E0-245: ASP

Lecture 14+15: Location Services

Dipanjan Gope

Module 2: Android Sensor Applications

- Location Sensors
  - Theory of location sensing
  - Package android.location
- Physical Sensors
  - Sensor Manager
  - Accelerometer
  - Gyroscope
  - Magnetometer
  - Sensor fusion
- Multimedia
  - Camera
  - Microphone
- NFC

References

- Greg Milette, Adam Stroud: Professional Android Sensor Programing, 2012, Wiley India

Popular Apps with Location Sensing

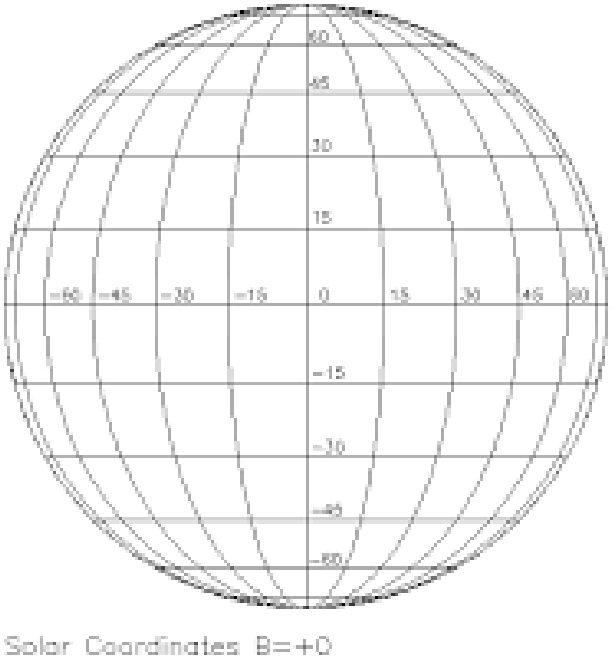
MAPS AND DIRECTIONS

GEOLOCATION

FITNESS TRAINING

LOCATION BASED SERVICES

Latitude and Longitude



Solar Coordinates B=+0

Haversine Formula

$$a = \sqrt{\sin^2\frac{(\phi_2 - \phi_1)}{2} + \cos(\phi_1)\cos(\phi_2)\sin^2\frac{(\lambda_2 - \lambda_1)}{2}}$$

$$d = 2 \times r \times \sin^{-1}(a)$$

What is the distance corresponding to 1 degree difference in latitude (assume same longitude)?

Latitude and Longitude

Degree precision versus length

decimal places	decimal degrees	DMS	qualitative scale that can be identified	N/S or E/W at equator	E/W at 23N/S	E/W at 45N/S	E/W at 67N/S
0	1.0	1° 00′ 0″	country or large region	111.32 km	102.47 km	78.71 km	43.496 km
1	0.1	0° 06′ 0″	large city or district	11.132 km	10.247 km	7.871 km	4.3496 km
2	0.01	0° 00′ 36″	town or village	1.1132 km	1.0247 km	787.1 m	434.96 m
3	0.001	0° 00′ 3.6″	neighborhood, street	111.32 m	102.47 m	78.71 m	43.496 m
4	0.0001	0° 00′ 0.36″	individual street, land parcel	11.132 m	10.247 m	7.871 m	4.3496 m
5	0.00001	0° 00′ 0.036″	individual trees	1.1132 m	1.0247 m	787.1 mm	434.96 mm
6	0.000001	0° 00′ 0.0036″	individual humans	111.32 mm	102.47 mm	78.71 mm	43.496 mm
7	0.0000001	0° 00′ 0.00036″	practical limit of commercial surveying	11.132 mm	10.247 mm	7.871 mm	4.3496 mm
8	0.00000001	0° 00′ 0.000036″	specialized surveying (e.g. tectonic plate mapping)	1.1132 mm	1.0247 mm	787.1 µm	434.96 µm

http://en.wikipedia.org/wiki/Decimal_degrees

Location Sensing: Theory

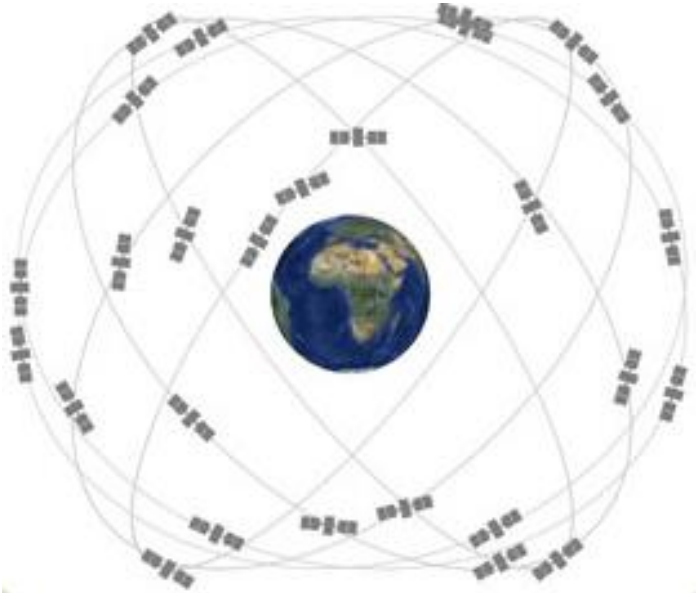
- GPS
- Wifi access points
- Cell tower

GPS Data

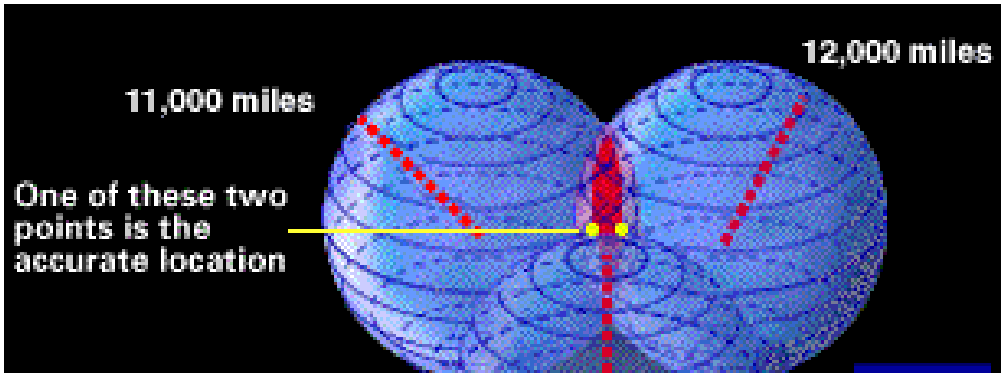
- Almanac
  - Time and status of entire constellation
  - A-GPS (mobile network)
- Ephimeris
  - Current position of particular satellite
- Time
  - Coarse acquisition code

GPS location

- ~1100 functioning satellites
- 24 GPS satellites



Triangulation

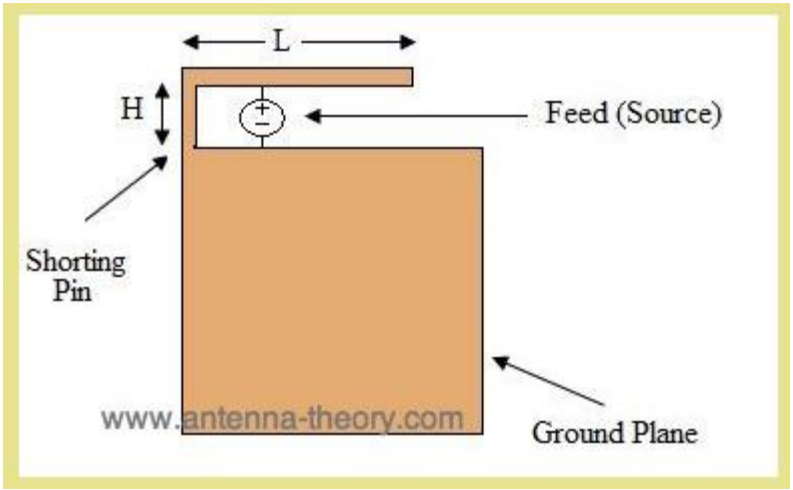


http://www.trimble.com/gps_tutorial/howgps-triangulating-anim02.aspx

http://www.gps.gov/systems/gps/space/

GPS Antenna

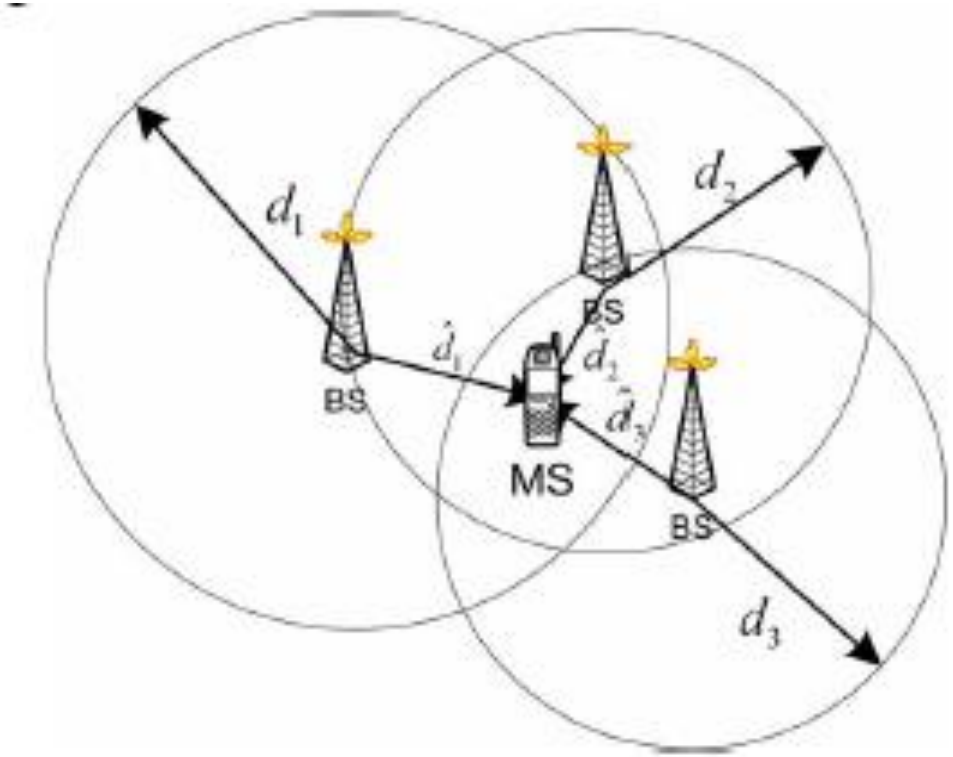
- 1.575 GHz
- Receive only – no SAR issues
- Typically Inverted-F antennas (IFA) / PIFA
- Omni-directional radiation pattern



Dipanjan Gope

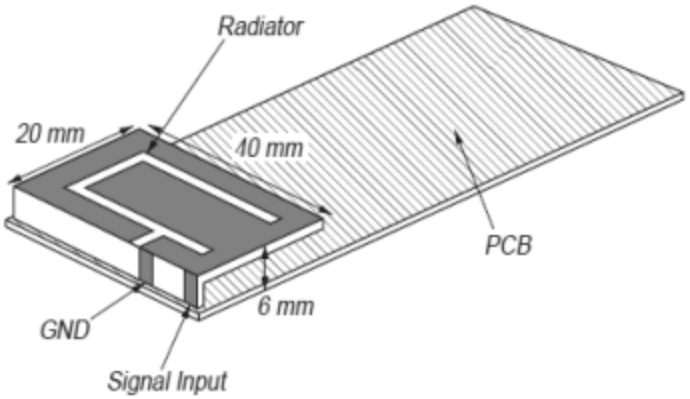
Cellular Network

- Received Signal Strength Intensity (RSSI)
- Propagation Model

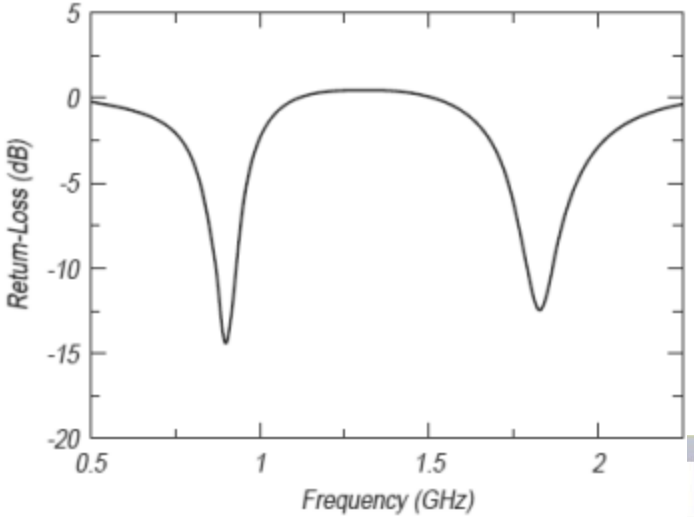


Cellular Antenna

- ~900MHz and ~1800MHz
- SAR: 2mW/gm
- Dual band IFA or PIFA



http://www.qsl.net/va3iul/



Dipanjan Gope

WiFi-based Localization

RADAR: An In-Building RF-based User Location and Tracking System

Paramvir Bahl and Venkata N. Padmanabhan
Microsoft Research
{bahl, padmanab}@microsoft.com

Abstract

The proliferation of mobile computing devices and local-area wireless networks has fostered a growing interest in location-aware systems and services. In this paper we present RADAR, a radio-frequency (RF) based system for locating and tracking users inside buildings. RADAR operates by recording and processing signal strength information at multiple base stations positioned to provide overlapping coverage in the area of interest. It combines empirical measurements with signal propagation modeling to determine user location and thereby enable location-aware services and applications. We present experimental results that demonstrate the ability of RADAR to estimate user location with a high degree of accuracy.

Keywords: *location-aware services, user location and tracking, wireless LAN, radio-frequency wireless network.*

uses signal strength information gathered at multiple receiver locations to *triangulate* the user's coordinates. Triangulation is done using both empirically-determined and theoretically-computed signal strength information.

Our experimental results are quite encouraging. With high probability, RADAR is able to estimate a user's location to within a few meters of his/her actual location. This suggests that a large class of location-aware services can be built over an RF local-area wireless data network.

The remainder of this paper is organized as follows. In Section 2, we survey related work in location determination technologies. In Section 3, we discuss our research methodology. Section 4 contains the core of the paper where we present and analyze the empirical and the signal propagation modeling methods. A discussion of extensions to the base RADAR system appears in Section 5. Finally, we present our conclusions in Section 6.

Dipanjan Gope

Covered in lab

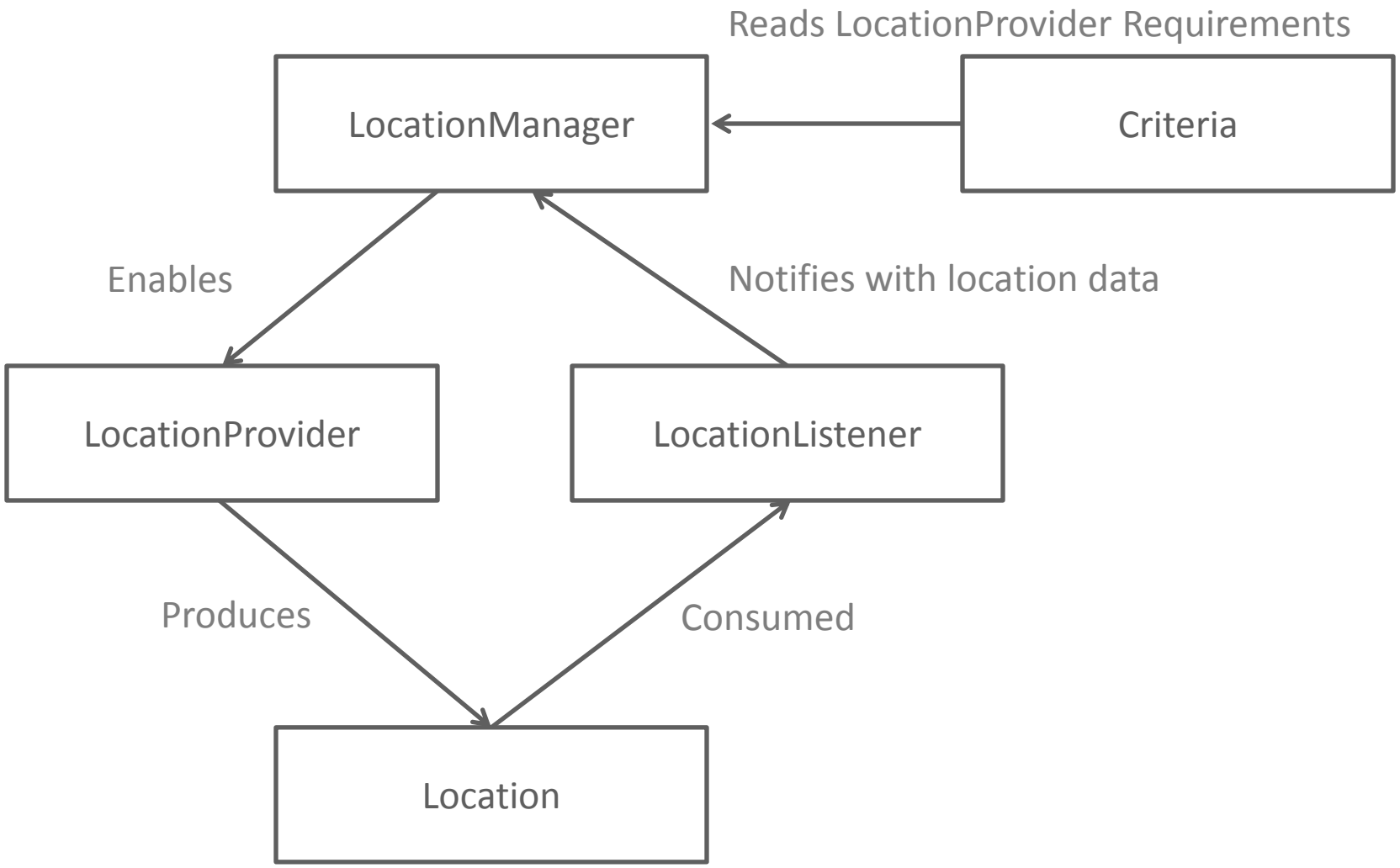
WiFi Antenna

- ~2.4GHz
- SAR:
- Dual or single band PIFA

Package: Android.location

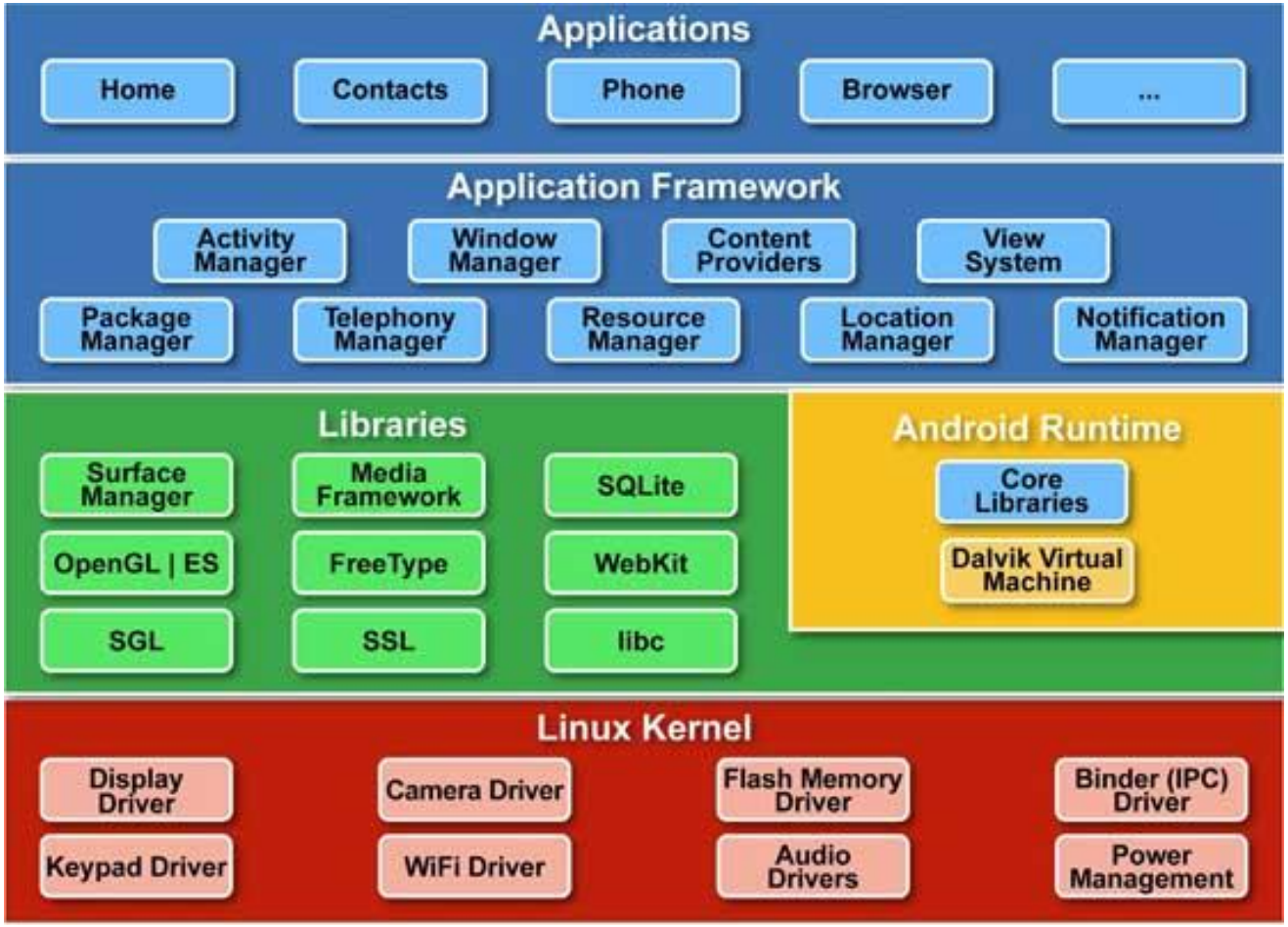
- Classes:
  - LocationManager
  - LocationProvider
  - Location
  - Criteria
- Interface
  - LocationListener

Package: Android.location



Greg Milette, Adam Stroud: Professional Android Sensor Programing, 2012, Wiley India

Location Manager in Android Stack



LocationManager

```
private LocationManager locationManager;

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

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

List<String> enabledProviders = locationManager.getProviders(criteria, true);
```

```
locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, MIN_TIME_BW_UPDATES,
    MIN_DISTANCE_CHANGE_FOR_UPDATES, this);

locationManager.requestSingleUpdate(criteria, this, null);

Location locationReturn = locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
```

```
locationManager.removeUpdates(this);
```

LocationListener

- abstract void onLocationChanged(Location location)
- abstract void onProviderDisabled(String provider)
- abstract void onProviderEnabled(String provider)
- abstract void onStatusChanged(String provider, int status, Bundle extras)

LocationProvider

- GPS location provider
- Network location provider
- Passive location provider

Location Provider

LOCATION PROVIDER	REQUIRED PERMISSION	BATTERY CONSUMPTION	ACCURACY
GPS Provider	`android.permission.ACCESS_FINE_LOCATION` or `android.permission.ACCESS_COARSE_LOCATION`	Consumes more battery power than other location providers	Provides the most accurate location data
Network Provider	`android.permission.ACCESS_COARSE_LOCATION`	Consumes less battery power than the GPS provider	Provides less accuracy than the GPS provider
Passive Provider	`android.permission.ACCESS_FINE_LOCATION`	N/A	N/A

Greg Milette, Adam Stroud: Professional Android Sensor Programing, 2012, Wiley India

Manifest File: Permission

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

- GPS
- Passive

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

- Network
- GPS

Location Provider: Battery Life

Battery Duration for Different Sensor Settings (in Hours)

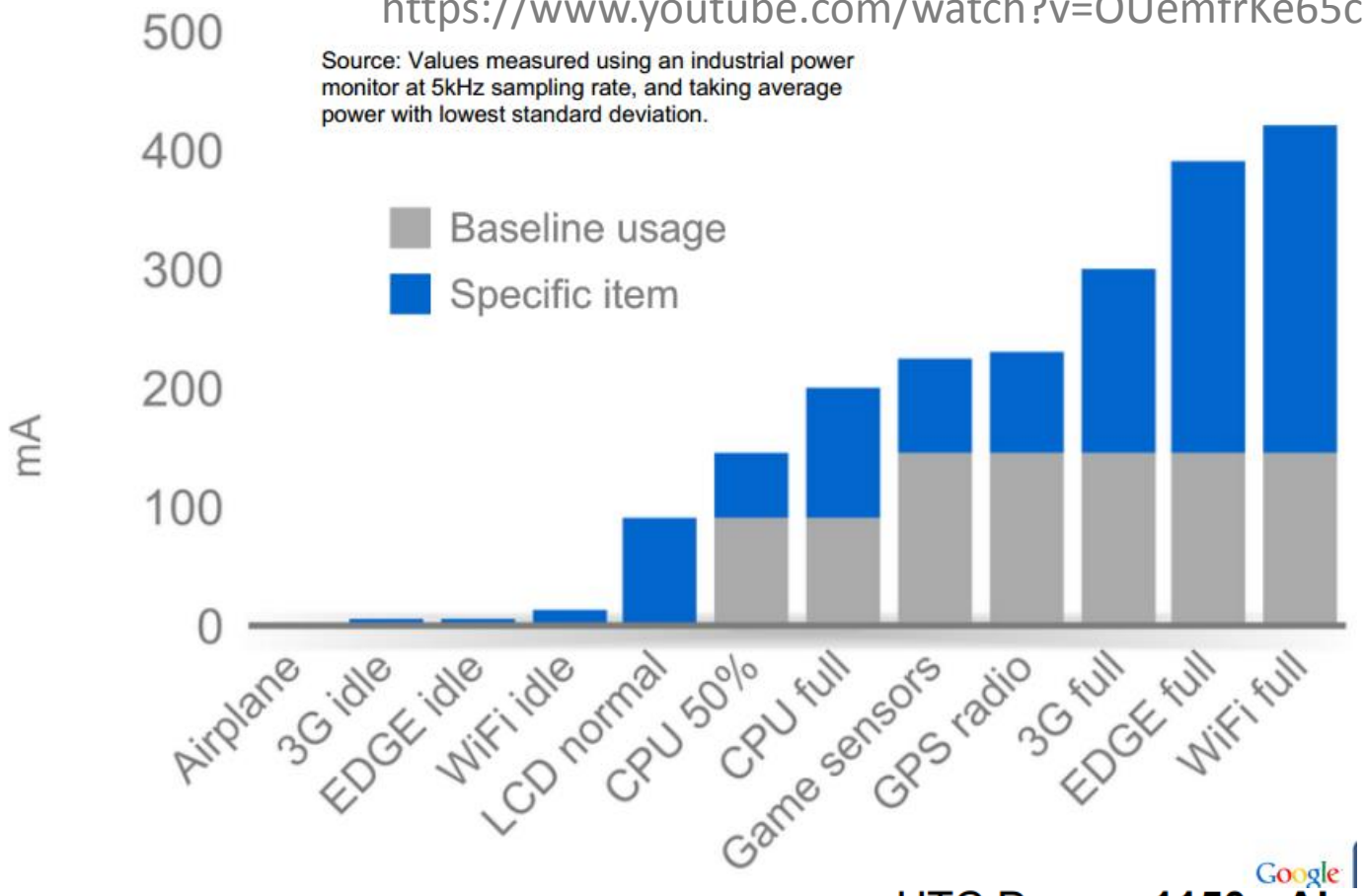
Sensors	Samsung i9100	HTC Desire
No sensor	18.2	15.3
Accelerometer 20Hz	18.0	15.2
Microphone 8kHz+FFT	17.5	14.9
Cell tower 1Hz	17.8	15.0
GPS 1Hz	9.7	6.4

How Long to Wait? Predicting Bus Arrival Time With Mobile Phone Based Participatory Sensing

Pengfei Zhou, *Student Member, IEEE,* Yuanqing Zheng, *Student Member, IEEE,* and Mo Li, *Member, IEEE*

Location Provider: Battery Life

https://www.youtube.com/watch?v=OUemfrKe65c



Google

- HTC Dream: **1150mAh**
- HTC Magic: **1350mAh**
- Samsung I7500: **1500mAh**
- Asus Eee PC: **5800mAh**

Coding for Life--Battery Life, That Is

Jeff Sharkey
May 27, 2009

Location

...

float	getAccuracy () Get the estimated accuracy of this location, in meters.
double	getAltitude () Get the altitude if available, in meters above the WGS 84 reference ellipsoid.
float	getBearing () Get the bearing, in degrees.
long	getElapsedRealtimeNanos () Return the time of this fix, in elapsed real-time since system boot.
Bundle	getExtras () Returns additional provider-specific information about the location fix as a Bundle.
double	getLatitude () Get the latitude, in degrees.
double	getLongitude () Get the longitude, in degrees.
String	getProvider () Returns the name of the provider that generated this fix.

...

Criteria

ATTRIBUTE	EXPLANATION	POSSIBLE VALUES
`accuracy`	Indicates the overall level of accuracy for a location provider.	`Criteria.ACCURACY_FINE` or `Criteria.ACCURACY_COURSE`
`altitudeRequired`	Indicates whether a location provider needs to provide altitude information.	True or false
`bearingRequired`	Indicates whether a location provider needs to provide bearing (the direction being traveled) information.	True or false
`bearingAccuracy`	Required accuracy for bearing information.	`Criteria.ACCURACY_HIGH` or `Criteria.ACCURACY_LOW`
`costAllowed`	Indicates whether the location provider is allowed to cost the user money.	True or false
`horizontalAccuracy`	Required accuracy for latitude and longitude values.	`Criteria.ACCURACY_LOW`, `Criteria.ACCURACY_MEDIUM`, or `Criteria.ACCURACY_LOW`

Greg Milette, Adam Stroud: Professional Android Sensor Programing, 2012, Wiley India

Criteria

ATTRIBUTE	EXPLANATION	POSSIBLE VALUES
`powerRequirement`	Amount of battery power required by the location provider.	`Criteria.POWER_LOW`, `Criteria.POWER_MEDIUM`, or `Criteria.POWER_LOW`
`speedRequired`	Indicates whether a location provider needs to provide speed information.	True or false
`speedAccuracy`	Required accuracy for speed information.	`Criteria.ACCURACY_HIGH` or `Criteria.ACCURACY_LOW`
`verticalAccuracy`	Required accuracy for altitude information.	`Criteria.ACCURACY_HIGH` or `Criteria.ACCURACY_LOW`

Greg Milette, Adam Stroud: Professional Android Sensor Programing, 2012, Wiley India

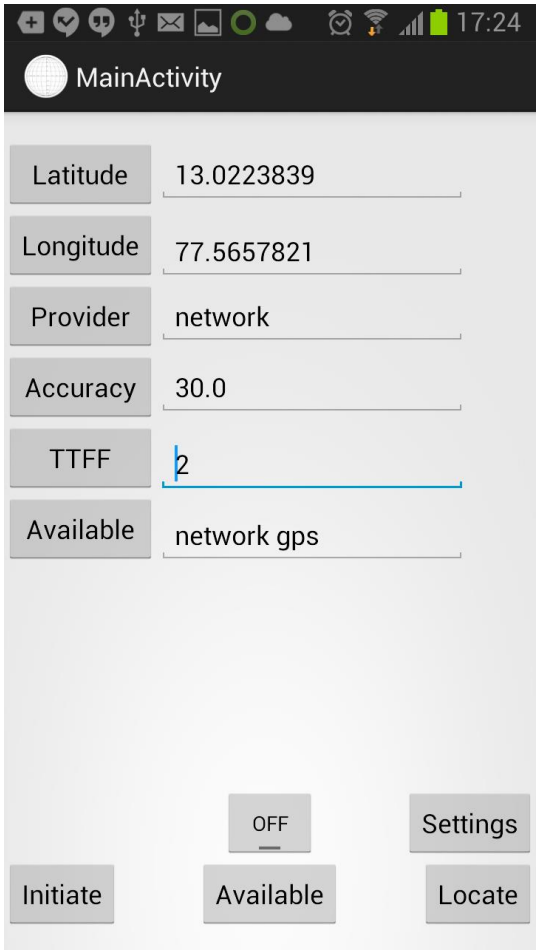
Lets Code …

```
case R.id.GPSbutton:
    timeAtStart = SystemClock.uptimeMillis();
    locationManager.requestSingleUpdate(criteria, this, null);
    break;
```

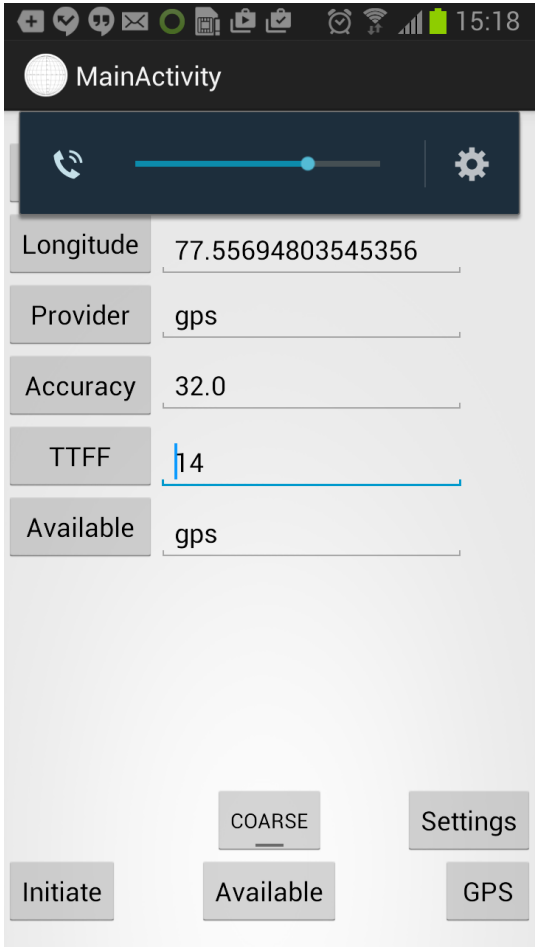
```
public void onLocationChanged(Location location)
{
    // TODO Auto-generated method stub
    long timeToFix = SystemClock.uptimeMillis() - timeAtStart;

    latitudeValue.setText("" + location.getLatitude());
    longitudeValue.setText(String.valueOf(location.getLongitude()));
    providerValue.setText(String.valueOf(location.getProvider()));
    TTFFvalue.setText(String.valueOf(timeToFix/1000));
    accuracyValue.setText(String.valueOf(location.getAccuracy()));
    locationManager.removeUpdates(this);
}
```

Lets Code …



SPW office



On the road