CONNECTING ANY SMARTPHONE ANDROID GIS AND GNSS APP TO EXTERNAL HIGH PRECISION GNSS RECEIVERS USING MOCK LOCATION.

Prof. Joël van Cranenbroeck, Managing Director
CGEOS CREATIVE GEOSENSING SPRL
Belgium, Europe

There are just more and more applications available for Smartphone and Tablet running Android OS. They are using positions from the internal GNSS chipset and WIFI and 4G network that are not accurate (generally around 10 meters depending on the multipath environment, obstruction and the body of the operator that can block the GNSS signals) but for more and more applications, high precision positioning is requested.

It can be either by using SBAS services such EGNOS or GNSS Network RTK such WALCORS and FLEPOS in Belgium. In that case sub-meter or even Float and Fixed RTK is obtained by survey grade GNSS receiver such the SinoGNSS G100, G200, T300, T30 and N5 IMU RTK by ComNav Technology Ltd. Shanghai, PR CHINA, to deliver cm accuracy positioning.

If you want to use the G100 with for instance MapIt Application http://mapit-gis.com, you do need first to configure the G100 to operate in Differential/RTK mode by using SURVEY MASTER application from ComNav Technology Ltd. Shanghai - freely available for Android OS devices – and then activate “Mock Location” function.

Mock Location is a virtual GPS simulator generated by Android R&D to help developers using a Smartphone that doesn’t have a GNSS Chipset to check their developments.

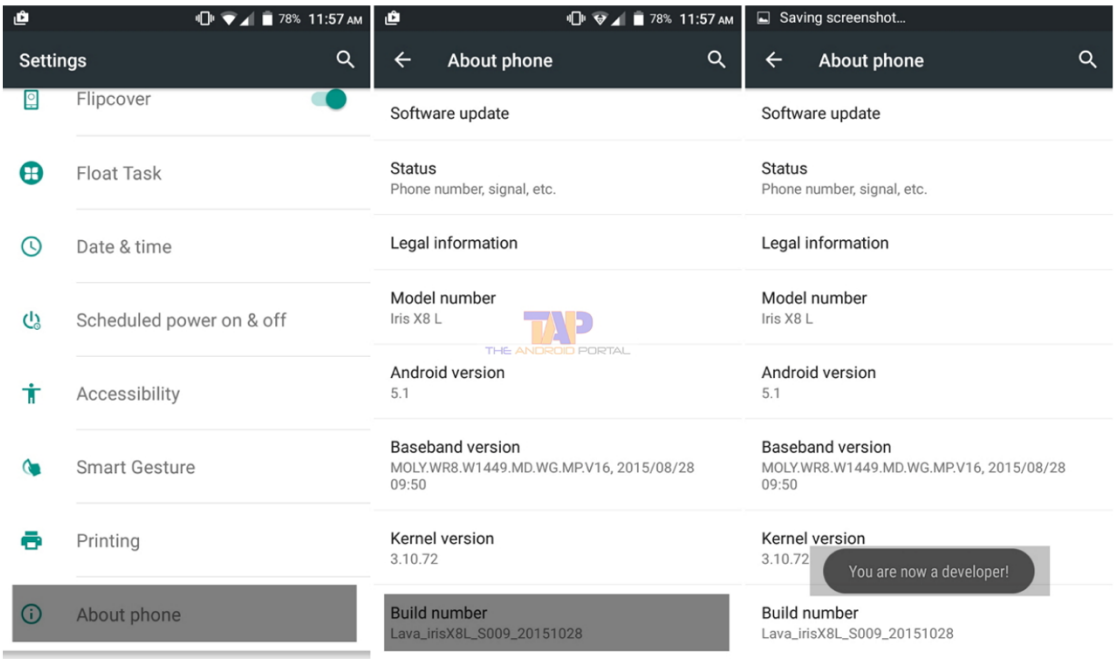
By enabling “mock location” your Smartphone will generate fake positions. If you allow SURVEY MASTER to be linked with “mock location”, then the external GNSS receiver will be used instead of the Smartphone GNSS chipset.

In summary:
You must enable “mock location” on your Smartphone to force an external GNSS to be paired with. Otherwise, the Smartphone will use its internal GNSS and WIFI as primary LBS device.

To enable or disable mock location on your phone, you have first to **enable developer option** on your Android. By default, developer option is not enabled on any of device. Users have to enable the developer option manually.

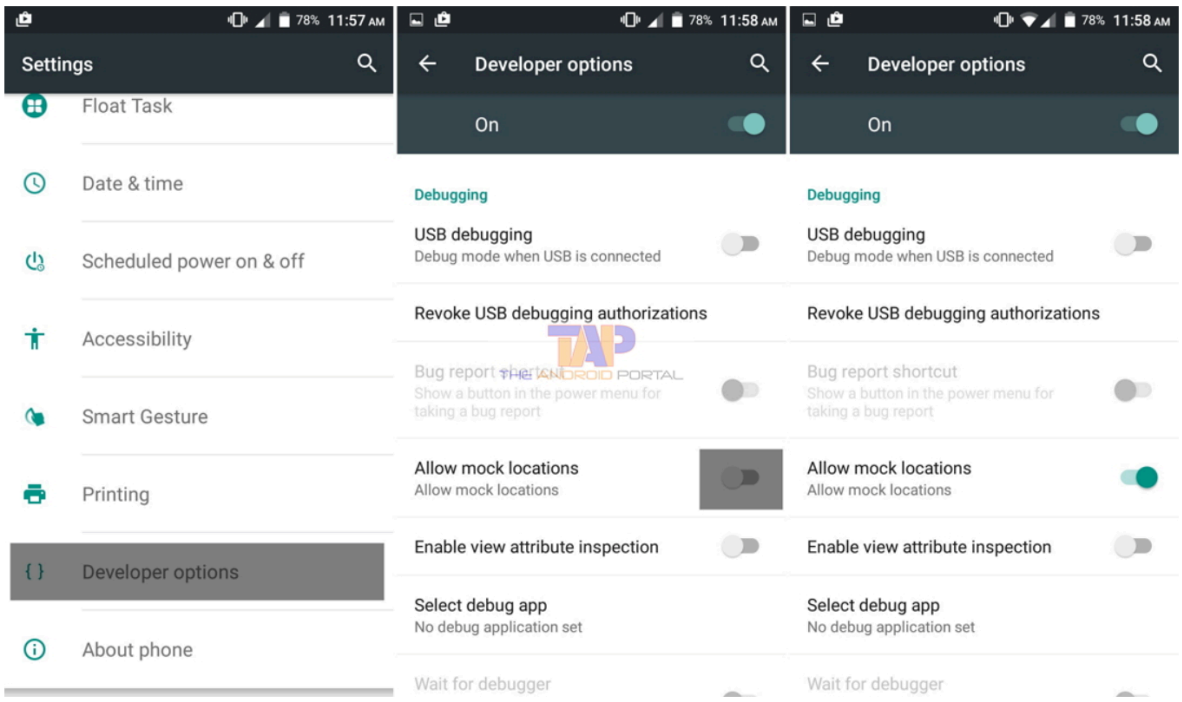
How to enable Developer option on any smartphone

1. First launch the Setting on your phone
2. Scroll down to bottom and select About Phone
3. In the About Phone section, Scroll down and find Build Number options
4. Now Click on seven times (7) on Build Number options
5. After seven times tapping on it, you will see small popup message on your screen “You are now a Developer!”
6. After this message, you can access developer option on your phone. You can now be able to access your mock locations on your phone and enable/disable it. Follow these steps to enable and disable mock location on your Android Smartphone.

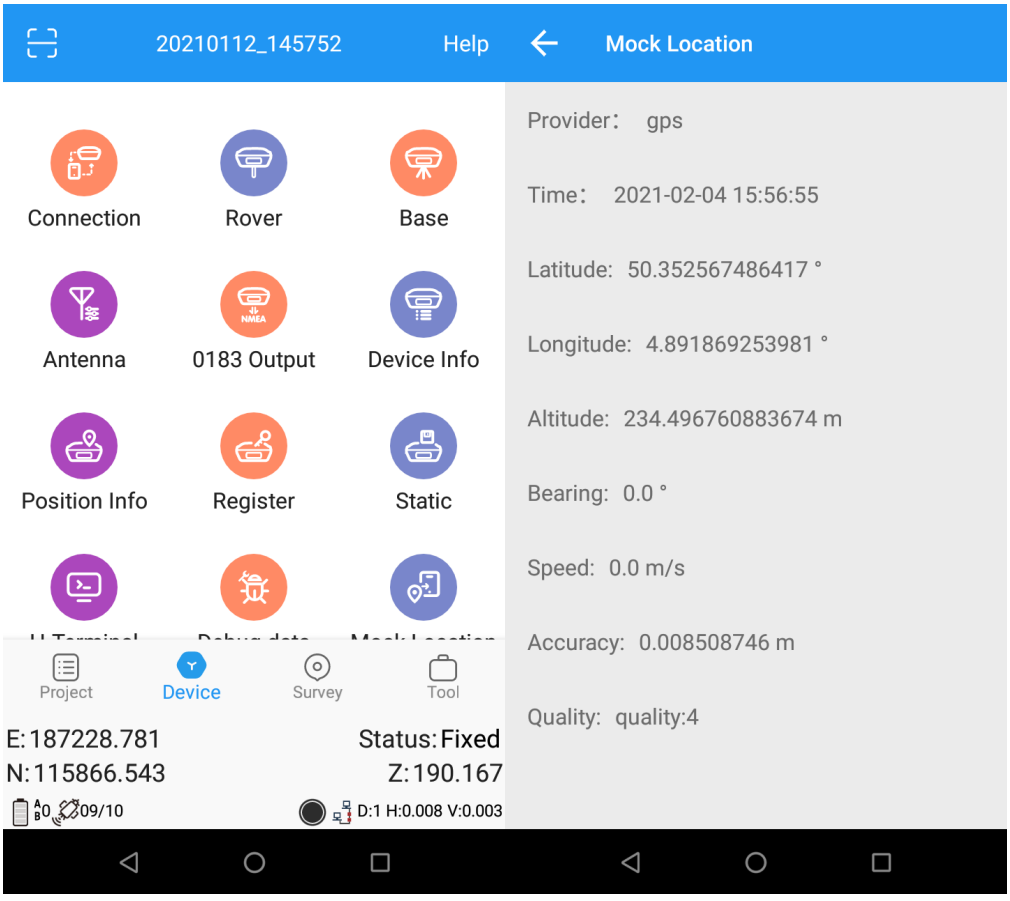


How to Turn Off Mock Locations on your Android

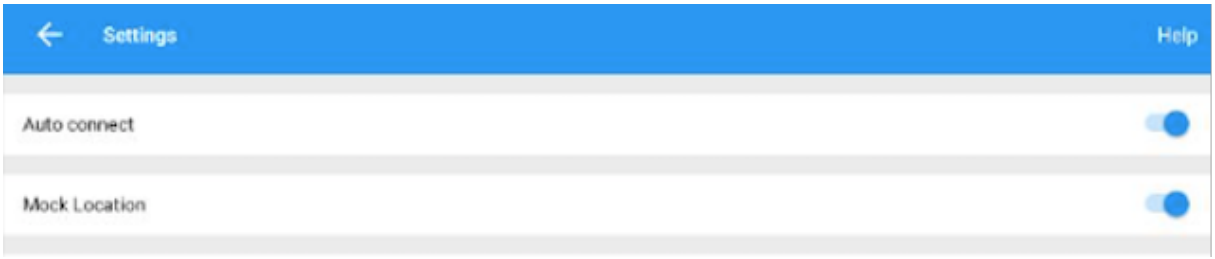
1. First, Open “Setting” app on your phone
2. Scroll down and select “Developer Options” and open it.
3. In the Developer Options, Scroll down and find “Allow Mock Location” option
4. Here you can enable and disable Mock Location using “Slider” button or assign Mock Location to a given app.



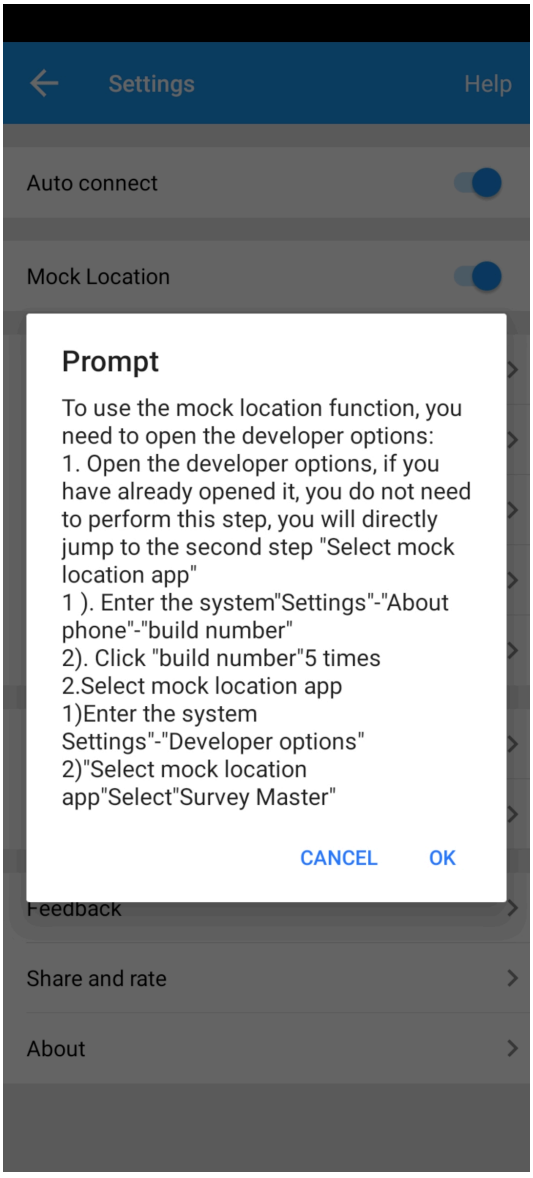
Start SURVEY MASTER, connect by BT the GNSS receiver, apply for GNSS CORS corrections and check the settings to use “Mock Location”, then in the Device TAB engage “Mock Location”. A panel will pop up to summarize the necessary steps.



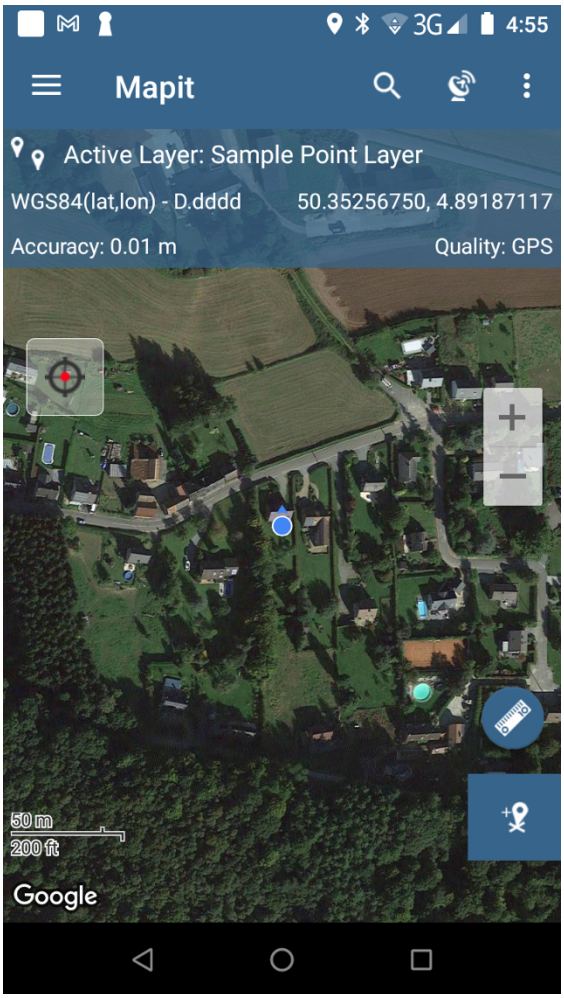
SURVEY MASTER in the Devise Tab, allows to switch “Mock Location ON” and check the settings

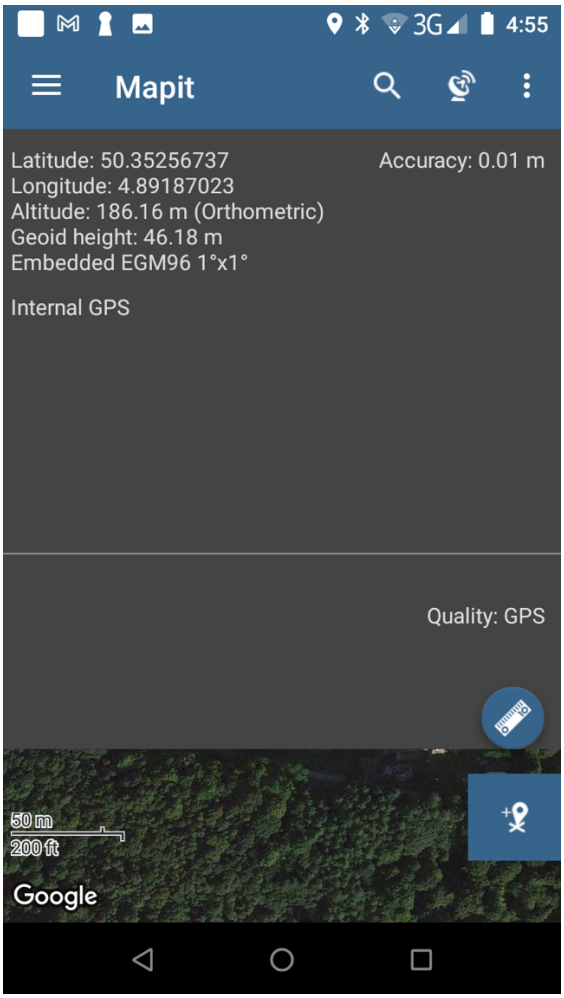


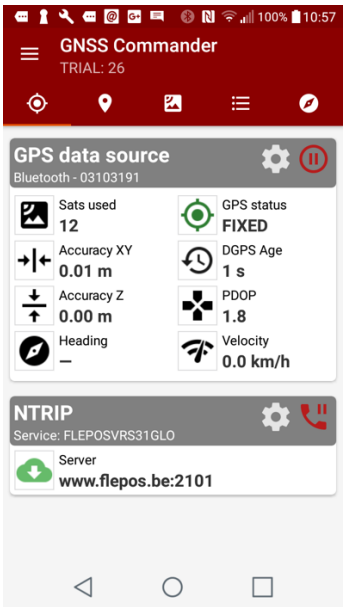
If the setting is not properly configure, “Mock Location” will refer to the Smartphone GNSS chipset with an accuracy of ± 10 meters.

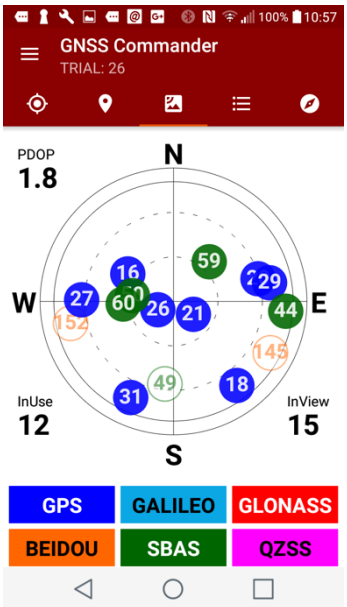


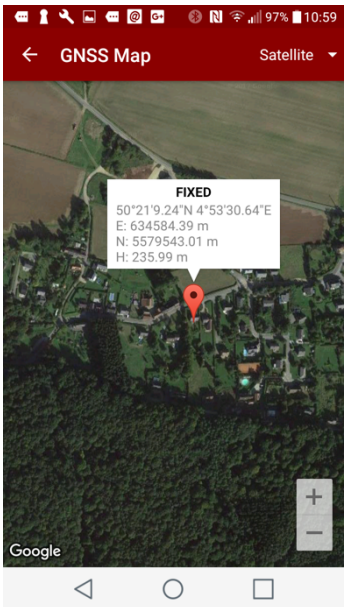
Now any application using the Smartphone location will get high precision GNSS fixes.











Some applications however using API, are not connecting to the GNSS coordinates but well on the LBS coordinates and therefore location is based on LBS (WIFI and Telecom networks).

Set up a location request

To store parameters for requests to the fused location provider, create a `LocationRequest`. The parameters determine the level of accuracy for location requests. For details of all available location request options, see the `LocationRequest` class reference. This lesson sets the update interval, fastest update interval, and priority, as described below:

Update interval

`setInterval()` - This method sets the rate in milliseconds at which your app prefers to receive location updates. Note that the location updates may be somewhat faster or slower than this rate to optimize for battery usage, or there may be no updates at all (if the device has no connectivity, for example).

Fastest update interval

`setFastestInterval()` - This method sets the **fastest** rate in milliseconds at which your app can handle location updates. Unless your app benefits from receiving updates more quickly than the rate specified in `setInterval()`, you don't need to call this method.

Priority

`setPriority()` - This method sets the priority of the request, which gives the Google Play services location services a strong hint about which location sources to use. The following values are supported:

- `PRIORITY_BALANCED_POWER_ACCURACY` - Use this setting to request location precision to within a city block, which is an accuracy of approximately 100 meters. This is considered a coarse level of accuracy, and is likely to consume less power. With this setting, the location services are likely to use WiFi and cell tower positioning. Note, however, that the choice of location provider depends on many other factors, such as which sources are available.
- `PRIORITY_HIGH_ACCURACY` - Use this setting to request the most precise location possible. With this setting, the location services are more likely to use GPS to determine the location.
- `PRIORITY_LOW_POWER` - Use this setting to request city-level precision, which is an accuracy of approximately 10 kilometers. This is considered a coarse level of accuracy, and is likely to consume less power.
- `PRIORITY_NO_POWER` - Use this setting if you need negligible impact on power consumption, but want to receive location updates when available. With this setting, your app does not trigger any location updates, but receives locations triggered by other apps.

And also consider the following:

LocationProvider

Added in API level 1

Kotlin | **Java**

```
public class LocationProvider
extends Object
```

java.lang.Object
↳ android.location.LocationProvider

An abstract superclass for location providers. A location provider provides periodic reports on the geographical location of the device.

Each provider has a set of criteria under which it may be used; for example, some providers require GPS hardware and visibility to a number of satellites; others require the use of the cellular radio, or access to a specific carrier's network, or to the internet. They may also have different battery consumption characteristics or monetary costs to the user. The `Criteria` class allows providers to be selected based on user-specified criteria.

In summary, “Mock Location” is the mechanism to connect all Android apps to an external GNSS high precision receiver.

Many thanks for your interest in our GNSS solutions.

Prof. Joël van Cranenbroeck
CGEOS Creative Geosensing SPRL
Rue du Tienne de Mont, 11
BE-5530 Mont, BELGIUM
Mobile +32 474 98 61 93
www.cgeos.be
E-mail cgeos2014@gmail.com