

Bluetooth back clip API description

One、Construction of development environment

- 1.1 Complete the construction of the development environment according to the jar files in the folder “required files for integrated development” in the development materials.
- 1.2 Create a new “myjar” folder under the Android project, copy sr_android.jar and sr_uhf.jar into the folder, right-click the project build path, configure the build path, and add jar package.
- 1.3 Clean up and refresh the project.

Two、Program flow description

Start Android program => scan Bluetooth device => connect Bluetooth device => connect and register service successfully => operate => end of operation => stop reading mark => close Bluetooth connection => exit program

Three、Development specification

3.1 Define the object

```
public static Reader Readercontroller;
```

3.2 Instantiate object

```
Readercontroller = new Reader("BlueTooth",this);
```

3.3 Register the callback,

```
Readercontroller.SetCallBack(this);
```

3.4 Turn on the timing of the heartbeat

```
ReaderController.SetHeartThread(true);
```

3.5 Implementation MultiLableCallBck interface

```
public class MainHandleActivity extends Activity
    implements MultiLableCallBack
```

Add unimplemented methods:

```
@Override
public void method(byte[] data)//Receive normal group read & group read tag data with password. See 5.1 for c
```

```
@Override
public void CmdRespond(String[] data)//Receive temperature label reading & with User group reading label data
```

```
@Override
public void BlueToothBtnNew(int state)//Bluetooth handle button callback interface (0: stop reading mark 1: s
```

```
@Override
public void BlueToothVoltageNew(int voltage, byte BAT_STATUS,
    byte WORK_STATUS, int Voltseg1, int Voltseg0)//Bluetooth handle voltage callback
//The handle battery status, the working status of the RFID module and the battery power information
```

```
@Override
```

```
@Override
    public void SdkActionResult(String ResultStr)//Bluetooth device connection result interface. See 5.3 for data
```

Note :Android 10 phones need to enable GPS permissions and enable GPS function to scan Bluetooth.

Android Bluetooth Permission

```
<uses-permission android:name="android.permission.BLUETOOTH" />
```

```
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

GPS Permission

```
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

Four、 Function description

4.1 Scan Bluetooth devices

Example of an interface function call

```
Readercontroller._bles.ScanTime = 6000;//Set scan time; The unit is ms
Readercontroller._bles.scanBle();//scan
```

- Readercontroller._bles.isScanFinsh. Indicates whether the scan is complete or not: True indicates that the scan is over and false indicates that the scan is in progress

Gets the currently scanned Bluetooth object

```
List<BleDevice> BleList = ReaderController._bles.GetBleList();
```

- The recommended interval between two scans of Bluetooth devices is $\geq 6s$

4.2 Stop scanning the Bluetooth device

```
ReaderController._bles.StopsCanBle();
```

4.3 Connecting Bluetooth devices

Example of an interface function call

```
Readercontroller._bles.ConnectDev(bleDevice);
```

4.4 Disconnect from Bluetooth device

Example of an interface function call

```
Readercontroller._bles.disconnect();
```

4.5 Get module version

Example of an interface function call

```
Ware mWare = new Ware(CommandType.GET_FIRMWARE_VERSION, 0, 0, 0);
Boolean ret = ReaderController.UHF_CMD(CommandType.GET_FIRMWARE_VERSION, mWare);
```

Returns a description of the value ret

- Is true: obtain success; False: Failed to obtain
If true, then the version is mWare.major_version + "." + mWare.minor_version + "." + mWare.revision_version

4.6 Start Mult Read Tags

4.6.1 Common Multi

Example of an interface function call

```
Multi_query_epc mMulti_query_epc = new Multi_query_epc();
mMulti_query_epc.query_total = 0;
ReaderController.UHF_CMD(CommandType.MULTI_QUERY_TAGS_EPC, mMulti_query_epc);
```

4.6.2 Specified password read

Example of an interface function call

```
Multi_query_epc mMulti_query_epc = new Multi_query_epc();
mMulti_query_epc.query_total = 0;
mMulti_query_epc.inventory_type = 0x06;
mMulti_query_epc.access_pwd = DemoConfig.AccessPwd;
ReaderController.UHF_CMD(CommandType.MULTI_QUERY_TAGS_EPC, mMulti_query_epc);
```

Parameter specification

Parameter names	Will choose	Type	Instructions
query_total	YES	int	for 0
inventory_type	YES	byte	Group reading mode; 0x06 is denoted as reading with password
access_pwd	YES	String	Tag access password

4.6.3 With the user multi

Example of an interface function call(User start address <= 255)

```
Multi_query_epc mMulti_query_epc = new Multi_query_epc();
mMulti_query_epc.query_total = 0;
mMulti_query_epc.inventory_type = 0x08;
mMulti_query_epc.access_pwd = DemoConfig.AccessPwd;
mMulti_query_epc.user_startAddr = (byte)DemoConfig.UserStartAddr;
mMulti_query_epc.user_Len = (byte)DemoConfig.UserLen;
ReaderController.UHF_CMD(CommandType.MULTI_QUERY_TAGS_EPC, mMulti_query_epc);
```

Example of an interface function call(User start address > 255)

```
Multi_query_epc mMulti_query_epc = new Multi_query_epc();
mMulti_query_epc.query_total = 0;
mMulti_query_epc.inventory_type = 0x09;
mMulti_query_epc.access_pwd = DemoConfig.AccessPwd;
mMulti_query_epc.Int_User_startAddr = DemoConfig.UserStartAddr;
mMulti_query_epc.user_Len = (byte)DemoConfig.UserLen;
ReaderController.UHF_CMD(CommandType.MULTI_QUERY_TAGS_EPC, mMulti_query_epc);
```

Parameter specification

Parameter names	Will choose	Type	Instructions
query_total	YES	int	for 0

Parameter names	Will choose	Type	Instructions
inventory_type	YES	byte	Group reading mode; 0x08 is denoted as read with User
access_pwd	YES	String	Tag access password
user_startAddr	YES	byte	The User data reads the starting address
Int_User_startAddr	YES	int	The User data reads the starting address
user_Len	YES	byte	Length of User data read

4.6.4 Read temperature tags

Example of an interface function call

```
Multi_TempTags mMulti_temptags = new Multi_TempTags();
mMulti_temptags.TempTagsType = 0x02;
ReaderController.UHF_CMD(CommandType.Read_TagTemp_Start, mMulti_temptags);
```

Parameter specification

Parameter names	Will choose	Type	Instructions
TempTagsType	YES	byte	Temperature label type; 0x01:LTU27, 0x02:LTU3x, (byte)0x20:RFM, (byte)0x30:NMV2D

4.7 Stop Mult Read Tags

4.7.1 Common Multi/Specified password read/With the user multi => STOP

Example of an interface function call

```
Boolean ret = ReaderController.UHF_CMD(CommandType.STOP_MULTI_QUERY_TAGS_EPC, null);
```

Returns a description of the value ret

- Is true: stop successful; False: Stop failure

4.7.2 Read temperature tags => STOP

Example of an interface function call

```
Multi_TempTags mMulti_temptags = new Multi_TempTags();
mMulti_temptags.TempTagsType = 0x02;
ReaderController.UHF_CMD(CommandType.Read_TagTemp_Stop, mMulti_temptags);
```

Parameter specification

Parameter names	Will choose	Type	Instructions
TempTagsType	YES	byte	Temperature label type; 0x01:LTU27, 0x02:LTU3x, (byte)0x20:RFM, (byte)0x30:NMV2D

Returns a description of the value ret

- Is true: stop successful; False: Stop failure

4.8 Get Power

Example of an interface function call

```
Power mPower = new Power();
mPower.com_type = CommandType.GET_POWER;
mPower.loop = 0;
mPower.read = 0;
mPower.write = 0;
Boolean ret = ReaderController.UHF_CMD(CommandType.GET_POWER, mPower);
```

Returns a description of the value ret

- Is true: obtain success; False: Failed to obtain
If true, the power is mPower.read

4.9 Set Power

Example of an interface function call

```
Power mPower = new Power();
mPower.com_type = CommandType.SET_POWER;
mPower.loop = 0;
mPower.read = ReadPower;
mPower.write = ReadPower;
Boolean ret = ReaderController.UHF_CMD(CommandType.SET_POWER, mPower);
```

Parameter specification

Parameter names	Will choose	Type	Instructions
ReadPower	YES	int	ReadPower for the power value you want to set. Values range from 5-30 to 5-25, depending on the type of device.

Returns a description of the value ret

- true: successful; false: Failed to set

4.10 Get RF Area

Example of an interface function call

```
Frequency_region mFrequency_region = new Frequency_region(CommandType.GET_FREQUENCY_REGION, 0, 0);
Boolean ret = MainActivity.ReaderController.UHF_CMD(CommandType.GET_FREQUENCY_REGION, mFrequency_region);
```

Returns a description of the value ret

- Is true: obtain success; False: Failed to obtain
If is true, frequency region is determined according to Mfrequency_regionally. Region: when is 1, it is China1; when is 2, it is China2; when is 3, it is Europe; when is 4, it is USA; when is 5, it is Korea; and when is 6, it is Japan.

4.11 Set RF Area

Example of an interface function call

```
Frequency_region mFrequency_region = new Frequency_region(CommandType.SET_FREQUENCY_REGION, int_save,int_region_t
Boolean ret = MainActivity.ReaderController.UHF_CMD(CommandType.SET_FREQUENCY_REGION, mFrequency_region);
```

--

Parameter specification

Parameter names	Will choose	Type	Instructions
int_save	YES	int	Save the mark; When it is 0, it is saved for continuous power; when it is 1, it is saved for power failure
int_region_tmp	YES	int	RF Area; Region: when is 1, it is China1; when is 2, it is China2; when is 3, it is Europe; when is 4, it is USA; when is 5, it is Korea; and when is 6, it is Japan.

Returns a description of the value ret

- true: successful; false: Failed to set

4.12 Get RF Link

Example of an interface function call

```
RfLink mRfLink = new RfLink();
mRfLink.com_type = CommandType.GET_RF_LINK;
mRfLink.rflink_Type = 0;
Boolean ret = MainActivity.ReaderController.UHF_CMD(CommandType.GET_RF_LINK, mRfLink);
```

Returns a description of the value ret

- Is true: obtain success; False: Failed to obtain
If true, the RF link is determined according to mRfLink.rflink_type; when 0, it is DENoted as DSB/ fm0/40khz; when 1, it is denoted as PR/ m4/250khz; when 2, it is denoted as PR/ m4/300khz; when 3, it is denoted as DSB/ fm0/400khz.

4.13 Set RF Link

Example of an interface function call

```
RfLink mRfLink = new RfLink();
mRfLink.com_type = CommandType.SET_RF_LINK;
mRfLink.save = int_save;
mRfLink.rflink_Type = int_rflink_temp;
Boolean ret = MainActivity.ReaderController.UHF_CMD(CommandType.SET_RF_LINK, mRfLink);
```

Parameter specification

Parameter names	Will choose	Type	Instructions
int_save	YES	int	Save the mark; When it is 0, it is saved for continuous power; when it is 1, it is saved for power failure

Parameter names	Will choose	Type	Instructions
int_rflink_temp	YES	int	RF Link; when 0, it is DENOted as DSB/fm0/40khz; when 1, it is denoted as PR/m4/250khz; when 2, it is denoted as PR/m4/300khz; when 3, it is denoted as DSB/fm0/400khz.

Returns a description of the value ret

- true: successful; false: Failed to set

4.14 Get working time and interval time

Example of an interface function call

```
Multi_interval mMulti_interval = new Multi_interval();
mMulti_interval.com_type =CommandType.GET_MULTI_QUERY_TAGS_INTERVAL;
mMulti_interval.work_time = 0;
mMulti_interval.interval = 0;
Boolean ret = MainActivity.ReaderController.UHF_CMD(CommandType.GET_MULTI_QUERY_TAGS_INTERVAL,mMulti_interval);
```

Returns a description of the value ret

- Is true: obtain success; False: Failed to obtain
If true, the working time is mMulti_interval. Work_time, and the interval is mMulti_interval. Interval.

4.15 Set working time and interval time

Example of an interface function call

```
Multi_interval mMulti_interval = new Multi_interval();
mMulti_interval.com_type = CommandType.SET_MULTI_QUERY_TAGS_INTERVAL;
mMulti_interval.work_time = int_worktime_temp;
mMulti_interval.interval = int_interval_temp;
Boolean ret = MainActivity.ReaderController.UHF_CMD(CommandType.SET_MULTI_QUERY_TAGS_INTERVAL,mMulti_interval);
```

Parameter specification

Parameter names	Will choose	Type	Instructions
int_worktime_temp	YES	int	Working time; Values range from 0 to 65535.
int_interval_temp	YES	int	Interval time; Values range from 0 to 65535.

Returns a description of the value ret

- true: successful; false: Failed to set

4.16 Get EPCAndTid State

Example of an interface function call

```
EPCandTID mepcandtid = new EPCandTID();
mepcandtid.com_type = CommandType.GET_EPCandTID;
mepcandtid.state = 0;
Boolean ret = MainActivity.ReaderController.UHF_CMD(CommandType.GET_EPCandTID, mepcandtid);
```

Returns a description of the value ret

- Is true: obtain success; False: Failed to obtain
If true, then read both EPC and TID state according to the value of mepcandtid.state. A value of 0 indicates a closed state, and a value of 1 indicates an open state.

4.17 Set EPCAndTid State

Example of an interface function call

```
EPCAndTID mepcandtid = new EPCAndTID();
mepcandtid.com_type = CommandType.SET_EPCAndTID;
mepcandtid.state = int_startq_temp;
Boolean ret = MainActivity.ReaderController.UHF_CMD(CommandType.SET_EPCAndTID, mepcandtid);
```

Parameter specification

Parameter names	Will choose	Type	Instructions
int_startq_temp	YES	int	Read the functional status of EPC and TID simultaneously; A value of 0 is closed, and a value of 1 is on

Returns a description of the value ret

- true: successful; false: Failed to set

4.18 Get 9200 Moudle/-L Module Read Tags Rssi State

Example of an interface function call

```
ModuleConfig moduleConfig = new ModuleConfig();
moduleConfig.com_type = CommandType.MODULE_CONFIG;
moduleConfig.subCmd = 0x01;
moduleConfig.funcCmd = 0x01;
moduleConfig.rssiState = 0x00;
Boolean ret = MainActivity.ReaderController.UHF_CMD(CommandType.MODULE_CONFIG, moduleConfig);
```

Parameter specification

Parameter names	Will choose	Type	Instructions
subCmd	YES	byte	Command type; 0x01 is the 9200Rssi parameter
funcCmd	YES	byte	Operation type; 0x01 is for get and 0x00 is for set
rssiState	YES	byte	Open status of 9200Rssi; The initial value 0x00 is assigned when getting

Returns a description of the value ret

- Is true: obtain success; False: Failed to obtain
If true, the Rssi status of 9200 is determined by the value of moduleConfig. RssiState. A value of (byte)0x55 is indicated as closed, and a value of (byte)0xAA is indicated as open.

4.19 Set 9200 Moudle/-L Module Read Tags Rssi State

Example of an interface function call


```
ModuleConfig moduleConfig = new ModuleConfig();
moduleConfig.com_type = CommandType.MODULE_CONFIG;
moduleConfig.subCmd = 0x01;
moduleConfig.funcCmd = 0x00;
moduleConfig.rssiState = rssiState;
Boolean ret = MainActivity.ReaderController.UHF_CMD(CommandType.MODULE_CONFIG, moduleConfig);
```

Parameter specification

Parameter names	Will choose	Type	Instructions
subCmd	YES	byte	Command type; 0x01 is the 9200Rssi parameter
funcCmd	YES	byte	Operation type; 0x01 is for get and 0x00 is for set
rssiState	YES	byte	Open status of 9200Rssi; A value of (byte)0x55 means closed, and a value of (byte)0xAA means open

Returns a description of the value ret

- true: successful; false: Failed to set

4.20 Read Tags Data

Example of an interface function call

```
Tags_data mTags_data = new Tags_data();
mTags_data.password = pwd;
mTags_data.FMB = int_filter;
mTags_data.filterData_len = data.length;
mTags_data.filterData = filterdata;
mTags_data.start_addr = int_offset;
mTags_data.data_len = int_length;
mTags_data.mem_bank = int_bank;
Boolean ret = MainActivity.ReaderController.UHF_CMD(CommandType.READ_TAGS_DATA, mTags_data);
```

Parameter specification

Parameter names	Will choose	Type	Instructions
pwd	YES	String	Access password; The default value is "00000000"
int_filter	YES	int	Filter/Specify type; A value of 0 represents filtering/specifying EPC, and a value of 1 represents filtering/specifying TID
data.length	YES	int	Filter/Specify data byte length; If there is no filtering, the value is 0
filterdata	NO	byte[]	Filter/specify data; If unfiltered, no assignment is required for this value

Parameter names	Will choose	Type	Instructions
int_offset	YES	int	Read data area starting address; In words (1 word =2 bytes)
int_length	YES	int	Length of reading data area; In words (1 word =2 bytes)
int_bank	YES	int	Read data area type; 0 is RFU region, 1 is EPC region, 2 is TID region, and 3 is USER region

Returns a description of the value ret

- Is true: read successfully; False: Failed to read
If true, then mTags_data.data is read data

4.21 Write Tags Data

Example of an interface function call

```
Tags_data mTags_data = new Tags_data();
mTags_data.password = psd;
mTags_data.FMB = int_filter;
mTags_data.filterData_len = data.length;
mTags_data.filterData = filterdata;
mTags_data.start_addr = int_offset;
mTags_data.data_len = int_length;
mTags_data.mem_bank = int_bank;
mTags_data.data = data;
Boolean ret = MainActivity.ReaderController.UHF_CMD(CommandType.WRITE_TAGS_DATA, mTags_data);
```

Parameter specification

Parameter names	Will choose	Type	Instructions
pwd	YES	String	Access password; The default value is "00000000"
int_filter	YES	int	Filter/Specify type; A value of 0 represents filtering/specifying EPC, and a value of 1 represents filtering/specifying TID
data.length	YES	int	Filter/Specify data byte length; If there is no filtering, the value is 0
filterdata	NO	byte[]	Filter/specify data; If unfiltered, no assignment is required for this value
int_offset	YES	int	Read data area starting address; In words (1 word =2 bytes)
int_length	YES	int	Length of reading data area; In words (1 word =2 bytes)

Parameter names	Will choose	Type	Instructions
int_bank	YES	int	Read data area type; 0 is RFU region, 1 is EPC region, 2 is TID region, and 3 is USER region
data	YES	byte[]	Write data

Returns a description of the value ret

- Is true: successful write; False: Failed to write

Five、Callback interface function parsing instructions

5.1 Callback interface function method data parse example

```
byte msb = data[0];
byte lsb = data[1];
int pc = (msb & 0x00ff) << 8 | (lsb & 0x00ff);
pc = (pc & 0xf800) >> 11;

byte[] epc = new byte[pc * 2];
System.arraycopy(data, 2, epc, 0, epc.length);
String str_epc = ShareData.CharToString(epc, epc.length);
str_epc = str_epc.replace(" ", "");

byte[] tid = new byte[data.length - 2 - (pc * 2) - 3];
System.arraycopy(data, 2 + (pc * 2), tid, 0, tid.length);
String str_tid = ShareData.CharToString(tid, tid.length);
str_tid = str_tid.replace(" ", "");

String str_rssi = "" + (~((short) (((data[2 + pc * 2 + tid.length] & 0xFF) << 8)
    | (data[2 + pc * 2 + 1 + tid.length] & 0xFF) - 1)) / -10.0);
```

5.2 Callback interface function CmdRespond data parsing instructions

Value	Instructions
String[0]	Reserve
String[1]	Cmd command type;" F7" is LTU27 temperature label data, "F8" is LTU3x/RFM/NMV2D temperature label data, and "79" is tag data read with User group
String[2]	Tags the EPC
String[3]	Tags the Rssi
String[4]	Tags the temperature
String[5]	Temperature label type
String[6]	Tags the TID

5.3 Callback interface function SDKActionResult data parsing description

Start by converting String to String[]

```
String[] result = ResultStr.split("\\\\,");
```

Value	Instructions
-------	--------------

Value	Instructions
String[0]	Message type; The value is BluetoothTypes.BlueToothMsgType said this message for bluetooth connection results related news
String[1]	Message content; The value of BLUETOOTHTYPES.CONNECT_SUCCESS indicates that the connection was successful. The value of BLUETOOTHTYPES.NOTIFY_SUCCESS indicates that the registration of the transceiver service was successful. Notify_fail for BLUETOOTHTYPES.NOTIFY_FAIL to register the transceiver service

- After the execution of the connection action, if the connection message is received successfully and the receipt and dispatch service is registered successfully, the connection communication is OK and the operation can be started. If the connection is successful and the registered transceiver service fails, it means that the connection communication is abnormal. The application layer should prompt the user to wait for about 10s before connecting again.

API Ver : 0.1.0

API Compile Time : 2021-06-24