Much more than lat/lon

A guided adventure through location in Android
Android

Google Maps, etc.
Places,
Activity Recognition,
Geofencing, FLP,
Sensor &
GNSS Raw Measurements
Abstraction to the physical
Hardware
Physical Drivers

Higher levels of abstraction

Raw measurements begin here

GNSS, WiFi, BLE, and Sensor chips live down here
Selected Public methods

- `getAccumulatedDeltaRangeMeters()`
- `getCarrierCycles()`
- `getCarrierFrequencyHz()`
- `getCarrierPhase()`
- `getCn0DbHz()`
- `getConstellationType()`
- `getPseudorangeRateMetersPerSecond()`
- `getReceivedSvTimeNanos()`
- `getSnrInDb()`
- `getSvid()`
Example

GNSS Logger

for logging raw

GNSS measurements
Colorful lines come from delta-pseudorange. Gray lines come from the measured Doppler: “PseudorangeRateMetersPerSecond”.

<table>
<thead>
<tr>
<th>Svid</th>
<th>PseudorangeRateMetersPerSecond</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>-484.2007006282</td>
</tr>
<tr>
<td>6</td>
<td>-242.9837335047</td>
</tr>
<tr>
<td>12</td>
<td>-371.3782264499</td>
</tr>
<tr>
<td>13</td>
<td>761.5554497739</td>
</tr>
<tr>
<td>15</td>
<td>578.0615023675</td>
</tr>
<tr>
<td>17</td>
<td>538.242900247</td>
</tr>
<tr>
<td>19</td>
<td>288.7567963982</td>
</tr>
<tr>
<td>24</td>
<td>-198.4411495391</td>
</tr>
<tr>
<td>28</td>
<td>475.0425059834</td>
</tr>
<tr>
<td>30</td>
<td>846.2004575168</td>
</tr>
</tbody>
</table>
Some useful analysis is now easy, just by looking:

For example: what’s up with the measurements for Svid: 13,15 and 2?
Common bias and frequency states from WLS

Clock rate is \(\approx 0.5 \text{ ppm}\), typical of a TCXO.
public final class GnssNavigationMessage
extends Object implements Parcelable

java.lang.Object
- android.location.GnssNavigationMessage

getData

byte[] getData()

Gets the data of the reported GPS message.

The bytes (or words) specified using big endian format (MSB first).

- For GPS L1 C/A, Beidou D1 & Beidou D2, each subframe contains 10 30-bit words. Each word (30 bits) should be fit into the last 30 bits in a 4-byte word (skip B31 and B32), with MSB first, for a total of 40 bytes, covering a time period of 6, 6, and 0.6 seconds, respectively.

- For Glonass L1 C/A, each string contains 85 data bits, including the checksum. These bits should be fit into 11 bytes, with MSB first (skip B86-B88), covering a time period of 2 seconds.

- For Galileo F/NAV, each word consists of 238-bit (sync & tail symbols excluded). Each word should be fit into 30-bytes, with MSB first (skip B239, B240), covering a time period of 10 seconds.

- For Galileo I/NAV, each page contains 2 page parts, even and odd, with a total of 2x114 = 228 bits, (sync & tail excluded) that should be fit into 29 bytes, with MSB first (skip B229-B232).
Decoded Nav data, in GnssLogger:

And in log file:

```plaintext
# Header Description:
#
# Version: 1.4.0.0, Platform: N
#
# Nav, Svid, Type, Status, MessageId, Sub-messageId, Data(Bytes)
Nav, 2, 257, 1, 0, 3, 34, -61, 121, 25, 12, -108, 107, 35, 0, 33, -42, 115, 35, 46, -77, -78, 63, -5, -55, -81, 29, 76, 25, -91, 8, -23, 106, -113, 14, -98, 104, 52, 51, -111, -36, -78, 63, -107, 66, 47, 9, 120, -1, -100, 2, 53, -24, 112, 3, 63, -84, -97, 9, -85, 31, -49
Nav, 12, 257, 1, 0, 3, 34, -61, 121, 25, 12, -108, 107, 35, 0, 33, -42, 115, 35, 46, -77, -78, 63, -5, -55, -81, 29, 76, 25, -91, 8, -23, 106, -113, 14, -98, 104, 52, 51, -111, -36, -78, 63, -107, 66, 47, 9, 120, -1, -100, 2, 53, -24, 112, 3, 63, -84, -97, 9, -85, 31, -49
Nav, 98, 769, 1, 0, 1, 8, 87, -128, 22, -95, 96, -81, -109, -100, 30, -104
```

GnssLogger settings interface is also shown in the image.
Raw GNSS Measurements – uses:

• Scientific & pedagogical.
• Improved PNT: DGNSS, PPP, RTK?
• Improved phones - system testing.
• Jammer detection & location.
GNSS, WiFi, BLE, and Sensor chips live down here

Google Maps, etc.

Places,

Activity Recognition,

Geofencing, FLP,

Sensor &

GNSS Raw Measurements

Abstraction to the physical Hardware

Physical Drivers
### Sensor data

<table>
<thead>
<tr>
<th>int</th>
<th>TYPE_GYROSCOPE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A constant describing a gravity sensor type.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>int</th>
<th>TYPE_GYROSCOPE_UNCALIBRATED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A constant describing an uncalibrated gyroscope sensor type.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>int</th>
<th>TYPE_HEART_BEAT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A constant describing a motion detect sensor.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>int</th>
<th>TYPE_HEART_RATE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A constant describing a heart rate monitor.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>int</th>
<th>TYPE_LIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A constant describing a light sensor type.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>int</th>
<th>TYPE_LINEAR_ACCELERATION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A constant describing a linear acceleration sensor type.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>int</th>
<th>TYPE_MAGNETIC_FIELD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A constant describing a magnetic field sensor type.</td>
</tr>
</tbody>
</table>

Get this data in a similar way to how you get the GNSS data. There are many freely available apps for logging sensor data.
Activity Recognition

Example:
Detected activities

**Fitness**
- Walking
- Running
- Biking
- Stairs
- Push-ups, Sit-ups, Squats
- Strength training (Curls etc.)

**Device Context**
- In-Vehicle
- Personal Vehicle
- Exiting Vehicle
- Still
- On/Off Body

**Gestures**
- Double Twist
- Tilt-to-wake
Double-twist for selfies
Clustering data

This is Walking & Running
## Detected activities

<table>
<thead>
<tr>
<th>Fitness</th>
<th>Device Context</th>
<th>Gestures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walking</td>
<td>In-Vehicle</td>
<td>Double Twist</td>
</tr>
<tr>
<td>Running</td>
<td>Personal Vehicle</td>
<td>Tilt-to-wake</td>
</tr>
<tr>
<td>Biking</td>
<td>Exiting Vehicle</td>
<td></td>
</tr>
<tr>
<td>Stairs</td>
<td>Still</td>
<td></td>
</tr>
<tr>
<td>Push-ups, Sit-ups, Squats</td>
<td>On/Off Body</td>
<td></td>
</tr>
<tr>
<td>Strength training (Curls etc.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Custom exercise</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Google
WiFi: RSSI and RTT (IEEE 802.11mc)

RSSI

Accuracy: 30 to 50 m

RTT

2*distance = ((t4-t1) - (t3-t2))*c

Accuracy: 1 to 3 m
Android Emergency Location, deployed across UK

Network Location

Android Emergency Location
FLP - Fused Location Provider

FLP

Location

GPS

Network Location (NLP)

Accel Gyro Mag Baro

GNSS
Geofencing API

Creates fences defined by lat/lon and radius

KEY FEATURES
● Up to 100 active geofences per app
● Reports: entrance, exit and dwell

EXAMPLES
● Smartlock: unlock phone at home, lock when away

VERY LOW POWER
● On newer phones (especially Pixel):
  ● Less than 1% impact on battery life.
A beacon is a physical analog to a geofence, but especially suitable for close proximity, and indoors.

Eddystone is an open format to enable a BLE beacon to broadcast:

- A unique ID number
  - Including cryptographically secure frames
- A URL
- Telemetry data
  - E.g. low battery data, for managing fleets of beacons
Nearby

**Bluetooth**
- 2.4 GHz radio
- Range to ~100 feet
- And goes through walls

**Ultrasound**
- 18.5-20 kHz audio
- Range to ~5 feet
- Doesn’t go through walls
Ultrasound
Higher levels of abstraction

Physically

GNSS, WiFi, BLE, and Sensor chips

Abstraction to the physical Hardware

Sensor & GNSS Raw Measurements

Activity Recognition, Geofencing, FLP,

Places

Applications

Android Framework

Native Libraries

Android Runtime

Linux Kernel

Physical Drivers
example:
meet me here
GetCurrentPlace

Android
PlaceDetectionApi
GetCurrentPlace

Android
PlaceDetectionApi

Morocco's Restaurant
GetCurrentPlace

Output Predictions

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.45</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.10</td>
<td></td>
</tr>
</tbody>
</table>

Morocco’s Restaurant
“We’re at Morocco’s Restaurant”
Places *
Tango
Nearby
Ultrasound
Beacons
Geofences
Fused Location
Emergency Location
Indoor Location
Activity Recognition *
Sensor Measurements
GNSS Raw Measurements

* Examples of Machine Learning
Thank You