

手持机条码SDK API说明 V1.0

Handheld Barcode SDK API Documentation V1.0

一、SDK概览

I.SDK Overview

二、详细API说明

II. Detail API Documentation

2.1 open(Context context)

2.2 close()

2.3 startScan()

2.4 stopScan()

2.5 setDecodeCallback(DecodeCallback decodeCallback)

2.6 setTimeout(int timeout)

2.7 isOpen()

2.8 setParameter(int paramNum, int paramVal)

2.9 DecodeCallback Interface

2.10 BarcodeEntity Class

2.10 BarcodeDecoder Constants

一、SDK 概览

I. SDK Overview

`com.rscja.barcode.BarcodeFactory` 作为条码解码器的单例工厂类，核心职责是依据设备底层硬件平台类型，动态创建并返回对应平台的 `com.rscja.barcode.BarcodeDecoder` 实例；而 `com.rscja.barcode.BarcodeDecoder` 作为条码解码功能的抽象基类，不仅定义了条码扫描全流程的核心API接口(包括扫描设备的打开/关闭、扫描触发/终止、解码结果回调、超时设置、参数配置等)，还封装了解码过程中的状态常量（如成功、失败、超时、取消等），为各硬件平台的具体解码实现提供了统一的规范与标准。

`com.rscja.barcode.BarcodeFactory` serves as a singleton factory class for barcode decoders, whose core responsibility is to dynamically create and return the corresponding `com.rscja.barcode.BarcodeDecoder` instance based on the device's underlying hardware platform type. Meanwhile, `com.rscja.barcode.BarcodeDecoder` acts as an abstract base class for barcode decoding functionality—it not only defines the core API interfaces for the entire barcode scanning process (including turning on/off the scanning device, triggering/terminating scanning, decoding result callbacks, timeout settings, parameter configuration, etc.) but also encapsulates status constants during decoding (such as success, failure, timeout, cancellation, etc.), providing a unified specification and standard for the specific decoding implementations of various hardware platforms.

1. 简单易用的单例设计

1. Simple and Easy-to-Use Singleton Design

您无需手动创建该类的实例，只需通过 `BarcodeFactory.getInstance()` 方法即可获取全局唯一的管理对象，确保调用过程简单且资源高效。

You do not need to manually create an instance of this class. You can obtain the globally unique management object only through the `BarcodeFactory.getInstance()` method, ensuring that the calling process is simple and resource-efficient.

2. 使用步骤

2. Usage Steps

第一步: 通过 `open(Context context)` 打开扫描设备（扫描头）。

Step 1: Open the scanning device (scanner engine) via `open(Context context)`.

第二步: 通过 `setDecodeCallback(DecodeCallback decodeCallback)` 设置解码结果回调，用于接收扫描成功后的条码数据。

Step 2: Set the decoding result callback via `setDecodeCallback(DecodeCallback decodeCallback)` to receive barcode data after successful scanning.

第三步: 通过 `startScan()` 触发扫描（启动解码流程），`stopScan()` 终止当前扫描（停止解码流程）

Step 3: Trigger scanning (start the decoding process) via `startScan()`, and terminate the current scanning (stop the decoding process) via `stopScan()`.

第四步: 退出应用时，调用 `close()` 关闭条码设备并释放硬件资源，避免资源泄漏。

Step 4: When exiting the application, call `close()` to shut down the barcode device and release hardware resources to avoid resource leaks.

3.使用示例

3.Usage Examples

```
public void test(){
    // Get BarcodeDecoder instance based on device hardware platform
    BarcodeDecoder decoder= BarcodeFactory.getInstance().getBarcodeDecoder();

    // Open barcode scanning device
    boolean result= decoder.open(context);
    if(!result){
        // Open barcode device failed, exit method
        return;
    }

    // Set callback for receiving decoding result
    decoder.setDecodeCallback(new BarcodeDecoder.DecodeCallback() {
        @Override
        public void onDecodeComplete(BarcodeEntity barcodeEntity) {
            // Callback when decoding is completed (contains barcode data)
            barcodeEntity.getBarcodeData();
        }
    });

    // Start barcode scanning
    decoder.startScan();

    //..... (Other business logic during scanning)

    // Stop barcode scanning
    decoder.stopScan();

    // Exit app: close barcode device and release resources
    decoder.close();
}
```

二、详细API说明

II.Detailed API description

2.1 open(Context context)

项目 Item	说明 Description
原型 Prototype	boolean open(Context context)
描述 Description	打开二维扫描设备 Switch on 2D scanning device
参数 Parameters	context: Android上下文 context: Android Context
返回 Return	true:成功 false:失败 true:success false:failure

2.2 close()

项目 Item	说明 Description
原型 Prototype	void close()
描述 Description	关闭二维扫描设备 Switch off 2D barcode scanning device
参数 Parameters	无 N/A
返回 Return	无 N/A

2.3 startScan()

项目 Item	说明 Description
原型 Prototype	boolean startScan()
描述 Description	触发二维条码扫描 Trigger 2D barcode scanning function
参数 Parameters	无 N/A
返回 Return	true:成功 false:失败 true:success false:failure

2.4 stopScan()

项目 Item	说明 Description
原型 Prototype	void stopScan()
描述 Description	终止扫描 terminate scanning
参数 Parameters	无 N/A
返回 Return	无 N/A

2.5 setDecodeCallback(DecodeCallback decodeCallback)

项目 Item	说明 Description
原型 Prototype	void setDecodeCallback(DecodeCallback decodeCallback)
描述 Description	设置回调对象接收条码数据,主线程回调 Setup call-back target to acquire barcode data (callback in main thread)
参数 Parameters	decodeCallback: 条码数据回调对象 decodeCallback: Barcode data callback object
返回 Return	无 N/A

2.6 setTimeOut(int timeOut)

项目 Item	说明 Description
原型 Prototype	void setTimeOut(int timeOut)
描述 Description	设置超时时间 Setup time-out duration
参数 Parameters	timeOut: 超时时间,单位:秒 timeOut: Time-out duration (Unit: second)
返回 Return	无 N/A

2.7 isOpen()

项目 Item	说明 Description
原型 Prototype	boolean isOpen()

项目 Item	说明 Description
描述 Description	扫描头是否已经打开 Check if the scanner engine is turned on
参数 Parameters	无 N/A
返回 Return	true:扫描头已打开, false:扫描头已关闭 true: Scanner engine is opened, false: Scanner engine is closed

2.8 setParameter(int paramNum, int paramVal)

项目 Item	说明 Description
原型 Prototype	boolean setParameter(int paramNum, int paramVal)
描述 Description	设置扫描头参数（只支持斑马扫描头） Set scanning engine parameters (supports only Zebra scanning engines)
参数 Parameters	paramNum: 参数编号 paramNum: Parameter number paramVal: 参数值 paramVal: Parameter value
返回 Return	true:成功 false:失败 true:success false:failure

2.9 DecodeCallback接口

2.9 DecodeCallback Interface

```
com.rscja.barcode.BarcodeDecoder.DecodeCallback
```

条码解码完成的回调接口，用于接收解码结果
Callback interface for barcode decoding completion, used to receive decoding results

方法 Method	描述 Description	参数 Parameters	返回值 Return
onDecodeComplete(BarcodeEntity barcodeEntity)	解码完成时的回调方法，返回条码解码结果实体 Callback method when decoding is completed, returns barcode decoding result entity	barcodeEntity: 条码解码结果实体（包含结果码、条码数据等信息） barcodeEntity: Barcode decoding result entity (contains result code, barcode data, etc.)	无 N/A

2.10 BarcodeEntity类

2.10 BarcodeEntity Class

```
com.rscja.barcode.BarcodeEntity
```

用于存储条码解码结果相关信息的实体类
Entity class used to store information related to barcode decoding results

方法/属性相关方法 Method	描述 Description	参数 Parameters	返回值 Return
getResultCode()	获取解码结果码（参考BarcodeDecoder的常量定义： DECODE_SUCCESS/DECODE_FAILURE等） Get decoding result code (refer to BarcodeDecoder constant definitions : DECODE_SUCCESS/DECODE_FAILURE, etc.)	无 N/A	int: 解码结果码 int: Decoding result code
getDecodeTime()	获取解码耗时 Get decoding time consumption	无 N/A	int: 解码耗时 int: Decoding time consumption
getBarcodeSymbology()	获取条码类型编码 Get barcode symbology code	无 N/A	int: 条码类型编码 int: Barcode symbology code
getBarcodeName()	获取条码类型名称 Get barcode type name	无 N/A	String: 条码类型名称 String: Barcode type name
getBarcodeBytesData()	获取条码原始字节数据 Get barcode raw byte data	无 N/A	byte[]: 条码原始字节 数据 byte[]: Barcode raw byte data
getBarcodeData()	获取条码字符串数据 Get barcode string data	无 N/A	String: 条码字符串数 据 String: Barcode string data

2.11 BarcodeDecoder常量

2.11 BarcodeDecoder Constants

`com.rscja.barcode.BarcodeDecoder` 类中定义的解码状态常量：

`com.rscja.barcode.BarcodeDecoder` Decoder state constants defined in the class：

常量名称 Constant Name	值 Value	描述 Description
com.rscja.barcode.BarcodeDecoder.DECODE_TIMEOUT	0	解码超时 Decode timeout
com.rscja.barcode.BarcodeDecoder.DECODE_SUCCESS	1	解码成功 Decode success
com.rscja.barcode.BarcodeDecoder.DECODE_CANCEL	-1	解码取消 Decode cancel
com.rscja.barcode.BarcodeDecoder.DECODE_FAILURE	-2	解码失败 Decode failure
com.rscja.barcode.BarcodeDecoder.DECODE_ENGINE_ERROR	-3	扫描头出现错误 Scanner engine error