Mobile Programming

Ian Batten
igb@batten.eu.org

Today’s Content

- Anything you need to know about the exercise
- Sensors
- An example of a messy sensor: GPS
- Android sensor abstractions and coding

Exercise Issues?

Sensors

- A computer in your pocket with no sensors isn't useless. In the 1990s, we all (for some value of all) had Palm Pilots.
- But if a device knows where it is, you can do useful things with that knowledge (mapping and directions).
- Similarly there are applications for temperature, speed, altitude or pressure, orientation, acceleration…phones now have many sensors.

GPS: Some Background

- Everyone now has GPS enabled devices.
- But it’s taken a long time for it to become ubiquitous.
- As a result, GPS is quite messy to program, and has a lot of hidden limitations.   Let’s look behind the abstractions for a few minutes!
- (GLONASS, the Russian equivalent, is very similar. Galileo has some differences, but isn’t operational yet).

History

- US military project of the 1960–1990s, with some usability by civilians (official after the KAL007 incident, probably intended earlier)
- Precision available to civilians is deliberately worse than that available to the military.
- Up until 2000, it was a **lot** worse.
- Intention is to make it useless for ballistic missile boost-phase guidance (limitations on accuracy, altitude, acceleration and velocity).

GPS Theory (quick!)

- The **space segment** consists of satellites in a 12-hour (26600km radius, 20200km altitude) orbit. They have atomic clocks on board which need to be accurate to ~1ns as seen from earth.
- Each broadcasts time, an *almanac* of which satellites are in service, and *ephemeris* data predicting satellite location.
- The **user segment** consists of the boxes in your pocket.
- Given the time and position of four satellites, you can solve for *x*, *y*, *z* and *t* (modulo some degenerate cases known as "bad geometry")
- Accuracy degraded by atmospheric conditions (*ionospheric dispersion*) and by the US DoD's concerns.

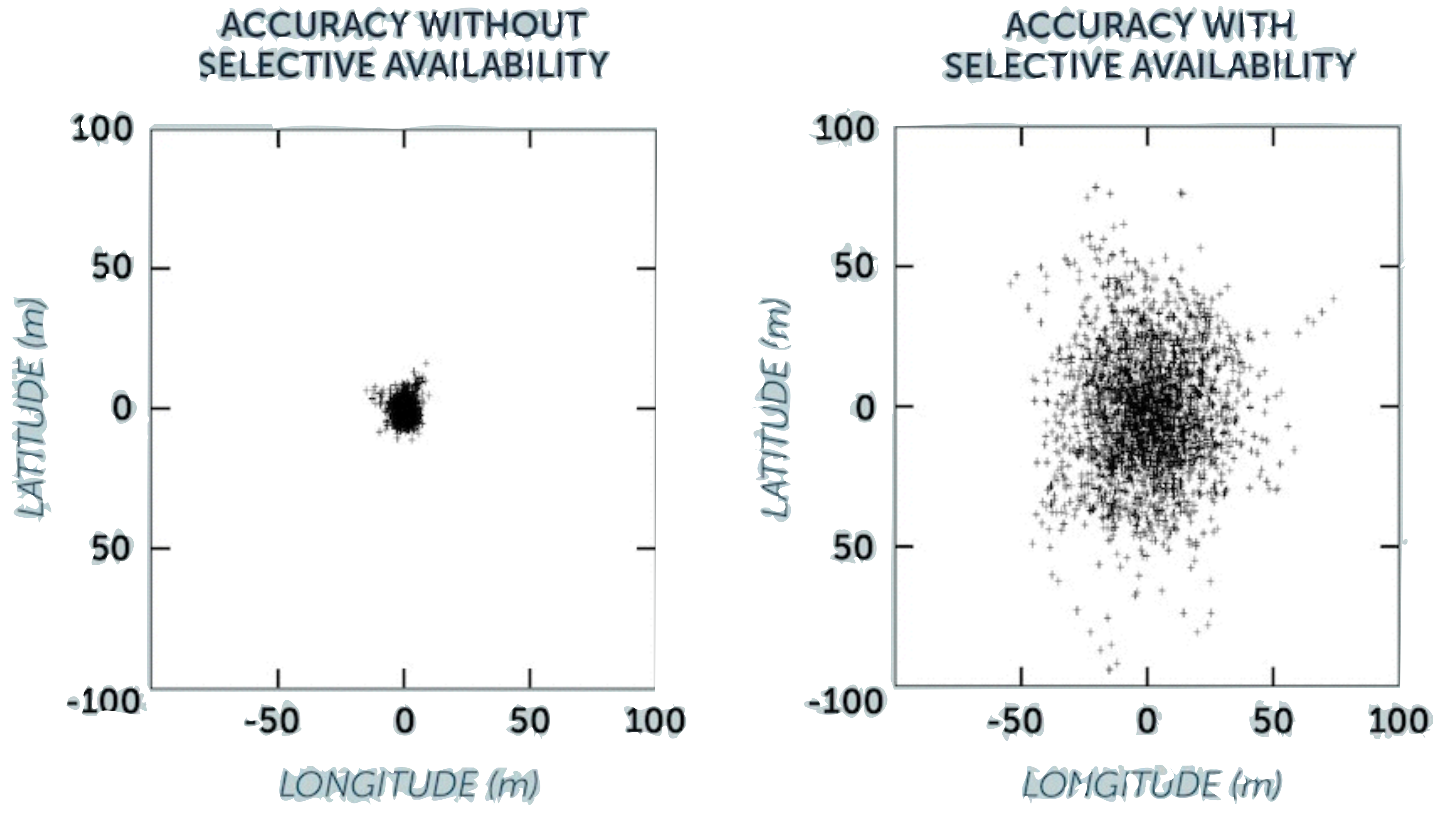
Clock Accuracy

- Speed of light is (roughly) one foot (30cm) per nanosecond
- So 3ns of timing error will cost you a metre of location error.
- Relativity
  - Special Relativity: clocks run 7μs per day slow, because of relative motion.
  - General Relativity: clocks run 45μs per day fast, because of difference in gravity.
  - 38μs per day fast = 10km per day increase in positioning error.
- Because all this is corrected for, we recover a ~1ns accuracy clock from GPS, as well as our location. This is often used as a clock reference for computer networks.

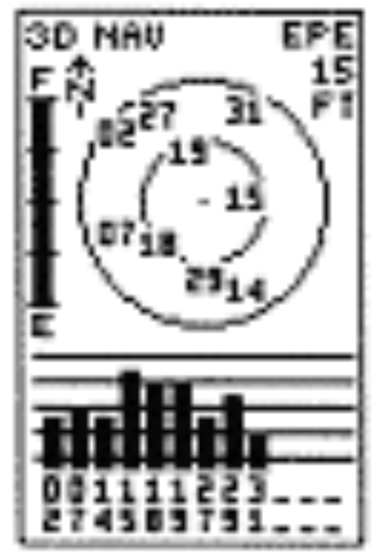
Positioning Accuracy

- Signal on L1 (1.57542 GHz) contains both the C/A *Coarse/Acquisition Service* and P(Y) *Precise Positioning Service.*
- Signal on L2 (1.2276 GHz) contains only P(Y).
- If you have access to P(Y) you can get ~2m accuracy in real time. The two frequencies allow you to correct for atmospheric conditions, and P(Y) is higher precision (faster bit rate, so more accurate clock information).
- Differential GPS allows you to correct for this, given access to a DGPS service.
- Unfortunately, PPS is encrypted and only available to the military. C/A on L1 alone has accuracy of ~5m or worse
- Until 1 May 2000, C/A was deliberately further degraded.

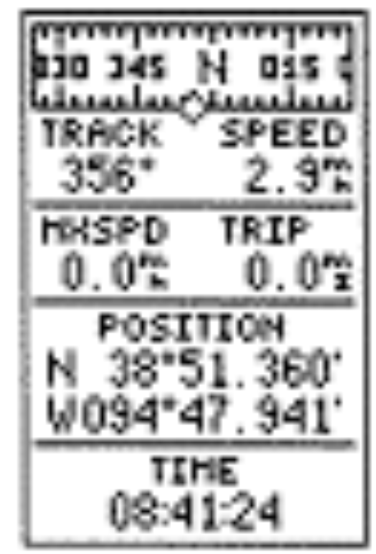
Until May 2000: Useless for real-time navigation



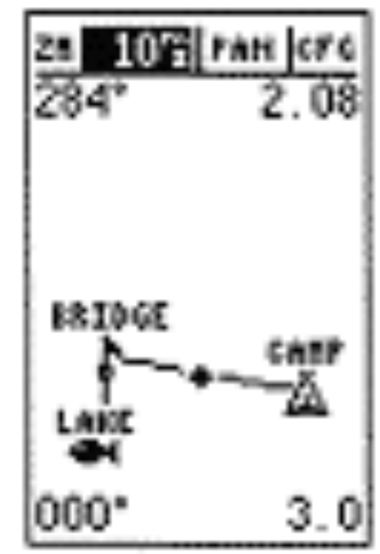
1990s: Clunky Devices



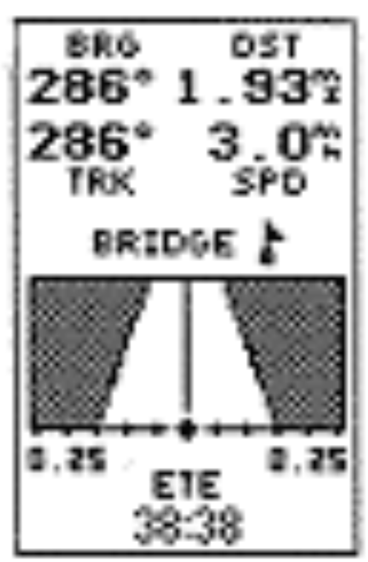
Satellite Status



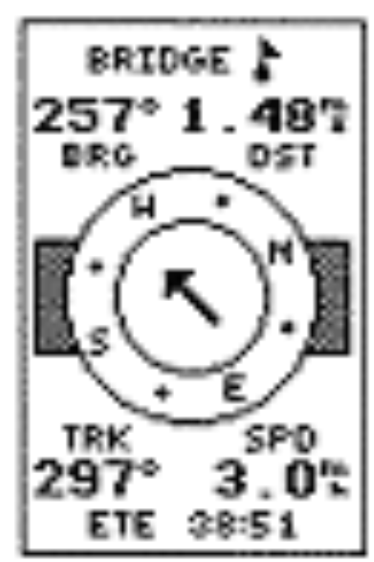
Position



Moving Map



Navigation

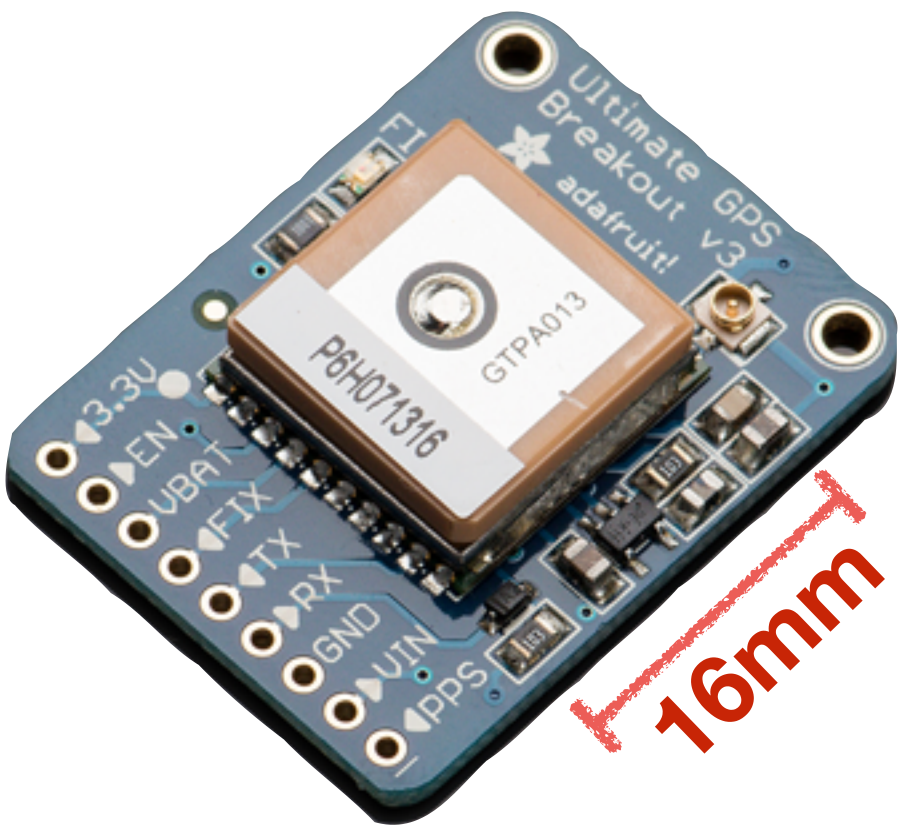


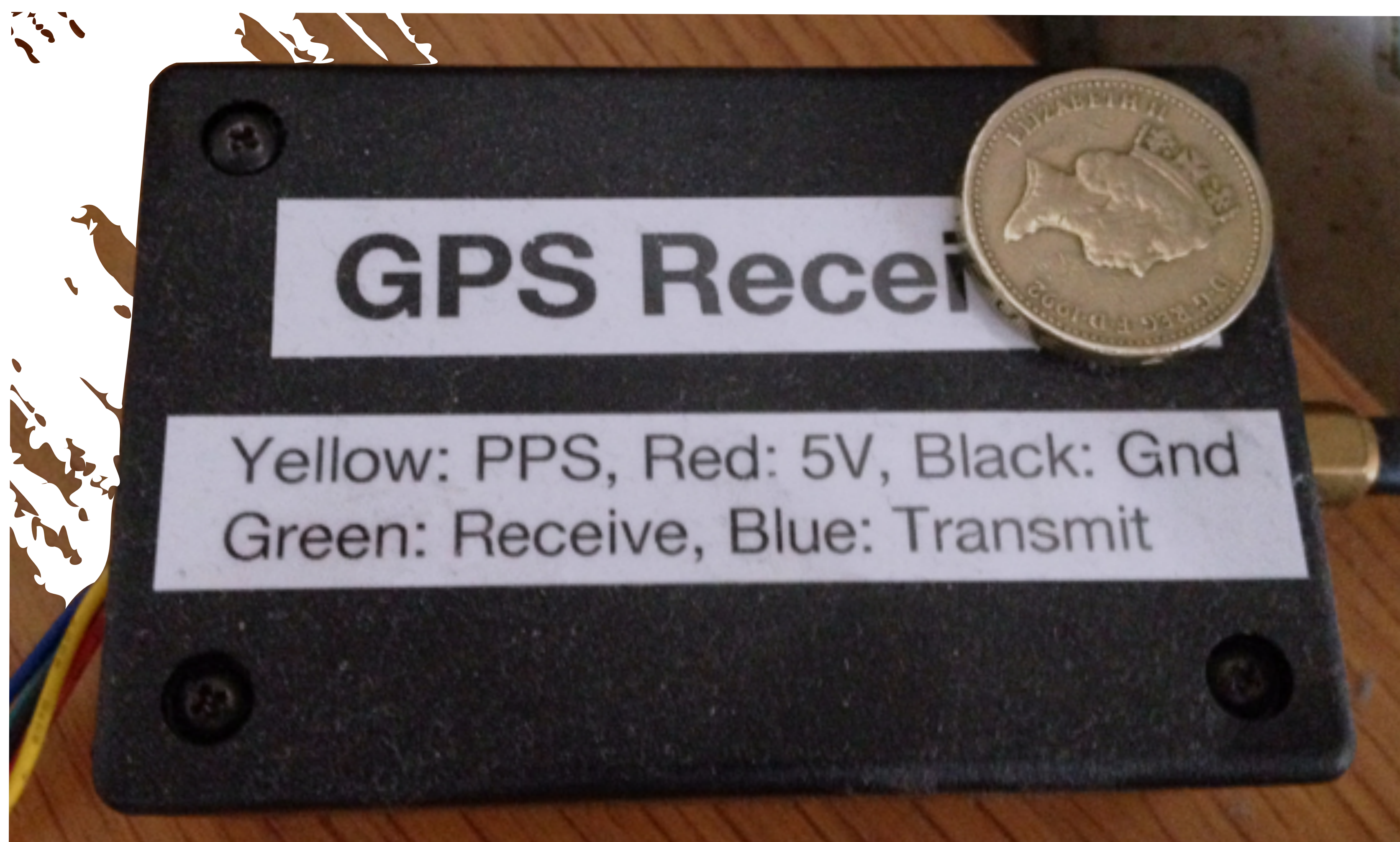
Compass

Menu

GPS Today

16mm x 16mm, $15 quantity one, built-in aerial, includes 1pps output for timing, DGPS, operates on 3.3V





NMEA

- ~100m precision was of limited use for terrestrial navigation, other than as a backup for a map and compass.
- But is fantastically more accurate than you can achieve with a sextant and clock from a moving ship or boat, and significantly better than LORAN and other maritime navigation systems (most of which dated back to WW2).
- So the maritime community got in first with GPS to computer integration. And it’s like 1990s never finished.

NMEA “Sentences”

- Physical NMEA is not of relevance to us
  - It’s a really weird version of RS232, at strange speeds and voltages, designed to work in electrically noisy boats).
- But the structure is still used by GPS receivers, and Smartphones have to deal with it directly.

7.7m

```
$GPGGA,111130.000,5225.0094,N,00157.5017,W,2,7,1.27,145.1,M,48.2,M,0000,0000*4F
$GPGGA,111146.000,5225.0137,N,00157.5007,W,2,8,1.15,146.1,M,48.2,M,0000,0000*4A
$GPGGA,111202.000,5225.0180,N,00157.4993,W,2,8,1.15,146.7,M,48.2,M,0000,0000*46
$GPGGA,111218.000,5225.0219,N,00157.4969,W,2,8,1.15,148.6,M,48.2,M,0000,0000*44
$GPGGA,111234.000,5225.0240,N,00157.4974,W,2,8,1.15,149.4,M,48.2,M,0000,0000*49
$GPGGA,111250.000,5225.0254,N,00157.4989,W,2,8,1.15,150.0,M,48.2,M,0000,0000*40
$GPGGA,111306.000,5225.0257,N,00157.5014,W,2,8,1.15,150.5,M,48.2,M,0000,0000*48
$GPGGA,111322.000,5225.0257,N,00157.5015,W,2,8,1.15,151.1,M,48.2,M,0000,0000*4A
$GPGGA,111338.000,5225.0278,N,00157.4997,W,2,8,1.14,151.7,M,48.2,M,0000,0000*49
$GPGGA,111354.000,5225.0297,N,00157.4984,W,2,8,1.14,152.8,M,48.2,M,0000,0000*4C
```

Note Drift

Sentence Identifier	$GPGGA	Global Positioning System Fix Data
Time	170834	17:08:34 Z
Latitude	4124.8963, N	41d 24.8963' N or 41d 24' 54" N
Longitude	08151.6838, W	81d 51.6838' W or 81d 51' 41" W
Fix Quality: 0 = Invalid 1 = GPS fix 2 = DGPS fix	1	Data is from a GPS fix
Number of Satellites	5	5 Satellites are in view
Horizontal Dilution of Precision (HDOP)	1.5	Relative accuracy of horizontal position
Altitude	280.2, M	280.2 meters above mean sea level
Height of geoid above WGS84 ellipsoid	-34.0, M	-34.0 meters
Time since last DGPS update	blank	No last update
DGPS reference station id	blank	No station id
Checksum	*75	Used by program to check for transmission errors

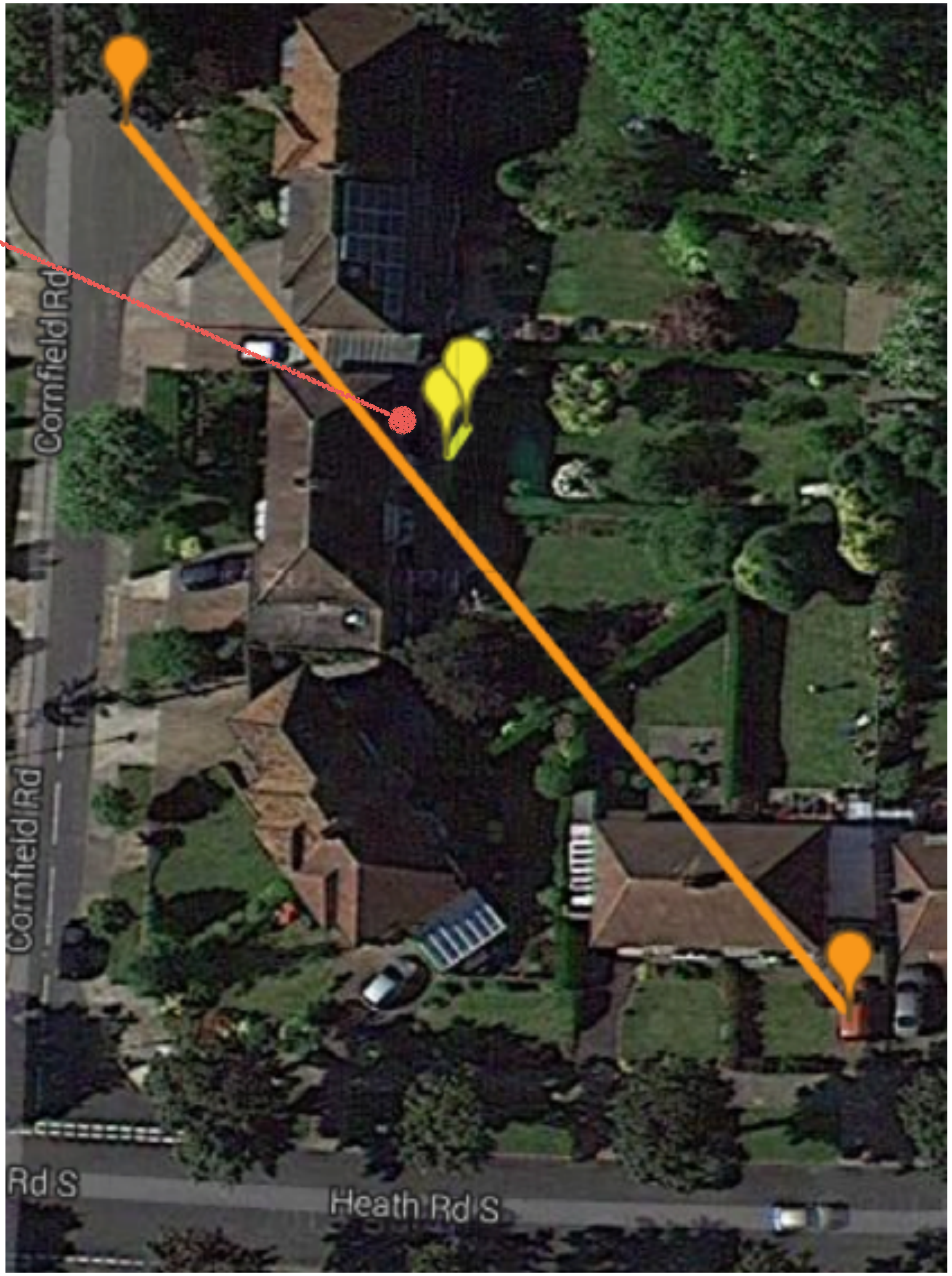
Real-World Accuracy

Antenna is really here

This is a GPS receiver that is in a fixed location, with Differential GPS, and hasn't been moved for some months. Note the accuracy is still not good enough for (say) surveying land boundaries.

3 Minute Wander

12 Hour Wander



NMEA Processing

```
NmeaListener nmeaListener = new NmeaListener(){
        @Override
        public void onNmeaReceived(long timestamp, String nmea) {
            Log.w ("gps", nmea);
            if (nmea.startsWith ("$G")) {
                String[] fields = nmea.split (",|\\*");

                if (fields[0].equals ("$GPGGA")) {
                    try {
                        here = new MyLocation (fields[2], fields[3],
                                               fields[4], fields[5],
                                               fields[9], fields[1]);
                    } catch (Exception e) {
                        here = new MyLocation ();
                    }
                } else if (fields[0].equals ("$GPRMC")) {
                    // nothing
```

Installing the handler

```
location = (LocationManager) this.getSystemService(Context.LOCATION_SERVICE);


protected void onResume () {
    super.onResume ();

    try {
        if (!listenersRegistered) {
            Sensor countSensor = sensor.getDefaultSensor (Sensor.TYPE_STEP_COUNTER);
            if (countSensor != null) {
                sensor.registerListener(sensorListener, countSensor,
                                        SensorManager.SENSOR_DELAY_NORMAL);
            }
            location.requestLocationUpdates(LocationManager.GPS_PROVIDER, 1 , 1, locationListener);
            location.addNmeaListener (nmeaListener);
            startRepeatingTask ();
            wl.acquire ();
            listenersRegistered = true;
        }
        gps_status = "init OK";
    } catch (Exception e) {
        gps_status = "bad init";
    }
}
```

Alternatively

```
LocationListener locationListener = new LocationListener(){
        @Override
        public void onLocationChanged(Location location) {
            // TODO Auto-generated method stub
            ;
        }

        @Override
        public void onProviderDisabled(String provider) {
            gps_status = "disabled " + provider;
            // TODO Auto-generated method stub

        }

        @Override
        public void onProviderEnabled(String provider) {
            gps_status = "enabled " + provider;
            // TODO Auto-generated method stub

        }

        @Override
        public void onStatusChanged(String provider, int status, Bundle extras) {
            gps_status = "changed " + provider;
            // TODO Auto-generated method stub

        }
    };
```

Another sensor

- Nexus 5 has a pedometer, just like the £5 device you can buy to monitor how much you exercise.
- Avoids having to integrate an accelerometer, runs in a low-power state (just returns a count when asked)

Sensor Framework

```
SensorEventListener sensorListener = new SensorEventListener () {
        @Override
        public void onSensorChanged (SensorEvent event) {
            steps = (int) Float.parseFloat (String.valueOf (event.values[0]));
        }
        @Override
        public void onAccuracyChanged (Sensor sensor, int accuracy) {
        }
    };
```

Installing the handler

```
        sensor = (SensorManager) getSystemService (Context.SENSOR_SERVICE);

protected void onResume () {
    super.onResume ();

    try {
        if (!listenersRegistered) {
            Sensor countSensor = sensor.getDefaultSensor (Sensor.TYPE_STEP_COUNTER);
            if (countSensor != null) {
                sensor.registerListener(sensorListener, countSensor,
                                        SensorManager.SENSOR_DELAY_NORMAL);
            }
            location.requestLocationUpdates(LocationManager.GPS_PROVIDER, 1 , 1, locationListener);
            location.addNmeaListener (nmeaListener);
            startRepeatingTask ();
            wl.acquire ();
            listenersRegistered = true;
        }
        gps_status = "init OK";
    } catch (Exception e) {
        gps_status = "bad init";
    }
}
```