Mobile Applications for Business

Master SIA/SDBIS

Octavian Dospinescu

2021

General topics

- Spatial positioning and geographical landmarks
- Latitude
- Longitude
- Satelits
- GPS
- Interception and using coordinates
- Other facilities offered by GPS technology: speed, azimuth, precision
- Specific issues for GPS positioning (false reflected waves, undetectable waves…)

Geographical coordinates

- A **geographic coordinate system** defines any location on Earth by 2 or 3 coordinates of **a spherical coordinate system** that is aligned with the axis around which **Earth** spins . Theories from ancient Babylonians, later expanded by **Ptolemeu**, assigned 360° to a full circle.

Latitude

- **Latitude** (Lat.) is the angle between any point on the Earth and equator. Lines with a constant latitude are called parallels. They trace circles on the surface of the Earth, but the only parallel that is a big circle is the **equator** (latitude = 0 degrees), with each **geographical pole located at 90 degrees** (90° N North Pole; South Pole 90° S).

Longitude

- **Longitude** (Long.) is the angle to east or west of an arbitrary point on the Earth: **Greenwich Observatory** (UK) is the international point with **0 degrees longitude**. Anti-Greenwich is both 180° W and 180° E .

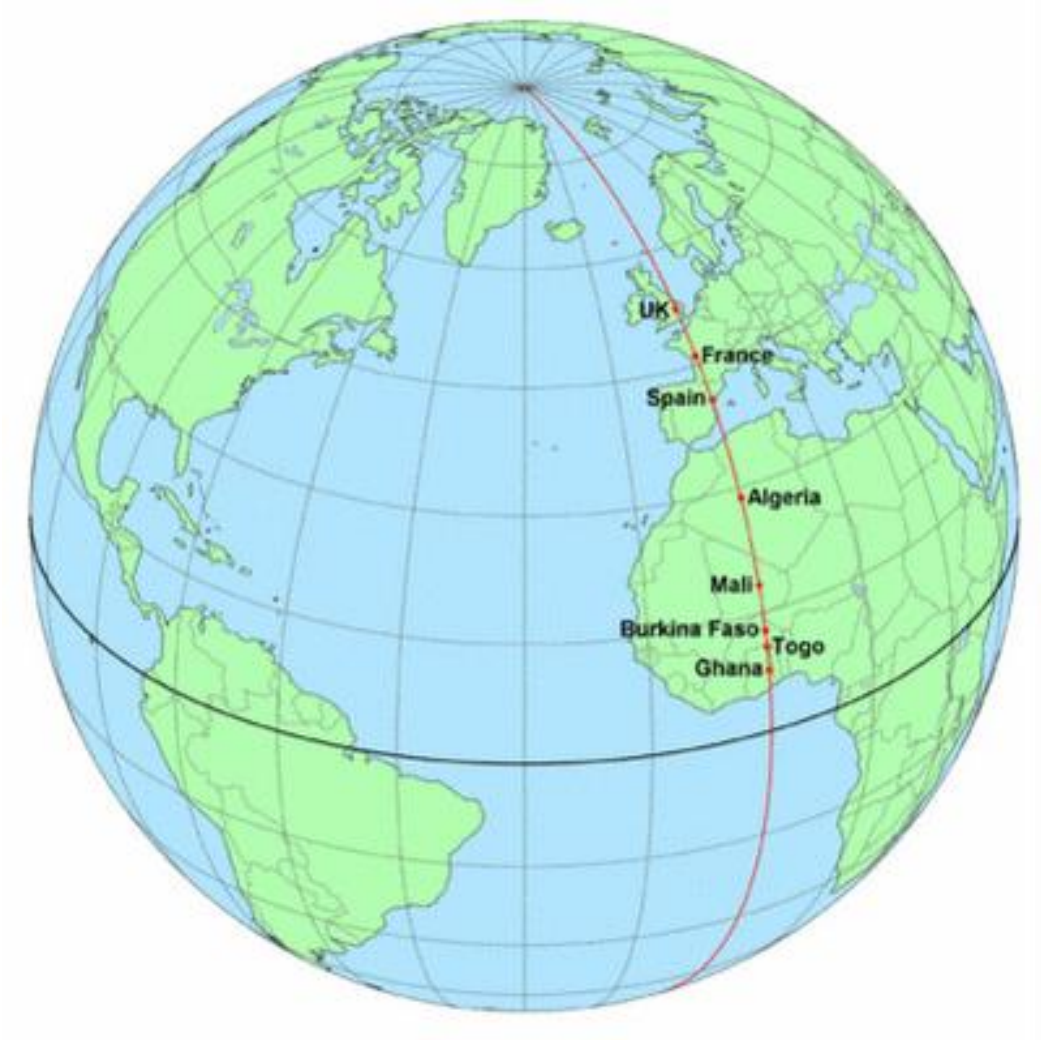
Longitude

- Lines of constant longitude are called **meridians** . Meridian passing through Greenwich is the **primary (first) meridian** . Unlike parallels, all meridians are half of full circles and they are not parallel: they intersect at the north pole and south pole.

Latitude and longitude

- By combining these two angles, it can be specified the horizontal position of any point on Earth.
- For example, Baltimore, Maryland (USA) has a latitude of 39.3° North and a longitude of 76.6° West. So, a vector drawn from the center of the Earth to a point disposed at 39.3° north of the equator and 76.6° west of Greenwich will pass through Baltimore.

Prime meridian - Greenwich



Degrees, minutes, seconds

- The degrees are divided in **minutes** (') and **seconds** ("). There are several formats for degrees, all in order Lat.-Long.:

- **GM** Degrees:Minutes (49:30.0-123:30.0)
- **GMS** Degrees:Minutes:Seconds (49:30:00-123:30:00)
- **GZ** Decimal Degrees(49.5000-123.5000), usually with 4 decimals.

Converting degrees, minutes, seconds to decimal degrees

- To convert from the first two formats to the last, decimal degrees are equal to the whole number of degrees, plus minutes divided by 60, plus seconds divided by 3600. Currently, decimal degrees are the most used format .

- There are cases where novice users better perceive the GMS system coordinates.

Geostationary Coordinates

- Geostationary satellites, such as the television, are arranged above the equator.
- Their position on earth is expressed in decimal degrees, their **latitude** is not changing and **is always zero** over the equator.

- Advantage: they emit in both hemispheres (northern and southern).

The third dimension: altitude, height, depth

- To specify an absolute point on or above the Earth, it must also be specified **the elevation**. This defines the vertical position of the point to the surface.

Elevation

- It can be expressed as the vertical distance to Earth below, but because of the ambiguity of the terms "surface" and "vertical" it is preferred expression relative to a set of more precise data, such as the sea level. The distance from the center of the earth is also a practical coordinate position for deep or space locations.

How to express the elevation

- Usual coordinates of the elevation / height from the surface or other data are **altitude**, **height** and **depth**.

Simulation based on geographical location address

https://www.latlong.net/

http://ro.wikipedia.org/wiki/Nod_(unitate)

Global Positioning System

- **GPS** stands for **Global Positioning System** (the military name is NAVSTAR), a global positioning system. GPS is the main satellite positioning system .

- This system, initiated by the Department of Defense of the United States may enable providing the position of an object on the earth's surface if the object is equipped with the necessary material for the functioning of this system .

- This object may be a person, allowing it to find orientation on land, water, air or space (near Earth) . GPS uses geodetic system WGS84.

How does GPS work… ☺

- The operating principle of GPS is to use satellites in space as reference points for locating ground. Through a very precise measurement of the distance in a straight line between the receiver and **at least 4 satellites**, GPS can determine the position of any point on Earth (latitude, longitude, altitude).
- Normally, determining the 3D position of a point on the terrestrial position using satellites, only 3 distances are needed (three satellites), as the method to be used is that of triangulation.
- A fourth distance is determined **to minimize positioning errors** due clocks receivers; they are not atomic as the satellites and are not very accurate.

GPS position calculation

- The distance between the satellite and the receiver is calculated by timing the time it needs to reach the radio signal from the satellite to the receiver.
- Knowing that radio signal travels 300 000 km / sec (speed of light), if we measure the time of propagation from the satellite to the receiver, we can deduce the distance between them.
- Each satellite has its own signal (Pseudo Random Code) so that the receiver will know exactly what satellite is.

GPS position calculation modes - estimated accuracies

- The reception of satellite signals and the calculation of position can be done in two modes: **absolute mode** and **differential mode**.

- **The absolute mode** uses a single GPS receiver and positioning accuracy is about 10-15 m.

- **The differential mode** involves the using of two receivers one of which will be *base station* that is installed within a point with known coordinates so that it can measure the difference between the known coordinates of the same point of analysis results for GPS signals.

GPS position calculation modes - estimated accuracies

- In **differential mode**, the differences calculated will be used to correct coordinates determined by a mobile receiver in other points in the area.

- This mode is highly accurate (1-5 cm) , but the distance between the mobile handset and the base station should not exceed 30 km.

GPS receivers

•They are used to get the current position of the receiver's owner.

•The range of applications that can be developed is very wide

• Mobile applications use increasingly more GPS technology

• GPS receivers tend to be hardware included in GSM terminals

• The power and precision of the receivers depend on the quality of the receiver hardware, positioning in space and the weather.

• The GPS antenna is the basic component.

Problems with the infrastructure

- Interesting details here:

http://www.realitatea.net/sistemul-de-sateliti-gps-ar-putea-ceda-pana-in-anul-2010_521316.html

About GPS and geographical positioning

Useful addresses and basic terms about GPS:

https://www.findlatitudeandlongitude.com/

http://ro.wikipedia.org/wiki/Nod_(unitate)

Mobile applications integrated with GPS capabilities

- Prerequisites :
  - Emulator mobile device (Android Virtual Device)
  - Simulator / emulator to generate the current GPS position (ADM - Android Device Monitor)
  - Gps functional (to be on the device **on**)
  - A library that offers the ability to " intercept " and use the GPS device
  - Permission **android.permission.ACCESS_FINE_LOCATION** (type uses permission) in file **AndroidManifest.xml**

Mobile applications integrated with GPS capabilities

- Android can access a mobile device coordinates and altitude, direction or speed through ***LocationProvider*** object .

- with this manager we "subscribe" to notifications of the service that handles our location.

Mobile applications integrated with GPS capabilities

Almost all of the Android systems **use the same mechanism Publish/Subscribe**, that an application must subscribe to notifications of service, and must implement methods that will be called when a certain event will occur.

The general methodology

General steps to find out the current location are:

- we create an object of type **Manager** with which **we access the services** that provide location information ;
- using the method *getLastKnownLocation()* we check if there is information about the last location known by the system and display it;
- always check if the location has changed and display information.

Access system services in Android

To get access to the Android system we will have to use the method ***getSystemService()*** , which returns an object of type Manager, with which we can access information about the desired service.

This method accepts as a parameter **the name of the desired service**; in our case we want to access location information. Therefore we send **LOCATION_SERVICE** parameter.

Access system services in Android – useful remarks

We must have in mind during the creation of applications that **required updates from a particular service are consuming battery**.

Access system services in Android – useful remarks

- We must use the lifecycle of activities and ask these updates only when the user uses the application.

- To achieve this we will overwrite methods *onPause()* and *onResume()* where we will subscribe respectively unsubscribe from updates received from the system service.

Practical implementation

AndroidManifest.XML

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

Practical implementation

```
public class PrjGPSActivity extends Activity implements LocationListener {
   /** Called when the activity is first created. */

TextView afisareCoordonate;
LocationManager locManager;

   @Override
   public void onCreate (Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.main);

      //codul meu
      afisareCoordonate = (TextView) findViewById(R.id.lblLocatie);

      //cu ajutorul acestui manager voi primi notificari cand locatia se schimba
      locManager = (LocationManager) getSystemService(LOCATION_SERVICE);

      Location ultimaLocatie = locManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);

      if (ultimaLocatie != null)
            onLocationChanged(ultimaLocatie);

   }
```

Practical implementation

```
@Override
public void onLocationChanged (Location location) {
        //s-a schimbat locatia, deci afisez noile valori
        this.afisareCoordonate.setText(Double.toString(location.getLatitude()));
}

@Override
protected void onResume() {
        super.onResume();
        locManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 1000, 10,
this);
}
```

Practical implementation

- In *onResume()* method we ask the system to resume notifications. To achieve this we used the method *requestLocationUpdates()* that can receive the following parameters:

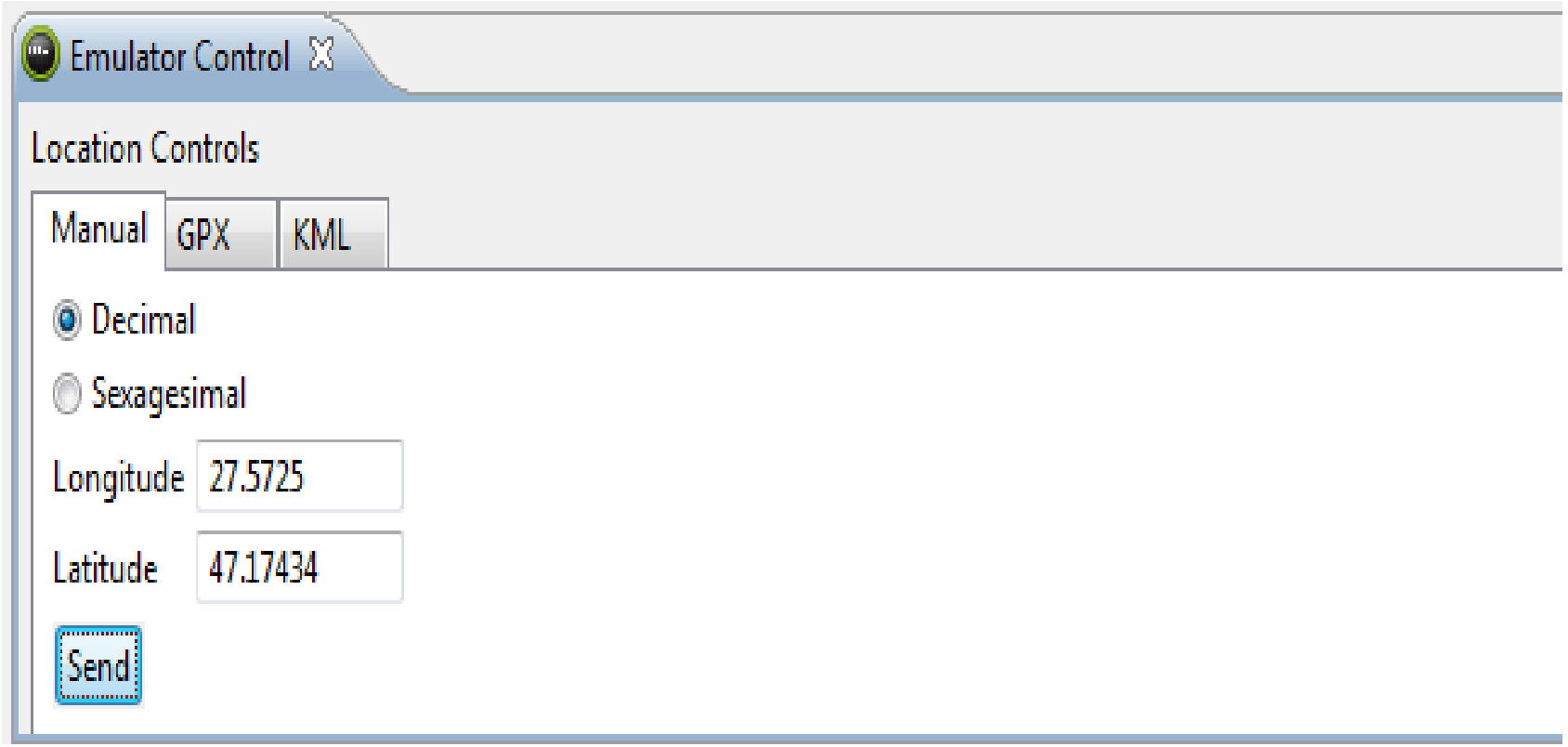
public void ***requestLocationUpdates*** *(String provider, long minTime, float minDistance, LocationListener listener)*

Simulating satellite for GPS

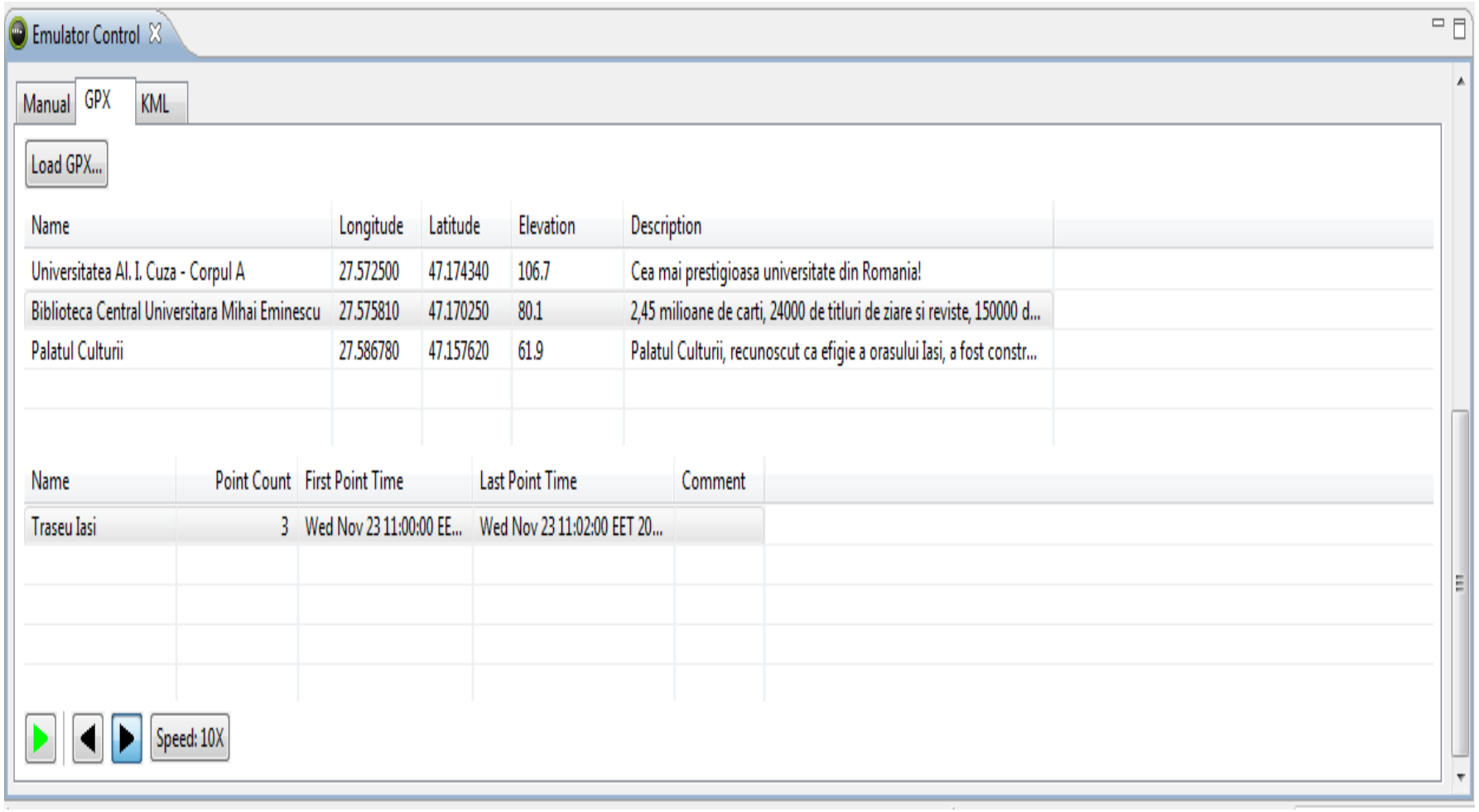
In an emulator, to simulate GPS coordinates we must manually send them in decimal or hexadecimal format.

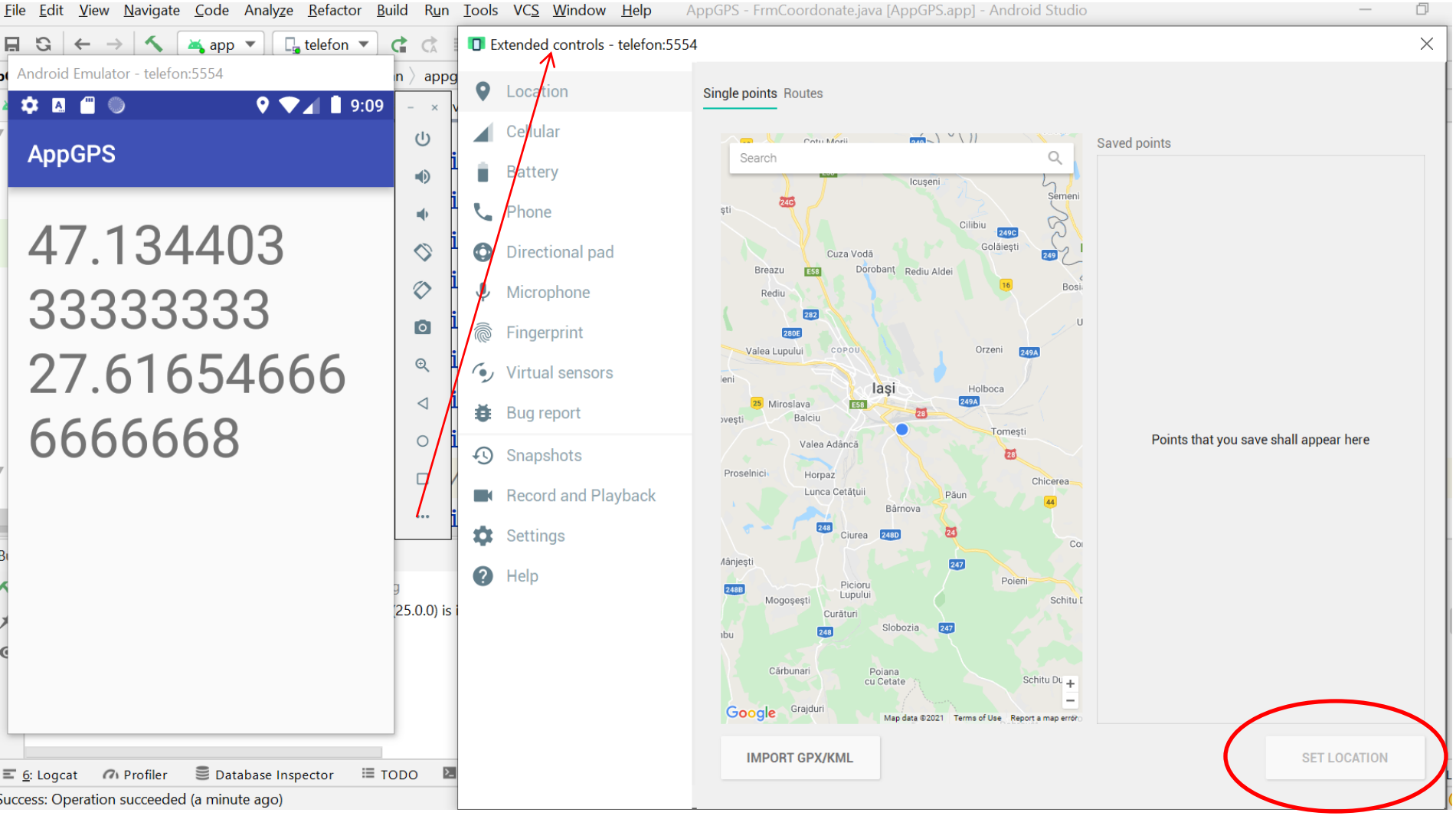
This can be done either from the command line or using the tool **ADM** - Android Device Monitor.

ADM – Android Device Monitor



ADM – Android Device Monitor




File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
AppGPS - FrmCoordonate.java [AppGPS.app] - Android Studio
app
telefon
Android Emulator - telefon:5554
9:09
AppGPS
47.134403
33333333
27.61654666
6666668
Extended controls - telefon:5554
Location
Cellular
Battery
Phone
Directional pad
Microphone
Fingerprint
Virtual sensors
Bug report
Snapshots
Record and Playback
Settings
Help
Single points
Routes
Search
Saved points
Points that you save shall appear here
IMPORT GPX/KML
SET LOCATION
Iași
Google
6: Logcat
Profiler
Database Inspector
TODO
Success: Operation succeeded (a minute ago)

Complete code – for the lab ☺

AndroidManifest.xml

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

Complete code – for the lab ☺

```
package com.example.user.appgeographic;

    import android.Manifest;
    import android.app.Activity;
    import android.content.pm.PackageManager;
    import android.location.Location;
    import android.location.LocationListener;
    import android.location.LocationManager;
    import android.support.v4.app.ActivityCompat;
    import android.support.v7.app.AppCompatActivity;
    import android.os.Bundle;
    import android.util.Log;
```

```
public class FrmCoordonate extends AppCompatActivity implements LocationListener {

    LocationManager locManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_frm_coordonate);

        ActivityCompat.requestPermissions(this,new String[]{Manifest.permission.ACCESS_FINE_LOCATION},1);

        locManager = (LocationManager) getSystemService(LOCATION_SERVICE);
        if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            // TODO: Consider calling
            //    ActivityCompat#requestPermissions
            // here to request the missing permissions, and then overriding
            //   public void onRequestPermissionsResult(int requestCode, String[] permissions,
            //                                          int[] grantResults)
            // to handle the case where the user grants the permission. See the documentation
            // for ActivityCompat#requestPermissions for more details.
            Log.i("TAVY","probleme cu permisiunea in onCreate");
            return;
        }
        Location ultimaLocatie = locManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
        if (ultimaLocatie != null) {
            onLocationChanged(ultimaLocatie);
        }
    }
```

```
    @Override
    public void onLocationChanged(Location location) {
        Log.i("TAVY", Double.toString(location.getLatitude()) + "  " + Double.toString(location.getLongitude()));
    }

    @Override
    protected void onResume() {
        super.onResume();
        if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            // TODO: Consider calling
            //    ActivityCompat#requestPermissions
            // here to request the missing permissions, and then overriding
            //   public void onRequestPermissionsResult(int requestCode, String[] permissions,
            //                                          int[] grantResults)
            // to handle the case where the user grants the permission. See the documentation
            // for ActivityCompat#requestPermissions for more details.
            Log.i("TAVY","probleme cu permisiunea in onResume");
            return;
        }
        locManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 1000, 10, this);
    }
}
```