**Indoor Positioning system**

**FastBle**

(https://github.com/FlyAndNotDown/BluetoothLocation)

Android Bluetooth Low Energy

* Filtering, scanning, linking, reading, writing, notification subscription and cancellation in a simple way.
* Supports acquiring signal strength and setting the maximum transmission unit.
* Support custom scan rules
* Support multi device connections
* Support reconnection
* Support configuration timeout for conncet or operation

# Usage

#### Init

* BleManager.getInstance().init(getApplication());

#### Determine whether the current Android system supports BLE

* boolean isSupportBle()

#### Open or close Bluetooth

* void enableBluetooth()
* void disableBluetooth()

#### Initialization configuration

* BleManager.getInstance()
* .enableLog(true)
* .setReConnectCount(1, 5000)
* .setSplitWriteNum(20)
* .setConnectOverTime(10000)
* .setOperateTimeout(5000);

#### Configuration scan rules

void initScanRule(BleScanRuleConfig scanRuleConfig)

BleScanRuleConfig scanRuleConfig = new BleScanRuleConfig.Builder()

.setServiceUuids(serviceUuids)

.setDeviceName(true, names)

.setDeviceMac(mac)

.setAutoConnect(isAutoConnect)

.setScanTimeOut(10000)

.build();

BleManager.getInstance().initScanRule(scanRuleConfig);

Tips：

* + Before scanning the device, scan rules can be configured to filter out the equipment matching the program.
  + What is not configured is the default parameter

#### Scan

void scan(BleScanCallback callback)

BleManager.getInstance().scan(new BleScanCallback() {

@Override

public void onScanStarted(boolean success) {

}

@Override

public void onScanning(BleDevice bleDevice) {

}

@Override

public void onScanFinished(List<BleDevice> scanResultList) {

}

});

Tips:

* + The scanning and filtering process is carried out in the worker thread, so it will not affect the UI operation of the main thread. Eventually, every callback result will return to the main thread.。

#### Connect with device

BluetoothGatt connect(BleDevice bleDevice, BleGattCallback bleGattCallback)

BleManager.getInstance().connect(bleDevice, new BleGattCallback() {

@Override

public void onStartConnect() {

}

@Override

public void onConnectFail(BleDevice bleDevice, BleException exception) {

}

@Override

public void onConnectSuccess(BleDevice bleDevice, BluetoothGatt gatt, int status) {

}

@Override

public void onDisConnected(boolean isActiveDisConnected, BleDevice bleDevice, BluetoothGatt gatt, int status) {

}

});

**Kiba Library(**https://github.com/KiBa1215/CoordinateAxisChart**)**

CoordinateAxisChart

* A simple Math coordinate system in Android.
* Using this library to draw function lines and points to a coordinate system.

## Add a point

SinglePoint point = new SinglePoint(new PointF(1f, 2f));

point.setPointColor(Color.RED);

coordinateAxisChart.addPoint(point);

coordinateAxisChart.invalidate();

## Reset

coordinateAxisChart.reset();

coordinateAxisChart.invalidate();

**Overall Followed code Reference:**

<https://github.com/Triple-Z/Indoor-Positioning-DEMO>