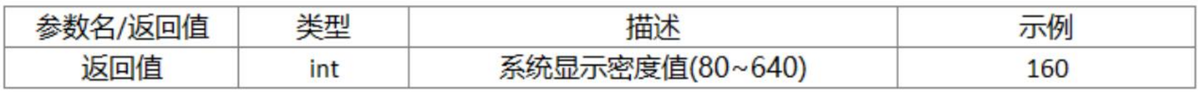
3.1.5 Setting the display density

ÿ void setDisplayDensity(int density)

Description: Set the display density value of the Android system. **After the API call is completed, the system will automatically restart. After the restart, the new density value can be**

Common density values are 120 - ldpi, 160 - mdpi, 240 - hdpi, 320 - xhdpi, 480 - xxhdpi. Note: This **API requires**

Need root permissions!



参数名/返回值	类型	描述	示例
返回值	int	系统显示密度值(80~640)	160

3.2 Navigation Bar

Function overview: Show navigation bar, hide navigation bar, turn on/off slide switch, set auto-hide time.

Machine Translated by Google

3.2.1 Display navigation bar

ÿ void showNavigationBar ()

Description: Display the Android bottom navigation bar. If the slide-out navigation bar switch is turned on and the timeout is set to hide, the system is idle.

After the timeout, the navigation bar will be automatically hidden.

3.2.2 Hide the navigation bar

ÿ void hideNavigationBar ()

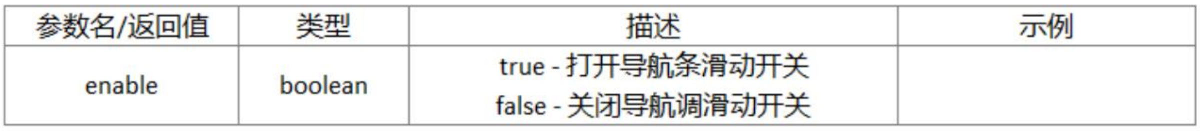
Description: Hide the Android bottom navigation bar.

3.2.3 Open/Close Slide

ÿ void navigationBarSlideShow(boolean enable)

Description: Turn on or off the sliding display system navigation bar property. After turning on the sliding switch, the system navigation bar can be displayed even if it is hidden.

You can swipe up from the bottom of the screen to display the navigation bar.



参数名/返回值	类型	描述	示例
enable	boolean	true - 打开导航条滑动开关 false - 关闭导航调滑动开关	

3.2.4 Navigation bar automatically hides

ÿ void navigationBarMaxIdle(int seconds)

Description: Set the time after which the navigation bar will be automatically hidden when the system is idle. If the timeout is greater than 0,

The navigation bar is automatically hidden if there is no operation for the corresponding number of seconds, and the navigation bar can be called out by sliding subsequently; if the set timeout seconds is less than or equal to

If the navigation bar is already displayed, it will not be automatically hidden.

Machine Translated by Google

3.3 Status Bar

The Android status bar carries the centralized display of dynamic information such as the Android system's network, time, settings, messages, notifications, etc.

Provides methods for dynamic hiding and display control, and only provides special system versions that either do not display the status bar at all or display the standard

And please note: if you use the version that displays the status bar, you must also display the navigation bar, otherwise the sliding of the status bar will

The meeting could not proceed normally.

3.4 Application Management

Function overview: read application list (system/normal/all), silently install applications, silently uninstall applications, and automatically start applications.

3.4.1 Read the application list

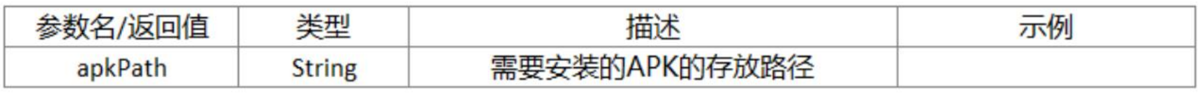
Please use the system API.

3.4.2 Silent application installation

ÿ void installApplication(String apkPath)

Description: Specify the storage path of the **APK** to be installed to automatically complete the installation without user interaction.

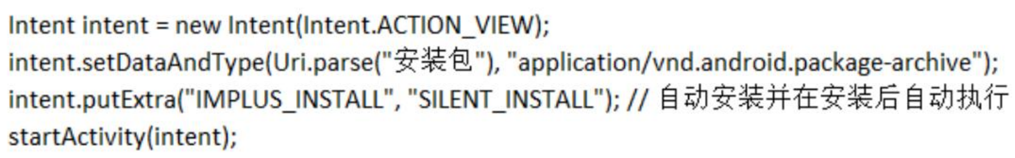
Requires root privileges!



参数名/返回值	类型	描述	示例
apkPath	String	需要安装的APK的存放路径	

3.4.3 Automatic application installation

Please use the following method for automatic installation and execution. During the installation process, an installation progress bar will pop up but no confirmation is required.



```
Intent intent = new Intent(Intent.ACTION_VIEW);
intent.setDataAndType(Uri.parse("安装包"), "application/vnd.android.package-archive");
intent.putExtra("IMPLUS_INSTALL", "SILENT_INSTALL"); // 自动安装并在安装后自动执行
startActivity(intent);
```

Machine Translated by Google

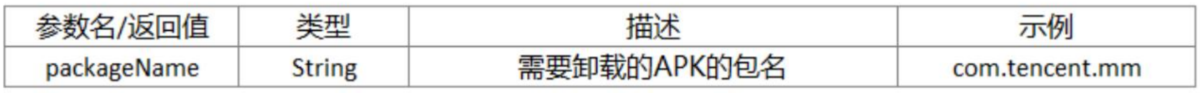
3.4.4 Silent application uninstallation

ÿ void uninstallApplication(String packageName)

Description: Specify the APK package name to be uninstalled to automatically complete the uninstallation, without user interaction.

The parameter is not the APK file name but the actual package name. For example, if you want to uninstall the WeChat program, the package name is usually com.tencent.mm.

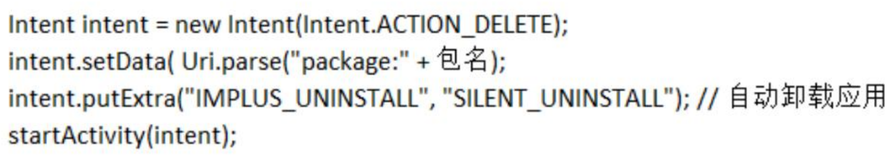
Note: This API requires root permissions!



参数名/返回值	类型	描述	示例
packageName	String	需要卸载的APK的包名	com.tencent.mm

3.4.5 Automatic application uninstallation

Please use the following method to automatically uninstall. No confirmation is required during the uninstallation process.



```
Intent intent = new Intent(Intent.ACTION_DELETE);
intent.setData( Uri.parse("package:" + 包名);
intent.putExtra("IMPLUS_UNINSTALL", "SILENT_UNINSTALL"); // 自动卸载应用
startActivity(intent);
```

3.4.6 Automatic application startup

Reference method: Implement a Receiver in the program, listen for the "android.intent.action.BOOT_COMPLETED" action and process it

For self-start, the application needs to add the "android.permission.RECEIVE_BOOT_COMPLETED" permission.

It provides a startup program in which you can specify third-party programs that need to be started automatically.

3.5 System Information

Function overview: read Android version number, read Android serial number, read system version number, read kernel version, read API version

number, read WiFi hardware address, read Ethernet hardware address.

3.5.1 Read Android version number

It is recommended to use the system API: android.os.Build.VERSION.RELEASE, such as: "4.4.4".

Machine Translated by Google

3.5.2 Read Android serial number

It is recommended to use the system API: android.os.Build.SERIAL, such as "QK2141IQAK". This serial number is generated by the system based on

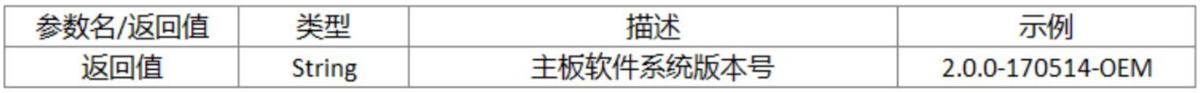
A hardware serial number generated by the WiFi MAC address. Different hardware serial numbers are usually different and can be used as a unique identifier for the hardware.

3.5.3 Reading the system version number

ÿ String getSystemVersion()

Description: Read the OEM release version number of the Android system of the motherboard, such as "2.0.0-170514-OEM". It is recommended to use the system method directly

Such as android.os.Build.DISPLAY read this information.

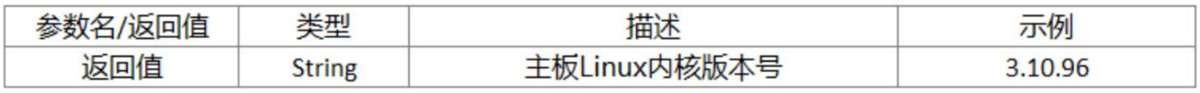


参数名/返回值	类型	描述	示例
返回值	String	主板软件系统版本号	2.0.0-170514-OEM

3.5.4 Read the kernel version

ÿ String getKernelVersion()

Description: Read the mainboard Linux kernel version number, such as "3.10.96".

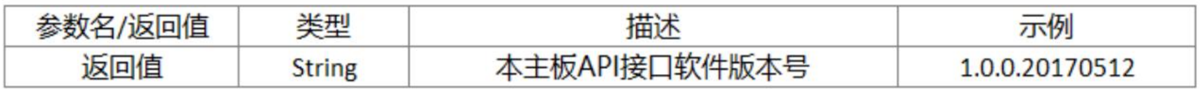


参数名/返回值	类型	描述	示例
返回值	String	主板Linux内核版本号	3.10.96

3.5.5 Reading API Version

ÿ String getApiVersion()

Description: Read the unified software version number of the API interface described in this document, such as "1.0.0.20170512".



参数名/返回值	类型	描述	示例
返回值	String	本主板API接口软件版本号	1.0.0.20170512

3.5.6 Read WiFi hardware address

It is recommended to use the system API to read the WifiInfo object through the getConnectionInfo() method of WifiManager and use WifiInfo

The object's getMacAddress() gets the WiFi hardware address.

Machine Translated by Google

3.5.7 Read Ethernet hardware address

Please refer to the "Reading Ethernet MAC Address" section in 2.1.6.

3.6 System Upgrade

Function Overview: Specify the system OTA upgrade file and complete the system upgrade.

3.6.1 System OTA Upgrade

ÿ void updateSystem(String updateFilePath)

Description: Specify the Android system OTA upgrade package (usually the OTA package update.zip provided by the motherboard manufacturer) and automatically reboot to enter

OTA upgrade mode completes the system upgrade. During the upgrade process, an Android robot animation and progress bar interface will appear. Wait for the upgrade progress to complete.

After the system is completed, it will automatically restart and enter Android. Note: This **API requires root permission! Please make sure that the upgrade file is**

To ensure the integrity of the file, it is recommended that you verify the file size and content after copying the file to the specified path. This will ensure that the file is

The upgrade is completed reliably and no content is in the hardware cache (not yet synchronized to the physical memory).

The level package is no longer needed, please delete the file yourself to save storage space.

参数名/返回值	类型	描述	示例
updateFilePath	String	需要升级的安卓系统包路径	

3.7 Shell command execution

Function Overview: Execute the specified Shell command.

3.7.1 Shell command execution

ÿ void suExec(String command)

Description: Execute Linux Shell commands with root privileges . **Note: This API requires root privileges!**

The superuser privileges are implemented through a background service program. The execution of shell commands in this interface is not completely blocked synchronously.

Machine Translated by Google

For example, using this interface to perform a file copy operation does not guarantee that the call will return after the file copy is completed.

Please refer to the following code to implement strict synchronous execution, but please strictly avoid waiting deadlock between processes.

参数名/返回值	类型	描述	示例
command	String	需要执行的Shell命令	chmod 0666 /dev/video0

ÿ Strictly synchronous shell command calling method (no API provided):

```
private static String suExecWait (String command, File workingDirectory) {
        if (command == null || (command=command.trim()).length() == 0)
                return null;
        if (workingDirectory == null)
                workingDirectory = new File("/");
        java.io.OutputStream out = null;
        java.io.InputStream in = null;
        java.io.InputStream err = null;
        try {
                Runtime runtime = Runtime.getRuntime(); Process
                process = runtime.exec("su", null, workingDirectory); StringBuffer inString = new
                StringBuffer();
                StringBuffer errString = new StringBuffer();
                out = process.getOutputStream();

                out.write(command.endsWith("\n")? command.getBytes() : (command + "\n").getBytes());
                out.write(new byte[]{'e', 'x', 'i', 't', '\n'});

                in = process.getInputStream(); err =
                process.getErrorStream();

                process.waitFor(); // This line will block execution until the command returns

                while (in.available() > 0)
                        inString.append((char)in.read()); while
                (err.available() > 0)
                        errString.append((char)err.read());
                return inString.toString();
        } catch (Exception ioex) {
                return null;
        } finally {
                closeStream(out);
                closeStream(in);
                closeStream(err);
```

Machine Translated by Google

```
        }

    }
```

Machine Translated by Google

4 Scheduled power on/off broadcast interface

[Function Description]: Scheduled power on/off setting broadcast interface (need to install the Scheduled power on/off Apk program)

Note: Due to the static broadcast permission restrictions of Android 8.0 or above, the corresponding package name must be specified when sending a broadcast.

Intent.setPackage("com.lztek.bootmaster.poweralarm7") must be added.

• Daily mode timed on/off com.lztek.tools.action.ALARM_DAILY

onTime -- boot time, type String, format HH:mm, 24-hour hour and minute, such as 08:05, an empty string represents

No power-on time setting

offTime -- shutdown time, type String, format HH:mm, 24-hour hour and minute, such as 20:30, empty string means no

Shutdown time setting

Calling example (start up at 08:05 and shut down at 20:30 every day):

```
Intent intent = new Intent("com.lztek.tools.action.ALARM_DAILY");
intent.putExtra("onTime", "08:05");
intent.putExtra("offTime", "20:30");
intent.setPackage("com.lztek.bootmaster.poweralarm7"); // Android 8.0 or above
context.sendBroadcast(intent);
```

• Weekly mode timer switch com.lztek.tools.action.ALARM_WEEKLY

onTime -- boot time, type String[], array length must be 7 (Sunday to Saturday), array element format HH:mm,

An empty string means no power-on time setting for that day

offTime -- shutdown time, type String[], array length must be 7 (Sunday to Saturday), array element format HH:mm,

An empty string means no computer time setting for the day

Machine Translated by Google

Calling example (start at a specified time on Monday, Tuesday, Wednesday, and Thursday; shut down at a specified time on Monday, Tuesday, Wednesday, and Friday):

```
Intent intent = new Intent("com.lztek.tools.action.ALARM_WEEKLY");
intent.putExtra("onTime", new String[]{"", "08:05", "09:15", "10:00", "10:00", "", ""});
intent.putExtra("offTime", new String[]{"", "21:45", "21:05", "21:00", "", "22:00", ""});
intent.setPackage("com.lztek.bootmaster.poweralarm7"); // Android 8.0 or above
context.sendBroadcast(intent);
```

• Clear the scheduled power on/off settings com.lztek.tools.action.ALARM_UNSET

Calling example:

```
Intent intent = new Intent("com.lztek.tools.action.ALARM_UNSET");
intent.setPackage("com.lztek.bootmaster.poweralarm7"); // Android 8.0 or above
context.sendBroadcast(intent);
```

Machine Translated by Google

5 Application startup management broadcast interface

[Function Description]: Application startup management - boot directly to the application guard broadcast interface (application startup management apk program needs to be installed)

Note: Due to the static broadcast permission restrictions of Android 8.0 or above, the corresponding package name must be specified when sending a broadcast.

Intent.setPackage("com.lztek.bootmaster.autoboot7") must be added.

• Set application guard com.lztek.tools.action.KEEPALIVE_SETUP

packageName -- the name of the application package that needs to be protected, type String

delaySeconds -- the number of seconds to delay the application from starting after exiting, type int, default value 0 seconds to start immediately

foreground -- the application remains running in the foreground (reopened if not in the foreground), type boolean, default value true

Calling example:

```
Intent intent = new Intent("com.lztek.tools.action.KEEPALIVE_SETUP");
intent.putExtra("packageName", "xxx.xxxx.xxx");
//intent.putExtra("delaySeconds", 5); // Restart the app 5 seconds after exiting
//intent.putExtra("foreground", false); // The application can run in the background and reopen after the process exits
intent.setPackage("com.lztek.bootmaster.autoboot7"); // Android 8.0 or above
context.sendBroadcast(intent);
```

• Cancel application guard com.lztek.tools.action.KEEPALIVE_UNSET

packageName -- the name of the application package that needs to be protected, type String

Calling example:

```
Intent intent = new Intent("com.lztek.tools.action.KEEPALIVE_UNSET");
```

Machine Translated by Google

```
intent.putExtra("packageName", "xxx.xxxx.xxx");

intent.setPackage("com.lztek.bootmaster.autoboot7"); // Android 8.0 or above

context.sendBroadcast(intent);
```

• Cancel all application guards com.lztek.tools.action.KEEPALIVE_UNSET_ALL

Calling example:

```
Intent intent = new Intent("com.lztek.tools.action.KEEPALIVE_UNSET_ALL");

intent.setPackage("com.lztek.bootmaster.autoboot7"); // Android 8.0 or above

context.sendBroadcast(intent);
```

• Set the application to boot directly to com.lztek.tools.action.BOOT_SETUP

packageName -- the name of the application package to be started directly at boot, type String

delaySeconds -- Delay the number of seconds to start directly, type int, default value 0 seconds means start immediately

Calling example:

```
Intent intent = new Intent("com.lztek.tools.action.BOOT_SETUP");

intent.putExtra("packageName", "xxx.xxxx.xxx");

//intent.putExtra("delaySeconds", 5); // Run the specified APK 5 seconds after booting up

intent.setPackage("com.lztek.bootmaster.autoboot7"); // Android 8.0 or above

context.sendBroadcast(intent);
```

• To cancel the application boot, go to com.lztek.tools.action.BOOT_UNSET

packageName -- the name of the application package to be started directly at boot, type String

Calling example:

```
Intent intent = new Intent("com.lztek.tools.action.BOOT_UNSET");
```

Machine Translated by Google

```
intent.putExtra("packageName", "xxx.xxxx.xxx");
intent.setPackage("com.lztek.bootmaster.autoboot7"); // Android 8.0 or above
context.sendBroadcast(intent);
```

• Cancel all applications and go directly to com.lztek.tools.action.BOOT_UNSET_ALL

Calling example:

```
Intent intent = new Intent("com.lztek.tools.action.BOOT_UNSET_ALL");
intent.setPackage("com.lztek.bootmaster.autoboot7"); // Android 8.0 or above
context.sendBroadcast(intent);
```