

TCS1 串口指纹 SDK API 说明 V1.0

TCS1 UART Fingerprint SDK API Documentation V1.0

一、SDK概览

I.SDK Overview

二、详细API说明

II. Detailed API Documentation

2.1 init(Context)

2.2 free()

2.3 startPtCapture()

2.4 startGRAB()

2.5 startTemplateVerify(char[])

2.6 startEnroll()

2.7 startIdentification()

2.8 stopGRAB()

2.9 stopEnroll()

2.10 stopPtCapture()

2.11 stopTemplateVerify()

2.12 stopIdentification()

2.13 setGrabCallBack(GrabCallBack)

2.14 setEnrollCallBack(EnrollCallBack)

2.15 setPtCaptureCallBack(PtCaptureCallBack)

2.16 setIdentificationCallBack(IdentificationCallBack)

2.17 setTemplateVerifyCallBack(TemplateVerifyCallBack)

2.18 ptConvertTemplateEx(DataFormat, byte[], int)

2.19 getPTInfo()

2.20 generateImg(byte[], String)

2.21 deleteAllFingers()

2.22 storeFinger(byte[])

2.23 getFingersCount()

2.30 GRABCallback Interface

2.31 EnrollCallback Interface

2.32 PtCaptureCallback Interface

2.33 IdentificationCallback Interface

2.34 TemplateVerifyCallback Interface

2.35 FingerprintInfo Class

2.36 DataFormat Enumeration

一、SDK 概览

I. SDK Overview

`com.rscja.deviceapi.FingerprintWithFIPS` 是基于 Android 平台的 TCS1 指纹模块操作类，采用单例模式设计，核心作用是为用户提供标准化的 API，实现指纹模块的初始化与资源释放、指纹图像采集、特征值生成与存储（缓冲区 / Flash）、指纹特征上传下载、指纹比对（精确比对 / 库内搜索）、模板管理（删除 / 清空）及指纹图像回调获取等全流程指纹识别功能。

`com.rscja.deviceapi.FingerprintWithFIPS` is a TCS1 fingerprint module operation class based on the Android platform, designed with the singleton pattern. Its core function is to provide a standardized API for upper-layer applications, enabling the full-process fingerprint recognition functions of the fingerprint module including initialization and resource release, fingerprint image acquisition, feature generation and storage (buffer/Flash), fingerprint feature upload and download, fingerprint matching (exact matching/database search), template management (deletion/clearing), and callback acquisition of fingerprint images.

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 `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 `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 `init()` 初始化指纹相关资源。

Step 1: Initialize fingerprint-related resources via `init()`.

第二步: 初始化成功后，根据需求执行以下操作：

Step 2: After successful initialization, perform the following operations as required:

- `startEnroll()` 注册指纹
- `startEnroll()` Register fingerprint
-
- `startIdentification()` 识别指纹
- `startIdentification()` Identify fingerprint
-
- `startGRAB()` 生成指纹图片
- `startGRAB()` Acquire fingerprint image
-
- `storeFinger(byte[] templateData)` 上传模板
- `storeFinger(byte[] templateData)` Store template

- `startPtCapture()` 下载模板
- `startPtCapture()` Acquire template

第三步：退出app调用 `free()` 释放资源。

Step 3: Call `free()` to release resources when exiting the app.

3.使用示例

3.Usage Examples

```
public class FingerprintDemo {
    public void test(){
        int result = fingerprintWithFIPS.init();
        if(result != 0){
            //Initialization failed
            return;
        }
        //*****Register fingerprint*****
        fingerprintWithFIPS.setIdentificationCallBack(new FingerprintWithFIPS.IdentificationCallBack()
{
            @Override
            public void messageInfo(String s) { }
            @Override
            public void onComplete(boolean result, int id,int failureCode) {
                if (result) {
                    //Fingerprint already exists, do not register repeatedly
                } else {
                    final int RESULT_STATUS_NO_MATCH=-3;
                    if(failureCode==RESULT_STATUS_NO_MATCH){
                        //Fingerprint does not exist, start enrollment
                        fingerprintWithFIPS.startEnroll();
                    }
                }
            }
        });
        fingerprintWithFIPS.setEnrollCallBack(new FingerprintWithFIPS.EnrollCallBack() {
            @Override
            public void messageInfo(String s) { }

            @Override
            public void onComplete(boolean result, byte[] bytes, int id,int failureCode) {
                if (result) {
                    //Enrollment succeeded
                } else {
                    //Enrollment failed
                }
            }
        });
        fingerprintWithFIPS.startIdentification();

        //*****Identify fingerprint*****
        fingerprintWithFIPS.setIdentificationCallBack(new FingerprintWithFIPS.IdentificationCallBack()
{
            @Override
            public void messageInfo(String s) { }
            @Override
            public void onComplete(boolean result, int id, int failureCode) {
                if(result) {
                    //Identification succeeded
                }else{
                    //Identification failed
                }
            }
        });
    }
}
```

```

    }
});
fingerprintWithFIPS.startIdentification();

//*****Acquire fingerprint image*****
fingerprintWithFIPS.setGrabCallback(new FingerprintWithFIPS.GRABCallback() {
    @Override
    public void messageInfo(String s) { }

    @Override
    public void onComplete(boolean result, byte[] bytes,int failureCode) {
        if(result) {
            //Image acquisition succeeded
            String filePath = "sdcard/finger.bmp";
            //Save BMP image
            fingerprintWithFIPS.generateImg(bytes, filePath);
        }else{
            //Image acquisition failed
        }
    }

    @Override
    public void progress(int progress) {

    }
});
fingerprintWithFIPS.startGRAB();

//*****Get fingerprint template*****
fingerprintWithFIPS.setPtCaptureCallback(new FingerprintWithFIPS.PtCaptureCallback() {
    @Override
    public void messageInfo(String s) {

    }

    @Override
    public void onComplete(boolean result, byte[] bytes,int failureCode) {
        if (result) {
            //Template acquisition succeeded, bytes contains fingerprint template data
        } else {
            //Template acquisition failed
        }
    }
});
fingerprintWithFIPS.startPtCapture();

//*****Fingerprint template comparison*****
fingerprintWithFIPS.setTemplateVerifyCallback(new FingerprintWithFIPS.TemplateVerifyCallback()
{
    @Override
    public void messageInfo(String s) {}
    @Override
    public void onComplete(boolean result,int failureCode) {
        if(result){
            //Template comparison succeeded
        }else{
            //Template comparison failed
        }
    }
});

```

```
byte[] templateData = readFile();
fingerprintwithFIPS.startTemplateVerify(templateData);

//*****Store fingerprint features to specified Flash
Page*****
byte[] templateData2 = readFile();
int fingerprintID =fingerprintwithFIPS.storeFinger(templateData2);
if (fingerprintID>=0) {
    //Fingerprint template storage succeeded
} else {
    //Fingerprint template storage failed
}

//.....
fingerprintwithFIPS.free();
}
```

二、详细API说明

II.Detailed API Documentation

2.1 init(Context context)

项目 Item	说明 Description
原型 Prototype	boolean init(Context context)
描述 Description	初始化指纹读写器的相关资源 Initialize the relevant resources of the fingerprint reader
参数 Parameters	context: Android上下文对象 context: Android Context object
返回 Return	true:成功 false:失败 true:success false:failure

2.2 free()

项目 Item	说明 Description
原型 Prototype	boolean free()
描述 Description	释放指纹读写器已初始化的相关资源 Release the initialized relevant resources of the fingerprint reader
参数 Parameters	无 N/A
返回 Return	true:成功 false:失败 true:success false:failure

2.3 startPtCapture()

项目 Item	说明 Description
原型 Prototype	void startPtCapture()
描述 Description	启动指纹模板获取流程，需先调用 setPtCaptureCallback(FingerprintwithFIPS.PtCaptureCallback) 方法设置回调接口以接收模板数据，可通过调用 stopPtCapture() 方法终止该获取流程。 Start the fingerprint template capture process, need to call the setPtCaptureCallback(FingerprintwithFIPS.PtCaptureCallback) method first to set up a callback interface for receiving template data, and the capture process can be terminated by calling the stopPtCapture() method.
参数 Parameters	无 N/A
返回 Return	void

2.4 startGRAB()

项目 Item	说明 Description
原型 Prototype	void startGRAB()
描述 Description	启动指纹图像采集流程，需先调用 setGrabCallback(FingerprintwithFIPS.GRABCallback) 方法设置回调接口以接收图像数据，可通过调用 stopGRAB() 方法终止该采集流程。 Start the fingerprint image acquisition process, need to call the setGrabCallback(FingerprintwithFIPS.GRABCallback) method first to set up a callback interface for receiving image data, and the acquisition process can be terminated by calling the stopGRAB() method.
参数 Parameters	无 N/A
返回 Return	void

2.5 startTemplateVerify(char[])

项目 Item	说明 Description
原型 Prototype	void startTemplateVerify(char[] template)
描述 Description	启动指纹模板比对验证流程，需先调用 setTemplateVerifyCallback(FingerprintwithFIPS.TemplateVerifyCallback) 方法设置回调接口以接收验证结果数据，可通过调用 stopTemplateVerify() 方法终止该验证流程。 Start the fingerprint template comparison and verification process, need to call the setTemplateVerifyCallback(FingerprintwithFIPS.TemplateVerifyCallback) method first to set up a callback interface for receiving verification result data, and the verification process can be terminated by calling the stopTemplateVerify() method.
参数 Parameters	template: 待比对的指纹模板数据字符数组 template: Character array of fingerprint template data to be compared
返回 Return	void

2.6 startEnroll()

项目 Item	说明 Description
原型 Prototype	void startEnroll()
描述 Description	启动指纹采集注册流程，需先调用 setEnrollCallback(FingerprintwithFIPS.EnrollCallback) 方法设置回调接口以接收采集数据，可通过调用 stopEnroll() 方法终止该采集注册流程 Start the fingerprint acquisition and enrollment process, you need to call the setEnrollCallback(FingerprintwithFIPS.EnrollCallback) method first to set up a callback interface for receiving acquisition data, and the acquisition and enrollment process can be terminated by calling the stopEnroll() method
参数 Parameters	无 N/A
返回 Return	void
备注 Remark	同一枚指纹不支持重复注册，调用本方法前，需先调用 startIdentification() 方法检测该指纹是否已存在；若指纹已存在，将无法完成注册操作，本次注册会直接失败 The same fingerprint does not support repeated enrollment. Before calling this method, you need to call the startIdentification() method to detect whether the fingerprint already exists; if the fingerprint exists, the enrollment operation cannot be completed and will fail directly

2.7 startIdentification()

项目 Item	说明 Description
原型 Prototype	void startIdentification()
描述 Description	启动指纹身份验证流程，需先调用 setIdentificationCallback(FingerprintwithFIPS.IdentificationCallback) 方法设置回调接口以接收验证数据，可通过调用 stopIdentification() 方法终止该验证流程 Start the fingerprint identity verification process, need to call the setIdentificationCallback(FingerprintwithFIPS.IdentificationCallback) method first to set up a callback interface for receiving verification data, and the verification process can be terminated by calling the stopIdentification() method
参数 Parameters	无 N/A
返回 Return	void

2.8 stopGRAB()

项目 Item	说明 Description
原型 Prototype	void stopGRAB()
描述 Description	停止采集指纹图像 Stop acquiring fingerprint image

项目 Item	说明 Description
参数 Parameters	无 N/A
返回 Return	void

2.9 stopEnroll()

项目 Item	说明 Description
原型 Prototype	void stopEnroll()
描述 Description	停止采集指纹 Stop acquiring fingerprint
参数 Parameters	无 N/A
返回 Return	void

2.10 stopPtCapture()

项目 Item	说明 Description
原型 Prototype	void stopPtCapture()
描述 Description	停止获取指纹模板 Stop acquiring fingerprint template
参数 Parameters	无 N/A
返回 Return	void

2.11 stopTemplateVerify()

项目 Item	说明 Description
原型 Prototype	void stopTemplateVerify()
描述 Description	停止指纹模板比对 Stop fingerprint template comparison
参数 Parameters	无 N/A
返回 Return	void

2.12 stopIdentification()

项目 Item	说明 Description
原型 Prototype	void stopIdentification()
描述 Description	停止指纹验证 Stop fingerprint verification
参数 Parameters	无 N/A
返回 Return	void

2.13 setGrabCallBack(GrabCallBack)

项目 Item	说明 Description
原型 Prototype	void setGrabCallBack(FingerprintWithFIPS.GrabCallBack callBack)
描述 Description	设置指纹图像采集的回调接口，用于接收图像采集的实时数据 Set up callback interface for fingerprint image acquisition, used to receive real-time data of image acquisition
参数 Parameters	callBack: GrabCallBack回调接口实例 callBack: GrabCallBack callback interface instance
返回 Return	void

2.14 setEnrollCallBack(EnrollCallBack)

项目 Item	说明 Description
原型 Prototype	void setEnrollCallBack(FingerprintWithFIPS.EnrollCallBack callBack)
描述 Description	设置指纹采集的回调接口，用于接收指纹采集的实时数据 Set up callback interface for fingerprint acquisition, used to receive real-time data of fingerprint acquisition
参数 Parameters	callBack: EnrollCallBack回调接口实例 callBack: EnrollCallBack callback interface instance
返回 Return	void

2.15 setPtCaptureCallBack(PtCaptureCallBack)

项目 Item	说明 Description
原型 Prototype	void setPtCaptureCallBack(FingerprintWithFIPS.PtCaptureCallBack callBack)

项目 Item	说明 Description
描述 Description	设置指纹模板获取的回调接口，用于接收模板获取的实时数据 Set up callback interface for fingerprint template acquisition, used to receive real-time data of template acquisition
参数 Parameters	callBack: PtCaptureCallBack回调接口实例 callBack: PtCaptureCallBack callback interface instance
返回 Return	void

2.16 setIdentificationCallBack(IdentificationCallBack)

项目 Item	说明 Description
原型 Prototype	void setIdentificationCallBack(FingerprintWithFIPS.IdentificationCallBack callBack)
描述 Description	设置指纹验证的回调接口，用于接收指纹验证的实时数据 Set up callback interface for fingerprint verification, used to receive real-time data of fingerprint verification
参数 Parameters	callBack: IdentificationCallBack回调接口实例 callBack: IdentificationCallBack callback interface instance
返回 Return	void

2.17 setTemplateVerifyCallBack(TemplateVerifyCallBack)

项目 Item	说明 Description
原型 Prototype	void setTemplateVerifyCallBack(FingerprintWithFIPS.TemplateVerifyCallBack callBack)
描述 Description	设置指纹模板比对的回调接口，用于接收模板比对的实时数据 Set up callback interface for fingerprint template comparison, used to receive real-time data of template comparison
参数 Parameters	callBack: TemplateVerifyCallBack回调接口实例 callBack: TemplateVerifyCallBack callback interface instance
返回 Return	void

2.18 ptConvertTemplateEx(DataFormat, byte[], int)

项目 Item	说明 Description
原型 Prototype	byte[] ptConvertTemplateEx(FingerprintWithFIPS.DataFormat jtarget_type, byte[] srcbuf, int jsrclen)
描述 Description	转换指纹模板数据格式，将原数据转换为指定格式的指纹模板数据 Convert fingerprint template data format, transform the original data into fingerprint template data in the specified format

项目 Item	说明 Description
参数 Parameters	jtarget_type: 转换后的指纹数据格式类型（DataFormat枚举值） jtarget_type: Fingerprint data format type after conversion (DataFormat enumeration value) srcbuf: 转换前的指纹模板原始数据字节数组 srcbuf: Byte array of original fingerprint template data before conversion jsrclen: 转换前的指纹模板原始数据长度 jsrclen: Length of original fingerprint template data before conversion
返回 Return	字节数组: 转换后的指纹模板数据 Byte array: Converted fingerprint template data

2.19 getPTInfo()

项目 Item	说明 Description
原型 Prototype	FingerprintWithFIPS.FingerprintInfo getPTInfo()
描述 Description	获取FIPS指纹模块的信息（包含版本号、模块ID等） Get the information of FIPS fingerprint module (including version number, module ID, etc.)
参数 Parameters	无 N/A
返回 Return	FingerprintInfo对象: FIPS指纹模块信息 FingerprintInfo object: FIPS fingerprint module information

2.20 generateImg(byte[], String)

项目 Item	说明 Description
原型 Prototype	boolean generateImg(byte[] data, String filePath)
描述 Description	根据指纹图像数据生成BMP格式图片，并保存到指定文件路径 Generate BMP format image based on fingerprint image data and save it to the specified file path
参数 Parameters	data: 指纹图像数据字节数组 data: Byte array of fingerprint image data filePath: 生成的BMP图片保存的文件路径 filePath: File path for saving the generated BMP image
返回 Return	true:生成并保存成功；false:生成或保存失败 true: Generation and saving success; false: Generation or saving failure

2.21 deleteAllFingers()

项目 Item	说明 Description
原型 Prototype	int deleteAllFingers()

项目 Item	说明 Description
描述 Description	删除指纹模块中存储的所有指纹信息 Delete all fingerprint information stored in the fingerprint module
参数 Parameters	无 N/A
返回 Return	0:操作成功; 非0值:操作失败 0:Operation success; Non-zero value: Operation failed

2.22 storeFinger()

项目 Item	说明 Description
原型 Prototype	int storeFinger(byte[] templateData)
描述 Description	将指纹模板数据导入到指纹模块中并完成存储 Import fingerprint template data into the fingerprint module and complete storage
参数 Parameters	templateData: 指纹模板数据字节数组 templateData: Byte array of fingerprint template data
返回 Return	>=0:操作成功, 返回指纹ID; -1:操作失败 >=0:Operation success, return fingerprint ID; -1: Operation failed

2.23 getFingersCount()

项目 Item	说明 Description
原型 Prototype	int getFingersCount()
描述 Description	获取指纹模块中已采集并存储的指纹数量 Get the number of acquired and stored fingerprints in the fingerprint module
参数 Parameters	无 N/A
返回 Return	>=0:获取成功, 返回指纹数量; -1:获取失败 >=0:Acquisition success, return fingerprint count; -1: Acquisition failed

2.30 GRABCallback 接口

2.30 GRABCallback Interface

指纹图像采集回调接口，用于实时反馈指纹图像采集过程中的消息、完成状态及相关数据。

Callback interface for fingerprint image acquisition, used to real-time feed back messages, completion status and related data during fingerprint image acquisition.

方法 Method	描述 Description	参数 Parameters	返回值 Return
messageInfo(String msg)	接收图像采集的实时状态消息 Receive real-time status messages of image acquisition	msg: 采集图像的实时描述信息 msg: Real-time description information of image acquisition	无 N/A
onComplete(boolean b, byte[] data, int Param)	图像采集完成后的回调方法，返回采集结果及相关数据 Callback method after image acquisition is completed, returns acquisition result and related data	b: true=采集成功，false=采集失败 b: true=Acquisition success, false=Acquisition failure data: 采集到的图像数据（失败时为空） data: Acquired image data (empty if failed) Param: 附加参数，返回采集失败的原因（成功时无实际意义） Param: Error code indicating acquisition status (no practical meaning if successful)	无 N/A
progress(int progress)	返回指纹图像采集的实时进度 Returns the real-time progress of fingerprint image acquisition	progress: 采集进度值（0-100） progress: Acquisition progress value (0-100)	无 N/A

2.31 EnrollCallBack 接口

2.31 EnrollCallBack Interface

指纹采集回调接口，用于实时反馈指纹采集过程中的消息、完成状态及指纹模板数据。

Callback interface for fingerprint enrollment, used to real-time feed back messages, completion status and fingerprint template data during fingerprint acquisition.

方法 Method	描述 Description	参数 Parameters	返回值 Return
messageInfo(String msg)	接收指纹采集的实时状态消息 Receive real-time status messages of fingerprint enrollment	msg: 采集指纹的实时描述信息 msg: Real-time description information of fingerprint enrollment	无 N/A

方法 Method	描述 Description	参数 Parameters	返回值 Return
onComplete(boolean b, byte[] data, int fingerprintID, int Param)	指纹采集完成后的回调方法，返回采集结果及相关数据 Callback method after fingerprint enrollment is completed, returns acquisition result and related data	b: true=采集成功，false=采集失败 b: true=Enrollment success, false=Enrollment failure data: 指纹模板数据（返回null表示采集失败，返回byte[0]表示取消操作） data: Fingerprint template data (return null for enrollment failure, return empty byte array for operation cancellation) fingerprintID: 指纹ID（采集成功时有效） fingerprintID: Fingerprint ID (valid when enrollment succeeds) Param: 附加参数，返回采集失败的原因（成功时无实际意义） Param: Error code indicating enrollment status (no practical meaning if successful)	无 N/A

2.32 PtCaptureCallBack 接口

2.32 PtCaptureCallBack Interface

指纹模板数据获取回调接口，用于实时反馈模板数据获取过程中的消息、完成状态及模板数据。

Callback interface for fingerprint template data acquisition, used to real-time feed back messages, completion status and template data during template data acquisition.

方法 Method	描述 Description	参数 Parameters	返回值 Return
messageInfo(String msg)	接收指纹模板数据获取的实时状态消息 Receive real-time status messages of fingerprint template data acquisition	msg: 获取指纹模板数据的实时描述信息 msg: Real-time description information of fingerprint template data acquisition	无 N/A
onComplete(boolean b, byte[] data, int Param)	指纹模板数据获取完成后的回调方法，返回获取结果及相关数据 Callback method after fingerprint template data acquisition is completed, returns acquisition result and related data	b: true=获取成功，false=获取失败 b: true=Acquisition success, false=Acquisition failure data: 指纹模板数据（失败时为空） data: Fingerprint template data (empty if failed) Param: 附加参数，返回获取失败的原因（成功时无实际意义） Param: Error code indicating acquisition status (no practical meaning if successful)	无 N/A

2.33 IdentificationCallBack 接口

2.33 IdentificationCallBack Interface

指纹验证回调接口，用于实时反馈指纹验证过程中的消息、验证结果及指纹ID。

Callback interface for fingerprint verification, used to real-time feed back messages, verification results and fingerprint ID during fingerprint verification.

方法 Method	描述 Description	参数 Parameters	返回值 Return
messageInfo(String msg)	接收指纹验证的实时状态消息 Receive real-time status messages of fingerprint verification	msg: 验证指纹的实时描述信息 msg: Real-time description information of fingerprint verification	无 N/A
onComplete(boolean b, int id, int Param)	指纹验证完成后的回调方法，返回验证结果及相关数据 Callback method after fingerprint verification is completed, returns verification result and related data	b: true=验证成功, false=验证失败 b: true=Verification success, false=Verification failure id: 指纹ID (id>=0表示验证成功; id=-1表示验证失败; id=-2表示取消操作; id=-3表示未找到指纹) id: Fingerprint ID (id>=0 for verification success; id=-1 for verification failure; id=-2 for operation cancellation; id=-3 for fingerprint not found) Param: 附加参数, 返回验证失败的原因 (成功时无实际意义) Param: Additional parameter, returns the reason for verification failure (no practical meaning if successful)	无 N/A

2.34 TemplateVerifyCallBack 接口

2.34 TemplateVerifyCallBack Interface

指纹模板比对回调接口，用于实时反馈模板比对过程中的消息及比对结果。

Callback interface for fingerprint template comparison, used to real-time feed back messages and comparison results during template comparison.

方法 Method	描述 Description	参数 Parameters	返回值 Return
messageInfo(String msg)	接收指纹模板比对的实时状态消息 Receive real-time status messages of fingerprint template comparison	msg: 指纹模板比对的实时描述信息 msg: Real-time description information of fingerprint template comparison	无 N/A
onComplete(boolean b, int Param)	指纹模板比对完成后的回调方法，返回比对结果 Callback method after fingerprint template comparison is completed, returns comparison result	b: true=比对成功, false=比对失败 b: true=Comparison success, false=Comparison failure Param: 附加参数, 返回比对失败的原因 (成功时无实际意义) Param: Additional parameter, returns the reason for comparison failure (no practical meaning if successful)	无 N/A

2.35 FingerprintInfo 类

2.35 FingerprintInfo Class

指纹信息实体类，用于存储指纹模块的版本号和ID信息。

Fingerprint information entity class, used to store the version number and ID information of the fingerprint module.

方法 Method	描述 Description	参数 Parameters	返回值 Return
getVersions()	获取指纹模块版本号 Get the version number of the fingerprint module	无 N/A	字符串：指纹模块版本号 String: Version number of the fingerprint module
setVersions(String version)	设置指纹模块版本号 Set the version number of the fingerprint module	version: 指纹模块版本号字符串 version: Version number string of the fingerprint module	无 N/A
getId()	获取指纹模块ID（十六进制） Get the fingerprint module ID (hexadecimal)	无 N/A	字符串：十六进制格式的指纹模块ID String: Fingerprint module ID in hexadecimal format
setId(String id)	设置指纹模块ID（十六进制） Set the fingerprint module ID (hexadecimal)	id: 十六进制格式的指纹模块ID字符串 id: Fingerprint module ID string in hexadecimal format	无 N/A

3.36 DataFormat 枚举类

3.36 DataFormat Enumeration Class

指纹数据格式枚举类，定义了指纹数据支持的四种标准格式类型。

Enumeration class for fingerprint data formats, defines four standard format types supported by fingerprint data.

枚举值 Enumeration Value	描述 Description
ANSI	ANSI标准指纹数据格式 ANSI standard fingerprint data format
ISO_FMR	ISO FMR格式指纹数据 ISO FMR format fingerprint data
ISO_FMC_NORMAL	ISO FMC标准格式指纹数据 ISO FMC normal format fingerprint data
ISO_FMC_COMPACT	ISO FMC紧凑格式指纹数据 ISO FMC compact format fingerprint data