

键盘助手SDK API说明 V1.0

keyboardemulator SDK API Documentation V1.0

一、SDK概览

I.SDK Overview

二、详细API说明

II. Detail API Documentation

2.1 getScannerInerface()

2.2 open(Context context)

2.3 close(Context context)

2.4 enableFunction(Context context, int function)

2.5 disableFunction(Context context, int function)

2.6 startScan(Context context, int function)

2.7 stopScan(Context context, int function)

2.8 enablePlaySuccessSound(Context context, boolean successSound)

2.9 enablePlayFailureSound(Context context, boolean failureSound)

2.10 enableVibrate(Context context, boolean vibrate)

2.11 setOutputMode(Context context, int outputMode)

2.12 setBarcodeEncodingFormat(Context context, int format)

2.13 enableEnter(Context context, boolean enter)

2.14 enableTAB(Context context, boolean tab)

2.15 setSuffix(Context context, String suffix)

2.16 setPrefix(Context context, String prefix)

2.17 filterCharacter(Context context, String chars)

2.18 interceptTrimLeft(Context context, int num)

2.19 interceptTrimRight(Context context, int num)

2.20 setScanOutTime(Context context, int time)

2.21 setContinuousScan(Context context, boolean isContinuous)

2.22 setContinuousScanIntervalTime(Context context, int intervalTime)

2.23 setContinuousScanTimeOut(Context context, int timeOut)

2.24 setScanKey(Context context, int type, int[] scanKey)

2.25 setScanResultBroadcast(Context context, String broadcastAction, String data)

2.26 ScannerUtility Constants

一、SDK 概览

I. SDK Overview

ScannerUtility是键盘助手工具类，用于控制键盘助手，采用单例模式设计，统一封装了条码扫描的控制接口。该类提供扫描设备的全生命周期管理（打开 / 关闭）、功能配置（扫描键设置、声音 / 振动反馈、输出模式调整）、数据格式处理（前缀 / 后缀、编码格式、字符过滤）、连续扫描参数配置。

ScannerUtility is a keyboard emulator utility class designed for controlling the keyboard emulator. It adopts a singleton pattern and provides a unified encapsulation for the control interfaces of barcode scanning. This class provides full lifecycle management for the scanning device (opening/closing), function configuration (scan key settings, sound/vibration feedback, output mode adjustment), data format processing (prefix/suffix, encoding format, character filtering), and continuous scanning parameter configuration.

1. 使用步骤

1. Usage Steps

第一步: 通过 `ScannerUtility.getScannerInterface()` 获取扫码器工具类实例。

Step 1: Obtain an instance of the barcode scanner utility class via `ScannerUtility.getScannerInterface()`.

第二步: 通过 `setOutputMode(context, 2)` 设置条码数据输出模式为「广播接收模式」。

Step 2: Set the barcode data output mode to "Broadcast Receiving Mode" by calling `setOutputMode(context, 2)`.

第三步: 通过 `setScanResultBroadcast(context, barcodeDataAction, intentExtraKey)` 配置条码数据广播的接收参数（用于上层监听扫码结果）

Step 3: Configure the receiving parameters of the barcode data broadcast by calling `setScanResultBroadcast(context, barcodeDataAction, intentExtraKey)` (for the application layer to receive scan results).

第四步:通过 `enableFunction(context, 2)` 启用扫码功能

Step 4: Enable the barcode scanning function by calling `enableFunction(context, 2)`.

第五步:通过 `open(context)` 打开键盘助手

Step 5: Open the keyboardemulator by calling `open(context)`.

第六步:通过 `startScan(context, 2)` 启动扫码（设备进入扫码就绪状态，识别到条码后触发广播）

Step 6: Initiate barcode scanning by calling `startScan(context, 2)` (the device enters the scan-ready state, and a broadcast will be triggered once a barcode is recognized).

第七步:通过 `stopScan(context, 2)` 停止扫码

Step 7: Terminate barcode scanning by calling `stopScan(context, 2)`.

第八步:通过 `disableFunction(context, 2)` 禁用扫码功能

Step 8: Disable the barcode scanning function by calling `disableFunction(context, 2)`.

第九步:通过 `close(context)` 关闭键盘助手

Step 9: Close the keyboardemulator by calling `close(context)`.

3.使用示例

3.Usage Examples

```
/**
 * Test method for barcode scanner core operations
 * Function: Configure the scanner to broadcast output mode, enable 2D scanning,
 * execute scan start/stop operations, and release resources finally
 */
public void testBarcodeScanner() {
    // 1. Get the singleton instance of ScannerUtility
    ScannerUtility scannerUtility = ScannerUtility.getScannerInterface();

    // 2. Set barcode data output mode to [Broadcast Receiving Mode]
    // Output mode description: 2-Broadcast push
    final int OUTPUT_MODE_BROADCAST = 2;
    scannerUtility.setOutputMode(context, OUTPUT_MODE_BROADCAST);

    // 3. Configure broadcast receiving parameters for barcode data (for upper layer to monitor scan
    results)
    // Broadcast Action: Must be consistent with the Action registered by the receiver,
    // it is recommended to fill in a custom business Action (e.g., "com.xxx.scanner.BARCODE_DATA")
    String barcodeDataAction = "com.xxx.scanner.BARCODE_DATA";
    // Intent Extra key: Get barcode data from Intent via this Key when receiving broadcast (e.g.,
    "barcode_data")
    String intentExtraKey = "barcode_data";
    scannerUtility.setScanResultBroadcast(context, barcodeDataAction, intentExtraKey);

    //*****
    IntentFilter intentFilter = new IntentFilter();
    // The barcodeDataAction configured via the method: setScanResultBroadcast(context,
    barcodeDataAction, intentExtraKey)
    intentFilter.addAction("com.xxx.scanner.BARCODE_DATA");
    context.registerReceiver(new ResultDataBroadcastReceiver(), intentFilter);
    //*****

    // 4. Enable 2D scanning function (function=2 stands for 2D scanning function)
    final int SCAN_FUNCTION_2D = 2;
    scannerUtility.enableFunction(context, SCAN_FUNCTION_2D);

    // 5. Open the scanner keyboardemulator service
    scannerUtility.open(context);

    // 6. Start 2D scanning (device enters scan ready state, triggers broadcast when barcode is
    recognized)
    scannerUtility.startScan(context, SCAN_FUNCTION_2D);

    // 7. Stop 2D scanning (terminate scan state regardless of exceptions)
    scannerUtility.stopScan(context, SCAN_FUNCTION_2D);
    //.....
    //.....
    //.....

    // 8. Disable 2D scanning function (release 2D scanning hardware resources)
    scannerUtility.disableFunction(context, SCAN_FUNCTION_2D);

    // 9. Close the scanner keyboardemulator service (release core scanner resources)
    scannerUtility.close(context);
}
```

```
}

class ResultDataBroadcastReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        // The intentExtraKey configured via the method: setScanResultBroadcast(context,
        barcodeDataAction, intentExtraKey)
        String intentExtraKey = "barcode_data";
        String barcodeStringData=intent.getStringExtra(intentExtraKey);
        if(barcodeStringData!=null){
            // Get the original barcode data (the parameter "barcodeByteData" is fixed)
            byte[] barcodeByteData=intent.getByteArrayExtra("barcodeByteData");
            int barcodeSymbology=intent.getIntExtra("barcodeCode", -1);
        }
    }
};
```

二、详细API说明

II.Detailed API description

2.1 getScannerInterface()

项目 Item	说明 Description
原型 Prototype	static ScannerUtility getScannerInterface()
描述 Description	通过单例模式创建ScannerUtility对象 Create an ScannerUtility object via the singleton pattern
参数 Parameters	无 N/A
返回 Return	ScannerUtility 实例 ScannerUtility instance

2.2 open(Context context)

项目 Item	说明 Description
原型 Prototype	void open(Context context)
描述 Description	打开键盘助手总开关 Enable keyboardemulator
参数 Parameters	context: Android上下文 context: Android Context
返回 Return	void

2.3 close(Context context)

项目 Item	说明 Description
原型 Prototype	void close(Context context)
描述 Description	关闭键盘助手总开关 Disable keyboardemulator
参数 Parameters	context: Android上下文 context: Android Context
返回 Return	void

2.4 enableFunction(Context context, int function)

项目 Item	说明 Description
原型 Prototype	void enableFunction(Context context, int function)
描述 Description	启用指定功能模块 Enable specific function module
参数 Parameters	context: Android上下文 context: Android Context function: 功能模块标识 FUNCTION_2D function: Function module identifier FUNCTION_2D
返回 Return	void

2.5 disableFunction(Context context, int function)

项目 Item	说明 Description
原型 Prototype	void disableFunction(Context context, int function)
描述 Description	禁用指定功能模块 Disable specific function module
参数 Parameters	context: Android上下文 context: Android Context function: 功能模块标识 FUNCTION_2D function: Function module identifier FUNCTION_2D
返回 Return	void

2.6 startScan(Context context, int function)

项目 Item	说明 Description
原型 Prototype	void startScan(Context context, int function)
描述 Description	开始扫描或者读卡 start scanning or reading card
参数 Parameters	context: Android上下文 context: Android Context function: 功能模块标识FUNCTION_2D function: Function module identifier FUNCTION_2D
返回 Return	void

2.7 stopScan(Context context, int function)

项目 Item	说明 Description
原型 Prototype	void stopScan(Context context, int function)
描述 Description	停止扫描 Stop scanning
参数 Parameters	context: Android上下文 context: Android Context function: 功能模块标识 FUNCTION_2D function: Function module identifier FUNCTION_2D
返回 Return	void

2.8 enablePlaySuccessSound(Context context, boolean successSound)

项目 Item	说明 Description
原型 Prototype	void enablePlaySuccessSound(Context context, boolean successSound)
描述 Description	扫描成功是否播放提示音 Scan success sound ON/OFF
参数 Parameters	context: Android上下文 context: Android Context successSound: true:播放声音(play sound), false不播放声音(do not play sound) successSound: true:play sound, false:do not play sound
返回 Return	void

2.9 enablePlayFailureSound(Context context, boolean failureSound)

项目 Item	说明 Description
原型 Prototype	void enablePlayFailureSound(Context context, boolean failureSound)
描述 Description	扫描失败是否播放提示音 scan failure sound ON/OFF
参数 Parameters	context: Android上下文 context: Android Context failureSound: true:播放声音(play sound), false不播放声音(do not play sound) failureSound: true:play sound, false:do not play sound
返回 Return	void

2.10 enableVibrate(Context context, boolean vibrate)

项目 Item	说明 Description
原型 Prototype	void enableVibrate(Context context, boolean vibrate)
描述 Description	扫描成功是否震动提示 Enable or disable vibration feedback upon successful scan
参数 Parameters	context: Android上下文 context: Android Context vibrate: true:表示震动(vibrate), false表示不震动(NO vibrate) vibrate: true:vibrate, false:NO vibrate
返回 Return	void

2.11 setOutputMode(Context context, int outputMode)

项目 Item	说明 Description
原型 Prototype	void setOutputMode(Context context, int outputMode)
描述 Description	设置扫描结果的输出模式 Set the output mode of scan results
参数 Parameters	context: Android上下文 context: Android Context outputMode: 输出模式标识, 0:扫描内容到光标位置、1:剪切板、2:广播、3:模拟键盘 outputMode: Output mode identifier, 0:scan content to cursor, 1: clipboard, 2: broadcast, 3: simulated keyboard
返回 Return	void

项目 Item	说明 Description
备注 Remark	当输出模式设置为2（广播输出）时，需通过 <code>setScanResultBroadcast(Context context, String broadcastAction, String data)</code> 方法配置广播接收的相关参数（包括广播Action和Extra数据名称），否则无法正常接收条码扫描结果的广播。 When the output mode is set to 2 (broadcast output), you need to configure the broadcast reception parameters (including broadcast Action and Extra data name) through the <code>setScanResultBroadcast(Context context, String broadcastAction, String data)</code> method; otherwise, the broadcast of barcode scanning results cannot be received properly.

2.12 `setBarcodeEncodingFormat(Context context, int format)`

项目 Item	说明 Description
原型 Prototype	<code>void setBarcodeEncodingFormat(Context context, int format)</code>
描述 Description	设置条码编码格式 Set barcode decoding format
参数 Parameters	context: Android上下文 context: Android Context format: 条码编码格式标识 <code>FORMAT_DEFAULT</code> 、 <code>FORMAT_ASCII</code> 、 <code>FORMAT_GB2312</code> 、 <code>FORMAT_UTF8</code> 、 <code>FORMAT_UNICODE</code> 、 <code>FORMAT_GBK</code> 、 <code>FORMAT_GB18030</code> format: Barcode encoding format identifier <code>FORMAT_DEFAULT</code> 、 <code>FORMAT_ASCII</code> 、 <code>FORMAT_GB2312</code> 、 <code>FORMAT_UTF8</code> 、 <code>FORMAT_UNICODE</code> 、 <code>FORMAT_GBK</code> 、 <code>FORMAT_GB18030</code>
返回 Return	<code>void</code>

2.13 `enableEnter(Context context, boolean enter)`

项目 Item	说明 Description
原型 Prototype	<code>void enableEnter(Context context, boolean enter)</code>
描述 Description	是否启用回车结束符；启用后，会在条码数据末尾自动添加回车字符 Whether to enable the carriage return terminator; if enabled, a carriage return character will be automatically added at the end of the barcode data
参数 Parameters	context: Android上下文 context: Android Context enter: true: 启用回车, false:不启用回车 enter: true: Enter ON, false: Enter OFF
返回 Return	<code>void</code>

2.14 `enableTAB(Context context, boolean tab)`

项目 Item	说明 Description
原型 Prototype	<code>void enableTAB(Context context, boolean tab)</code>

项目 Item	说明 Description
描述 Description	是否启用TAB结束符；启用后，会在条码数据末尾自动添加TAB字符 Whether to enable TAB terminator; if enabled, a TAB character will be automatically added at the end of the barcode data
参数 Parameters	context: Android上下文 context: Android Context tab: true: 启用, false:不启用 tab: true: Tab ON, false: Tab OFF
返回 Return	void

2.15 setSuffix(Context context, String suffix)

项目 Item	说明 Description
原型 Prototype	void setSuffix(Context context, String suffix)
描述 Description	设置条码数据的后缀字符 Set the suffix character for barcode data
参数 Parameters	context: Android上下文 context: Android Context suffix: 后缀字符 suffix: suffix characters
返回 Return	void

2.16 setPrefix(Context context, String prefix)

项目 Item	说明 Description
原型 Prototype	void setPrefix(Context context, String prefix)
描述 Description	设置条码数据的前缀字符 Set the prefix characters of barcode data
参数 Parameters	context: Android上下文 context: Android Context prefix: 前缀字符 prefix: prefix characters
返回 Return	void

2.17 filterCharacter(Context context, String chars)

项目 Item	说明 Description
原型 Prototype	void filterCharacter(Context context, String chars)
描述 Description	过滤条码数据中的指定字符 Filter specified characters in barcode data
参数 Parameters	context: Android上下文 context: Android Context chars: 需要过滤的字符 chars: Characters to be filtered
返回 Return	void

2.18 interceptTrimLeft(Context context, int num)

项目 Item	说明 Description
原型 Prototype	void interceptTrimLeft(Context context, int num)
描述 Description	截取条码数据左侧指定数量的字符 Intercept the specified number of characters from the left side of barcode data
参数 Parameters	context: Android上下文 context: Android Context num: 左侧需要截取的字符数量 num: Number of characters to intercept from the left
返回 Return	void

2.19 interceptTrimRight(Context context, int num)

项目 Item	说明 Description
原型 Prototype	void interceptTrimRight(Context context, int num)
描述 Description	截取条码数据右侧指定数量的字符 Intercept the specified number of characters from the right side of barcode data
参数 Parameters	context: Android上下文 context: Android Context num: 右侧需要截取的字符数量 num: Number of characters to intercept from the right
返回 Return	void

2.20 setScanOutTime(Context context, int time)

项目 Item	说明 Description
原型 Prototype	void setScanOutTime(Context context, int time)
描述 Description	设置扫码超时时间 Set the barcode scanning timeout duration
参数 Parameters	context: Android上下文 context: Android Context time: 超时时间(1-10), 单位为秒 time: Timeout duration(1-10), unit: second
返回 Return	void

2.21 setContinuousScan(Context context, boolean isContinuous)

项目 Item	说明 Description
原型 Prototype	void setContinuousScan(Context context, boolean isContinuous)
描述 Description	设置条码连续扫描模式 Set barcode continuous scanning mode
参数 Parameters	context: Android上下文 context: Android Context isContinuous: true: 连续扫描, false: 单次扫描 isContinuous: true: Continuous scanning, false: Single scanning
返回 Return	void

2.22 setContinuousScanIntervalTime(Context context, int intervalTime)

项目 Item	说明 Description
原型 Prototype	void setContinuousScanIntervalTime(Context context, int intervalTime)
描述 Description	设置条码连续扫描间隔时间 Set the interval time of barcode continuous scanning
参数 Parameters	context: Android上下文 context: Android Context intervalTime: 间隔时间(0-60000), 单位为毫秒 intervalTime: Interval time(0-60000), unit: millisecond
返回 Return	void

2.23 setContinuousScanTimeOut(Context context, int timeOut)

项目 Item	说明 Description
原型 Prototype	void setContinuousScanTimeOut(Context context, int timeOut)
描述 Description	设置条码连续扫描超时时间 Set the timeout duration of barcode continuous scanning
参数 Parameters	context: Android上下文 context: Android Context timeOut: 超时时间(1-99999), 单位为秒 timeOut: Timeout duration(1-99999), unit:second
返回 Return	void

2.24 setScanKey(Context context, int type, int[] scanKey)

项目 Item	说明 Description
原型 Prototype	void setScanKey(Context context, int type, int[] scanKey)
描述 Description	设置扫描/读卡功能的触发按键值 Set the trigger keycode for scanning or card-reading functions
参数 Parameters	context: Android上下文 context: Android Context scanKey: 触发按键值; 条码功能支持4个按键值, UHF (超高频) 功能仅支持2个按键值, 且所有按键值必须大于0 scanKey: Trigger keycode; Barcode function supports 4 keycodes, UHF (Ultra High Frequency) function supports only 2 keycodes, and all keycodes must be greater than 0 type: 功能类型, 0: 条码、3: 超高频 (UHF) type: Function type, 0: Barcode, 3: UHF (Ultra High Frequency)
返回 Return	void
备注 Remark	若需禁用按键触发的扫描功能, 可将所有按键值设置为非扫描按键对应的数值。例如, 要禁止条码扫描按键触发条码扫描, 可将按键值全部设为1, 示例代码如下: <code>setScanKey(context,0,new int[]{1,1,1,1})</code> If you need to disable the key-triggered scanning function, you can set all keycodes to the values corresponding to non-scanning keys. For example, to prevent barcode scanning from being triggered by barcode scan keys, you can set all keycodes to 1, with the sample code as follows: <code>setScanKey(context,0,new int[]{1,1,1,1})</code>

2.25 setScanResultBroadcast(Context context, String broadcastAction, String data)

项目 Item	说明 Description
原型 Prototype	void setScanResultBroadcast(Context context, String broadcastAction, String data)

项目 Item	说明 Description
描述 Description	设置条码扫描结果的广播接收配置 Set the broadcast receiving configuration for barcode scanning results
参数 Parameters	context: Android上下文 context: Android Context broadcastAction: 广播的Action名称 broadcastAction: Action name of the broadcast data: 广播的Extra数据名称 data: Extra data name of the broadcast
返回 Return	void
备注 Remark	仅当 <code>setOutputMode(Context context, int outputMode)</code> 方法设置的输出模式值为2时, 才会有条码扫描结果的广播输出 Only when the output mode value set by the <code>setOutputMode(Context context, int outputMode)</code> method is 2 will the barcode scan results be broadcast Sample code: <pre>// Configure broadcast receiving parameters for barcode data (for the application layer to receive scan results) // Broadcast Action: Must be consistent with the Action registered by the receiver, // it is recommended to fill in a custom business Action (e.g., "com.xxx.scanner.BARCODE_DATA") String barcodeDataAction = "com.xxx.scanner.BARCODE_DATA"; // Intent Extra key: Get barcode data from Intent via this Key when receiving broadcast (e.g., "barcode_data") String intentExtraKey = "barcode_data"; setScanResultBroadcast(context, barcodeDataAction, intentExtraKey); IntentFilter intentFilter = new IntentFilter(); // The barcodeDataAction configured via the method: setScanResultBroadcast(context, barcodeDataAction, intentExtraKey); intentFilter.addAction("com.xxx.scanner.BARCODE_DATA"); context.registerReceiver(new ResultDataBroadcastReceiver(), intentFilter); class ResultDataBroadcastReceiver extends BroadcastReceiver { @Override public void onReceive(Context context, Intent intent) { // The intentExtraKey configured via the method: setScanResultBroadcast(context, barcodeDataAction, intentExtraKey) String intentExtraKey = "barcode_data"; String barcodeStringData=intent.getStringExtra(intentExtraKey); if(barcodeStringData!=null){ // Get the original barcode data (the parameter "barcodeByteData" is fixed) byte[] barcodeByteData=intent.getByteArrayExtra("barcodeByteData"); int barcodeSymbology=intent.getIntExtra("barcodeCode", -1); } } }</pre>

2.26 ScannerUtility常量

2.26 ScannerUtility Constants

com.rscja.scanner.utility.ScannerUtility 类中定义的解码状态常量:

com.rscja.scanner.utility.ScannerUtility Decoder state constants defined in the class:

常量名称 Constant Name	值 Value	描述 Description
com.rscja.scanner.utility.ScannerUtility.FUNCTION_2D	2	功能：二维条码 Function: Barcode 2D
com.rscja.scanner.utility.ScannerUtility.FORMAT_DEFAULT	0	编码格式：默认格式 Decoding format: default format
com.rscja.scanner.utility.ScannerUtility.FORMAT_ASCII	1	编码格式：ASCII Decoding format: ASCII
com.rscja.scanner.utility.ScannerUtility.FORMAT_GB2312	2	编码格式：GB2312 Decoding format: GB2312
com.rscja.scanner.utility.ScannerUtility.FORMAT_UTF8	3	编码格式：UTF8 Decoding format: UTF8
com.rscja.scanner.utility.ScannerUtility.FORMAT_UNICODE	4	编码格式：UNICODE Decoding format: UNICODE
com.rscja.scanner.utility.ScannerUtility.FORMAT_GBK	5	编码格式：GBK Decoding format: GBK
com.rscja.scanner.utility.ScannerUtility.FORMAT_GB18030	6	编码格式：GB18030 Decoding format: GB18030