

**Dati raw GNSS da
dispositivi Android: analisi
delle performance del
dispositivo xiaomi mi 9 e
sviluppo di una nuova app
open source per
l'acquisizione dei dati**

Lorenzo Benvenuto^{1,2}, Manuel Viaggi², Tiziano Cosso¹, Giorgio Delzanno²

1)

Gter **Innovazione**
in Geomatica, Gnss e Gis

2)

Dibris
www.dibris.unige.it



Torino, 20 Febbraio 2020

Indice

- **Dati grezzi GNSS da smartphone**
- **Test su dispositivo Xiaomi mi 9**
- **GNSS raw: app FOSS per la raccolta dati**

Dati Grezzi GNSS da Smartphone

Tipologie di ricevitori GNSS comunemente utilizzati



Ricevitori Geodetici

Tipicamente in doppia frequenza

Accuratezza nell'ordine dei cm

Molto costosi (<10000 €)

Ricevitori Mass Market

Tipicamente in singola frequenza

Accuratezza nell'ordine del metro

Poco costosi (100-300 €)

Smartphone

Singola ma anche doppia frequenza

Accuratezza metrica? (molto variabile)

Prezzo 300-900 €

Dati Grezzi GNSS da Smartphone

Da Maggio 2016 Google ha annunciato la disponibilità dei dati grezzi GNSS da dispositivi Android.

In particolare si ha accesso a:

GNSS Clock

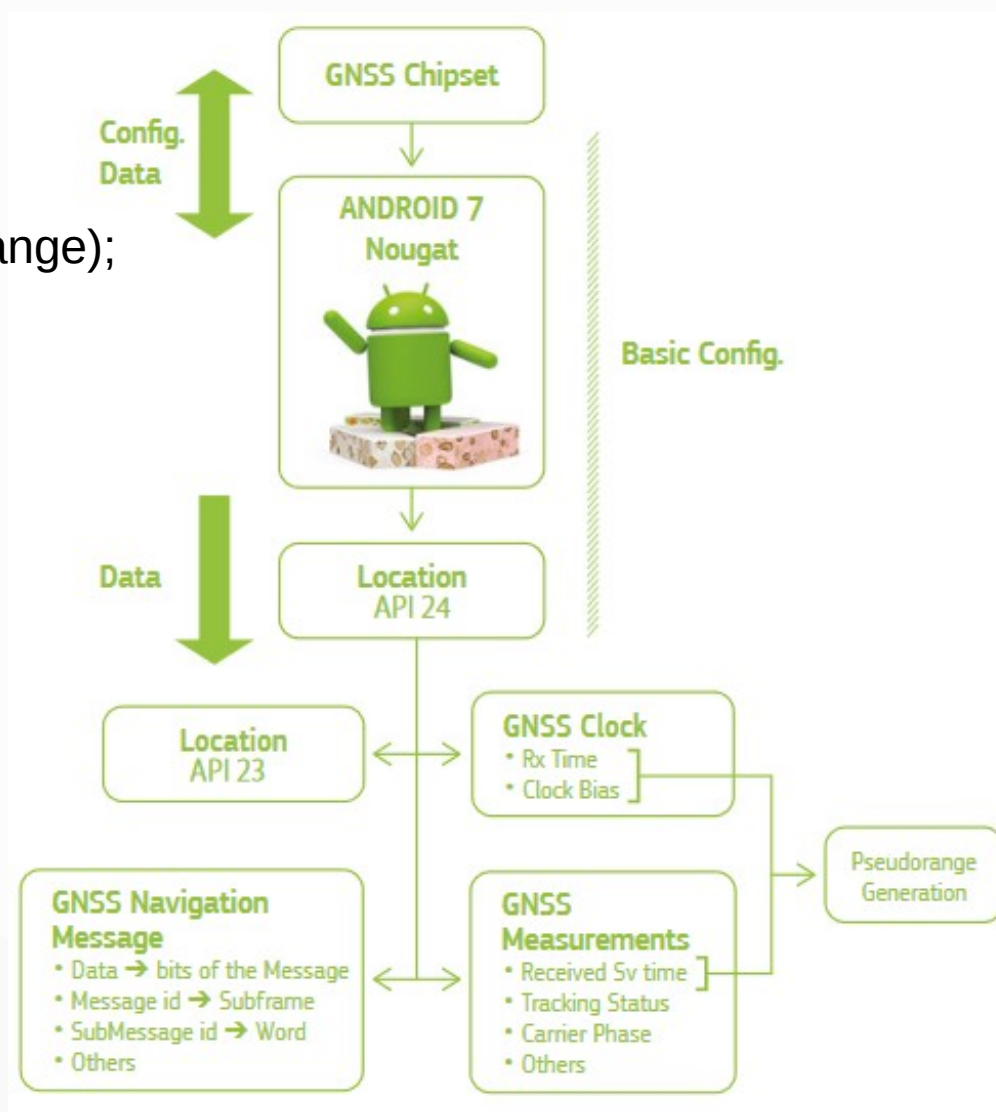
- Receiver time (usato per gli pseudorange);
- Clock bias

GNSS Navigation Message

- Navigation Message bits;
- Navigation message status.

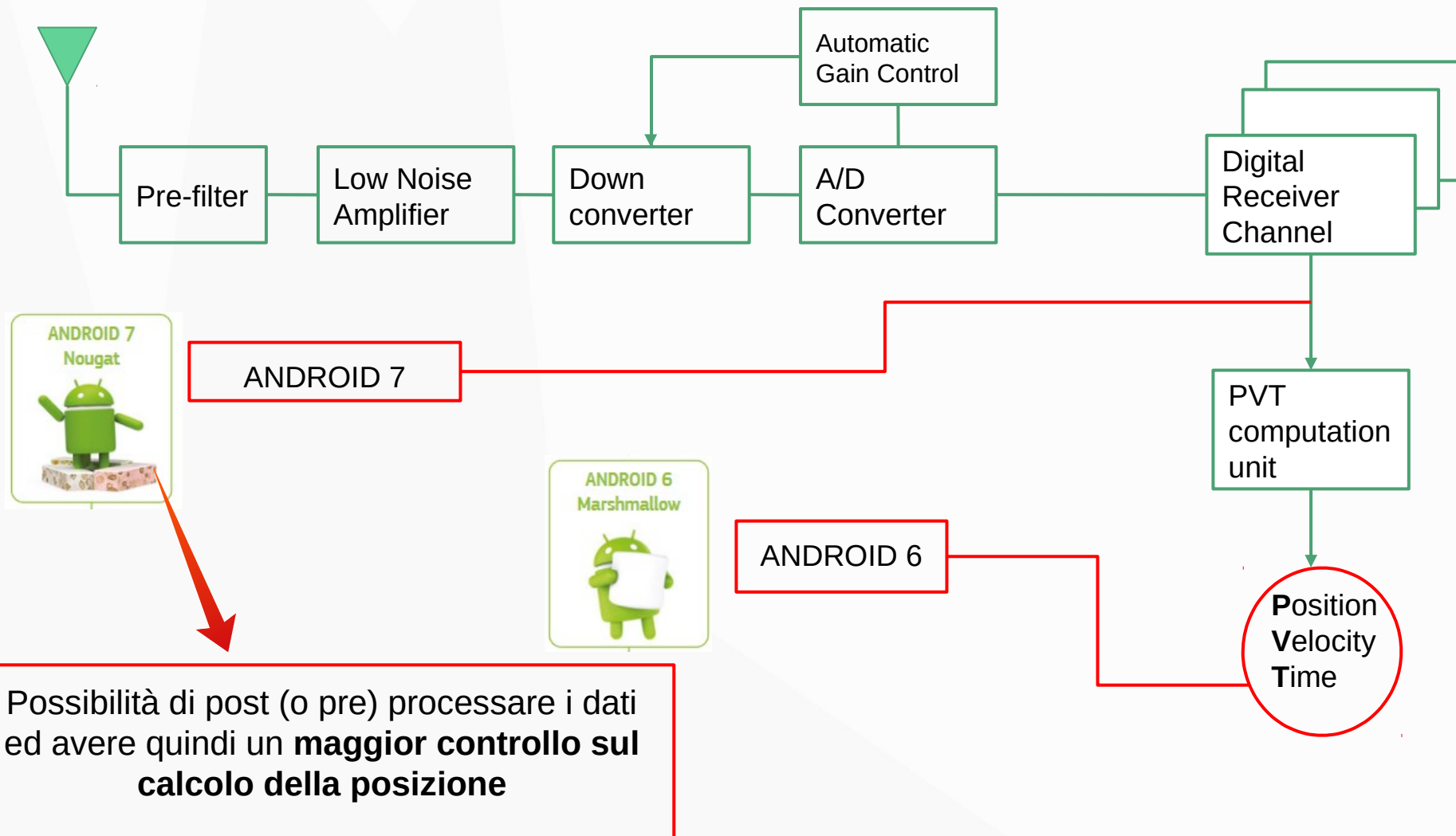
GNSS Measurement

- Received Satellite Time (usato per gli pseudorange);
- Doppler;
- Carrier phase.



Dati Grezzi GNSS da Smartphone

In altre parole...



Dati Grezzi GNSS da Smartphone

Caratteristiche Hardware

La maggior parte degli smartphone è multiconstellazione (GPS, GLONASS, Galileo, BeiDou)

Tipicamente in singola frequenza, ma esistono sul mercato smartphone in doppia frequenza (L1/E1 - L5/E5a)

Vantaggi:

- ✓ Dimensioni ridotte e portabilità
- ✓ Diffusione capillare
- ✓ Buona potenza di calcolo (smartphone con < 3 GB RAM molto diffusi)

Svantaggi:

- ✗ Scarsa qualità dell'antenna
- ✗ Duty cycle – consumo batteria

Non tutti gli smartphone hanno accesso ai dati grezzi GNSS. Si può verificare se un dispositivo è abilitato al seguente [link](#)

Alternativamente è disponibile l'app GNSS Check [link](#)

Test su dispositivo Xiaomi mi 9

Scelta dispositivo:

Xiaomi mi8:



- Chip Broadcom BCM47755 (multicast. L1/L5). Tale chip è presente solo in questo dispositivo;
- Anno uscita 2018: minore durata del supporto (< livello API)

Xiaomi mi9:



- Chip Qualcomm Sanpdragon 855 (multicast. L1/L5). Tale chip è presente diversi dispositivi;
- Anno uscita 2019: maggior durata del supporto (> livello API)

Obiettivi:

- Caratterizzare le osservazioni del dispositivo xiaomi mi 9 (il cui processore è inglobato in diversi smatrphone sul mercato).
- Indagare la possibilità di intervenire sul pre-processing dei dati, con particolare attenzione al multipath per migliorare il posizionamento (in termini di accuratezza e integrità)

Test:

- Test1: acquisizione di dati grezzi da dispositivo xiaomi mi 9 con 4 app diverse
- Test2: acquisizione di 90 minuti da 3 diversi ricevitori e post processing con le osservazioni di una stazione permanente

Test su dispositivo Xiaomi mi 9

Test 1: Considerazioni

Modalità: 30 minuti di acquisizione dati con dispositivo xiaomi mi 9

Luogo: Tetto lab idraulica Università degli Studi di Genova

App utilizzate

- Geo++ Rinex logger, NSL RINEX On
- Google GNSS Logger, GADIP3



Formato RINEX

Formato csv

In entrambi i casi
mancano sia le **misure di fase**
che il **messaggio navigazionale**

Test su dispositivo Xiaomi mi 9

Test 1: Considerazioni

GNSS Measurement variables	Unit	Format
AccumulatedDeltaRangeState	[-]	int
AccumulatedDeltaRangeMeters	[m]	double
AccumulatedDeltaRangeUncertaintyMeters	[m]	double

Misure di fase → ADR (e relativa incertezza espressa in metri)

ADR State:

0: invalid or unknown

1: detected a rest

2: cycle slip detected

Test su dispositivo Xiaomi mi 9

Test 1: Considerazioni

GNSS Measurement variables	Unit	Format
MultipathIndicator	[-]	int

Indicatore di multipath:

0: the indicator is not available or the multipath is unknown

1: the measurement shows signs of multipath

2: the measurement no shows signs of multipath

Il **valore atteso** per le condizioni al contorno del test sarebbe stato **1**

Test su dispositivo Xiaomi mi 9

Test 2

Modalità: acquisizione dati da 3 ricevitori in parallelo

Luogo: cortile Villa Cambiaso, Università degli Studi di Genova



Topcon Hiper Pro

(Geodetico)

GPS, GLONASS

L1, L2



Ublox neo m8t

(Mass Market)

GPS, GLONASS

L1



Xiaomi mi9

(Smartphone)

GPS, GLONASS, Galileo, BeiDou

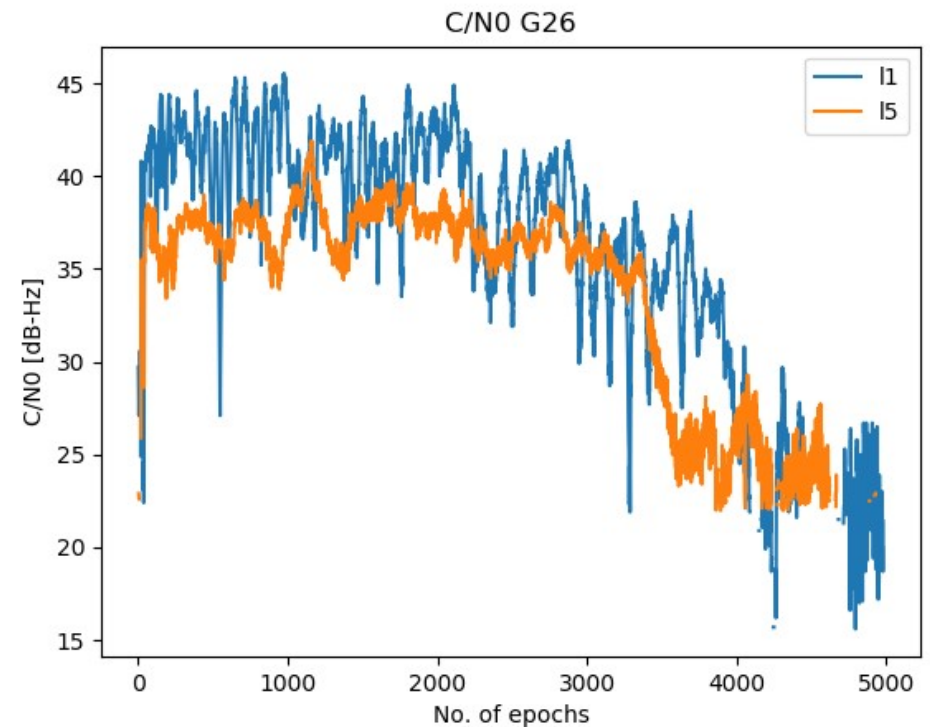
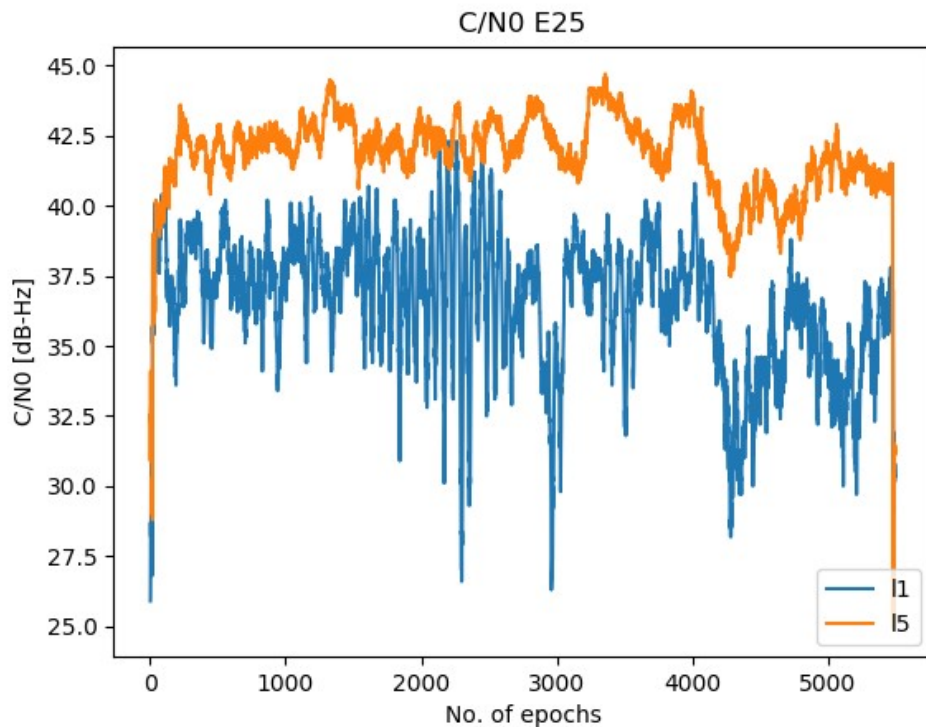
L1, L5



Test su dispositivo Xiaomi mi 9

Test 2: Considerazioni sul rapporto Sengale Rumore C/N₀

Confronto frequenze L1 e L5 (xiaomi mi9)

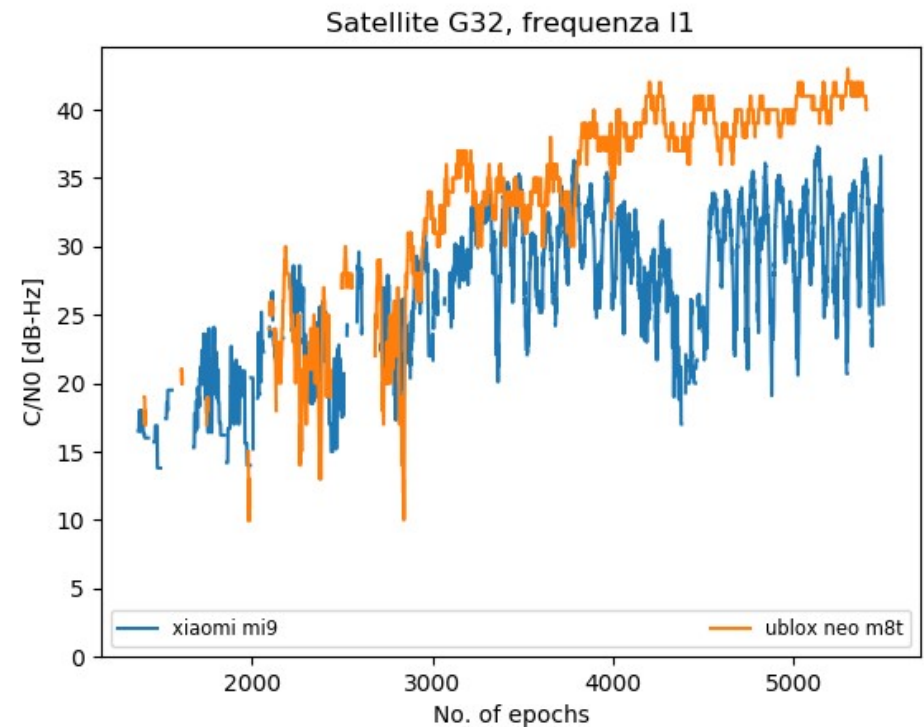
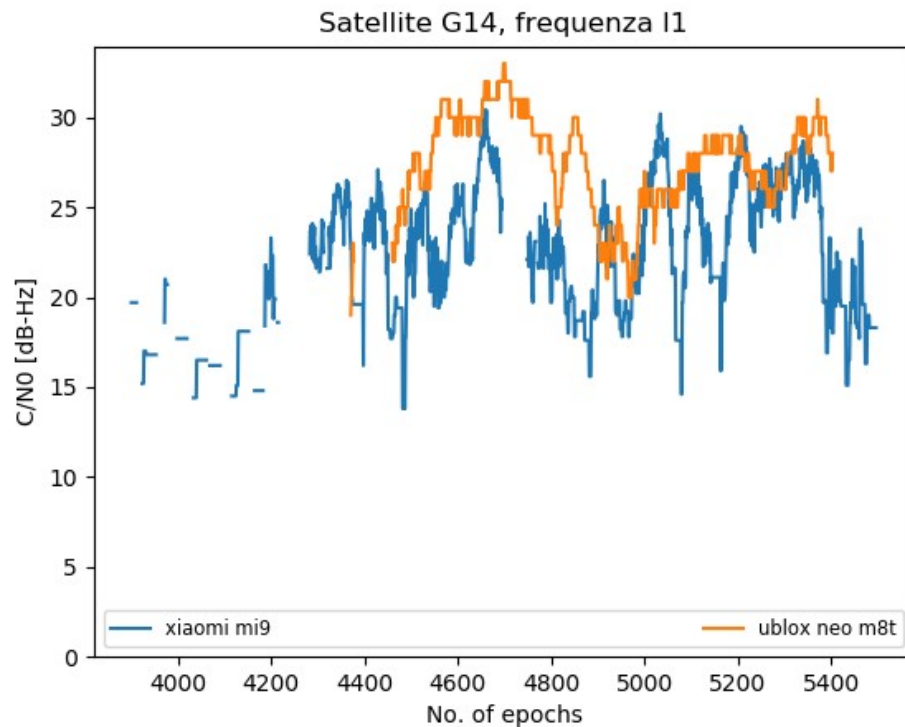


Frequenza L5 meno rumorosa rispetto a L1, quindi migliore per posizionamento in ambienti ostili (ad es urban canyon)

Test su dispositivo Xiaomi mi 9

Test 2: Considerazioni sul rapporto Segnale Rumore C/N_0

Confronto frequenze ricevitori xiaomi mi 9 e ublox neo m8t (frequenza L1)



Ricevitore ublox ha osservabili meno rumorose rispetto a ricevitore xiaomi

Ricevitore ublox non acquisisce segnali troppo rumorosi, i quali potrebbero generare più problemi che benefici in fase di elaborazione)

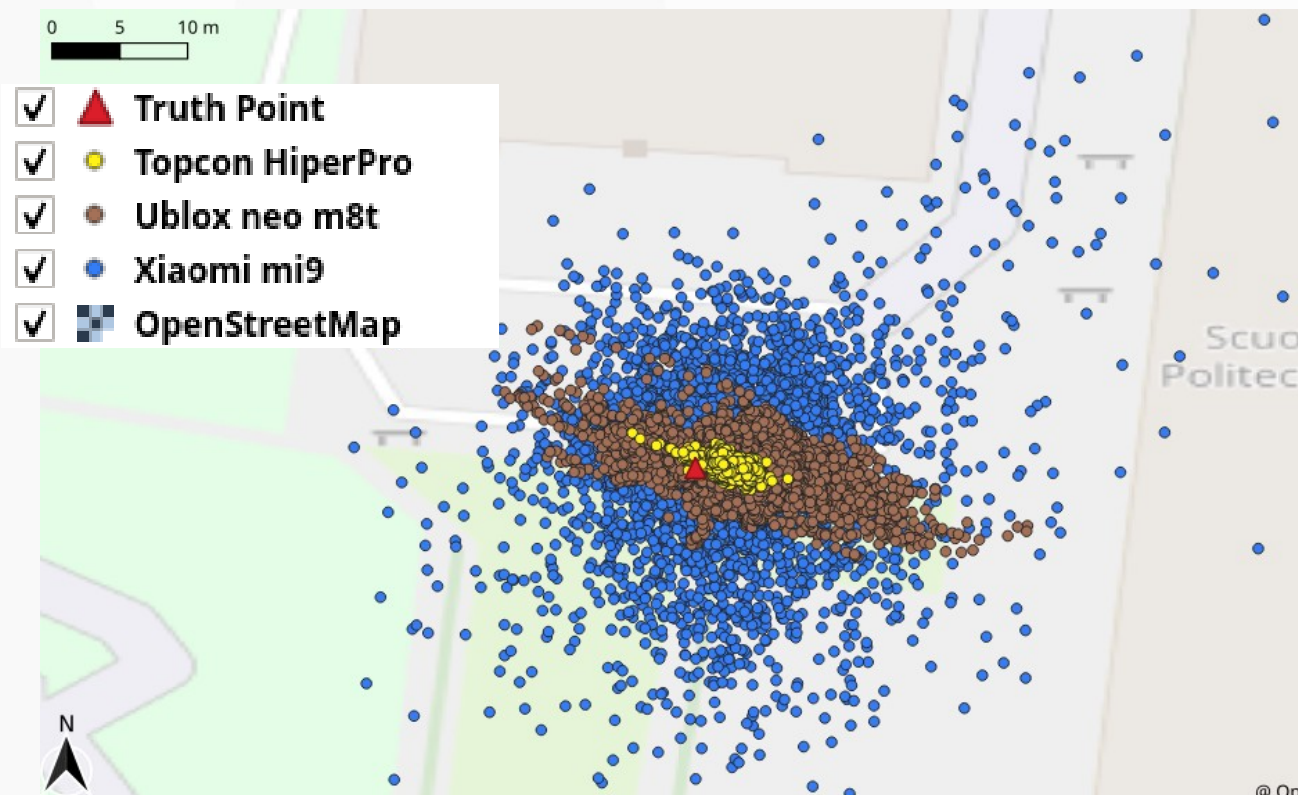
Test su dispositivo Xiaomi mi 9

Test 2: Considerazioni sul posizionamento

Posizionamento effettuato con sw FOSS rtklib

Confronto solo su posizionamento Stand Alone

Xiaomi mi 9 presenta diversi outlier



Topcon Hiper Pro

STD E = 0.6279 m
STD N = 0.4072 m
STD H = 1.1152 m

Ublox neo m8t

STD E = 3.0444 m
STD N = 1.8844 m
STD H = 4.9172 m

Xiaomi mi9

STD E = 5.3445 m
STD N = 7.0903 m
STD H = 14.3606 m

Test su dispositivo Xiaomi mi 9

Conclusioni:

Mancanza delle misure di fase, rende di fatto superflua la seconda frequenza, e **preclude** l'utilizzo del dispositivo per il **posizionamento di precisione**:

- Posizionamento di solo codice ha precisione metrica;
- Non si possono applicare algoritmi per il miglioramento della qualità del posizionamento (ad esempio combinazione ionofree per errore ionosferico o variabile “codice meno fase” per la multipath detection)

Osservabili dello **smartphone** xiaomi mi 9 (ma di tutti smartphone in generale) risultano **molto rumorose** rispetto ai tradizionali ricevitori GNSS

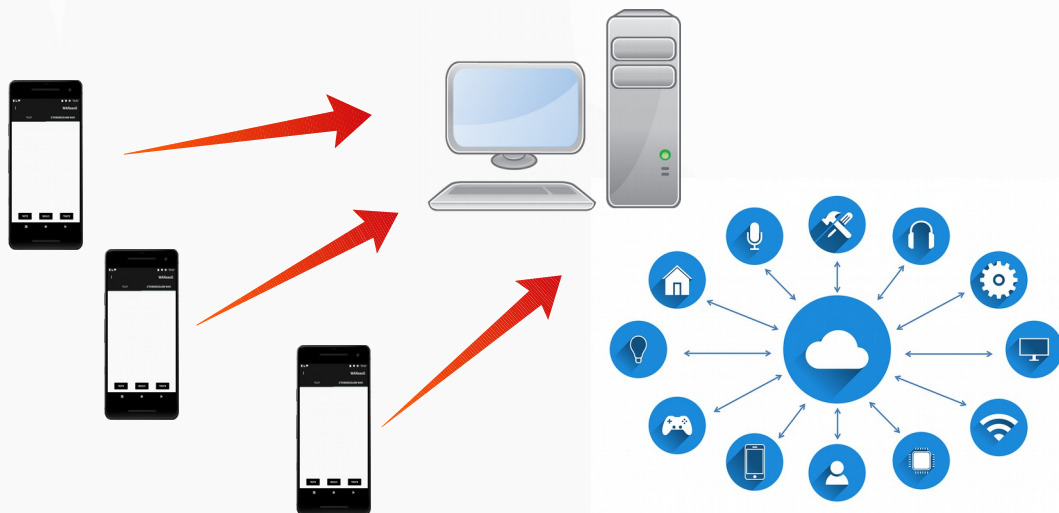


Diventa molto importante un efficace pre elaborazione delle osservabili al fine di evitare la generazione di outlier nella soluzione

GNSS raw: app FOSS per la raccolta dati

Applicazione Free and Open Source per la registrazione di dati provenienti da sensori integrati nello smartphone.

- Principalmente dati grezzi GNSS
- Dati provenienti da altri sensori → Fusion integration per il calcolo della posizione in modalità Stand Alone
- Invio dei dati ad un server: porre le basi per l'implementazione di algoritmi di posizionamento in relativo in maniera "IoT like"



GNSS raw: app FOSS per la raccolta dati

Inizio registrazione (tasto start):

Viene creata una cartella nella memoria interna del dispositivo al cui interno vengono aperti 3 file:

- GNSS_log_RAW che contiene le osservabili GNSS;
- GNSS_log_NAV che contiene i dati di navigazione;
- GNSS_log_FIX che contiene la posizione (“Android Fine”)

Le osservabili GNSS vengono inoltre visualizzate in tempo reale sullo schermo, che rimarrà acceso durante tutta la registrazione.

Clear: cancella la schermata visualizzata, ma non interrompe la scrittura dei file.

Fine registrazione (tasto stop): termina della scrittura, e salvataggio dei file aperti



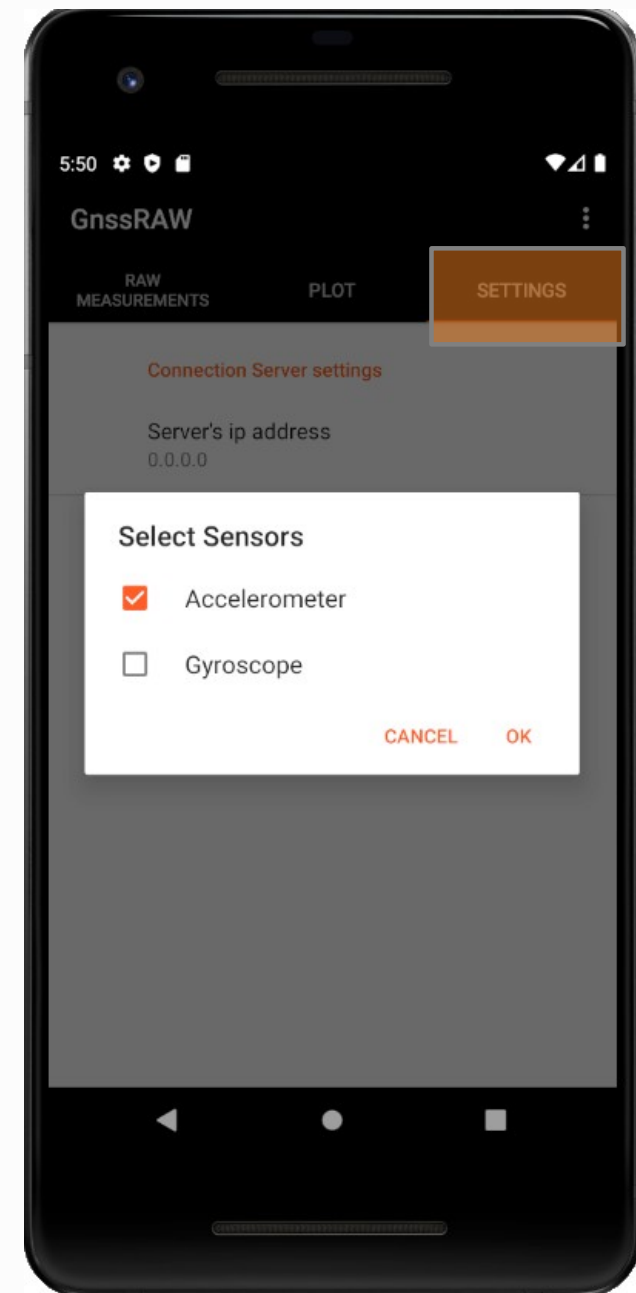
GNSS raw: app FOSS per la raccolta dati

Registrazione dei Dati Sensori

Attraverso il Fragment “Settings” è possibile scegliere alcuni sensori per i quali attivare la registrazione dei dati contestualmente a quella delle osservabili GNSS.

La registrazione dei dati provenienti dai diversi sensori inizia nell'istante in cui viene acquisita la prima osservabile GNSS: in tal modo i diversi file di log risultano sincronizzati tra di loro.

I file di log dei sensori vengono salvati nella cartella “Sensor Measurements” con denominazione log_«nome_del_sensore»_data.txt



GNSS raw: app FOSS per la raccolta dati

Invio dati al server (beta)

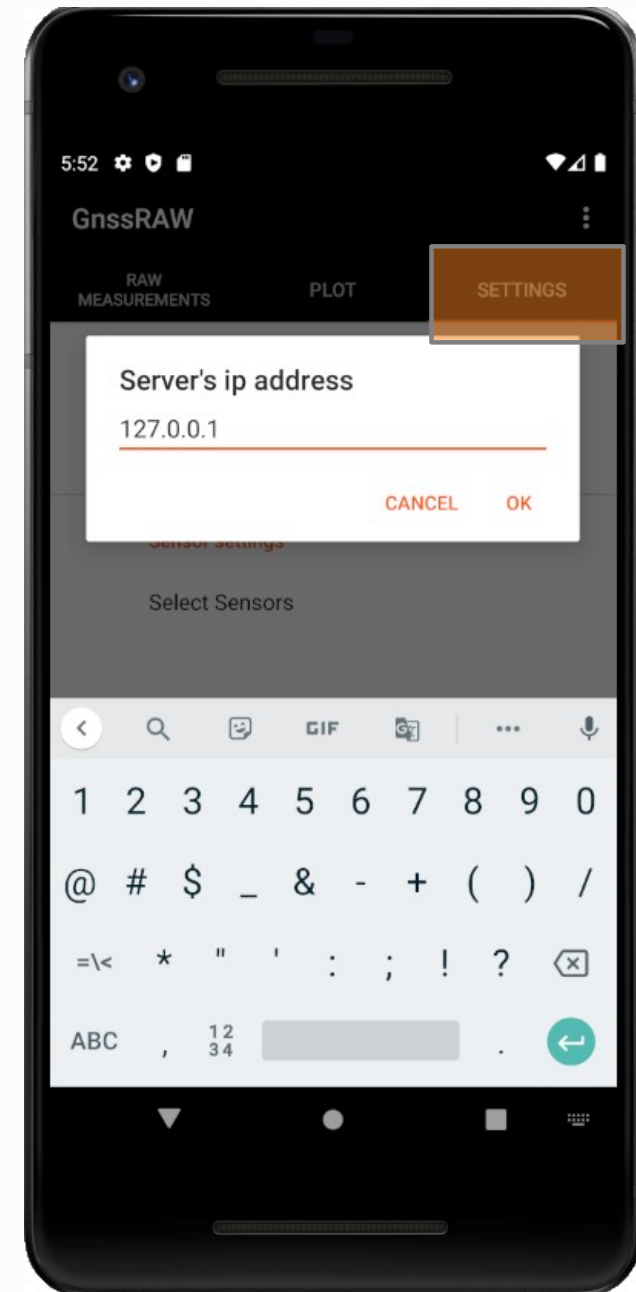
Per avviare la comunicazione con un server bisogna inserire l'indirizzo ip del server nell'apposito menù (Settings → Server's ip address) e poi avviare la registrazione dei dati come nei casi precedenti.

Ogni dispositivo connesso avrà un identificativo col quale il server potrà riconoscerlo.

Vantaggi del server:

- Processare osservabili di più dispositivi per il calcolo della posizione (posizionamento in relativo)
- Si sfrutta la potenza di calcolo del server per il processamento e non quella dello smartphone

Funzione “beta” ancora in fase di test e sviluppo



GNSS raw: app FOSS per la raccolta dati

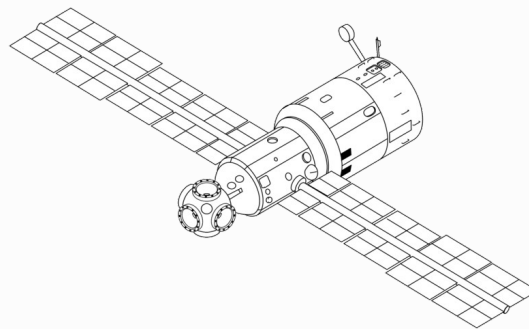
Implementazioni future

- Riorganizzazione della parte server con implementazione di un'architettura IoT like
- Cambio dei formati di output delle osservabili GNSS (ad. es. salvataggio in formato RINEX)
- Implementazione di altri sensori
- Implementazione del modulo plot per visualizzazione e analisi dati
-



[Link](#) al repository con il codice sorgente dell'applicazione





**GRAZIE PER
L'ATTENZIONE**

Contatti:

Lorenzo Benvenuto: lorenzo.benvenuto@gter.it

Manuel Viaggi: manuelviaggi5@gmail.com