About This Article

The GPS functionality has become highly popular, but on device it has some constrain it does not work indoors and not all devices having built-in GPS receiver. This document describes how to get the latitude and longitude for non GPS can cell phone using Open Cell-ID, followed by sample code.

Scope:

This document gives introduction about retrieving latitude and longitude on Android cell phone. It assumes that user has already install Android and necessary tools to develop application. It also assumes that the user to be familiar with Basic knowledge of Android

Introduction

A GSM Cell ID (CID) is a generally unique number used to identify each Base transceiver station (BTS) or sector of a BTS within a Location area code (LAC) if not within a GSM network.

User's phone is always connected to a cellular network, and by knowing CellID number, we know the Cell, and by knowing the position of the cell, we know where user is. There is some accuracy issue’s, as the cell can cover from several hundreds of meters to several kilometers, but this could be a very good starting point to locate user.

Overview

To get the current location using cell-id we need to access telephony services of the device which confer by TelephonyManager class. In Android CELL_ID, Wi-Fi etc. are permission protected services to access this information we need to add uses permission "android.permission.ACCESS_COARSE_LOCATION" into the manifest.xml file, it allows to access coarse location.

```
...

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

...
```

Initialize Telephony Services

After adding the uses-permission into the manifest.xml we need to initialize the telephony services by taking the reference of TelephonyManager through Context.getSystemService(String)

```
...

telephoneyMg = (TelephonyManager) mContext.getSystemService(Context.TELEPHONY_SERVICE);

...
```

TELEPHONY_SERVICE Use with getSystemService(String) to retrieve a TelephonyManager for handling management the telephony features of the device.

Get Cell-ID & LAC

To get the GSM location of the device we need to take reference of GsmCellLocation, it gives access to cell-id & lac.

```
...
```

```
gsmCellLocation = (GsmCellLocation) telephoneyMg.getCellLocation();
iCellID = gsmCellLocation.getCid();
iLAC = gsmCellLocation.getLac();

...
```

If cell-id and lac max value is belong 0 to 0xffff then its legal otherwise unknown.

Getting cell position

To get cell position (latitude and longitude) using cell-id & lac we need to do http connection, to access http connection service we need to add internet permission into the manifest.xml file.

```
...

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

...
```

After adding internet permission now we can fetch latitude & longitude using http connection

```
...

String urlmmap = "http://www.google.com/glm/mmap";

         URL url = new URL(urlmmap);

         URLConnection conn = url.openConnection();

         HttpURLConnection httpConn = (HttpURLConnection) conn;

         httpConn.setRequestMethod("POST");

         httpConn.setDoOutput(true);

         httpConn.setDoInput(true);

         httpConn.connect();

         OutputStream outputStream = httpConn.getOutputStream();

         WriteData(outputStream, iCellID, iLac);//user method

         InputStream inputStream = httpConn.getInputStream();
         DataInputStream dataInputStream = new DataInputStream(inputStream);
         dataInputStream.readShort();

         dataInputStream.readByte();

         int code = dataInputStream.readInt();

         if(code == 0){

                  mLatitude = dataInputStream.readInt() / 1000000D;//latitude
                  mLongitude = dataInputStream.readInt() / 1000000D;//longitude

         }



...
```

Getting cell position

Once we get latitude and longitude now we are ready to launch the google map as a separate activity. This code snippet displays the current position of cell.

```
...


String uri = "geo:" + httpConn.getLatitude() + ","+ httpConn.getLongitude() + "?z=" + 10;
startActivity(new Intent(android.content.Intent.ACTION_VIEW, Uri.parse(uri)));
...
```

Sample Code Snippet

The following code snippet demonstrates how to get latitude and longitude using open cell-id.

Class: AndroidCellID

```
...


import android.app.Activity;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.view.Window;
import android.widget.Button;
import android.widget.TextView;
import android.widget.EditText;
import android.widget.Toast;
import android.view.View.OnClickListener;
import android.app.ProgressDialog;
import android.content.Intent;
```

```
public class AndroidCellID extends Activity implements OnClickListener,
AsyncCallBack {

        public final int CELL_ID = 1;
        public final int HTTPCONN = 2;
        private Button mButtonUpdate;
        private Button mButtonLocateMe;
        private TextView mTextViewCell_ID;
        private TextView mTextViewLAC;
        private EditText mEditTextCell_ID;
        private EditText mEditTextLAC;
        private GetCellID mCellId;
        private HttpConn httpConn;
        private ProgressDialog progressDialog;

        /** Called when the activity is first created. */
        @Override
        public void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                requestWindowFeature(Window.FEATURE_NO_TITLE);
                setContentView(R.layout.main);
                init();

        }

        private void init() {

                mButtonUpdate = (Button) findViewById(R.id.updateMe);
                mButtonUpdate.setOnClickListener(this);
                mButtonLocateMe = (Button) findViewById(R.id.locateMe);
                mButtonLocateMe.setOnClickListener(this);
                mTextViewCell_ID = (TextView) findViewById(R.id.cell_id);
                mTextViewLAC = (TextView) findViewById(R.id.lac);
                mEditTextCell_ID = (EditText) findViewById(R.id.edCellID);
                mEditTextCell_ID.setText("-1");
                mEditTextLAC = (EditText) findViewById(R.id.edlac);
                mEditTextLAC.setText("-1");
                mCellId = new GetCellID(this, CELL_ID, this);
                progressDialog = new ProgressDialog(this);
                progressDialog.setCancelable(true);
                progressDialog.setMessage("Featching info...");
        }
```

```
         @Override
         public void onClick(View v) {

                  int id = v.getId();

                  switch (id) {

                  case R.id.updateMe:
                           progressDialog.show();
                           mCellId.updateCellLocation();
                           break;
                  case R.id.locateMe:
                           String cellid = ("" + mEditTextCell_ID.getEditableText()).trim();
                           String lac = ("" + mEditTextLAC.getEditableText()).trim();
                           if ((!cellid.equals("-1")) && (!lac.equals("-1"))) {
                                    progressDialog.show();
                                    httpConn = new HttpConn(cellid, lac, this, HTTPCONN);
                                    httpConn.start();
                           }else{

                  Toast.makeText(this,"update for cell id!!!", Toast.LENGTH_LONG).show();
                           }
                           break;
                  }
         }
         @Override
         public void updateStatus(boolean f, int id) {
                  progressDialog.cancel();
                  switch (id) {
                  case CELL_ID:
                           if (mCellId != null) {
                                    mEditTextCell_ID.setText("" + mCellId.getCellID());
                                    mEditTextLAC.setText("" + mCellId.getLAC());
                           }
                           break;
                  case HTTPCONN:
                           if (f == true) {
                                    if (httpConn != null) {
                                    String uri = "geo:" + httpConn.getLatitude() + ","
                                    + httpConn.getLongitude() + "?z=" + 10;
                                    startActivity(new
Intent(       android.content.Intent.ACTION_VIEW, Uri.parse(uri)));
                           }else{
```

```
                                    //Unable to found cell id
                                    Toast.makeText(this,"Unable to found cell id",
Toast.LENGTH_LONG).show();
                                    }
                           } else {
                                    //http connection fail
                                    Toast.makeText(this,"Http Connection fail",
Toast.LENGTH_LONG).show();
                           }
                           break;
                  }
         }
}
```

ref:http://developer.samsung.com/android/technical-docs/LBS-for-non-GPS-device-through-Android