Bluetooth ESL API Introduction
V1.0.6

Dalian Sertag Technology Co., Ltd

Content

1. Overview

This Bluetooth tag SDK involves three classes and two interfaces. The following describes each interface call in detail through different chapters.

2. Calling class

BluetoothLEManager Bluetooth tag search management class
BluetoothTransferClient Bluetooth tag screen update client class
BluetoothLEDevice Bluetooth tag device class

2.1. Introduction of BluetoothLEManager Class

2.1.1. Constructed Prototype

```
public static BluetoothLEManager getInstance(Context c);
```

Reinitialize routines

```
BluetoothLEManager bluetoothLEManager = BluetoothLEManager.getInstance(this);
        bluetoothLEManager.prepareForScanForDevices(this);
        bluetoothLEManager.setBluetoothLEManagerCB(new
BluetoothLEManager.BluetoothLEManagerCB() {
            @Override
            public void deviceFound(BluetoothLEDevice bluetoothLEDevice) {}

            @Override
            public void deviceFound(List<BluetoothLEDevice> list) { }
});
```

2.1.2. Start Searching

```
bluetoothLEManager.scanForDevices();
```

2.1.3. Stop Searching

```
bluetoothLEManager.stopScan();
```

2.2. Introduction of BluetoothTransferClient Class

2.2.1. Constructed Prototype

```
Public BluetoothTransferClient(Context context, BluetoothTransferClientProgressCallback callback);
```

2.2.2. Reinitialize routines

```
BluetoothTransferClientProgressCallback clientProgressCallback = new
BluetoothTransferClientProgressCallback() {
        @Override
        public void progressUpdate(String address, float percent, int currentBlock) {}

        @Override
        public void statusUpdate(String address,
BluetoothTransferDefinitions.StatusEnumeration status) {}
};
BluetoothTransferClient bluetoothTransferClient = new BluetoothTransferClient(this,
clientProgressCallback);
```

2.2.3. Prototype of Update picture function

```
public void start(String address, List<Bitmap> bitmaps, int type) throws IOException, InterruptedException
```

1)Parameter descripton

#	Parameter	Type	Description
1	address	String	Tag MAC address
2	bitmaps	Bitmap	Imag
3	type	Integer	Tag type，get it with BluetoothLEDevice function

2)Call routine

```
try {
    int type = device.device.getDeviceType();

    InputStream in = getAssets().open(png);
    Bitmap bitmap = BitmapFactory.decodeStream(in);

    bluetoothTransferClient.start(device.device.getAddress(), bitmap, type );
```

```
} catch (IOException e) {

} catch (InterruptedException e) {

}
```

2.3. Introduction of BluetoothLEDevice Class

2.3.1. Get Device Name

```
public String getName();
```

2.3.2. Get Device Address

```
public String getAddress();
```

2.3.3. Get Tag Device Type

```
public int getDeviceType();
```

2.3.4. Get the battery voltage of the tag device

```
public int getPowerLevel();
```

2.3.5. Get the signal intensity of the tag device

```
public int getRssi()
```

3. Interface call

BluetoothLEManager.BluetoothLEManagerCB Bluetooth search callback function
BluetoothTransferClientProgressCallback Status callback during Bluetooth tag screen update

3.1. Callback function of Bluetooth searching device

3.1.1. Interface prototype

```
public interface BluetoothLEManagerCB {

    void deviceFound(BluetoothLEDevice device);

    void deviceFound(List<BluetoothLEDevice> devices);
}
```

3.2. Picture data update callback function

3.2.1. Interface prototype

```
public interface BluetoothTransferClientProgressCallback {
    void progressUpdate(String address, float percent, int currentBlock);
    void statusUpdate(String address, BluetoothTransferDefinitions.StatusEnumeration status);
}
```

3.2.1.1. statusUpdate Status callback function

1) Parameter Description

#	Parameter	Type	Description
1	address	String	Tag MAC address
2	status	BluetoothTransferDefinitions.StatusEnumeration	Current status

2) BluetoothTransferDefinitions.StatusEnumeration type and description

#	BluetoothTransferDefinitions.StatusEnumeration	value	Description
1	ClientInitializing	6	initialization
2	ClientDeviceReady	9	Preparing for connection establishment
3	ClientDeviceDiscoverTimeout	3	Discovery service timeout failed
4	ClientDeviceConnectTimeout	0	Connection establishment timeout
5	ClientServiceMissingOnPeripheral	8	The UUID service could not be found, resulting in failure
6	ClientReady	10	Connection established

			successfully, ready to update pictures
7	ClientCompleteDeviceDisconnectedPositive	30	Normally disconnected from connected peripherals
8	ClientGotEnableUpdateScreenResponseFail	19	Failed to enable update of picture data
9	ClientProcessStartCommandSentFail	21	The command to start transferring picture data failed
10	ClientImageTransfer	22	During picture data transmission
11	ClientImageTransferFailed	23	Picture data transmission process failed
12	ClientProgrammingAbortedByUser	32	Picture data transmission is interrupted by the user, failed
13	ClientCompleteFeedbackOK	25	Picture data transmission completed
14	ClientCompleteDeviceDisconnectedDuringProgramming	31	Picture data transmission failed

3.2.1.2. progressUpdate Progress callback of picture data transmission

parameter description

#	Parameter	Type	Description
1	address	String	ag MAC address
2	percent	float	Percentage of data sent currently（0-100）
3	currentBlock	Int	Serial number of the currently sent package（0 1 2...n）