

# 《嵌入式综合设计项目》

# “Loc”软件文档

项目名称: “Loc”

**学号.姓名**  2012101010 李晗静

**学号.姓名**  2012101027 李鑫

**任课教师** 文斌

**评分**

2015年5 月31 日

目录

[《嵌入式综合设计项目》 1](#_Toc420876345)

[一·软件简介 2](#_Toc420876347)

[二·软硬件环境说明 2](#_Toc420876348)

[三·软件架构设计 2](#_Toc420876349)

[1.定位模块 2](#_Toc420876350)

[2.网络模块 3](#_Toc420876351)

[3.UI模块 5](#_Toc420876352)

[四·软件测试 6](#_Toc420876353)

# 一·软件简介

Loc是一款手机定位软件，双方同时使用该软件可以实时地获取对方的位置信息并在地图上显示，方便快捷地获知对方位置的实时变化情况。

# 二·软硬件环境说明

1. **cpu：高通**MSM7627（核心频率600MHz）以上（推荐MSM8225以上）
2. **内存：50MB以上（推荐512MB以上）**
3. **操作系统：Android2.2以上（推荐Android4.0以上）**
4. **网络：网络链接正常**

# 三·软件架构设计

本软件主要分为定位模块，网络模块和UI交互模块。

## 1.定位模块

该模块用于获取自己的位置信息，调用百度地图提供的SDK的API结合手机方向传感器来进行定位，获取定位经纬度。

**public** **void** onReceiveLocation(BDLocation location) {

**if** (location == **null** || mMapView == **null**) {

System.*out*.println("null");

**return**;

}

// 获取自己的定位信息

mLatitude = location.getLatitude();

mLongitude = location.getLongitude();

mRadius = location.getRadius();

## 2.网络模块

该模块使用百度云存储提供的网络接口，从百度云端获取对方的位置信息并将自己的位置信息上传到云端供对方查询 。网络操作在子线程中异步执行，完成后将数据通过回调函数handleMessage（）返回UI主线程。

Thread requestThread = **new** Thread() {

**public** **void** run() {

**try** {

// 根据过滤选项拼接请求URL

String requestURL = "";

**if** (searchType == *SEARCH\_TYPE\_CREATE*) {

requestURL = *URI\_CREATE*;

} **else** **if** (searchType == *SEARCH\_TYPE\_QUERY*) {

requestURL = *URI\_QUERY*;

} **else** **if** (searchType == *SEARCH\_TYPE\_UPDATE*) {

requestURL = *URI\_UPDATE*;

}

requestURL = requestURL + "ak=" + *ak* + "&geotable\_id="

+ *geotable\_id*;

String filter = **null**;

Iterator iter = filterParams.entrySet().iterator();

**new** BasicNameValuePair("str", "I am Post String");

List<NameValuePair> params = **new** ArrayList<NameValuePair>();

**while** (iter.hasNext()) {

Map.Entry entry = (Map.Entry) iter.next();

String key = entry.getKey().toString();

String value = entry.getValue().toString();

// 如需要POST请求则可以用此处的params设置参数

params.add(**new** BasicNameValuePair(key, value));

requestURL = requestURL + "&" + key + "=" + value;

}

Log.*e*("LocationShare", "request url:" + requestURL);

// 根据请求类型判断使用GET或POST

HttpClient httpclient = **new** DefaultHttpClient();

HttpResponse httpResponse = **null**;

**if** (searchType == *SEARCH\_TYPE\_QUERY*) {

HttpGet httpGet = **new** HttpGet(requestURL);

httpResponse = httpclient.execute(httpGet);

Log.*e*("LocationShare", "GET Response is ok!");

} **else** {

// 创建POST请求并绑定参数

params.add(**new** BasicNameValuePair("ak",

LBSCloudSearch.*ak*));

params.add(**new** BasicNameValuePair("geotable\_id",

LBSCloudSearch.*geotable\_id*));

HttpPost httpPost = **new** HttpPost(requestURL);

httpPost.setEntity(**new** UrlEncodedFormEntity(params,

HTTP.*UTF\_8*));

httpResponse = httpclient.execute(httpPost);

**if** (searchType == LBSCloudSearch.*SEARCH\_TYPE\_CREATE*) {

Log.*e*("LocationShareCreate",

"------------>Create--->"

+ httpResponse.getStatusLine()

.getStatusCode());

}

}

**int** status = httpResponse.getStatusLine().getStatusCode();

**if** (status == HttpStatus.*SC\_OK*) {

String result = EntityUtils.*toString*(

httpResponse.getEntity(), "utf-8");

// Header a = httpResponse.getEntity().getContentType();

// Message msgTmp = handler

// .obtainMessage(MainActivity.MSG\_NET\_SUCC);

// msgTmp.obj = result;

// msgTmp.sendToTarget();

Message msg = **new** Message();

msg.what = MainActivity.*MSG\_NET\_SUCC*;

msg.obj = result;

handler.sendMessage(msg);

} **else** {

// httpRequest.abort();

Message msgTmp = handler

.obtainMessage(MainActivity.*MSG\_NET\_STATUS\_ERROR*);

msgTmp.obj = "HttpStatus error";

msgTmp.sendToTarget();

}

} **catch** (Exception e) {

Log.*e*("LocationShare", "网络异常，请检查网络后重试！");

e.printStackTrace();

}

}

};

## 3.UI模块

该模块使用百度地图SDK提供的mapView，将从云端获取的对方位置信息实时显示在地图上提供给用户查看获知对方目前的位置，在网络请求模块的回调函数中获取位置信息并更新界面。

// 地图初始化

mMapView = (MapView) findViewById(R.id.*bmapView*);

mBaiduMap = mMapView.getMap();

// 开启定位图层

mBaiduMap.setMyLocationEnabled(**true**);

// 设置定位图层的配置（定位模式，是否允许方向信息，用户自定义定位图标）

mCurrentMode = MyLocationConfiguration.LocationMode.*NORMAL*;

MyLocationConfiguration config = **new** MyLocationConfiguration(

mCurrentMode, **true**, **null**);

mBaiduMap.setMyLocationConfigeration(config);

myListener = **new** MyLocationListener();

mLocationClient = **new** LocationClient(getApplicationContext()); // 声明LocationClient类

mLocationClient.registerLocationListener(myListener); // 注册监听函数

// 设置LocationClient的option

LocationClientOption option = **new** LocationClientOption();

option.setLocationMode(LocationMode.*Hight\_Accuracy*);// 设置定位模式

option.setCoorType("bd09ll");// 返回的定位结果是百度经纬度,默认值gcj02

option.setScanSpan(5000);// 设置发起定位请求的间隔时间为5000ms

option.setIsNeedAddress(**true**);// 返回的定位结果包含地址信息

option.setNeedDeviceDirect(**true**);// 返回的定位结果包含手机机头的方向

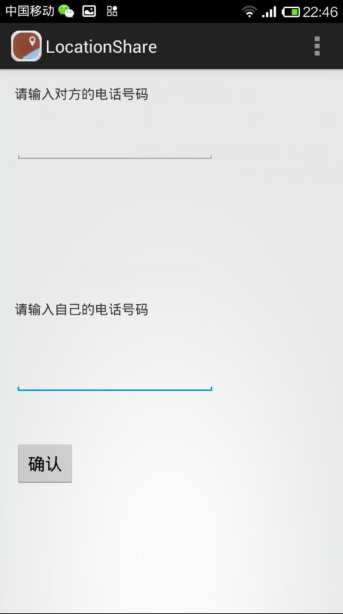
option.setOpenGps(**true**);// 打开gps

mLocationClient.setLocOption(option);

# 四·软件测试

（测试地点：组员所在宿舍）

输入双方的电话。



己方手机界面显示



对方的手机界面显示

