目录

序言 5

1. TrustFinger™ SDK 简介 6

1.1. SDK 系统概述 6

1.2. 指纹数据格式和标准 6

1.3. 支持设备 7

2. 快速开发指引 7

2.1. SDK 目录结构 7

2.2. 在 Android Studio 中使用 TrustFinger Android SDK 8

3. 基本应用开发流程 11

3.1. 指纹图像采集 11

3.2. 指纹注册 11

3.3. 指纹比对 12

4. 接口描述 14

4.1. TrustFinger 类 14

4.1.1 getInstance(Context context) 14

4.1.2 initialize() 14

4.1.3 release() 15

4.1.4 getDeviceCount() 15

4.1.5 openDevice(int, DeviceOpenListener) 15

4.1.6 getSdkVersion() 16

4.1.7 getDeviceList() 16

4.1.8 setDeviceListener(DeviceListener) 16

4.2. TrustFingerDevice 类 17

4.2.1 getImageInfo () 17

4.2.2 getDeviceDescription() 17

4.2.3 captureRawData() 18

4.2.4 captureRawData(long) 18

4.2.5 captureRawDataLfd(int[]) 19

4.2.6 captureBmpData() 19

4.2.7 captureBmpDataLfd(int[]) 20

4.2.8 captureISOData(FingerPosition, ImgCompressAlg) 20

4.2.9 captureISODataLfd(FingerPosition, ImgCompressAlg,int[]) 21

4.2.10 captureANSIData(FingerPosition, ImgCompressAlg) 22

4.2.11 captureANSIDataLfd(FingerPosition, ImgCompressAlg,int[]) 23

4.2.12 captureWSQData() 24

4.2.13 captureWSQDataLfd(int[]) 24

4.2.14 getLedStatus(LedIndex) 24

4.2.15 setLedStatus(LedIndex, LedStatus) 25

4.2.16 extractFeature(byte[], FingerPosition) 26

4.2.17 extractANSIFeature(byte[], FingerPosition) 26

4.2.18 extractISOFeature(byte[], FingerPosition) 27

4.2.19 generalizeTemplate(byte[], byte[], byte[]) 27

4.2.20 verify(SecurityLevel, byte[], byte[]) 28

4.2.21 rawToBmp(byte[], int, int, int) 29

4.2.22 bmpToRaw(byte[]) 30

4.2.23 rawToWsq(byte[], int, int, int) 30

4.2.24 rawToANSI(byte[], int, int, int, int, int) 31
4.2.25 rawToISO(byte[], int, int, int, int, int) 32
4.2.26 rawDataQuality(byte[]) 33
4.2.27 bmpDataQuality(byte[]) 33
4.2.28 close() 34
4.2.29 setLFDLevel(int) 34
4.2.30 getLFDLevel() 34
4.3. VerifyResult 类 35
4.4. ScannerImageInfo 类 35
4.5. DeviceDesciption 类 35
4.6. DeviceOpenListener 接口 36
4.6.1 openSuccess(TrustFingerDevice) 36
4.6.2 openFail(String) 37
4.7. DeviceListener 接口 37
4.7.1 deviceAttached(List<String>) 37
4.7.2 deviceDetached(List<String>) 37
4.8. ImgComCompressAlg 枚举类型 38
4.9. LedIndex 枚举类型 38
4.10. LedStatus 枚举类型 38
4.11. SecurityLevel 枚举类型 39
4.12. LfdLevel 枚举类型 39
4.13. LfdStatus 枚举类型 39
4.14. FingerPosition 枚举类型 39
4.15. TrustFingerException.Type 枚举类型 40

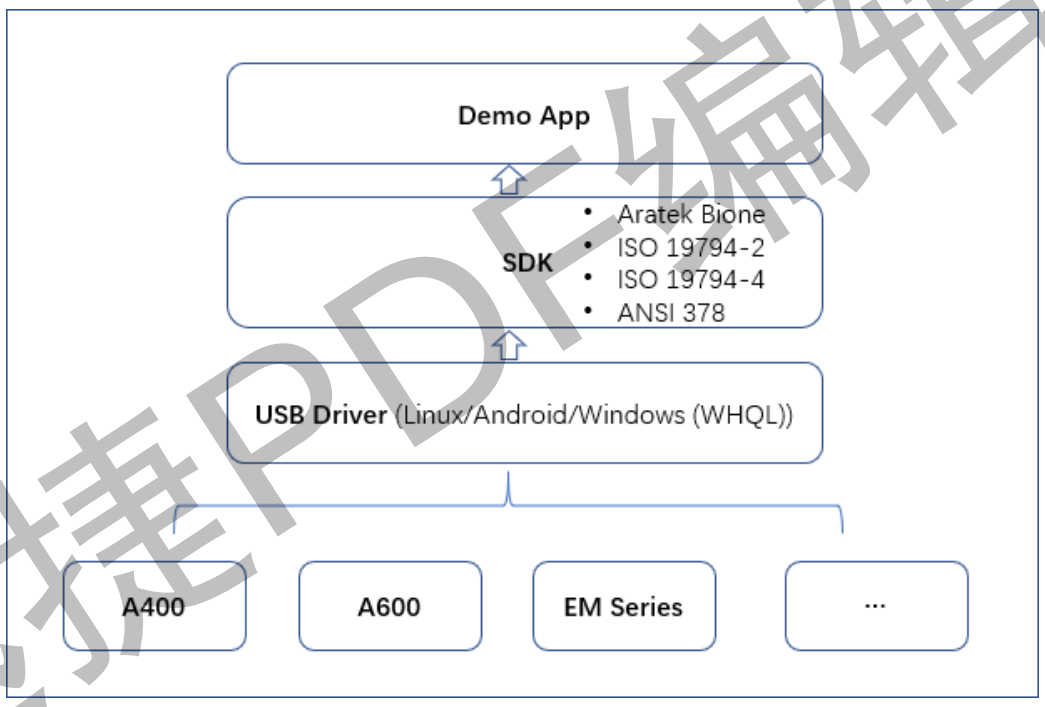
1. TrustFinger™ SDK 简介

Aratek TrustFinger™ SDK 是深圳市亚略特生物识别科技有限公司（以下简称亚略特）为第三方开发者提供的应用程序开发包。通过 Aratek TrustFinger™ SDK，应用程序开发者能够快速集成指纹采集、注册、识别功能。

TrustFinger™ SDK 主要具有以下特点:

- 支持 ISO/ANSI 标准的指纹图像和指纹特征
- 兼容亚略特不同类型、不同尺寸的指纹采集设备

1.1. SDK 系统概述



1.2. 指纹数据格式和标准

亚略特 TrustFinger™ SDK 支持以下 3 种指纹数据格式[1]:

- ARATEK Bione®
- ANSI

[1] 实际支持标准以硬件规格说明书为准

- ISO

当指纹数据存储为 ISO/ANSI 格式时,支持以下指纹格式标准。

	ANSI 标准	ISO 标准
指纹图像数据(FIR)	ANSI INCITS 381-2004	ISO/IEC 19794-4:2005
指纹特征数据(FMR)	ANSI INCITS 378-2004	ISO/IEC 19794-2:2005

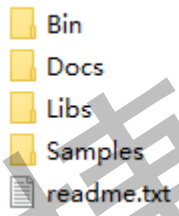
1.3. 支持设备

TrustFinger™ SDK 兼容半导体，光学等多款不同类型的指纹采集设备。目前版本支持 A600，A400 指纹仪。详情参考附录 A。

2. 快速开发指引

2.1. SDK 目录结构

下图为 SDK 总体的目录结构



- Bin

  demo 安装包文件。

- Docs

  开发文档。

- Libs

  SDK 库文件（*.jar 和*.so 文件）。

- Samples

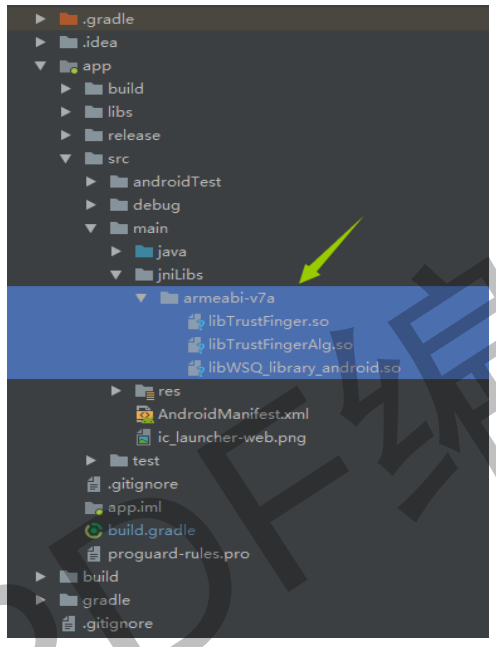
  demo 工程源码。

- Readme.txt

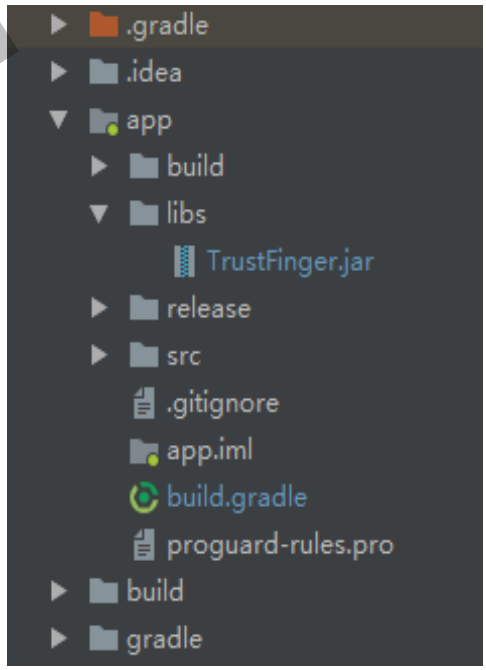
文档结构以及版本更新信息说明。

2.2. 在 Android Studio 中使用 TrustFinger Android SDK

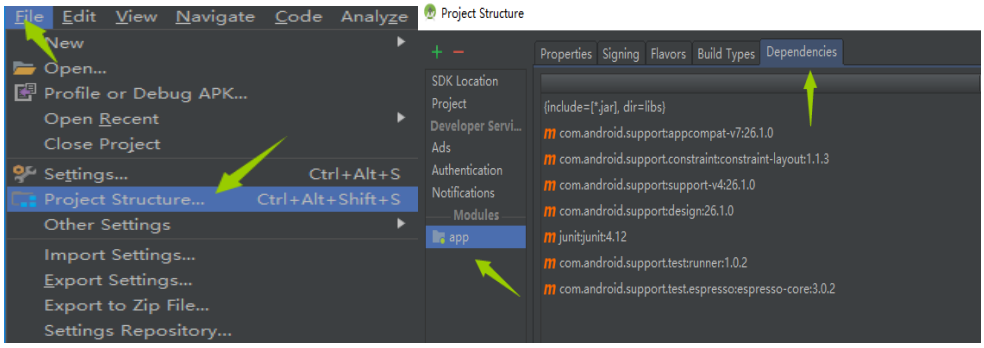
步骤一，创建文件夹/ app / src / main / jniLibs，然后将 libTrustFinger.so，libTrustFingerAlg.so 和 libWSQ_library_android.so 放在相应 abi 平台类型的文件夹中。



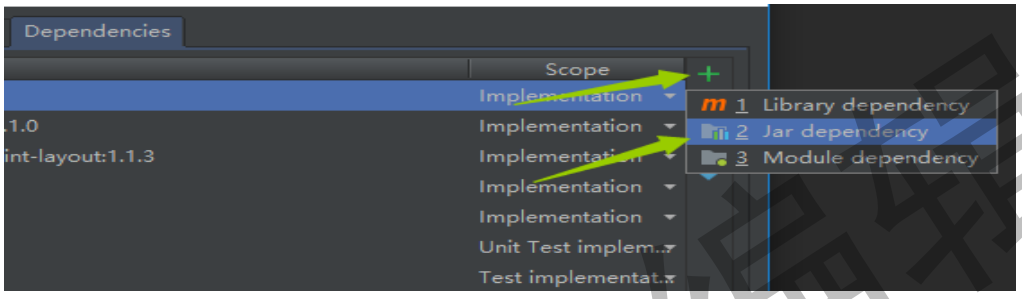
步骤二，将 TrustFinger.jar 文件复制到项目的 app 文件夹下的 libs 文件夹中。



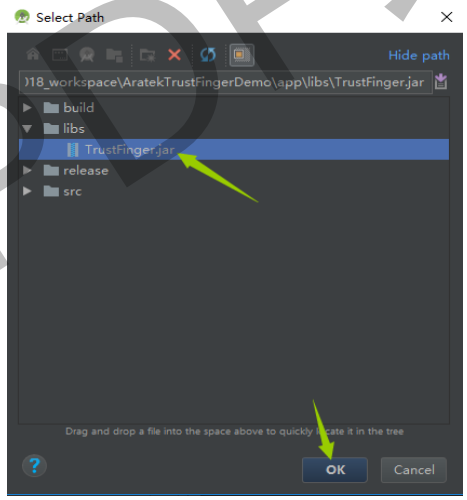
步骤三，单击 File > Project Structure >Select app > Dependencies 选项卡。



步骤四，单击右侧给出的（+）加号按钮，然后选择 Jar Dependency。



步骤五，这时将弹出一个用于选择路径的对话框。在这个打开的 libs 文件夹下，添加 TrustFinger.jar。



步骤六，选择 TrustFinger.jar 文件后，单击“确定”按钮，Gradle 将开始构建。

注意：

如果您无法找到库，那么您必须在“Android”视图中查看项目。 在 Android 选项卡的右侧，您将看到<>符号。 单击它并选择 Project。 此选项在 Android Studio 2.1 中可用。

如果您使用的是旧版本，可能找不到<>符号，则只需单击 Android 选项卡，它将显示该列表选择项目的下拉列表。

3. 基本应用开发流程

3.1. 指纹图像采集

- 1、调用 getInstance() 来获取设备实例，然后调用 initialize() 来初始化 SDK。在一个进程中，只需要调用 initialize() 一次。

```
TrustFinger mTrustfinger = TrustFinger.getInstance(getApplicationContext());
mTrustFinger.initialize();
```

- 2、调用 openDevice()打开设备。

```
int deviceIndex = 0;
TrustFingerDevice mTrustFingerDevice = null;
mTrustFinger.openDevice(deviceIndex, new DeviceOpenListener(){
    @Override
    public void openSuccess(TrustFingerDevice device){
     mTrustFingerDevice = device;
    }
    @Override
    public void openFail(String msg){
    }
});
```

- 3、采集指纹图像。

```
byte[] rawData = mTrustFingerDevice.captureRawData();
```

3.2. 指纹注册

- 1、调用 getInstance() 来获取设备实例，然后调用 initialize() 来初始化 SDK。在一个进程中，只需要调用 initialize() 一次。

```
TrustFinger mTrustfinger = TrustFinger.getInstance(getApplicationContext());
mTrustFinger.initialize();
```

- 2、调用 openDevice() 打开设备。

```
int deviceIndex = 0;
TrustFingerDevice mTrustFingerDevice = null;
mTrustFinger.openDevice(deviceIndex, new DeviceOpenListener(){
    @Override
    public void openSuccess(TrustFingerDevice device){
        mTrustFingerDevice = device;
    }
    @Override
    public void openFail(String msg){
    }
});
```

- 3、调用 captureRawData() 来采集图像原始数据。
- 4、图像质量判断和指纹特征提取。
- 5、重复步骤 3 和 4 以获得 3 个指纹特征，然后将它们合成一个模板。
- 6、调用 generalizeTemplate() 将步骤 5 中的 3 个指纹特征合成为一个模板。
- 7、保存模板，指纹注册完成。

3.3. 指纹比对

- 1、调用 getInstance() 来获取设备实例，然后调用 initialize() 来初始化 SDK。在一个进程中，只需要调用 initialize() 一次。
- 2、调用 openDevice() 打开设备。
- 3、调用 captureRawData() 来采集图像原始数据。
- 4、图像质量判断和指纹特征提取。

■ 5、调用 verify() 来比对两个指纹特征。

两个指纹特征进行比对时，SecurityLevel 有 5 个级别：从 1 到 5，等级越高安全性越好，推荐设置为 4;我们的算法在大量数据库的支持下进行了严格的测试，下表是我们 SDK 的在不同安全等级下 FAR、FRR 以及阈值信息。

等级、FAR 与阈值的关系表：

等级	FAR（认假率）	阈值分数
1	1%	24
2	0.5%	30
3	0.1%	36
4	0.01%	48
5	0.001%	60

4. 接口描述

4.1. TrustFinger 类

4.1.1 getInstance(Context context)

函数原型

Method	public static TrustFinger getInstance(Context context)

函数说明

获取 TrustFinger 类的单实例对象。

参数说明

参数名	描述
context	上下文。

返回值

TrustFinger 的单个实例。

4.1.2 initialize()

函数原型

Method	public void initialize() throws TrustFingerException

函数说明

初始化 SDK 运行环境，通常在应用程序进程开始时调用此 API，并且只需在释放设备之前调用一次。

4.1.3 release()

函数原型

Method	public void release()

函数说明

释放初始化时申请的资源。

4.1.4 getDeviceCount()

函数原型

Method	public int getDeviceCount() throws TrustFingerException

函数说明

检索已连接的 Aratek 指纹设备的数量。 只有获得许可的设备才会被计算在内。

返回值

Aratek 指纹设备的数量。

4.1.5 openDevice(int, DeviceOpenListener)

函数原型

Method	public void openDevice(final int deviceIndex, DeviceOpenListener deviceOpenListener) throws TrustFingerException

函数说明

给定特定的设备索引，异步初始化设备。此函数立即返回。 参数 DeviceOpenListener 的回调接口 openSuccess() 会返回一个 device 对象。 发生错误时，openFail() 接口回调并返回错误消息。

参数说明

参数名	描述
deviceIndex	设备索引值。（从 0 开始）
deviceOpenListener	打开设备后回调接口。（请参阅 DeviceOpenListener 接口）

4.1.6 getSdkVersion()

函数原型

Method	public native String getSdkVersion()

函数说明

获取 SDK 版本信息。

返回值

SDK 版本信息。

4.1.7 getDeviceList()

函数原型

Method	public List<String> getDeviceList()

函数说明

获取已连接的设备列表

返回值

设备列表信息

4.1.8 setDeviceListener(DeviceListener)

函数原型

Method	public void setDeviceListener(DeviceListener deviceListener) throws TrustFingerException

函数说明

对设备的插拔进行监听。此函数立即返回。 当有设备插入时，参数 DeviceListener 的回调接口 deviceAttached() 会返回新的设备列表。 当有设备拔出时，deviceDetached() 接口回调并返回新的设备列表。

参数说明

参数名	描述
deviceListener	插拔设备的回调接口。（请参阅 DeviceListener 接口）

4.2. TrustFingerDevice 类

4.2.1 getImageInfo ()

函数原型

Method	public ScannerImageInfo getImageInfo()

函数说明

获取当前指纹采集设备的图像像素的宽度，高度和分辨率信息。

返回值

设备的图像信息。（请参阅 ScannerImageInfo 类）

4.2.2 getDeviceDescription()

函数原型

Method	public DeviceDescription getDeviceDescription()

函数说明

获取指纹采集设备的详细设备信息。

返回值

设备的描述信息。（请参阅 DeviceDescription 类）

4.2.3 captureRawData()

函数原型

Method	public byte[] captureRawData() throws TrustFingerException

函数说明

采集一帧原始指纹图像数据。

返回值

原始指纹图像数据。

4.2.4 captureRawData(long)

函数原型

Method	public byte[] captureRawData(long timeout) throws TrustFingerException

函数说明

在超时时间内采集一帧原始指纹图像数据。

参数说明

参数名	描述
timeout	超时时间

返回值

原始指纹图像数据。

4.2.5 captureRawDataLfd(int[])

函数原型

Method	public byte[] captureRawDataLfd(int[] lfdStatus) throws TrustFingerException

函数说明

采集一帧原始指纹图像数据，采集过程中会进行活体检测。

参数说明

参数名	描述
lfdStatus	活体检测得到的手指状态（参阅 LfdStatus 枚举类）

返回值

原始指纹图像数据。

4.2.6 captureBmpData()

函数原型

Method	public byte[] captureBmpData() throws TrustFingerException

函数说明

采集一帧 BMP 格式指纹图像数据。

返回值

BMP 格式指纹图像数据。

4.2.7 captureBmpDataLfd(int[])

函数原型

Method	public byte[] captureBmpDataLfd(int[] lfdStatus) throws TrustFingerException

函数说明

采集一帧 BMP 格式指纹图像数据，采集过程中会进行活体检测。

参数说明

参数名	描述
lfdStatus	活体检测得到的手指状态（参阅 LfdStatus 枚举类）

返回值

BMP 格式指纹图像数据。

4.2.8 captureISOData(FingerPosition, ImgCompressAlg)

函数原型

Method	public byte[] captureISOData(FingerPosition fingerPosition, ImgCompressAlg imgCompressAlg) throws TrustFingerException

函数说明

采集一帧符合 ISO 标准格式的指纹图像数据。

参数说明

参数名	描述
fingerPosition	指位信息。（请参阅 FingerPosition 枚举类）

imgCompressAlg	图像压缩算法: （请参阅 ImgCompressAlg 枚举类） UnCompressed = 0 BitPacked = 1 WSQ = 2 JPEG = 3 JPEG2000 = 4 PNG = 5

返回值

ISO 标准格式的指纹图像数据。

4.2.9 captureISODataLfd(FingerPosition, ImgCompressAlg,int[])

函数原型

Method	public byte[] captureISOData(FingerPosition fingerPosition, ImgCompressAlg imgCompressAlg, int[] lfdStatus) throws TrustFingerException

函数说明

采集一帧符合 ISO 标准格式的指纹图像数据，采集过程中会进行活体检测。

参数说明

参数名	描述
fingerPosition	指位信息。（请参阅 FingerPosition 枚举类）
imgCompressAlg	图像压缩算法: （请参阅 ImgCompressAlg 枚举类） UnCompressed = 0 BitPacked = 1 WSQ = 2 JPEG = 3

	JPEG2000 = 4 PNG = 5
lfdStatus	活体检测得到的手指状态（参阅 LfdStatus 枚举类）

返回值

ISO 标准格式的指纹图像数据。

4.2.10 captureANSIData(FingerPosition, ImgCompressAlg)

函数原型

Method	public byte[] captureANSIData(FingerPosition fingerPosition, ImgCompressAlg imgCompressAlg) throws TrustFingerException

函数说明

采集一帧符合 ANSI 标准格式的指纹图像数据。

参数说明

参数名	描述
fingerPosition	指位信息。（请参阅 FingerPosition 枚举类）
imgCompressAlg	图像压缩算法: （请参阅 ImgCompressAlg 枚举类） UnCompressed = 0 BitPacked = 1 WSQ = 2 JPEG = 3 JPEG2000 = 4 PNG = 5

返回值

ANSI 标准格式的指纹图像数据。

4.2.11 captureANSIDataLfd(FingerPosition, ImgCompressAlg,int[])

函数原型

Method	public byte[] captureANSIData(FingerPosition fingerPosition, ImgCompressAlg imgCompressAlg, int[] lfdStatus) throws TrustFingerException

函数说明

采集一帧符合 ANSI 标准格式的指纹图像数据，采集过程中会进行活体检测。

参数说明

参数名	描述
fingerPosition	指位信息。（请参阅 FingerPosition 枚举类）
imgCompressAlg	图像压缩算法: （请参阅 ImgCompressAlg 枚举类） UnCompressed = 0 BitPacked = 1 WSQ = 2 JPEG = 3 JPEG2000 = 4 PNG = 5
lfdStatus	活体检测得到的手指状态（参阅 LfdStatus 枚举类）

返回值

ANSI 标准格式的指纹图像数据。

4.2.12 captureWSQData()

函数原型

Method	public byte[] captureWSQData() throws TrustFingerException

函数说明

采集一帧 WSQ 格式的指纹图像数据。

返回值

WSQ 格式的指纹图像数据。

4.2.13 captureWSQDataLfd(int[])

函数原型

Method	public byte[] captureWSQDataLfd(int[] lfdStatus) throws TrustFingerException

函数说明

采集一帧 WSQ 格式的指纹图像数据，采集过程中会进行活体检测。

参数说明

参数名	描述
lfdStatus	活体检测得到的手指状态（参阅 LfdStatus 枚举类）

返回值

WSQ 格式的指纹图像数据。

4.2.14 getLedStatus(LedIndex)

函数原型

Method	public LedStatus getLedStatus(LedIndex ledIndex) throws TrustFingerException

函数说明

获取指纹采集设备 LED 灯的状态。

参数说明

参数名	描述
ledIndex	LED 灯索引号。（参阅 LedIndex 枚举类）

返回值

指定 LED 灯的状态。（参阅 LedStatus 枚举类）

4.2.15 setLedStatus(LedIndex, LedStatus)

函数原型

Method	public int setLedStatus(LedIndex ledIndex, LedStatus ledStatus) throws TrustFingerException

函数说明

设置指纹采集设备 LED 灯的状态。

参数说明

参数名	描述
ledIndex	LED 灯索引号。（参阅 LedIndex 枚举类）
ledStatus	LED 灯状态。（参阅 LedStatus 枚举类）

返回值

返回值	描述

0	LED 灯状态设置成功。
Others	LED 灯状态设置失败。

4.2.16 extractFeature(byte[], FingerPosition)

函数原型

Method	public byte[] extractFeature(byte[] rawData, FingerPosition fingerPosition)

函数说明

从原始图像中提取特征数据。

参数说明

参数名	描述
rawData	原始图像数据。
fingerPosition	指位信息。（请参阅 FingerPosition 枚举类）

返回值

ARATEK Bione 格式指纹特征数据。

4.2.17 extractANSIFeature(byte[], FingerPosition)

函数原型

Method	public byte[] extractANSIFeature(byte[] rawData, FingerPosition fingerPosition)

函数说明

从原始图像中提取 ANSI 标准格式的特征数据。

参数说明

参数名	描述
rawData	原始图像数据。
fingerPosition	指位信息。（请参阅 FingerPosition 枚举类）

返回值

ANSI 标准格式的指纹特征数据。

4.2.18 extractISOFeature(byte[], FingerPosition)

函数原型

Method	public byte[] extractISOFeature(byte[] rawData, FingerPosition fingerPosition) throws TrustFingerException

函数说明

从原始图像中提取 ISO 标准格式的特征数据。

参数说明

参数名	描述
rawData	原始图像数据。
fingerPosition	指位信息。（请参阅 FingerPosition 枚举类）

返回值

ISO 标准格式的指纹特征数据。

4.2.19 generalizeTemplate(byte[], byte[], byte[])

函数原型

Method	public byte[] generalizeTemplate(byte[] featureData1, byte[] featureData2, byte[] featureData3) throws TrustFingerException

函数说明

把来自同一个手指的 3 个指纹特征值合成一个指纹模板。

参数说明

参数名	描述
featureData1	指纹特征数据 1。
featureData2	指纹特征数据 2。
featureData3	指纹特征数据 3。

返回值

指纹特征模板数据。

4.2.20 verify(SecurityLevel, byte[], byte[])

函数原型

Method	public VerifyResult verify(SecurityLevel securityLevel, byte[] feartureData1, byte[] feartureData2)

函数说明

两个指纹特征格式的比对。该函数有 3 个参数，securityLevel 参数为安全等级，范围从 1 到 5，等级越高，安全性越高，推荐等级设置为 4；另外两个参数 feartureData1 和 feartureData2 表示需要比对两个指纹特征，可以是 ARATEK Bione,ANSI，ISO 指纹特征中的任意组合。如果 VerifyResult.error 的值为 0，表示比对操作正常，否则，表示比对操作异常。

等级、FAR 与阈值的关系表：

等级	FAR（认假率）	FRR（拒真率）	阈值分数
1	5%	0.10%	18
2	1%	0.40%	24

3	0.1%	0.50%	36
4	0.01%	0.80%	48
5	0.001%	1.20%	60

参数说明

参数名	描述
securityLevel	安全等级。（请参阅 SecurityLevel 枚举类）
feartureData1	指纹特征数据 1。
feartureData2	指纹特征数据 2。

返回值

比对结果。（请参阅 VerifyResult 类）

4.2.21 rawToBmp(byte[], int, int, int)

函数原型

Method	public byte[] rawToBmp(byte[] rawData, int imageWidth, int imageHeight, int imageResoltion)

函数说明

将原始图像数据转换为 BMP 格式的图像数据。

参数说明

参数名	描述
rawData	指纹原始图像数据。
imageWidth	图像宽度。
imageHeight	图像高度。
imageResoltion	图像分辨率。

返回值

BMP 格式的图像数据。

4.2.22 bmpToRaw(byte[])

函数原型

Method	public byte[] bmpToRaw(byte[] bmpData)

函数说明

将 BMP 格式图像数据转为 RAW 原始图像数据。

参数说明

参数名	描述
bmpData	Bmp 格式指纹图像数据。

返回值

RAW 原始图像数据。

4.2.23 rawToWsq(byte[], int, int, int)

函数原型

Method	public byte[] rawToWsq(byte[] rawData, int imageWidth, int imageHeight, int imageResolution)

函数说明

将原始图像数据压缩为 WSQ 格式的图像数据。

参数说明

参数名	描述
rawData	指纹原始图像数据。

imageWidth	图像宽度。
imageHeight	图像高度。
imageResoltion	图像分辨率。

返回值

WSQ 格式的图像数据。

4.2.24 rawToANSI(byte[], int, int, int, int, int)

函数原型

Method	public byte[] rawToANSI(byte[] rawData, int imageWidth, int imageHeight, int imageResolution, FingerprintPosition fingerprintPosition, ImgCompressAlg imgCompressAlg)

函数说明

将原始图像数据转换为 ANSI 标准格式的图像数据。

参数说明

参数名	描述
rawData	指纹原始图像数据。
imageWidth	图像宽度。
imageHeight	图像高度。
imageResoltion	图像分辨率。
fingerprintPosition	指位信息。（请参阅 FingerPosition 枚举类）
imgCompressAlg	图像压缩算法: （请参阅 FingerPosition 枚举类） UnCompressed = 0 BitPacked = 1 WSQ = 2 JPEG = 3

	JPEG2000 = 4 PNG = 5

返回值

ANSI 标准格式的图像数据。

4.2.25 rawToISO(byte[], int, int, int, int, int)

函数原型

Method	public byte[] rawToISO(byte[] rawData, int imageWidth, int imageHeight, int imageResolution, int fingerprintPosition, int imgCompressAlg)

函数说明

将原始图像数据转换为符合 ISO 标准格式的图像数据。

参数说明

参数名	描述
rawData	指纹原始图像数据。
imageWidth	图像宽度。
imageHeight	图像高度。
imageResoltion	图像分辨率。
fingerprintPosition	指位信息。（请参阅 FingerPosition 枚举类）
imgCompressAlg	图像压缩算法: （请参阅 FingerPosition 枚举类） UnCompressed = 0 BitPacked = 1 WSQ = 2 JPEG = 3 JPEG2000 = 4

	PNG = 5

返回值

ISO 标准格式的图像数据。

4.2.26 rawDataQuality(byte[])

函数原型

Method	public int rawDataQuality(byte[] rawData) throws TrustFingerException

函数说明

获得原始图像质量，输出分数在 0 到 100 范围内，分数越高，图像质量越好。

参数说明

参数名	描述
rawData	原始图像数据。

返回值

指纹图像质量。

4.2.27 bmpDataQuality(byte[])

函数原型

Method	public int bmpDataQuality(byte[] bmpData) throws TrustFingerException

函数说明

获得 BMP 格式图像质量，输出分数在 0 到 100 范围内，分数越高，图像质量越好。

参数说明

参数名	描述
bmpData	BMP 格式图像数据。

返回值

指纹图像质量。

4.2.28 close()

函数原型

Method	public void close() throws TrustFingerException

函数说明

关闭指纹采集设备。

4.2.29 setLFDLevel(int)

函数原型

Method	public void setLFDLevel(int level)

函数说明

设置 LFD 检测等级（仅 A600 设备支持）

参数说明

参数名	描述
level	LFD 等级（参阅 LfdLevel 枚举类）

4.2.30 getLFDLevel()

函数原型

Method	public int getLFDLevel()

函数说明

获取目前的 LFD 检测等级（仅 A600 设备支持）

返回值

LFD 等级（参阅 LfdLevel 枚举类）

4.3. VerifyResult 类

如果 VerifyResult.error 的值为 0，表示比对操作正常，可以获取到相似度以及两个指纹特征是否匹配的结果。否则，表示比对操作异常，不能进一步获取比对结果。

属性	描述
int error	错误代码。0 表示操作成功，其它表示操作失败。
int similarity	相似度。
boolean isMatched	是否匹配。

4.4. ScannerImageInfo 类

属性	描述
int width	图像宽度。
int height	图像高度。
int resolution	图像分辨率。

4.5. DeviceDesciption 类

属性	描述
String infoVersion	设备信息版本号。
String manufacturer	制造厂商。

String productName	产品名称。
String productModel	产品型号。
String hwVersion	硬件版本号。
String bootVersion	BootLoader 版本号。
String fwVersion	固件版本号。
String serialNumber	序列号。
int imageWidth	传感器像素宽。
int imageHeight	传感器像素高。
String productionDate	生产日期。
int deviceId	设备 ID。
int resolution	传感器分辨率。
boolean isUSBSupported	是否支持 USB 接口。
boolean isUARTSupported	是否支持 UART 接口。
boolean isSPISupported	是否支持 SPI 接口。

4.6. DeviceOpenListener 接口

4.6.1 openSuccess(TrustFingerDevice)

函数原型

Method	void openSuccess(TrustFingerDevice trustFingerDevice)

函数说明

打开设备成功回调方法。

参数说明

参数名	描述
trustFingerDevice	设备对象。（请参阅 TrustFingerDevice 类）

4.6.2 openFail(String)

函数原型

Method	void openFail(String errorMessage)

函数说明

打开设备失败回调方法。

参数说明

参数名	描述
errorMessage	错误信息。

4.7. DeviceListener 接口

4.7.1 deviceAttached(List<String>)

函数原型

Method	void deviceAttached(List<String> mDevices)

函数说明

插入设备时的回调方法。

参数说明

参数名	描述
mDevice	设备列表

4.7.2 deviceDetached(List<String>)

函数原型

Method	void deviceDetachede(List<String> mDevices)

函数说明

拔出设备时的回调方法。

参数说明

参数名	描述
mDevice	设备列表

4.8. ImgComCompressAlg 枚举类型

枚举成员	描述
UNCOMPRESSED_NO_BIT_PACKING	UnCompressed
UNCOMPRESSED_BIT_PACKED	BitPacked
COMPRESSED_WSQ	WSQ
COMPRESSED_JPEG	JPEG
COMPRESSED_JPEG2000	JPEG2000
PNG	PNG

4.9. LedIndex 枚举类型

枚举成员	描述
RED	红灯
GREEN	绿灯

4.10. LedStatus 枚举类型

枚举成员	描述
OPEN	打开
CLOSE	关闭

4.11. SecurityLevel 枚举类型

枚举成员	描述
Level5	Level 5
Level4	Level 4
Level3	Level 3
Level2	Level 2
Level1	Level 1

4.12. LfdLevel 枚举类型

枚举成员	描述
OFF	数值为 0
EXTRA_LOW	数值为 1
LOW	数值为 2
MEDIUM	数值为 3
HIGH	数值为 4
ULTRA_HIGH	数值为 5

4.13. LfdStatus 枚举类型

枚举成员	描述
UNKOWN	未知手指
NORMAL	正常手指
FAKE	假体手指

4.14. FingerPosition 枚举类型

枚举成员	描述
RightThumb	右手大拇指

RightIndexFinger	右手食指
RightMiddleFinger	右手中指
RightRingFinger	右手环指
RightLittleFinger	右手小指
LeftThumb	左手大拇指
LeftIndexFinger	左手食指
LeftMiddleFinger	左手中指
LeftRingFinger	左手环指
LeftLittleFinger	左手小指
Unknown	未知指位

4.15. TrustFingerException.Type 枚举类型

枚举成员	描述
SUCCESS	函数操作成功。
FAIL	函数操作失败。
DEVICE_NOT_FOUND	设备未找到。
DEVICE_NOT_AUTHORILIZED	设备未认证。
DEVICE_NOT_INITIALIZED	设备未初始化。
DEVICE_NOT_CONNECTED	设备未连接。
DEVICE_NOT_OPENED	无法打开设备。
DEVICE_GET_INTERFACE_FAIL	获取设备接口失败。
DEVICE_GET_ENDPOINT_FAIL	获取 USB 结点失败。
DEVICE_GET_CONNECTION_FAIL	连接设备失败。
DEVICE_NO_USB_HOST_FEATURE	不支持 USB HOST 模式。
DEVICE_NOT_ACCESSIBLE	无法访问该设备。
DEVICE_ALREADY_OPENED	设备已打开。

DEVICE_ALREADY_CLOSED	设备已关闭。
API_NOT_SUPPORTED	该接口暂不支持。
FAKE_FINGER	假体手指
CAPTURE_FAIL	采集图像失败。
CAPTURE_ERROR	采集图像时发生错误。
TRANSFER_PACKET_ERROR	USB 数据传输打包失败。
TRANSFER_READ_ERROR	USB 数据传输信息读取失败。
TRANSFER_WRITE_ERROR	USB 数据传输信息写入失败。
TRANSFER_CONTROL_ERROR	USB 数据传输控制传输失败。
INVALID_PARAM_VALUE	无效的参数。
NOT_ENOUGH_MEMORY	内存分配不足。
INVALID_DEVICE_INDEX	无效的设备索引号。
FP_FEATURE_CONVERT_ERROR	格式转换错误。
FP_BAD_IMAGE	图像质量过低。
FP_INVALID_DATA	无效的数据。
INIT_ALGORITHM_ERROR	算法初始化失败。
UNKNOW_TYPE	未知异常。
GENERALIZE_TEMPLATE_FAIL_NOT_SAME_FINGER	用于合成模板的三个指纹特征不是来自同一根手指
UNKNOW_ERROR	未知错误。