

物联网导论作业

姓名：黄翔
班级：软件71
学号：2017013570

1. 基于蓝牙、WiFi 定位

基本信息 为实现基于蓝牙、WiFi 定位，或者其他类似的无线定位，首先我们必须明晰无线定位利用的基本信息：**接受信号强度 (RSSI)**。接受信号强度的衰减量可以正相关地反映空间距离，通过适当的模型分析与实际测试我们可以量化二者之间的关系，从而实现从 RSSI 到距离的精准映射。

基本设备 为获取 RSSI，我们需要设置一些**访问接入点 (AP)**。通过 AP 获取 RSSI、进行分析，凭借 AP 之间的空间位置及分析数据最终定位目标。此外，发出信号的**移动设备**（例如手机）也不可少。这些移动设备将不断发出信号，并被 AP 接受分析。

基本方法 有了之前的基础，我们便能构建出基于蓝牙、WiFi 定位基本方法的全貌了。首先，当开启定位功能后，移动设备需要主动发出信号，并被 AP 接受。AP 借此获取 RSSI，并分析出到移动设备的距离范围。多个 AP 共同协作，通过**三边测量法**，最终即可得到用户的位置范围。

2. GPS 定位

a. 基本原理：GPS 定位依靠 4 个（或 4 个以上）的卫星同时协作。对于各个卫星，它们位置已知，并不断地播报自己所在的位置坐标，附带发送信息时的时间戳。

定位目标携带的 GPS 接受器能够接受卫星播报的信息，通过自己接受信息时的时间戳与信息中卫星发送信息时的时间戳之差，结合信息的传播数据，便能得到自己距离卫星的位置。凭借到 4 个（或 4 个以上）的卫星的距离，结合立体几何知识并能求解出自己的位置坐标。。

存在问题：GPS 定位严重依赖于数量有限的卫星，外部因素（包括人为因素）及内部因素对卫星的状况带来的改变将为 GPS 定位的精度与可靠性带来影响。

同时，定位信号穿透性差，到达地面时通常较弱，GPS 接收器成本也较为高昂。

此外，从高度的角度而言，GPS 定位仅能得到目标的大地高 h 。要想得到目标的正高 H ，需利用公式 $H = h - N$ 获取。其中大地水准面高度 N 各地不一且难以精确测定。这就导致了目标的正高测量不准。

b. 实现步骤：

此程序实现思路主要参考了博客 <https://iot-book.github.io/> 。

1. 实现系统定位服务的调用。程序通过 Android 提供的服务管理类 `LocationManager` 获取当前的位置信息。为方便测试，在程序 demo 中添加按钮与文本框，通过点击按钮主动调用 `getLastKnownLocation()` 方法，并更新获得的位置信息于文本框内。

2. 实现位置更新的监听。为确保位置信息的实时自动更新，而非主动调用更新，采用 `LocationListener` 接口注册监听。此时，程序便能实时自动更新位置。

3. 实现百度地图的显示。参考博客的提供的思路，在获取百度地图移动版开发密钥（AK）后，依次配置 `AndroidManifest.xml` 文件、在布局文件中添加地图容器、进行地图初始化、创建地图 Activity，管理 `MapView` 生命周期。此时，已可在应用中显示地图控件。

4. 实现实时位置在地图上的显示。借助 BaiduMap 类即可在指定位置显示定位图标，从而实现实时位置在地图上的显示。

关键代码：

位置监听接口

```
private LocationListener locationListener = new LocationListener() {
    /**
     * 位置信息变化时触发:当坐标改变时触发此函数，如果 Provider 传进相同的坐标，它就不会被触发
     * @param location
     */
    @Override
    public void onLocationChanged(Location location) {
        Log.i(TAG, "时间: " + location.getTime());
        Log.i(TAG, "经度: " + location.getLongitude());
        Log.i(TAG, "纬度: " + location.getLatitude());
        Log.i(TAG, "海拔: " + location.getAltitude());
        tx.setText("当前经度: "+location.getLongitude()+"\n"+"当前纬度: "+location.getLatitude());
        // 地图控件管理对象
        BaiduMap mBaiduMap = mMapView.getMap();
        // 开启地图控件的定位功能
        mBaiduMap.setMyLocationEnabled(true);
        // 获取经纬度及海拔高度信息，存入百度地图指定的坐标信息类 MyLocationData
        MyLocationData locData = new MyLocationData.Builder()
            .accuracy(location.getAccuracy())
            .latitude(location.getLatitude())
            .longitude(location.getLongitude()).build();
        // 在地图上显示定位图标
        mBaiduMap.setMyLocationData(locData);
        Log.v("pcw","setUserMapCenter : lat : "+ location.getLatitude()+" lon : " + location.getLongitude());
        LatLng cenpt = new LatLng(location.getLatitude(),location.getLongitude());
        //定义地图状态
        MapStatus mMapStatus = new MapStatus.Builder()
            .zoom(18)
            .target(cenpt)
            .build();
        //定义 MapStatusUpdate 对象，以便描述地图状态将要发生的变化
```

```

        MapStatusUpdate mMapStatusUpdate = MapStatusUpdateFactory.newMapStatus(mMapStatus)
;

        //改变地图状态
        mBaiduMap.setMapStatus(mMapStatusUpdate);
    }

```

地图创建与显示

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    //获取地图控件引用
    mMapView = (MapView) findViewById(R.id.bmapView);
    tx = findViewById(R.id.textView);
    locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
    // 运行时权限管理
    if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION)
        != PackageManager.PERMISSION_GRANTED
        && ActivityCompat.checkSelfPermission(MainActivity.this, Manifest.permission.A
ACCESS_COARSE_LOCATION)
        != PackageManager.PERMISSION_GRANTED)
    {
        ActivityCompat.requestPermissions(MainActivity.this, MULTI_PERMISSIONS, 100);
    }
    location = locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER); // 通过
GPS 获取位置
    // 注册监听，当时间为间隔 100 毫秒或位移变化超过 0 米时更新
    locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 100, 0, locationL
istener);
    BaiduMap mBaiduMap = mMapView.getMap();
    // 开启地图控件的定位功能
    mBaiduMap.setMyLocationEnabled(true);
    // 获取经纬度及海拔高度信息，存入百度地图指定的坐标信息类 MyLocationData
    MyLocationData locData = new MyLocationData.Builder()
        .accuracy(location.getAccuracy())
        .latitude(location.getLatitude())
        .longitude(location.getLongitude()).build();
    // 在地图上显示定位图标
    mBaiduMap.setMyLocationData(locData);
    Log.v("pcw", "setUserMapCenter : lat : " + location.getLatitude() + " lon : " + location.g
etLongitude());
    LatLng cenpt = new LatLng(location.getLatitude(), location.getLongitude());

```

```

//定义地图状态
MapStatus mMapStatus = new MapStatus.Builder()
    .zoom(18)
    .target(cenpt)
    .build();

//定义 MapStatusUpdate 对象，以便描述地图状态将要发生的变化
MapStatusUpdate mMapStatusUpdate = MapStatusUpdateFactory.newMapStatus(mMapStatus);
//改变地图状态
mBaiduMap.setMapStatus(mMapStatusUpdate);
}

```

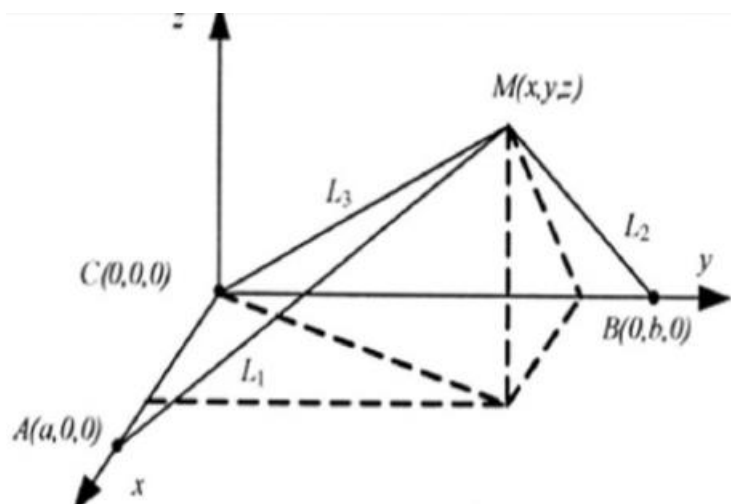
效果展示:



3. 超声波定位

距离获取 手机可通过信号频率分辨扬声器，并借由接受信号的时间间隔算出距离各个扬声器的距离。(利用波速乘以时间)

距离交会法 之后，可利用距离交会法算出手机的空间坐标位置。简单起见，我们假设三个扬声器的空间坐标分别为 $A(a, 0, 0)$, $B(0, b, 0)$, $C(0, 0, 0)$ ，并建立空间坐标系，如下：



(图片引自网络)

假设计算得到手机 (M) 离 A、B、C 点距离分别为 L_1 、 L_2 、 L_3 ，其坐标为(x, y, z)。则可以得到如下方程：

$$\begin{cases} L_1^2 = (x - a)^2 + y^2 + z^2 \\ L_2^2 = x^2 + (y - b)^2 + z^2 \\ L_3^2 = x^2 + y^2 + z^2 \end{cases}$$

解得：

$$\begin{cases} x = \frac{L_3^2 - L_1^2 + a^2}{2a} \\ y = \frac{L_3^2 - L_2^2 + b^2}{2b} \\ z = \sqrt{L_3^2 - x^2 - y^2} \end{cases}$$

此即为手机的空间坐标位置。