COMP4336/9336 Lab 3

Lab Objectives

1. Learn how to check the GPS status whether is enabled or not.
2. Learn how to enable GPS with user permission.
3. Learn how to read data from the GPS device.
4. Develop a simple GPS application.

Preparation

1. In Android, you can use *LocationManager/Location* classes to obtain location based on obtained data from GPS inside the mobile phone ore from network data. They have different methods to give all details of a location such as latitude, longitude, speed and accuracy of positioning.
   More details:
   http://developer.android.com/guide/topics/location/strategies.html
   http://developer.android.com/reference/android/location/LocationManager.html
   http://developer.android.com/reference/android/location/Location.html

2. Some useful notes:
   - To access GPS in your application you need to add required permissions in AndroidManifest.xml file. If you are getting location using GPS you need to add ACCESS_FINE_LOCATION (Which includes both ACCESS_FINE_LOCATION and ACCESS_COARSE_LOCATION). Also if you are getting network-based location then you need to add INTERNET permission too.

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

   - If you want to develop a specific class for your own GPS based application, other than default class of *MainActivity*, you should extend the new class from *Service* and also implements *LocationListener*. For example:

     ```
     public class GPSReader extends Service implements LocationListener.
     ```

   - In order to get the location service of mobile phone, you should use this line of code:

     ```
     locationManager =
     (LocationManager)mContext.getSystemService(LOCATION_SERVICE);
     ```

     Which mContext is your instance from mainactivity class. As you want to call GPS reader from *mainactivity* class, it would be better to develop a method for your GPSReader class to get the location.

- Getting GPS Status:

```
isGPSEnabled =
locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);
```

- Get Location from GPS Provider:

```
locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
MIN_TIME_BW_UPDATES,MIN_DISTANCE_CHANGE_FOR_UPDATES,
this)

location =
locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER)
```

- Get Location from Network Provider:

```
Location=
locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVI
DER);
```

- After obtaining the location, you can use methods of class *Location* such as getAltitude() , getLongitude(), … to obtain the details of detected position.

- In main activity class you can create an instance of GPS class as :

```
GPSReader GP = new GPSReader(this);
```

More details and samples:

http://www.androidhive.info/2012/07/android-gps-location-manager-tutorial/

https://sites.google.com/site/androidhowto/how-to-1/using-the-gps

Lab Task

Task1 : Enabling GPS (0.25 Marks)
Develop a simple GPS application program that can check whether GPS is active or not (Figure1).

Optional task: If GPS is not enabled you can ask user to direct him to setting in order to enable GPS (Figure2). You can use *alertDialog* class to create a dialog:

```
AlertDialog.Builder alertDialog = new AlertDialog.Builder(mContext);
```

To direct user to the setting of mobile phone, you can get an instance of *Intent* class as :

```
Intent intent = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
```

Task2 : A simple GPS reader (0.25 Marks)
Extend the previous application of *task1* that can read position data from the GPS and Network Provider and show the results on the screen after pressing a button by user (such as Figure 3).

Task3 : A simple GPS Tracker (0.5 Marks)
Develop a simple GPS based application that can give your position and speed in any arbitrary time when you are moving (Figure 4).
Note1: In order to develop a location sensitive program you have to override the *onLocationChanged(Location l)* method from *LocationManager* class.

Note2: In order to convert GPS time format to a usual format like YYYY/MM/DD you can use *SimpleDateFormat* class as:

```
long time = loc.getTime();
Date date = new Date(time);
SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
String textofDate = sdf.format(date);
```

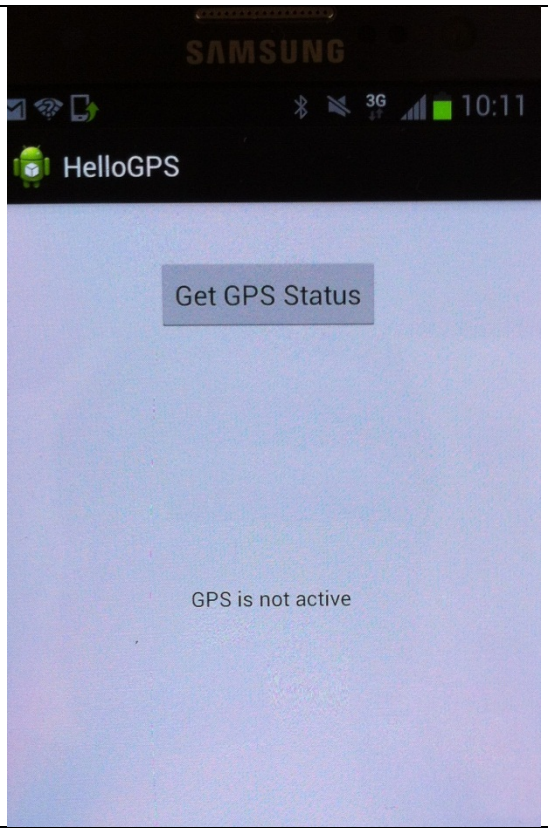

Figure1

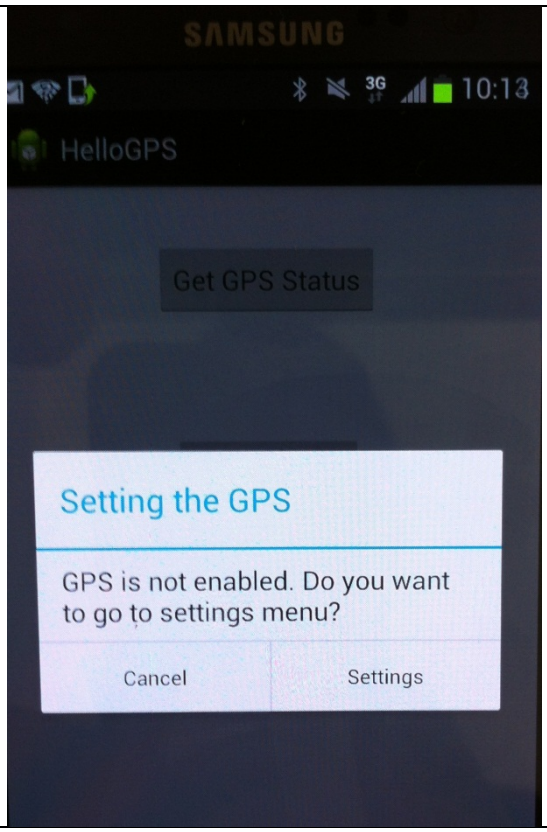

Figure2

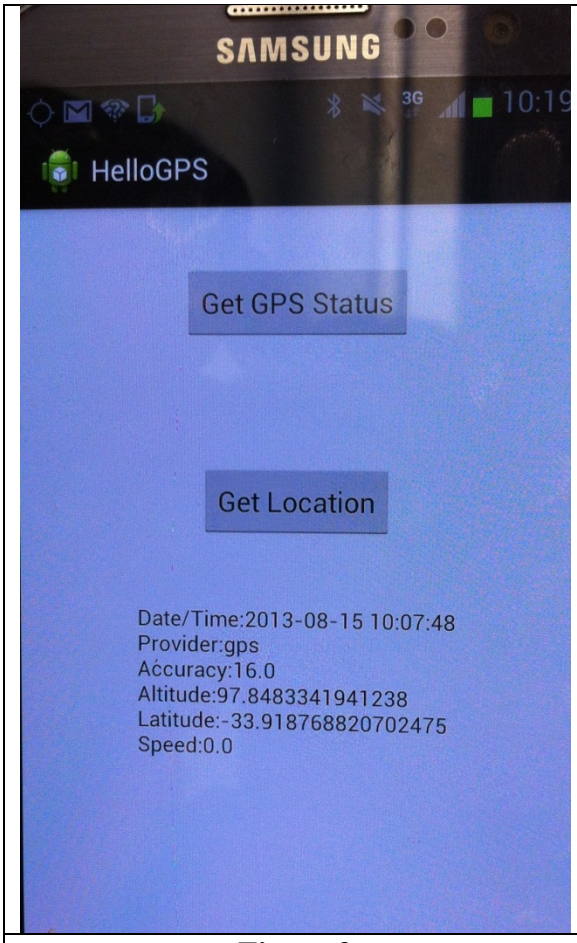

Figure 3

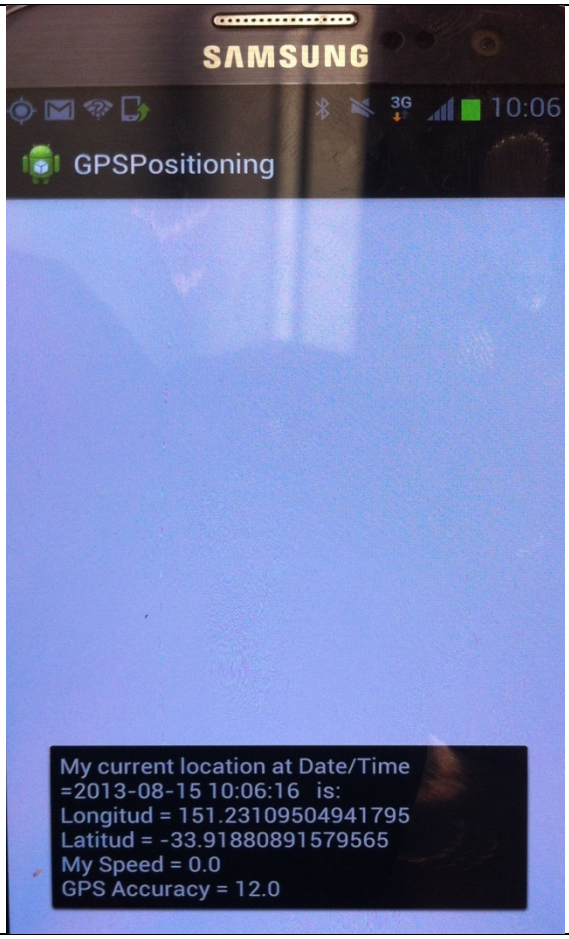

Figure 4