

蓝牙UHF SDK API 说明V1.1

Bluetooth UHF SDK API DocumentationV1.1

一、SDK概览

I.SDK Overview

二、详细API说明

II. Detail API Documentation

1 初始化连接

1 Initialize Connection

1.1 getInstance()

1.2 init(Context)

1.3 connect(String)

1.4 connect(String,ConnectionStatusCallback)

1.5 setConnectionStatusCallback(ConnectionStatusCallback btStatusCallback)

1.6 getConnectStatus()

1.7 startScanBTDevices()

1.8 stopScanBTDevices()

2 断开连接

2 Disconnect

2.1 disconnect

2.2 free

3 获取UHF信息

3 Get UHF Info

3.1 getVersion()

3.2 getEx10SDKFirmware()

3.3 getTemperature()

3.4 getBattery()

3.5 getSTM32Version()

4 功率控制

4 Power Control

4.1 getPower()

4.2 setPower(int power)

5 频率设置

5 Frequency Settings

5.1 getFrequencyMode()

5.2 setFrequencyMode(int freMode)

6 标签盘点

6 Tag Inventory

6.1 startInventoryTag()

6.2 startInventoryTag(InventoryParameter)

6.3 stopInventory()

6.4 inventorySingleTag()

6.5 setInventoryCallback(IUHFIInventoryCallback)

6.6 setEPCMode()

6.7 setEPCAndTIDMode()

6.8 setEPCAndTIDUserMode(int,int)

6.9 setEPCAndTIDUserMode(InventoryModeEntity)

6.10 getEPCAndTIDUserMode()

6.11 setFilter(int,int,int,String)

6.12 setFilter(List<FilterEntity>)

6.13 isInventorying()

6.14 setFastInventoryMode(FastInventoryEntity)

6.15 getFastInventoryMode()

6.16 getAllTagTotalFromFlash()

6.17 getNewTagTotalFromFlash()

6.18 deleteAllTagToFlash()

6.19 getTagDataFromFlash()

7 数据读写

7 Data Read/Write

7.1 readData(String,int,int,int)

7.2 readData(String,int,int,int,String,int,int,int)

7.3 writeData(String,int,int,int,String)

7.4 writeData(String,int,int,int,String,int,int,int,String)

7.5 writeDataToEpc(String,String)

7.6 writeDataToEpc(String,int,int,int,String,String)

7.7 blockWriteData(String,int,int,int,String,int,int,int,String)

7.8 eraseData(String,int,int,int)

7.9 eraseData(String,int,int,int,String,int,int,int)

7.10 uhfBlockPermalock(String,int,int,int,String,int, int,int,int,byte[])

8 标签销毁和锁定

8 Tag Kill、 Lock

8.1 killTag(String)

8.2 killTag(String,int,int,int,String)

8.3 lockMem(String,String)

8.4 lockMem(String,int,int,int,String,String)

8.5 generateLockCode(ArrayList,int)

9 RF链路、 FastID、 TagFocus、 GEN2

9 RFLink、 FastID、 TagFocus、 GEN2

9.1 setRFLink(int)

9.2 getRFLink()

9.3 setFastID(boolean)

9.4 getFastId()

9.5 setTagFocus(boolean)

9.6 getTagFocus()

9.7 setGen2(Gen2Entity)

9.8 getGen2()

10 定位功能

10 Location Functions

10.1 startLocation(Context,String,int,int,IUHFLocationCallback)

10.2 stopLocation()

10.3 startRadarLocation(Context,String,int,int,IUHF RadarLocationCallback)

10.4 stopRadarLocation()

10.5 setDynamicDistance()

11 条码

11 Barcode

11.1 scanBarcodeToBytes()

11.2 scanBarcode()

12 升级功能

12 Upgrade Functions

12.1 uhfJump2Boot(int)

12.2 uhfStartUpdate()

12.3 uhfUpdating(byte[] buff)

12.4 uhfStopUpdate()

13 其他功能

13 Other Functions

13.1 setBaudRate(int baudRate)

13.2 getBaudRate()

13.3 setUart(String uart)

13.4 setKeyEventCallback(KeyEventCallback)

13.5 factoryReset()

13.6 setReaderAwaitSleepTime()

13.7 getReaderAwaitSleepTime()

14 外部相关类

14 External related classes

14.1 InventoryParameter

14.2 InventoryParameter.ResultData

14.3 InventoryModeEntity

14.4 InventoryModeEntity.Builder

14.5 Gen2Entity

14.6 ErrorCode

14.7 FastInventoryEntity

14.8 IUHFLocationCallback

14.9 IUHFRadarLocationCallback

14.10 RadarLocationEntity

14.11 RadarLocationEntity.Builder

14.12 ConnectionStatusCallback

14.13 ScanBTCallback

14.14 KeyEventCallback

14.15 FilterEntity

一、SDK 概览

I. SDK Overview

`com.rscja.deviceapi.RFIDwithUHFBLE` 是基于 Android 平台的 UHF 模块蓝牙通信操作类，其核心作用是上层应用提供标准化的 API，实现 UHF 模块与设备的 BLE 连接、RFID 标签读写、参数配置、定位等功能。

`com.rscja.deviceapi.RFIDwithUHFBLE` is a UHF (Ultra-High Frequency) module serial communication operation class based on the Android platform. Its core function is to provide full-function control of the UHF module for handheld devices, such as tag inventory, data reading/writing, parameter configuration, and more.

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(Context context)` 初始化相关资源。

Step 1: Initialize related resources via `init(Context context)`.

第二步: 通过 `connect(String address, ConnectionStatusCallback btStatusCallback)` 连接蓝牙。

Step 2: Connect to Bluetooth via `connect(String address, ConnectionStatusCallback btStatusCallback)`.

第三步: 连接成功后，根据需求执行以下操作：

Step 3: After a successful connection, perform the following operations as needed:

- 若需设置参数：调用对应的函数进行参数配置及读写操作；
- If parameter setting is required: Call the corresponding function for parameter configuration and read/write operations.
- 若需盘点标签：
- If tag inventory is required:

先调用 `setInventoryCallback (IUHFInventoryCallback inventoryCallback)` 设置标签回调接口，当有标签数据时会触发该接口回调，再调用 `startInventoryTag()` 函数启动盘点。

First, call `setInventoryCallback(IUHFInventoryCallback inventoryCallback)` to set the tag callback interface, which will be triggered when there is tag data.

Then call the `startInventoryTag()` function to start the inventory.

注意：在标签盘点过程中，RFID 模块仅会响应 `stopInventory()` 函数。

Note: During the tag inventory process, the RFID module will only respond to the `stopInventory()` function.

第四步：调用 `disconnect()` 断开蓝牙连接，如果断开之前正在盘点，请先停止盘点，再断开连接。

Step 4: Call `disconnect()` to terminate the Bluetooth connection. If an inventory operation is in progress before disconnecting, stop the inventory first and then disconnect.

第五步：退出app调用 `free()` 释放相关资源。

Step 5: When exiting the app, call `free()` to release related resources.

3.使用示例

3.Usage Examples

```
public void Test(){
    RFIDWithUHFBLE rfid=RFIDWithUHFBLE.getInstance();
    boolean result=rfid.init(Context context);
    if(!result){
        return;
    }
    rfid.setConnectionStatusCallback(new ConnectionStatusCallback() {
        @Override
        public void getStatus(ConnectionStatus status, Object device) {
            if(status==ConnectionStatus.CONNECTED){
                // Connection successful
            }else if(status==ConnectionStatus.DISCONNECTED){
                // Connection failed
            }
        }
    });
    String address="xx:xx:xx:xx:xx:xx";
    rfid.connect(address);
    // Waiting for connection...
    // Connection successful
    // Configure parameters
    //rfid.setFrequencyMode(xx)
    //rfid.setPower(30);
    rfid.setInventoryCallback(new IUHFInventoryCallback() {
        @Override
        public void callback(UHFTAGInfo uhftagInfo) {
            String epc= uhftagInfo.getEPC();
            String rssi= uhftagInfo.getRssi();
            //.....
        }
    });
    if(rfid.startInventoryTag()){
        // Operation successful
    }else{
        // Operation failed
    }

    rfid.stopInventory();
    //.....
    // Disconnect Bluetooth
    rfid.disconnect();
    //.....
    // Exit application
    rfid.free();

}
```

二、详细API说明

II.Detailed API description

1 初始化连接

1 Initialize Connection

1.1 getInstance()

项目 Item	说明 Description
原型 Prototype	static RFIDWithUHFBLE getInstance() throws ConfigurationException
描述 Description	通过单例模式创建RFIDWithUHFBLE对象 Create an RFIDWithUHFBLE object via the singleton pattern
参数 Parameters	无 N/A
返回 Return	RFIDWithUHFBLE 实例 RFIDWithUHFBLE instance
备注 Remark	配置错误异常 Configuration err

1.2 init(Context context)

项目 Item	说明 Description
原型 Prototype	boolean init(Context context)
描述 Description	初始化蓝牙相关资源。该函数仅需调用一次，且必须在连接蓝牙前调用。 Initialize Bluetooth-related resources. This function only needs to be called once and must be invoked before connecting to Bluetooth.
参数 Parameters	context: Android上下文 context: Android Context
返回 Return	true:成功 false:失败 true:succcess false:failure

1.3 connect(String address)

项目 Item	说明 Description
原型 Prototype	void connect(String address)
描述 Description	以异步模式连接蓝牙读写器设备，需通过 <code>setConnectionStatusCallback(ConnectionStatusCallback)</code> 函数设置连接监听，连接结果将通过 <code>ConnectionStatusCallback</code> 回调返回。 Connect to the Bluetooth reader device in asynchronous mode. You need to set up the connection listener via the <code>setConnectionStatusCallback(ConnectionStatusCallback)</code> function, and the connection result will be returned through the <code>ConnectionStatusCallback</code> callback.

项目 Item	说明 Description
参数 Parameters	address: 蓝牙地址 address: Bluetooth address
返回 Return	无 N/A
备注 Remark	必须在调用该函数前，通过 <code>init(Context)</code> 函数完成蓝牙资源的初始化 You must complete the initialization of Bluetooth resources through the <code>init(Context)</code> function before calling this function. 需要如下权限: The following permissions are required: -android.permission.BLUETOOTH -android.permission.ACCESS_FINE_LOCATION -android.permission.ACCESS_COARSE_LOCATION -android.permission.BLUETOOTH_CONNECT

1.4 connect(String address, ConnectionStatusCallback btStatusCallback)

项目 Item	说明 Description
原型 Prototype	void connect(String address, cbtStatusCallback)
描述 Description	以异步模式连接蓝牙读写器设备，连接结果通过 <code>ConnectionStatusCallback</code> 回调返回。 Connect to the Bluetooth reader device in asynchronous mode, and the connection result is returned via the <code>ConnectionStatusCallback</code> callback.
参数 Parameters	address: 蓝牙地址 address: Bluetooth address btStatusCallback: 蓝牙连接状态回调接口，用于接收连接过程中的状态信息 btStatusCallback: Bluetooth connection status callback interface, used to receive status information during connection
返回 Return	无 N/A
备注 Remark	必须在调用该函数前，通过 <code>init(Context)</code> 函数完成蓝牙资源的初始化 You must complete the initialization of Bluetooth resources through the <code>init(Context)</code> function before calling this function. 需要如下权限: The following permissions are required: -android.permission.BLUETOOTH -android.permission.ACCESS_FINE_LOCATION -android.permission.ACCESS_COARSE_LOCATION -android.permission.BLUETOOTH_CONNECT

1.5 setConnectionStatusCallback(ConnectionStatusCallback btStatusCallback)

项目 Item	说明 Description
原型 Prototype	void setConnectionStatusCallback(ConnectionStatusCallback btStatusCallback)
描述 Description	监听连接状态回调 Listen for connection status callback
参数 Parameters	btStatusCallback: 连接状态回调 Connection status callback
返回 Return	无 N/A

1.6 getConnectStatus()

项目 Item	说明 Description
原型 Prototype	ConnectionStatus getConnectStatus()
描述 Description	获取连接状态 Get connection status
参数 Parameters	无 N/A
返回 Return	ConnectionStatus.DISCONNECTED, ConnectionStatus.CONNECTED, ConnectionStatus.CONNECTING

1.7 void startScanBTDevices(ScanBTCallback scanBTCallback)

项目 Item	说明 Description
原型 Prototype	void startScanBTDevices(ScanBTCallback scanBTCallback)
描述 Description	扫描蓝牙设备，扫描结果通过回调接口返回。 Scanning Bluetooth devices, and the scanning results are returned via the callback interface.
参数 Parameters	scanBTCallback: 扫描结果回调接口，用于接收蓝牙设备的扫描数据（如设备信息、信号值等） scanBTCallback: Scanning result callback interface, used to receive scanning data of Bluetooth devices (such as device information, signal value, etc.)
返回 Return	void
备注 Remark	需要如下权限: The following permissions are required: -android.permission.BLUETOOTH_ADMIN -android.permission.ACCESS_FINE_LOCATION -android.permission.ACCESS_COARSE_LOCATION -android.permission.BLUETOOTH_SCAN

1.8 void stopScanBTDevices()

项目 Item	说明 Description
原型 Prototype	void stopScanBTDevices()
描述 Description	停止蓝牙设备的扫描操作。 Stop the scanning operation of Bluetooth devices.
参数 Parameters	无 N/A
返回 Return	void

2 断开连接

2 Disconnect

2.1 disconnect()

项目 Item	说明 Description
原型 Prototype	void disconnect()
描述 Description	断开蓝牙连接 Disconnect the Bluetooth connection
参数 Parameters	无 N/A
返回 Return	void

2.2 free

项目 Item	说明 Description
原型 Prototype	boolean free()
描述 Description	释放蓝牙相关资源，退出app的时候调用 Release Bluetooth-related resources; this should be called when exiting the app.
参数 Parameters	无 N/A
返回 Return	true:成功 false:失败 true:success false:failure

3 获取UHF信息

3 Get UHF Info

3.1 getVersion()

项目 Item	说明 Description
原型 Prototype	String getVersion()
描述 Description	获取UHF模块版本 Get UHF module version
参数 Parameters	无 N/A
返回 Return	UHF模块版本 UHF module version

3.2 getEx10SDKFirmware()

项目 Item	说明 Description
原型 Prototype	String getEx10SDKFirmware()
描述 Description	获取Ex10SDK版本 Get Ex10SDK version
参数 Parameters	无 N/A
返回 Return	Ex10SDK版本 Ex10SDK version

3.3 getTemperature()

项目 Item	说明 Description
原型 Prototype	synchronized int getTemperature()
描述 Description	获取UHF模块温度 Get UHF module temperature
参数 Parameters	无 N/A
返回 Return	温度,单位: 摄氏度 Temperature,Unit:°C

3.4 getBattery()

项目 Item	说明 Description
原型 Prototype	int getBattery()
描述 Description	获取电池电量 Get battery capacity
参数 Parameters	无 None
返回 Return	电量， -1表示获取电量失败 battery capacity, -1 means acquire failure

3.5 getSTM32Version()

项目 Item	说明 Description
原型 Prototype	String getSTM32Version()
描述 Description	获取主板版本 Get mainboard version
参数 Parameters	无 None
返回 Return	主板信息字符串 String of mainboard version information

4 功率控制

4 Power Control

4.1 getPower()

项目 Item	说明 Description
原型 Prototype	int getPower()
描述 Description	获取天线功率 Get the antenna power
参数 Parameters	无 N/A
返回 Return	功率值 Power value

4.2 setPower(int power)

项目 Item	说明 Description
原型 Prototype	boolean setPower(int power)
描述 Description	设置天线功率 Set the antenna power
参数 Parameters	power: 功率值(1-30) power: Power value(1-30)
返回 Return	true:成功 false:失败 true:success false:failure

5 频率设置

5 Frequency Settings

5.1 getFrequencyMode()

项目 Item	说明 Description
原型 Prototype	int getFrequencyMode()
描述 Description	获取UHF模块频率模式 Get UHF module frequency mode
参数 Parameters	无 N/A
返回 Return	频率模式 Frequency mode 0x01： China Standard(840~845MHz) 0x02： China Standard2(920~925MHz) 0x04： Europe Standard(865~868MHz) 0x08： USA(902-928MHz) 0x16： Korea(917~923MHz) 0x32: Japan(916.8~920.8MHz) 0x33: South Africa(915~919MHz) 0x34: China Taiwan(920~928Mhz) 0x35:Vietnam(918~923MHz) 0x36:Peru(915-928MHz) 0x37:Russia(860-867.6MHz) 0x3B:Malaysia(919-923MHz) 0x3C:Brazil(902 ~ 907.5MHz, 915 ~ 928MHz) 0x3D:ETSI_UPPER(915 ~ 921MHz) 0x3E:Australia(920 ~ 926MHz) 0x3F:Indonesia(920-923MHz) 0x40:Israel(915 ~ 917MHz) 0x41:HK(920 ~ 925MHz) 0x42:New Zealand(920 ~ 928MHz)

5.2 setFrequencyMode(int freMode)

项目 Item	说明 Description
原型 Prototype	boolean setFrequencyMode(int freMode)
描述 Description	设置UHF模块频率模式 Set UHF module frequency mode
参数 Parameters	freMode: 频率模式 freMode: Frequency mode 0x01: China Standard(840~845MHz) 0x02: China Standard2(920~925MHz) 0x04: Europe Standard(865~868MHz) 0x08: USA(902-928MHz) 0x16: Korea(917~923MHz) 0x32: Japan(916.8~920.8MHz) 0x33: South Africa(915~919MHz) 0x34: China Taiwan(920~928Mhz) 0x35:Vietnam(918~923MHz) 0x36:Peru(915-928MHz) 0x37:Russia(860-867.6MHz) 0x3B:Malaysia(919-923MHz) 0x3C:Brazil(902 ~ 907.5MHz, 915 ~ 928MHz) 0x3D:ETSI_UPPER(915 ~ 921MHz) 0x3E:Australia(920 ~ 926MHz) 0x3F:Indonesia(920-923MHz) 0x40:Israel(915 ~ 917MHz) 0x41:HK(920 ~ 925MHz) 0x42:New Zealand(920 ~ 928MHz)
返回 Return	true:成功 false:失败 true:success false:failure

6 标签盘点

6 Tag Inventory

6.1 startInventoryTag()

项目 Item	说明 Description
原型 Prototype	boolean startInventoryTag()
描述 Description	开始循环识别标签,在盘点标之前先设置setInventoryCallback(IUHFInventoryCallback)回调接口接收标签数据。开始盘点后UHF模块只响应stopInventory()函数。 Start cyclically identifying tags. Before inventory counting, set the setInventoryCallback(IUHFInventoryCallback) callback interface first to receive tag data. After starting the inventory, the UHF module will only respond to the stopInventory() function.
参数 Parameters	无 N/A
返回 Return	true:成功 false:失败 true:success false:failure

6.2 startInventoryTag(InventoryParameter)

项目 Item	说明 Description
原型 Prototype	boolean startInventoryTag(InventoryParameter inventoryParameter)
描述 Description	开始循环识别标签,在盘点标之前先设置setInventoryCallback(IUHFIInventoryCallback)回调接口接收标签数据。开始盘点后UHF模块只响应stopInventory()函数。 Start cyclically identifying tags. Before inventory counting, set the setInventoryCallback(IUHFIInventoryCallback) callback interface first to receive tag data. After starting the inventory, the UHF module will only respond to the stopInventory() function.
参数 Parameters	inventoryParameter: 盘点参数 inventoryParameter: Inventory parameters
返回 Return	true:成功 false:失败 true:success false:failure
备注 Remark	可以通过InventoryParameter参数来配置盘点到的标签信息是否需要包含相位或者频点。 You can configure whether the inventoried tag information needs to include phase or frequency points through the InventoryParameter parameter. 比如需要盘点的标签信息包含相位数据： For example, the tag information required for inventory includes phase data: <code>rfidWithUHFUART.startInventoryTag(new InventoryParameter(new InventoryParameter.ResultData().setNeedPhase(true)))</code> ;

6.3 stopInventory()

项目 Item	说明 Description
原型 Prototype	boolean stopInventory()
描述 Description	停止盘点 Stop inventory
参数 Parameters	无 N/A
返回 Return	true:成功 false:失败 true:success false:failure

6.4 inventorySingleTag()

项目 Item	说明 Description
原型 Prototype	UHFTAGInfo inventorySingleTag()
描述 Description	单标签识别 Single tag identification
参数 Parameters	无 N/A

项目 Item	说明 Description
返回 Return	标签信息 Tag information

6.5 setInventoryCallback(IUHFIInventoryCallback)

项目 Item	说明 Description
原型 Prototype	void setInventoryCallback(IUHFIInventoryCallback inventoryCallback)
描述 Description	设置盘点回调接口,在开始盘点之调用 Set the inventory callback interface,Set the inventory callback interface and call it before starting the inventory.
参数 Parameters	inventoryCallback: 盘点回调接口 inventoryCallback: Inventory callback interface
返回 Return	void

6.6 setEPCMode()

项目 Item	说明 Description
原型 Prototype	boolean setEPCMode()
描述 Description	设置为EPC模式，盘点时仅输出标签的EPC数据 Set to EPC mode, only output EPC data of tags when inventorying
参数 Parameters	无 N/A
返回 Return	true:成功 false:失败 true:success false:failure

6.7 setEPCAndTIDMode()

项目 Item	说明 Description
原型 Prototype	boolean setEPCAndTIDMode()
描述 Description	设置为EPC+TID模式，盘点时输出标签的EPC和TID数据 Set to EPC+TID mode, output EPC and TID data of tags when inventorying
参数 Parameters	无 N/A
返回 Return	true:成功 false:失败 true:success false:failure

6.8 setEPCAndTIDUserMode(int,int)

项目 Item	说明 Description
原型 Prototype	boolean setEPCAndTIDUserMode(int user_prt, int user_len)
描述 Description	设置为EPC+TID+USER模式，盘点时输出EPC、TID、USER数据 Set to EPC+TID+USER mode, output EPC, TID, USER data of tags when inventorying
参数 Parameters	user_prt: 用户数据起始地址 user_prt: User data start address user_len: 用户数据长度 user_len: User data length
返回 Return	true:成功 false:失败 true:success false:failure

6.9 setEPCAndTIDUserMode(InventoryModeEntity)

项目 Item	说明 Description
原型 Prototype	boolean setEPCAndTIDUserMode(InventoryModeEntity entity)
描述 Description	设置盘点模式实体（如输出字段等） Set the inventory mode via entity (e.g., output fields)
参数 Parameters	entity: 盘点模式 entity: Inventory mode
返回 Return	true:成功 false:失败 true:success false:failure
备注 Remark	比如：设置盘点EPC+Reserved 数据模式 For example: Set up the EPC+Reserved data mode for inventory InventoryModeEntity inventoryModeEntity=new InventoryModeEntity.Builder() .setMode(InventoryModeEntity.MODE_EPC_RESERVED) .setReservedLength(0) .setReservedOffset(4).build(); rfidWithUHFUART.setEPCAndTIDUserMode(inventoryModeEntity);

6.10 getEPCAndTIDUserMode()

项目 Item	说明 Description
原型 Prototype	InventoryModeEntity getEPCAndTIDUserMode()
描述 Description	获取当前盘点模式实体（如当前输出字段配置） Get the current inventory mode entity (e.g., current output fields configuration)
参数 Parameters	无 N/A

项目 Item	说明 Description
返回 Return	盘点模式实体对象 Inventory mode entity object

6.11 setFilter(int,int,int,String)

项目 Item	说明 Description
原型 Prototype	boolean setFilter(int bank, int offset, int length, String data)
描述 Description	设置盘点过滤条件 Set the inventory count filtering conditions
参数 Parameters	bank: 过滤内存 (IUHF.BANK_EPC/IUHF.BANK_TID/IUHF.BANK_USER) bank: Filter memory (IUHF.BANK_EPC/IUHF.BANK_TID/IUHF.BANK_USER) offset: 偏移量 offset: Offset length: 长度 length: Length data: 数据 data: Data
返回 Return	true:成功 false:失败 true:success false:failure

6.12 setFilter(List<FilterEntity>)

项目 Item	说明 Description
原型 Prototype	boolean setFilter(List<FilterEntity> list)
描述 Description	设置盘点过滤条件，最多支持过滤8张标签 Set the inventory count filtering conditions, supports filtering up to 8 tags
参数 Parameters	list: 过滤器列表 (Filter list) ， 列表内元素为FilterEntity类型，每个FilterEntity对应一个标签的过滤规则 list: Filter list, elements in the list are of FilterEntity type, each FilterEntity corresponds to the filter rule of one tag
返回 Return	true:成功 (success) false:失败 (failure)

6.13 isInventorying()

项目 Item	说明 Description
原型 Prototype	boolean isInventorying()
描述 Description	判断设备是否正在盘点 Judge the device is inventorying or not
参数 Parameters	无 N/A

项目 Item	说明 Description
返回 Return	true:正在盘点 false:未盘点 true:inventorying false:not inventorying

6.14 setFastInventoryMode(FastInventoryEntity entity)

项目 Item	说明 Description
原型 Prototype	boolean setFastInventoryMode(FastInventoryEntity entity)
描述 Description	设置快速盘点模式 Set fast inventory mode
参数 Parameters	entity: 快速盘点实体对象 entity: Fast inventory entity object
返回 Return	true:成功 false:失败 true:success false:failure

6.15 getFastInventoryMode()

项目 Item	说明 Description
原型 Prototype	FastInventoryEntity getFastInventoryMode()
描述 Description	获取快速盘点模式 Get fast inventory mode
参数 Parameters	无 N/A
返回 Return	快速盘点实体对象 FastInventoryEntity

6.16 getAllTagTotalFromFlash()

项目 Item	说明 Description
原型 Prototype	int getAllTagTotalFromFlash()
描述 Description	获取Flash中存储的所有标签总数 Get the total number of all tags stored in Flash
参数 Parameters	无 N/A
返回 Return	Flash中所有标签的总数（int类型） Total number of all tags in Flash (int type)

项目 Item	说明 Description
备注 Remark	只支持 PDA_A Only supports Device_A

6.17 getNewTagTotalFromFlash()

项目 Item	说明 Description
原型 Prototype	int getNewTagTotalFromFlash()
描述 Description	获取 Flash 中存储的标签数，该数量对应最后一次盘点操作新增的标签 Get the number of tags stored in Flash, which corresponds to the new tags added in the last inventory operation.
参数 Parameters	无 N/A
返回 Return	Flash中新增标签的总数（int类型） Total number of new tags in Flash (int type)
备注 Remark	只支持 PDA_A Only supports Device_A

6.18 deleteAllTagToFlash()

项目 Item	说明 Description
原型 Prototype	boolean deleteAllTagToFlash()
描述 Description	删除Flash中存储的所有标签 Delete all tags stored in Flash
参数 Parameters	无 N/A
返回 Return	true:成功 false:失败 true:success false:failure
备注 Remark	只支持 PDA_A Only supports Device_A

6.19 List<UHFTAGInfo> getTagDataFromFlash()

项目 Item	说明 Description
原型 Prototype	List<UHFTAGInfo> getTagDataFromFlash()

项目 Item	说明 Description
描述 Description	从 Flash 中获取标签数据列表。该方法单次调用仅返回部分数据，需循环调用直至所有数据接收完成；调用前必须先通过 <code>getNewTagTotalFromFlash()</code> 或 <code>getAllTagTotalFromFlash()</code> 获取缓存标签总数，否则直接调用将无数据返回。 Get the list of tag data from Flash. This method returns only a portion of the data in a single call. Therefore, it needs to be called repeatedly until all the data is received. Before calling this method, you must first obtain the total number of cached tags through <code>getNewTagTotalFromFlash()</code> or <code>getAllTagTotalFromFlash()</code> ; otherwise, no data will be returned if you call it directly.
参数 Parameters	无 N/A
返回 Return	标签数据列表 (List<UHFTAGInfo>类型) ， 包含Flash中存储的标签详细信息 List of tag data (List<UHFTAGInfo> type) containing detailed information of tags stored in Flash.
备注 Remark	只支持 PDA_A Only supports Device_A

7 数据读写

7 Data Read/Write

7.1 readData(String,int,int,int)

项目 Item	说明 Description
原型 Prototype	String readData(String accessPwd, int bank, int offset, int length)
描述 Description	读取数据 Read data
参数 Parameters	accessPwd: 访问密码 accessPwd: Access password bank: 内存 (IUHF.BANK_RESERVED/EPC/TID/USER) bank: Memory (IUHF.BANK_RESERVED/EPC/TID/USER) offset: 偏移量 offset: Offset length: 长度 length: Length
返回 Return	数据 Data

7.2 readData(String,int,int,int,String,int,int,int)

项目 Item	说明 Description
原型 Prototype	String readData(String accessPwd, int filterBank, int filterOffset, int filterLength, String filterData, int bank, int offset, int length)
描述 Description	读取数据 (带过滤条件) Read data with filter condition

项目 Item	说明 Description
参数 Parameters	accessPwd: 访问密码 accessPwd: Access password filterBank: 过滤内存 (IUHF.BANK_EPC/IUHF.BANK_TID/IUHF.BANK_USER) filterBank: Filter bank (IUHF.BANK_EPC/IUHF.BANK_TID/IUHF.BANK_USER) filterOffset: 过滤偏移量 filterOffset:Filter offset filterLength: 过滤长度 filterLength:Filter length filterData: 过滤数据 filterData:Filter data bank: 目标内存 (IUHF.BANK_RESERVED/EPC/TID/USER) bank: Memory (IUHF.BANK_RESERVED/EPC/TID/USER) offset: 偏移量 offset: Offset length: 长度 length: Length
返回 Return	数据 Data

7.3 writeData(String,int,int,int,String)

项目 Item	说明 Description
原型 Prototype	boolean writeData(String accessPwd, int bank, int offset, int length, String data)
描述 Description	写入数据 Write data
参数 Parameters	accessPwd: 访问密码 accessPwd: Access password bank: 内存 bank: Memory offset: 偏移量 offset: Offset length: 长度 length: Length data: 数据 data: Data
返回 Return	true:成功 false:失败 true:succcess false:failure

7.4 writeData(String,int,int,int,String,int,int,int,String)

项目 Item	说明 Description
原型 Prototype	boolean writeData(String accessPwd, int filterBank, int filterOffset, int filterLength, String filterData, int bank, int offset, int length, String writeData)
描述 Description	写入数据 (带过滤条件) Write data with filter condition

项目 Item	说明 Description
参数 Parameters	accessPwd: 访问密码 accessPwd: Access password filterBank: 过滤内存 (IUHF.BANK_EPC/IUHF.BANK_TID/IUHF.BANK_USER) filterBank: Filter Bank (IUHF.BANK_EPC/IUHF.BANK_TID/IUHF.BANK_USER) filterOffset: 过滤偏移量 filterOffset:Filter offset filterLength: 过滤长度 filterLength:Filter length filterData: 过滤数据 filterData:Filter data bank: 目标内存 (IUHF.BANK_RESERVED/EPC/TID/USER) bank: Memory (IUHF.BANK_RESERVED/EPC/TID/USER) offset: 偏移量 offset:Offset length: 长度 length:Length writeData: 写入数据 writeData:Data
返回 Return	true:成功 false:失败 true:success false:failure

7.5 writeDataToEpc(String,String)

项目 Item	说明 Description
原型 Prototype	boolean writeDataToEpc(String accessPwd, String writeData)
描述 Description	写入数据到EPC Write data to the EPC
参数 Parameters	accessPwd: 访问密码 accessPwd: Access password writeData: 写入数据 writeData: Write data
返回 Return	true:成功 false:失败 true:success false:failure

7.6 writeDataToEpc(String,int,int,int,String,String)

项目 Item	说明 Description
原型 Prototype	boolean writeDataToEpc(String accessPwd, int filterBank, int filterOffset, int filterLength, String filterData, String writeData)
描述 Description	写入数据到EPC (带过滤条件) Write data to the EPC with filter condition

项目 Item	说明 Description
参数 Parameters	accessPwd: 访问密码 accessPwd: Access password filterBank: 过滤内存 (IUHF.BANK_EPC/IUHF.BANK_TID/IUHF.BANK_USER) filterBank: Filter Bank (IUHF.BANK_EPC/IUHF.BANK_TID/IUHF.BANK_USER) filterOffset: 过滤偏移量 filterOffset:Filter offset filterLength: 过滤长度 filterLength:Filter length filterData: 过滤数据 filterData:Filter data offset: 偏移量 offset:Offset length: 长度 length:Length writeData: 写入数据 writeData:Data
返回 Return	true:成功 false:失败 true:success false:failure

7.7 blockWriteData(String,int,int,int,String,int,int,int,String)

项目 Item	说明 Description
原型 Prototype	boolean blockWriteData(String accessPwd, int filterBank, int filterOffset, int filterLength, String filterData, int bank, int offset, int length, String writeData)
描述 Description	块写入数据（带过滤条件） Block write data with filter condition
参数 Parameters	accessPwd: 访问密码 accessPwd: Access password filterBank: 过滤内存 (IUHF.BANK_EPC/IUHF.BANK_TID/IUHF.BANK_USER) filterBank: Filter Bank (IUHF.BANK_EPC/IUHF.BANK_TID/IUHF.BANK_USER) filterOffset: 过滤偏移量 filterOffset:Filter offset filterLength: 过滤长度 filterLength:Filter length filterData: 过滤数据 filterData:Filter data bank: 目标内存 (IUHF.BANK_RESERVED/EPC/TID/USER) bank: Memory (IUHF.BANK_RESERVED/EPC/TID/USER) offset: 偏移量 offset:Offset length: 长度 length:Length writeData: 写入数据 writeData:Data

7.8 eraseData(String,int,int,int)

项目 Item	说明 Description
原型 Prototype	boolean eraseData(String accessPwd, int bank, int offset, int length)
描述 Description	擦除数据 Erase data
参数 Parameters	accessPwd: 访问密码 accessPwd: Access password bank: 内存 bank: Memory offset: 偏移量 offset: Offset length: 长度 length: Length
返回 Return	true:成功 false:失败 true:success false:failure

7.9 eraseData(String,int,int,int,String,int,int,int)

项目 Item	说明 Description
原型 Prototype	boolean eraseData(String accessPwd, int filterBank, int filterOffset, int filterLength, String filterData, int bank, int offset, int length)
描述 Description	擦除数据（带过滤条件） Erase data with filter condition
参数 Parameters	accessPwd: 访问密码 accessPwd: Access password filterBank: 过滤内存（IUHF.BANK_EPC/IUHF.BANK_TID/IUHF.BANK_USER） filterBank: Filter Bank（IUHF.BANK_EPC/IUHF.BANK_TID/IUHF.BANK_USER） filterOffset: 过滤偏移量 filterOffset:Filter offset filterLength: 过滤长度 filterLength:Filter length filterData: 过滤数据 filterData:Filter data bank: 目标内存（IUHF.BANK_RESERVED/EPC/TID/USER） bank: Memory（IUHF.BANK_RESERVED/EPC/TID/USER） offset: 偏移量 offset:Offset length: 长度
返回 Return	true:成功 false:失败 true:success false:failure

7.10 uhfBlockPermalock(String,int,int,int,String, int, int,int,int,byte[])

项目 Item	说明 Description
原型 Prototype	boolean uhfBlockPermalock(String accessPwd, int FilterBank, int FilterStartaddr, int FilterLen, String FilterData, int ReadLock, int uBank, int uPtr, int uRange, byte[] uMaskbuf)
描述 Description	数据块操作 Data block operation
参数 Parameters	accessPwd: 访问密码 accessPwd: Access password FilterBank: 过滤的存储区 FilterBank: Filtered storage area: IUHF.Bank_EPC、IUHF.Bank_TID、IUHF.Bank_USER FilterStartaddr: 过滤的起始地址 FilterStartaddr: Filter start address FilterLen: 过滤数据长度 FilterLen: Filter data length FilterData: 过滤的数据 FilterData: Filtered data ReadLock: 读写锁 ReadLock: Read write lock uBank: 存储区域 uBank: Storage area: IUHF.Bank_RESERVED、IUHF.Bank_EPC、IUHF.Bank_TID、IUHF.Bank_USER uPtr: 存储区域起始地址，默认为0，其中1个块单元为16个块 uPtr: Start address in storage area, default is 0, 1 unit has 16 blocks uRange: 存储区域结束范围，默认为1，其中1个块单元为16个块 uRange: End of storage, default is 1, 1 unit has 16 blocks uMaskbuf: 设置锁MSAK，高位在前，每个bit代表为1个块，默认为2字节 uMaskbuf: Setup lock MSAK, high level is at front, every bit has 1 block, default is 2 bytes.
返回 Return	true:成功 false:失败 true:success false:failure

8 标签销毁和锁定

8 Tag Kill、Lock

8.1 killTag(String)

项目 Item	说明 Description
原型 Prototype	boolean killTag(String killPwd)
描述 Description	杀死标签 Kill tag
参数 Parameters	killPwd: 杀死密码 killPwd: Kill password
返回 Return	true:成功 false:失败 true:success false:failure

8.2 killTag(String,int,int,int,String)

项目 Item	说明 Description
原型 Prototype	boolean killTag(String killPwd, int filterBank, int filterOffset, int filterLength, String filterData)
描述 Description	杀死标签（带过滤条件） Kill tag with filter condition
参数 Parameters	killPwd: 杀死密码 killPwd: Kill password filterBank: 过滤内存（IUHF.BANK_EPC/IUHF.BANK_TID/IUHF.BANK_USER） filterBank: Filter bank（IUHF.BANK_EPC/IUHF.BANK_TID/IUHF.BANK_USER） filterOffset: 过滤偏移量 filterOffset:Filter offset filterLength: 过滤长度 filterLength:Filter length filterData: 过滤数据 filterData:Filter data
返回 Return	true:成功 false:失败 true:success false:failure

8.3 lockMem(String,String)

项目 Item	说明 Description
原型 Prototype	boolean lockMem(String accessPwd, lockCode)
描述 Description	锁定标签 Lock tag
参数 Parameters	accessPwd: 访问密码 accessPwd: Access password lockCode: 锁码 lockCode: Lock code
返回 Return	true:成功 false:失败 true:success false:failure
备注 Remark	lockCode通过 generateLockCode() 函数获取 lockCode is obtained through the generateLockCode() function

8.4 lockMem(String,int,int,int,String,String)

项目 Item	说明 Description
原型 Prototype	boolean lockMem(String accessPwd, int filterBank, int filterOffset, int filterLength, String filterData, String lockCode)
描述 Description	锁定标签（带过滤与锁码） Lock tag with filter condition and lock code

项目 Item	说明 Description
参数 Parameters	accessPwd: 访问密码 accessPwd: Access password filterBank: 过滤内存 (IUHF.BANK_EPC/IUHF.BANK_TID/IUHF.BANK_USER) filterBank: Filter bank (IUHF.BANK_EPC/IUHF.BANK_TID/IUHF.BANK_USER) filterOffset: 过滤偏移量 filterOffset:Filter offset filterLength: 过滤长度 filterLength:Filter length filterData: 过滤数据 filterData:Filter data lockCode: 锁码 lockCode:Lock code
返回 Return	true:成功 false:失败 true:success false:failure
备注 Remark	lockCode通过 generateLockCode() 函数获取 lockCode is obtained through the generateLockCode() function

8.5 generateLockCode(ArrayList,int)

项目 Item	说明 Description
原型 Prototype	String generateLockCode(ArrayList lockBank, int lockMode)
描述 Description	生成锁码 Generate lock code
参数 Parameters	lockBank: IUHF.LockBank_ACCESS IUHF.LockBank_KILL IUHF.LockBank_EPC IUHF.LockBank_USER lockMode: IUHF.LockMode_LOCK IUHF.LockMode_OPEN IUHF.LockMode_PLOCK-永久锁定(Permanent LOCK) IUHF.LockMode_POPEN-永久开放(Permanent release)
返回 Return	锁码 Lock code
备注 Remark	比如:同时锁EPC和 USER For example: Lock both the EPC bank and the USER bank at the same time ArrayList lockBank= new ArrayList(); lockBank.add(IUHF.LockBank_EPC); lockBank.add(IUHF.LockBank_USER); String lockCode=rfidWithUHFUART.generateLockCode(lockBank,IUHF.LockMode_LOCK);

9 RFLink、FastID、TagFocus

9 RFLink、FastID、TagFocus

9.1 setRFLink(int)

项目 Item	说明 Description
原型 Prototype	boolean setRFLink(int mode)
描述 Description	设置RFLink Set RFLink
参数 Parameters	mode: 模式 mode: Mode 0-PR ASK/Miller8/160KHz 1-PR ASK/Miller4/250KHz 2-PR ASK/Miller4/320KHz 3-PR ASK/Miller4/640KHz 4-PR ASK/Miller2/320KHz 5-PR ASK/Miller2/640KHz 10-Gen2X/Miller8/160KHz 11-Gen2X/Miller4/250KHz 12-Gen2X/Miller4/320KHz 13-Gen2x/Miller4/640KHz 14-Gen2X/Miller2/320KHz 15-Gen2X/Miller2/640KHz 20-Autoset/Miller8/160KHz 21-Autoset/Miller4/250KHz 22-Autoset/Miller4/320KHz 23-Autoset/Miller4/640KHz 24-Autoset/Miller2/320KHz 25-Autoget/Miller2/640KHz
返回 Return	true:成功 false:失败 true:success false:failure
备注 Remark	Gen2X的链路只支持英频杰M830和M850等最新的标签。 The links of Gen2X only support the latest tags such as Impinj M830 and M850.

9.2 getRFLink()

项目 Item	说明 Description
原型 Prototype	synchronized int getRFLink()
描述 Description	获取RFLink模式 Get RFLink mode
参数 Parameters	无 N/A

项目 Item	说明 Description
返回 Return	-1:失败(failure) 0-PR ASK/Miller8/160KHz 1-PR ASK/Miller4/250KHz 2-PR ASK/Miller4/320KHz 3-PR ASK/Miller4/640KHz 4-PR ASK/Miller2/320KHz 5-PR ASK/Miller2/640KHz 10-Gen2X/Miller8/160KHz 11-Gen2X/Miller4/250KHz 12-Gen2X/Miller4/320KHz 13-Gen2x/Miller4/640KHz 14-Gen2X/Miller2/320KHz 15-Gen2X/Miller2/640KHz 20-Autoset/Miller8/160KHz 21-Autoset/Miller4/250KHz 22-Autoset/Miller4/320KHz 23-Autoset/Miller4/640KHz 24-Autoset/Miller2/320KHz 25-Autoget/Miller2/640KHz

9.3 setFastID(boolean)

项目 Item	说明 Description
原型 Prototype	synchronized boolean setFastID(boolean enalbe)
描述 Description	设置FastID Set Fast ID
参数 Parameters	enalbe: 是否启用 enalbe: Enable
返回 Return	true:成功 false:失败 true:success false:failure

9.4 getFastId()

项目 Item	说明 Description
原型 Prototype	synchronized int getFastId()
描述 Description	获取FastId Get FastId
参数 Parameters	无 N/A
返回 Return	0:关闭 1:打开 -1:失败 0:Close 1:Open -1:failure

9.5 setTagFocus(boolean)

项目 Item	说明 Description
原型 Prototype	synchronized boolean setTagFocus(boolean enalbe)
描述 Description	设置TagFocus Set Tag Focus
参数 Parameters	enalbe: 是否启用 enalbe: Enable
返回 Return	true:成功 false:失败 true:succcess false:failure

9.6 getTagFocus()

项目 Item	说明 Description
原型 Prototype	synchronized int getTagFocus()
描述 Description	获取TagFocus Get TagFocus
参数 Parameters	无 N/A
返回 Return	0:关闭 1:打开 -1:失败 0:Close 1:Open -1:failure

9.7 setGen2(Gen2Entity)

项目 Item	说明 Description
原型 Prototype	boolean setGen2(Gen2Entity gen2Entity)
描述 Description	设置GEN2参数值 Set GEN2 parameter value
参数 Parameters	gen2Entity: GEN2参数实体对象 gen2Entity: GEN2 parameter entity object
返回 Return	true:成功 false:失败 true:succcess false:failure
备注 Remark	设置之前先调用getGen2()获取Gen2Entity对象，在修改对应参数。目前只支持 Gen2Entity.setQuerySession(int)和Gen2Entity.setQueryTarget(int). Call getGen2() first to obtain the Gen2Entity object before setting, and then modify the corresponding parameters. Currently, only Gen2Entity.setQuerySession(int) and Gen2Entity.setQueryTarget(int) are supported.

9.8 getGen2()

项目 Item	说明 Description
原型 Prototype	Gen2Entity getGen2()
描述 Description	获取GEN2参数值 Get GEN2 parameter value
参数 Parameters	无 N/A
返回 Return	GEN2参数值实体对象 GEN2 parameter value entity object

10 定位功能

10 Location Functions

10.1 startLocation(Context,String,int,int,IUHFLocationCallback)

项目 Item	说明 Description
原型 Prototype	boolean startLocation(Context context, String label, int bank, int offset, IUHFLocationCallback locationCallback)
描述 Description	开始定位 Start location
参数 Parameters	context: 上下文 context: Context label: 要定位的标签 label: Tag to locate bank: 标签的存储区 bank: Storage area offset: 起始地址的偏移量 offset: Start address offset locationCallback: 定位回调 locationCallback: Location callback
返回 Return	true:成功 false:失败 true:success false:failure

10.2 stopLocation()

项目 Item	说明 Description
原型 Prototype	boolean stopLocation()
描述 Description	停止定位 Stop location

项目 Item	说明 Description
参数 Parameters	无 N/A
返回 Return	true:成功 false:失败 true:success false:failure

10.3 startRadarLocation(Context,String,int,int,IUHFRadarLocationCallback)

项目 Item	说明 Description
原型 Prototype	boolean startRadarLocation(Context context, String tag, int bank, int offset, IUHFRadarLocationCallback locationCallback)
描述 Description	开始雷达定位 Start radar location
参数 Parameters	context: 上下文 context: Context tag: 要定位的标签 tag: Tag to locate bank: 标签的存储区 bank: Storage area offset: 起始地址的偏移量 offset: Start address offset locationCallback: 雷达定位回调 locationCallback: Radar location callback
返回 Return	true:成功 false:失败 true:success false:failure

10.4 stopRadarLocation()

项目 Item	说明 Description
原型 Prototype	boolean stopRadarLocation()
描述 Description	停止雷达定位 Stop radar location
参数 Parameters	无 N/A
返回 Return	true:成功 false:失败 true:success false:failure

10.5 setDynamicDistance(int value)

项目 Item	说明 Description
原型 Prototype	boolean setDynamicDistance(int value)
描述 Description	雷达定位具备动态距离调整功能，在雷达定位过程中，可调用此函数动态调整定位距离，以提升标签寻找的精准度。 Radar positioning enables dynamic distance adjustment. During the radar positioning process, this function can be called to dynamically adjust the positioning distance, thereby improving the accuracy of tag detection.
参数 Parameters	value: 取值范围为1-30 value: Range is 1-30
返回 Return	true:成功 false:失败 true:success false:failure

11 条码

11 Barcode

11.1 scanBarcodeToBytes()

项目 Item	说明 Description
原型 Prototype	byte[] scanBarcodeToBytes()
描述 Description	扫描条形码并返回字节数组 Scan barcode and return byte array
参数 Parameters	无 None
返回 Return	条形码数据的字节数组 Byte array of barcode data

11.2 scanBarcode()

项目 Item	说明 Description
原型 Prototype	String scanBarcode()
描述 Description	扫描条形码并返回字符串 Scan barcode and return string
参数 Parameters	无 None
返回 Return	条形码数据的字符串 String of barcode data

12 升级功能

12 Upgrade Functions

12.1 uhfJump2Boot(int)

项目 Item	说明 Description
原型 Prototype	boolean uhfJump2Boot(int type)
描述 Description	跳转至BOOT模式 Jump to BOOT mode
参数 Parameters	type: 0,upgrade reader mainboard 1,upgrade UHF module 2,upgrade reader bootloader 3,upgrade Ex10 SDK firmware
返回 Return	true:成功 false:失败 true:success false:failure

12.2 uhfStartUpdate()

项目 Item	说明 Description
原型 Prototype	boolean uhfStartUpdate()
描述 Description	开始升级uhf模块 Start upgradfing the UHF module
参数 Parameters	无 N/A
返回 Return	true:成功 false:失败 true:success false:failure

12.3 uhfUpdating(byte[] buff)

项目 Item	说明 Description
原型 Prototype	boolean uhfUpdating(byte[] buff)
描述 Description	升级uhf模块，发送uhf固件数据 Upgrade the UHF module and send UHF firmware data
参数 Parameters	buff: 固件数据,最大支持64字节 buff: Firmware data, maximum support for 64 bytes
返回 Return	true:成功 false:失败 true:success false:failure

12.4 uhfStopUpdate()

项目 Item	说明 Description
原型 Prototype	synchronized boolean uhfStopUpdate()
描述 Description	停止升级uhf模块 Stop upgrading the UHF module
参数 Parameters	无 N/A
返回 Return	true:成功 false:失败 true:success false:failure

13 其他功能

13 Other Functions

13.1 setBaudRate(int baudRate)

项目 Item	说明 Description
原型 Prototype	boolean setBaudRate(int baudRate)
描述 Description	设置UHF通信波特率 Set the UHF communication baud rate
参数 Parameters	baudRate: 波特率 (2:115200, 3:460800) baudRate: Baud rate (2:115200, 3:460800)
返回 Return	true:成功 false:失败 true:success false:failure

13.2 getBaudRate()

项目 Item	说明 Description
原型 Prototype	int getBaudRate()
描述 Description	获取UHF通信波特率 Get UHF communication baud rate
参数 Parameters	无 N/A
返回 Return	波特率 (2:115200, 3:460800) Baud rate (2:115200, 3:460800)

13.3 setUart(String uart)

项目 Item	说明 Description
原型 Prototype	boolean setUart(String uart)
描述 Description	android设备连接UHF模块开发可以使用此函数设置uhf串口 Android devices can use this function to configure the UHF serial port when connecting to a UHF module for development.
参数 Parameters	uart: uart路径 uart: uart path
返回 Return	void
备注 Remark	设置uhf串口之后在调用init(Context); 如果初始化失败，请检查模块供电是否正常，以及串口收发数据是否正常。 After configuring the UHF serial port, call the init(Context)method. If initialization fails, check whether the module's power supply is normal and verify the serial port's data transmission/reception status.

13.4 void setKeyEventCallback(KeyEventCallback keyEventCallback)

项目 Item	说明 Description
原型 Prototype	void setKeyEventCallback(KeyEventCallback keyEventCallback)
描述 Description	设置按键事件回调接口，用于注册监听按键按下和松开的事件通知 Set the key event callback interface to register and listen for event notifications of key presses and releases
参数 Parameters	keyEventCallback： 按键事件回调接口实例，用于接收按键事件通知 keyEventCallback: Instance of the key event callback interface, used to receive key event notifications
返回 Return	无 N/A

13.5 factoryReset()

项目 Item	说明 Description
原型 Prototype	boolean factoryReset()
描述 Description	恢复出厂设置 factory reset
参数 Parameters	无 N/A
返回 Return	true:成功 false:失败 true:success false:failure

13.6 setReaderAwaitSleepTime(int time)

项目 Item	说明 Description
原型 Prototype	boolean setReaderAwaitSleepTime(int time)
描述 Description	设置读写器等待休眠的时间，该时间特指读写器在断开连接状态下即将进入休眠的时长 Set the time for the reader to wait before entering sleep mode. This time specifically refers to the duration during which the reader will enter sleep mode when it is disconnected from the connection.
参数 Parameters	time：等待休眠时间，单位为分钟 - 取值范围：1-254 - 默认值：5分钟 time: Hibernation waiting time, unit is minute - Value range: 1-254 - Default value: 5 minutes
返回 Return	true:成功 false:失败 true:success false:failure

13.7 int getReaderAwaitSleepTime()

项目 Item	说明 Description
原型 Prototype	int getReaderAwaitSleepTime()
描述 Description	获取读写器等待休眠的时间 Get the time for the reader to wait for hibernation
参数 Parameters	无 N/A
返回 Return	返回值为int类型 - 成功：返回读写器等待休眠的时间（单位：分钟） - 失败：返回-1，表示获取失败 Return value is of int type - Success: Returns the reader's hibernation await time (unit: minute) - Failure: Returns -1, indicating acquisition failure

14 外部相关类

14 External related classes

14.1 InventoryParameter类

14.1 InventoryParameter Class

盘点参数类，用于配置UHF标签盘点的相关参数，包括结果数据类型和标签去重规则。

Inventory parameter class, used to configure parameters related to UHF tag inventory, including result data type and tag deduplication rules.

方法 Method	描述 Description	参数 Parameters	返回值 Return
InventoryParameter()	无参构造函数 No-argument constructor	无 N/A	-
InventoryParameter(ResultData resultData)	带结果数据的构造函数 Constructor with result data	resultData:结果数据配置对象 resultData:Result data configuration object	-
InventoryParameter(boolean isUnique)	带去重标识的构造函数 Constructor with uniqueness flag	isUnique:是否去重标识 isUnique:Uniqueness flag	-
InventoryParameter(ResultData resultData, boolean isTagUnique)	带结果数据和去重标识的构造函数 Constructor with result data and uniqueness flag	resultData:结果数据配置对象 isTagUnique:是否去重标识	-
getResultData()	获取结果数据配置 Get result data configuration	无 N/A	结果数据配置对象 ResultData object
setResultData(ResultData resultData)	设置结果数据配置 Set result data configuration	resultData:结果数据配置对象 resultData:Result data configuration object	当前InventoryParameter实例 Current InventoryParameter instance
isTagUnique()	判断是否启用标签去重 Check if tag deduplication is enabled	无 N/A	布尔值（true:去重, false:不去重） Boolean (true:unique, false:not unique)
setTagUnique(boolean tagUnique)	设置是否启用标签去重 Set whether to enable tag deduplication	tagUnique:是否去重标识 tagUnique:Uniqueness flag	void

14.2 InventoryParameter.ResultData内部类

14.2 InventoryParameter.ResultData Inner Class

结果数据配置类，用于指定盘点结果中需要包含的额外数据（频率点和相位点）。

Result data configuration class, used to specify additional data (frequency point and phase point) to be included in inventory results.

方法 Method	描述 Description	参数 Parameters	返回值 Return
isNeedFrequencyPoint()	判断是否需要频率点数据 Check if frequency point data is needed	无 N/A	布尔值（true:需要, false:不需要） Boolean (true:need, false:not need)
isNeedPhase()	判断是否需要相位点数据 Check if phase point data is needed	无 N/A	布尔值（true:需要, false:不需要） Boolean (true:need, false:not need)
setNeedFrequencyPoint(boolean needFrequencyPoint)	设置是否需要频率点数据 Set whether frequency point data is needed	needFrequencyPoint:是否需要频率点标识 needFrequencyPoint:Frequency point need flag	当前ResultData实例 Current ResultData instance

方法 Method	描述 Description	参数 Parameters	返回值 Return
setNeedPhase(boolean needPhase)	设置是否需要相位点数据 Set whether phase point data is needed	needPhase:是否需要相位点标识 needPhase:Phase point need flag	当前ResultData实例 Current ResultData instance

14.3 InventoryModeEntity类

14.3 InventoryModeEntity Class

盘点模式实体类，用于定义UHF标签盘点时的输出数据模式，包括不同存储区数据的输出配置及访问密码等信息。

Inventory mode entity class, used to define the output data mode during UHF tag inventory, including output configuration of data in different storage areas and access password information.

常量 Constant	描述 Description
MODE_EPC	EPC模式，盘点时仅输出标签的EPC数据，不能指定EPC的起始地址和长度 EPC mode, only output EPC data of the tag, can not specify the start address and length of EPC
MODE_EPC_TID	EPC+TID模式，盘点时输出标签的EPC和TID数据，不能指定EPC和TID的起始地址和长度 EPC+TID mode, output EPC and TID data of the tag, can not specify the start address and length of EPC and TID
MODE_EPC_TID_USER	EPC+TID+USER模式，盘点时输出标签的EPC、TID和USER数据，只能指定USER的起始地址和长度 EPC+TID+USER mode, output EPC, TID and USER data of the tag, can only specify the start address and length of USER
MODE_EPC_TID_M775AUTHENTICATION	EPC+TID+M775模式，盘点时输出标签的EPC、TID和M775认证数据，不能指定EPC、TID和M775认证数据的起始地址和长度 EPC+TID+M775 mode, output EPC, TID and M775 authentication data of the tag, can not specify the start address and length of EPC, TID and M775 authentication data
MODE_EPC_TID_RESERVED	EPC+TID+RESERVED模式，盘点时输出标签的EPC、TID和RESERVED数据，且只能指定RESERVED数据的起始地址和长度 EPC+TID+RESERVED mode, output EPC, TID and RESERVED data of the tag, and can only specify the start address and length of RESERVED data
MODE_EPC_RESERVED	EPC+RESERVED模式，盘点时仅输出标签的EPC和RESERVED数据，且只能指定RESERVED数据的起始地址和长度 EPC+RESERVED mode, output EPC and RESERVED data of the tag, and can only specify the start address and length of RESERVED data
MODE_EPC_MEMORY_BANK_EPC	EPC+MEMORY_BANK_EPC模式，盘点时输出标签的EPC和MEMORY_BANK_EPC数据，且只能指定EPC数据的起始地址和长度 EPC+MEMORY_BANK_EPC mode, output EPC and MEMORY_BANK_EPC data of the tag, and can only specify the start address and length of EPC data

常量 Constant	描述 Description
MODE_EPC_MEMORY_BANK_TID	EPC+MEMORY_BANK_TID模式，盘点时输出标签的EPC和MEMORY_BANK_TID数据，且只能指定TID数据的起始地址和长度 EPC+MEMORY_BANK_TID mode, output EPC and MEMORY_BANK_TID data of the tag, and can only specify the start address and length of TID data
MODE_EPC_MEMORY_BANK_USER	EPC+MEMORY_BANK_USER模式，盘点时输出标签的EPC和MEMORY_BANK_USER数据，且只能指定USER数据的起始地址和长度 EPC+MEMORY_BANK_USER mode, output EPC and MEMORY_BANK_USER data of the tag, and can only specify the start address and length of USER data
MODE_LED_TAG	LED_TAG模式，盘点时点亮LED标签 LED_TAG mode, light up LED tag during inventory
MODE_TEMPERATURE_TAG	TEMPERATURE_TAG模式，盘点时输出标签的温度数据 TEMPERATURE_TAG mode, output temperature data of the tag during inventory

方法 Method	描述 Description	参数 Parameters	返回值 Return
getMode()	获取盘点模式 Get inventory mode	无 N/A	盘点模式（对应上述常量值） Inventory mode (corresponding to the above constant values)
getUserLength()	获取要盘点的User区长度 Get User area data length	无 N/A	User区长度 User area data length
getUserOffset()	获取要盘点的User区起始地址 Get User area start address	无 N/A	User区起始地址 User area start address
getReservedLength()	获取要盘点的RESERVED区长度 Get RESERVED area data length	无 N/A	RESERVED区长度 RESERVED area data length
getReservedOffset()	获取要盘点的RESERVED区起始地址 Get RESERVED area start address	无 N/A	RESERVED区起始地址 RESERVED area start address
getEpcLength()	获取要盘点的EPC区长度 Get EPC area data length	无 N/A	EPC区长度 EPC area data length
getEpcOffset()	获取要盘点的EPC区起始地址 Get EPC area start address	无 N/A	EPC区起始地址 EPC area start address
getTidLength()	获取要盘点的TID区长度 Get TID area data length	无 N/A	TID区长度 TID area data length

方法 Method	描述 Description	参数 Parameters	返回值 Return
getTidOffset()	获取要盘点的TID区起始地址 Get TID area start address	无 N/A	TID区起始地址 TID area start address
getHexAccessPassword()	获取访问密码 Get access password	无 N/A	访问密码（十六进制字符串） Access password (hex string)

14.4 Builder内部类

14.4 InventoryModeEntity.Builder Inner Class

InventoryModeEntity的构建器类，用于便捷地创建和配置InventoryModeEntity实例。

Builder class for InventoryModeEntity, used to conveniently create and configure InventoryModeEntity instances.

方法 Method	描述 Description	参数 Parameters	返回值 Return
setMode(int mode)	设置盘点模式 Set inventory mode	mode: 盘点模式（参考上述常量） mode: Inventory mode (refer to the above constants)	构建器实例（Builder） Builder instance
setUserLength(int userLen)	设置要盘点的User区长度 Set User area data length	userLen: User区长度 userLen: User area data length	构建器实例（Builder） Builder instance
setUserOffset(int userOffset)	设置要盘点的User区起始地址 Set User area start address	userOffset: User区起始地址 userOffset: User area start address	构建器实例（Builder） Builder instance
setReservedLength(int reservedLength)	设置要盘点的RESERVED区长度 Set RESERVED area data length	reservedLength: RESERVED区长度 reservedLength: RESERVED area data length	构建器实例（Builder） Builder instance
setReservedOffset(int reservedOffset)	设置要盘点的RESERVED区起始地址 Set RESERVED area start address	reservedOffset: RESERVED区起始地址 reservedOffset: RESERVED area start address	构建器实例（Builder） Builder instance
setEpcLength(int epcLength)	设置要盘点的EPC区长度 Set EPC area data length	epcLength: EPC区长度 epcLength: EPC area data length	构建器实例（Builder） Builder instance
setEpcOffset(int epcOffset)	设置要盘点的EPC区起始地址 Set EPC area start address	epcOffset: EPC区起始地址 epcOffset: EPC area start address	构建器实例（Builder） Builder instance
setTidLength(int tidLength)	设置要盘点的TID区长度 Set TID area data length	tidLength: TID区长度 tidLength: TID area data length	构建器实例（Builder） Builder instance

方法 Method	描述 Description	参数 Parameters	返回值 Return
setTidOffset(int tidOffset)	设置要盘点的TID区起始地址 Set TID area start address	tidOffset: TID区起始地址 tidOffset: TID area start address	构建器实例 (Builder) Builder instance
setHexAccessPassword(String pwd)	设置访问密码 Set access password	pwd: 访问密码 (十六进制字符串) pwd: Access password (hex string)	构建器实例 (Builder) Builder instance
build()	构建盘点模式实体 Build inventory mode entity	无 N/A	配置完成的InventoryModeEntity实例 Configured InventoryModeEntity instance

14.5 Gen2Entity类

14.5 Gen2Entity Class

Gen2协议参数类，用于配置与Gen2协议相关的各种参数，包括select命令和query命令的参数以及链路频率等。

Gen2 protocol parameter class, used to configure various parameters related to the Gen2 protocol, including parameters of the select command, query command, and link frequency, etc.

方法 Method	描述 Description	参数 Parameters	返回值 Return
checkParameter()	检查所有参数是否都已设置 (均不为-1) Check if all parameters have been set (none are -1)	无 N/A	布尔值 (true:所有参数均已设置, false:存在未设置的参数) Boolean (true:all parameters are set, false:there are unset parameters)
getSelectTarget()	获取select命令的Target参数 Get the Target parameter of the select command	无 N/A	整数 (0:s0, 1:s1, 2:s2, 3:s3, 4:SL) Integer (0:s0, 1:s1, 2:s2, 3:s3, 4:SL)
setSelectTarget(int selectTarget)	设置select命令的Target参数 Set the Target parameter of the select command	selectTarget:整数 (0:s0, 1:s1, 2:s2, 3:s3, 4:SL) selectTarget:Integer (0:s0, 1:s1, 2:s2, 3:s3, 4:SL)	void
getSelectAction()	获取select命令的Action参数 Get the Action parameter of the select command	无 N/A	整数 (0-7, 分别对应不同的匹配和非匹配操作) Integer (0-7, corresponding to different matching and non-matching operations)

方法 Method	描述 Description	参数 Parameters	返回值 Return
setSelectAction(int selectAction)	设置select命令的Action参数 Set the Action parameter of the select command	selectAction:整数 (0-7, 分别对应不同的匹配和非匹配操作) selectAction:Integer (0-7, corresponding to different matching and non-matching operations)	void
getSelectTruncate()	获取select命令的Truncate参数 Get the Truncate parameter of the select command	无 N/A	整数 (0:禁用截断, 1:启用截断) Integer (0:Disable truncation, 1:Enable truncation)
setSelectTruncate(int selectTruncate)	设置select命令的Truncate参数 Set the Truncate parameter of the select command	selectTruncate:整数 (0:禁用截断, 1:启用截断) selectTruncate:Integer (0:Disable truncation, 1:Enable truncation)	void
getQ()	获取Q值 Get the Q value	无 N/A	整数 (0:固定Q算法, 1:动态Q算法) Integer (0:fixed Q algorithm, 1:dynamic Q algorithm)
setQ(int q)	设置Q值 Set the Q value	q:整数 (0:固定Q算法, 1:动态Q算法) q:Integer (0:fixed Q algorithm, 1:dynamic Q algorithm)	void
getStartQ()	获取StartQ值 Get the StartQ value	无 N/A	整数 (0-15) Integer (0-15)
setStartQ(int startQ)	设置StartQ值 Set the StartQ value	startQ:整数 (0-15) startQ:Integer (0-15)	void
getMinQ()	获取MinQ值 Get the MinQ value	无 N/A	整数 (0-15) Integer (0-15)
setMinQ(int minQ)	设置MinQ值 Set the MinQ value	minQ:整数 (0-15) minQ:Integer (0-15)	void
getMaxQ()	获取MaxQ值 Get the MaxQ value	无 N/A	整数 (0-15) Integer (0-15)
setMaxQ(int maxQ)	设置MaxQ值 Set the MaxQ value	maxQ:整数 (0-15) maxQ:Integer (0-15)	void
getQueryDR()	获取query命令的DR参数 Get the DR parameter of the query command	无 N/A	整数 (0:8, 1:64/3) Integer (0:8, 1:64/3)

方法 Method	描述 Description	参数 Parameters	返回值 Return
setQueryDR(int queryDR)	设置query命令的DR参数 Set the DR parameter of the query command	queryDR:整数 (0:8, 1:64/3) queryDR:Integer (0:8, 1:64/3)	void
getQueryM()	获取query命令的M参数 Get the M parameter of the query command	无 N/A	整数 (0:FM0, 1:Miller2, 2:Miller4, 3:Miller8) Integer (0:FM0, 1:Miller2, 2:Miller4, 3:Miller8)
setQueryM(int queryM)	设置query命令的M参数 Set the M parameter of the query command	queryM:整数 (0:FM0, 1:Miller2, 2:Miller4, 3:Miller8) queryM:Integer (0:FM0, 1:Miller2, 2:Miller4, 3:Miller8)	void
getQueryTRext()	获取query命令的TRext参数 Get the TRext parameter of the query command	无 N/A	整数 (0:无导频, 1:使用导频) Integer (0:No pilot, 1:Use pilot)
setQueryTRext(int queryTRext)	设置query命令的TRext参数 Set the TRext parameter of the query command	queryTRext:整数 (0:无导频, 1:使用导频) queryTRext:Integer (0:No pilot, 1:Use pilot)	void
getQuerySel()	获取query命令的sel参数 Get the sel parameter of the query command	无 N/A	整数 (0:所有, 1:所有, 2:~SL, 3:SL) Integer (0:All, 1:All, 2:~SL, 3:SL)
setQuerySel(int querySel)	设置query命令的sel参数 Set the sel parameter of the query command	querySel:整数 (0:所有, 1:所有, 2:~SL, 3:SL) querySel:Integer (0:All, 1:All, 2:~SL, 3:SL)	void
getQuerySession()	获取query命令的session参数 Get the session parameter of the query command	无 N/A	整数 (0:S0, 1:S1, 2:S2, 3:S3) Integer (0:S0, 1:S1, 2:S2, 3:S3)
setQuerySession(int querySession)	设置query命令的session参数 Set the session parameter of the query command	querySession:整数 (0:S0, 1:S1, 2:S2, 3:S3) querySession:Integer (0:S0, 1:S1, 2:S2, 3:S3)	void

方法 Method	描述 Description	参数 Parameters	返回值 Return
getQueryTarget()	获取query命令的Target参数 Get the Target parameter of the query command	无 N/A	整数 (0:A, 1:B) Integer (0:A, 1:B)
setQueryTarget(int queryTarget)	设置query命令的Target参数 Set the Target parameter of the query command	queryTarget:整数 (0:A, 1:B) queryTarget:Integer (0:A, 1:B)	void
getLinkFrequency()	获取链路频率设置 Get the link frequency setting	无 N/A	-
setLinkFrequency(int linkFrequency)	设置链路频率 Set the link frequency	-	void
toString()	返回对象的字符串表示 Return the string representation of the object	无 N/A	包含所有参数信息的字符串 String containing all parameter information

14.6 ErrorCode

14.6 错误码定义

常量名 Constant Name	数值 Value	说明 Description
ERROR_NO_TAG	0x01	找不到标签 No tags found.
ERROR_INSUFFICIENT_PRIVILEGES	0x02	没有权限访问 The Interrogator did not authenticate itself with sufficient privileges for the Tag to perform the operation
ERROR_MEMORY_OVERRUN	0x03	数据区超限 Memory Overflow
ERROR_MEMORY_LOCK	0x04	数据区被锁定 Memory lock
ERROR_TAG_NO_REPLY	0x05	标签没有应答 Tag not responding
ERROR_PASSWORD_IS_INCORRECT	0x06	密码不正确 Incorrect Password
ERROR_RESPONSE_BUFFER_OVERFLOW	0x07	缓冲区溢出 The operation failed because the ResponseBuffer overflowed
ERROR_NO_ENOUGH_POWER_ON_TAG	0x0B	标签能量不足 no enough power on tag

常量名 Constant Name	数值 Value	说明 Description
ERROR_OPERATION_FAILED	0xFF	操作失败 Operation failure
ERROR_SEND_FAIL	0xFC	发送失败 Send failure
ERROR_RECV_FAIL	0xFD	接收失败 Receive failure
ERRCODE_SUCCESS	0	成功 Success
ERRCODE_FAILURE	-1	失败 Failure

14.7 FastInventoryEntity类

14.7 FastInventoryEntity Class

快速盘点相关参数类。

Fast Inventory related parameters.

方法 Method	描述 Description	参数 Parameters	返回值 Return
FastInventoryEntity(int cr)	带CR值的构造函数 Constructor with CR value	cr	-
getCr()	获取CR值 Get CR value	无 N/A	0x00:close 0x01:CrID16 0x02:CrStoredCRC 0x03:CrRN16 0x04:CrID32
setCr(int cr)	设置CR值 Set CR value	0x00:close 0x01:CrID16 0x02:CrStoredCRC 0x03:CrRN16 0x04:CrID32	当前FastInventoryEntity实例 Current FastInventoryEntity instance

14.8 IUHFLocationCallback接口

14.8 IUHFLocationCallback Interface

UHF定位回调接口，用于获取标签的定位数据。

UHF location callback interface, used to obtain location data of tags.

方法 Method	描述 Description	参数 Parameters	返回值 Return
getLocationValue(int value, boolean valid)	获取定位数据 Obtain location data	value:定位值（0-100），值越大表示距离标签越近 value:Location value (0-100), the larger the value, the closer it is to the label valid:标记定位值是否有效 valid:Flag whether the location value is valid	void

14.9 IUHFRadarLocationCallback接口

14.9 IUHFRadarLocationCallback Interface

回调定位过程产生的数据的接口。

Interface for callback of data generated by the positioning process.

方法 Method	描述 Description	参数 Parameters	返回值 Return
getLocationValue(List<RadarLocationEntity> radarLocationEntityList)	获取定位数据 Obtain location data	radarLocationEntityList:定位数据列表 radarLocationEntityList:List of location data	void
getAngleValue(int angle)	获取设备方位角 Obtain the device azimuth	angle:方位角 angle:Azimuth	void

14.10 RadarLocationEntity类

14.10 RadarLocationEntity Class

回调数据实体类型，用于存储雷达定位相关的标签数据，包括标签值、方位角、信号值和标签类型等信息。

Callback data entity type, used to store tag data related to radar positioning, including tag value, azimuth, signal value, tag type and other information.

方法 Method	描述 Description	参数 Parameters	返回值 Return
getTag()	获取标签值 Get the value of the label	无 N/A	标签值（字符串） The value of the label (String)
getAngle()	获取标签的方位角 Get the azimuth of the label	无 N/A	标签的方位角（整数） The azimuth of the label (Integer)
getValue()	获取标签的信号值 Get the signal value of the tag	无 N/A	标签的信号值（整数） The signal value of the tag (Integer)

方法 Method	描述 Description	参数 Parameters	返回值 Return
getUhfBank()	获取标签的类型 Get the type of label	无 N/A	标签类型（整数），参考{@link #IUHF.Bank_RESERVED}、{@link #IUHF.Bank_EPC}、{@link #IUHF.Bank_TID}、{@link #IUHF.Bank_USER} Tag type (Integer), refer to {@link #IUHF.Bank_RESERVED}, {@link #IUHF.Bank_EPC}, {@link #IUHF.Bank_TID}, {@link #IUHF.Bank_USER}
setValue(int value)	设置标签的信号值 Set the signal value of the label	value:标签的信号值 value:The signal value of the label	void
setAngle(int angle)	设置标签的方位角 Set the azimuth of the label	angle:标签的方位角 angle:The azimuth of the label	void
toString()	返回对象的字符串表示 Return the string representation of the object	无 N/A	包含所有属性信息的字符串 String containing all attribute information

14.11 RadarLocationEntity.Builder内部类

14.11 RadarLocationEntity.Builder Inner Class

标签实体构造器，用于构建RadarLocationEntity实例。

Constructor for label entities, used to build RadarLocationEntity instances.

方法 Method	描述 Description	参数 Parameters	返回值 Return
setValue(int value)	设置标签的信号值 Set the signal value of the label	value:标签的信号值 value:The signal value of the label	当前Builder实例 Current Builder instance
setAngle(int angle)	设置标签的方位角 Set the azimuth of the label	angle:标签的方位角 angle:The azimuth of the label	当前Builder实例 Current Builder instance
setUhfBank(int uhfBank)	设置标签的类型 Set the type of label	uhfBank:标签类型 uhfBank:The type of label	当前Builder实例 Current Builder instance
setTag(String tag)	设置标签值 Set the value of the label	tag:标签值 tag:The value of the label	当前Builder实例 Current Builder instance
build()	构建RadarLocationEntity实例 Build RadarLocationEntity instance	无 N/A	构建好的RadarLocationEntity实例 Built RadarLocationEntity instance

14.12 ConnectionStatusCallback接口

14.12 ConnectionStatusCallback Interface

连接状态回调接口，用于接收连接状态变化通知
Connection status callback interface, used to receive connection status change notifications

方法 Method	描述 Description	参数 Parameters	返回值 Return
getStatus(final ConnectionStatus status, final T device)	接收连接状态信息 Receive connection status information	status: 状态 status device: 目标设备 target device	无 N/A

14.13 ScanBTCallback接口

14.13 ScanBTCallback Interface

蓝牙扫描回调接口，用于接收扫描过程中获取的蓝牙设备信息、信号值及广播数据。
Bluetooth scanning callback interface, used to receive Bluetooth device information, signal values, and broadcast data obtained during the scanning process.

方法 Method	描述 Description	参数 Parameters	返回值 Return
getDevices(final BluetoothDevice device, final int rssi, final byte[] scanRecord)	接收蓝牙扫描到的设备 相关数据 Receive data related to scanned Bluetooth devices	device: 蓝牙设备对象，存储扫描到的蓝牙设备信息 device: Bluetooth device object, storing information of the scanned Bluetooth device rssi: 蓝牙设备的信号值（Received Signal Strength Indicator） rssi: Signal value of the Bluetooth device (Received Signal Strength Indicator) scanRecord: 广播和扫描应答数据的字节数组，存放设备广播及扫描反馈的原始数据 scanRecord: Byte array of broadcast and scanning response data, storing raw data of device broadcast and scanning feedback	无 N/A

14.14 KeyEventCallback接口

14.14 KeyEventCallback Interface

按键事件回调接口，用于接收按键按下和松开的事件通知
Key event callback interface, used to receive event notifications of key presses and releases

方法 Method	描述 Description	参数 Parameters	返回值 Return
onKeyDown(int keyCode)	接收按键按下事件通知 Receive key down event notification	keyCode: 按键值 keyCode: Key code	无 N/A

方法 Method	描述 Description	参数 Parameters	返回值 Return
onKeyUp(int keyCode)	接收按键松开事件通知 Receive key up event notification	keyCode: 按键值 keyCode: Key code	无 N/A

14.15 FilterEntity类

14.15 FilterEntity Class

标签过滤实体类，用于定义UHF标签盘点时的过滤规则，包括过滤的存储区、起始地址、数据长度及匹配的过滤数据等信息。

Tag filter entity class, used to define filter rules during UHF tag inventory, including the storage area, start address, data length of filtering, and matching filter data.

属性 Attribute	类型 Type	描述 Description
filterBank	int	标签存储区 Tag storage area 取值说明： 0x00: Reserve（保留区） 0x01: EPC（电子产品码区） 0x02: TID（标签识别码区） 0x03: USR（用户区）
filterOffset	int	过滤起始地址 Filter start address
filterLength	int	过滤数据长度 Filter data length 单位：bit（比特） 特殊值：0 = 无过滤（no filter）
filterData	String	过滤数据 Filter data 需与指定存储区、起始地址、长度匹配的标签数据