

Comunicaciones inalámbricas

Fing - Udelar

Docentes: Federico Larroca
Pablo Belzarena

GPS con SDR

Pablo Rodríguez

20-02-2015

Hardware usado

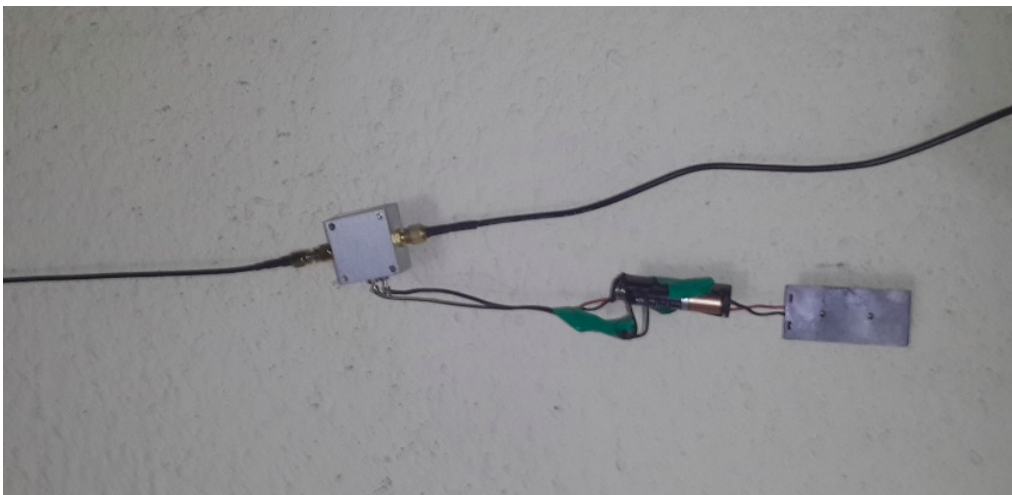
- Usrcp Ettus B100 y RTL-SDR



- Antena activa gps patch 1.57242 ghz



- Bias T, mini circuits



Descripción del trabajo

En este trabajo se busca simular el comportamiento de un gps usando tecnología sdr para la adquisición de la señal. Se utilizaron 2 proyectos opensource y se adaptaron según las necesidades, gnss-sdr y gpsprune.

Conceptos básicos sobre la señal gps

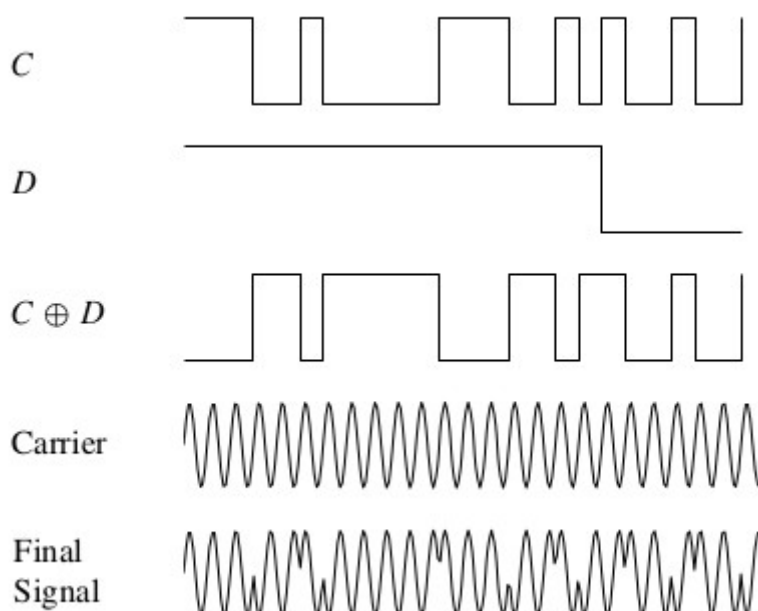
El sistema GPS usa DS-SS, espectro ensanchado por secuencia directa. Cada satélite tiene su código pseudo randomico(PRN)con el cual codifica su datos que luego son modulados por la portadora.

En este trabajo se trabajará solo el espectro comercial, por lo que los PRN son de 1023 chips. Los PRN se repiten cada 1 ms, 1.023Mhz cps. El 90% de la señal se concentra en un ancho de banda de 2 Mhz.

En este caso se tratará con la portadora primaria llamada L1 a 1572,42 Mhz.

La tasa de datos es de 50 bps, con lo que se demora 30 segundos en obtener un paquete básico de 1500 bits. Para obtener el mensaje de navegación completo se necesitan 12,5 minutos.

PRN, Datos, XOR, Portadora, Final



Code Tracking

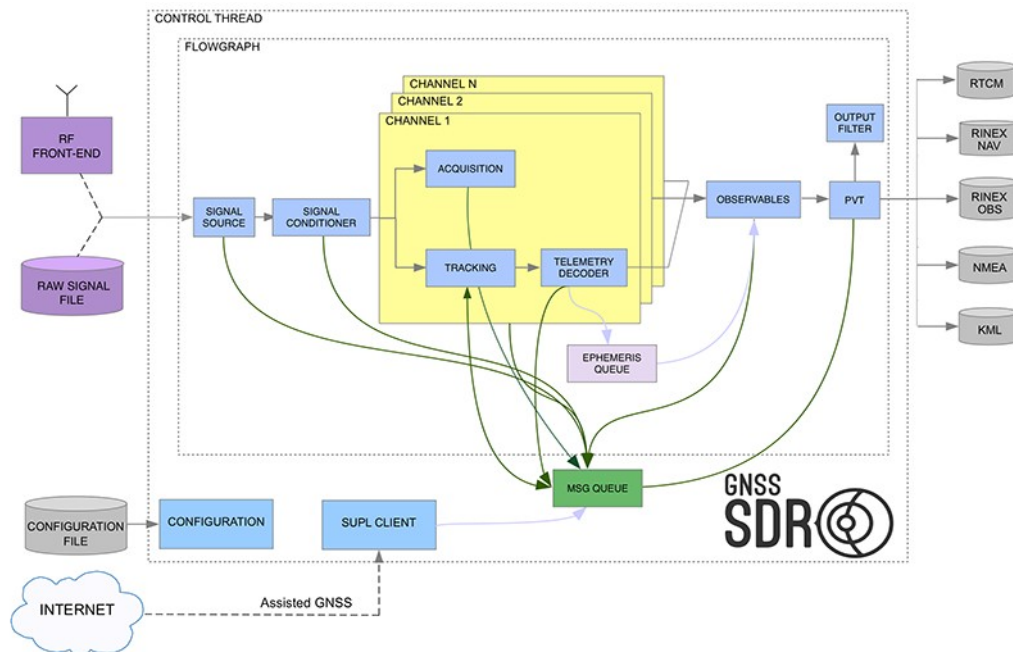
Es necesario estar sincronizado con los códigos PRN de cada satélite para poder decodificar los mensajes de navegación. Se usan dos tipos de algoritmos, DLL coherente y DLL no coherente. Se diferencian en que el coherente requiere previa sincronización en fase por medio de algún PLL.

Otro factor a tener en cuenta es el efecto doppler debido al movimiento del transmisor. Para un receptor estacionario se aproxima en alrededor 5khz, mientras que para uno moviéndose a alta velocidad es alrededor de 10khz.

GNSS-SDR

Este trabajo usó toda la infraestructura de adquisición y procesado desarrollada en el proyecto gnss-sdr, <http://gnss-sdr.org/>, la cual es altamente configurable según se requiera. Se optó esta alternativa debido a que escribir un programa de similares características saldría del alcance de este trabajo. Las instrucciones de compilación están en la web del proyecto, vale aclarar que no resultó trivial para ubuntu 14.04. También es necesario compilar una versión mayor o igual a 3.7.3 de gnuradio. (en la ultima versión del livedvd gnuradio esta incluido).

En la siguiente imagen se muestra un diagrama de bloques de gnss-sdr.



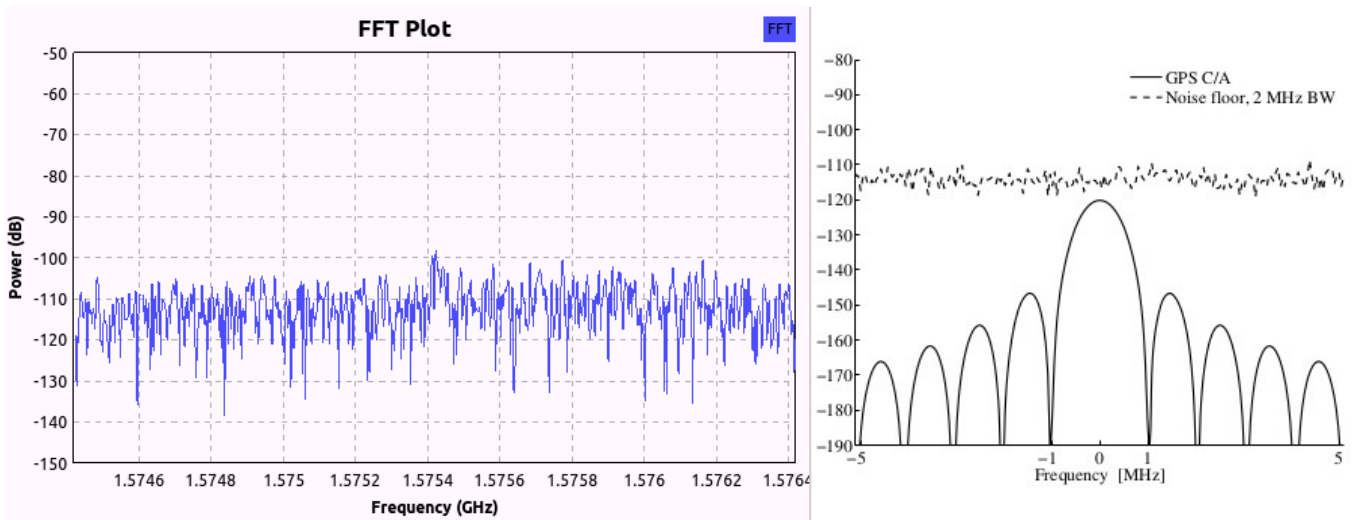
Visualización en mapa.

Para lograr una visualización de la posición mas inteligible, se modificó el programa java opensource GPSPRUNE, que sirve como visualizador de mapas. Se lo adapto para que parsee la entrada estándar buscando la secuencia esperada de coordenadas que imprime gnss-sdr y luego actualice el mapa.

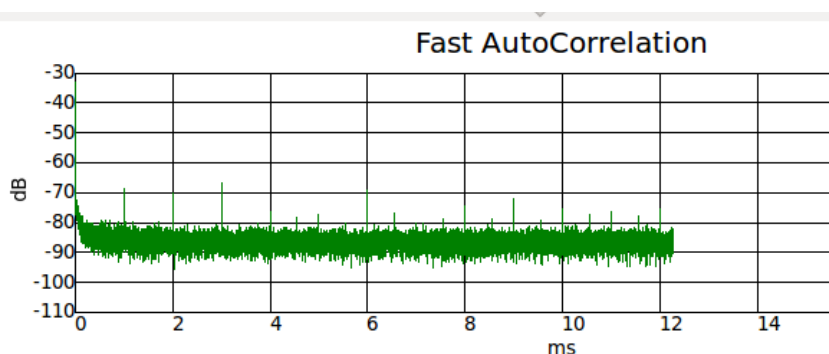
<http://activityworkshop.net/software/gpsprune/>

Pruebas

Primero se intentó una visualización del espectro L1, se obtuvo lo esperado. La señal de los satélites que llega al receptor esta por debajo del nivel de ruido. La antena se la posiciono al aire libre lo mas alto posible.

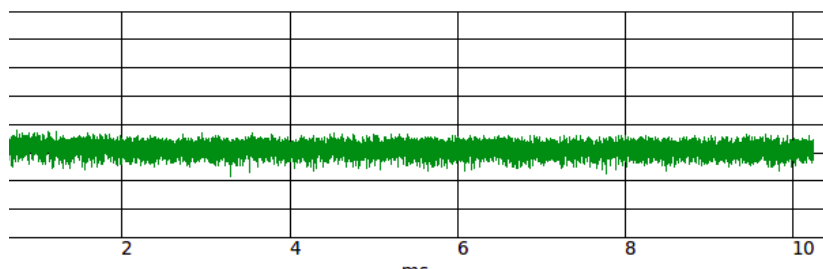


Para comprobar que efectivamente la señal buscada estaba siendo recibida, se le realizó una auto-correlación con el fin de detectar picos. Los bloques gnuradio se obtuvieron de <http://www.mattrobert.com/Wireless/Franks/USRPFastAutocorrelation.html>



Con el receptor secundario rtl-sdr sin antena activa, se logró constatar que la señal gps era demasiado débil con respecto al ruido como para ser detectada. Mientras que con el usrp con antena activa es posible.

Fast AutoCorrelation



Posicionamiento

Las modificaciones en la configuración de gns-sdr que tuvieron buen resultado para el escenario de prueba fueron:

```
#USRP B100
```

```
SignalSource.implementation=UHD_Signal_Source
```

```
#Ancho de banda
```

```
SignalSource.sampling_frequency=2000000
```

```
#GPS L1
```

```
SignalSource.freq=1575420000
```

```
#Ganancia receptor
```

```
SignalSource.gain=25
```

```
#Doppler máximo y paso o resolución entre distintas posibles frecuencias
```

```
Acquisition_GPS.doppler_max=10000
```

```
Acquisition_GPS.doppler_step=500
```

```
# PLL y DLL para el code tracking
```

```
Tracking_GPS.implementation=GPS_L1_CA_DLL_PLL_Optim_Tracking
```

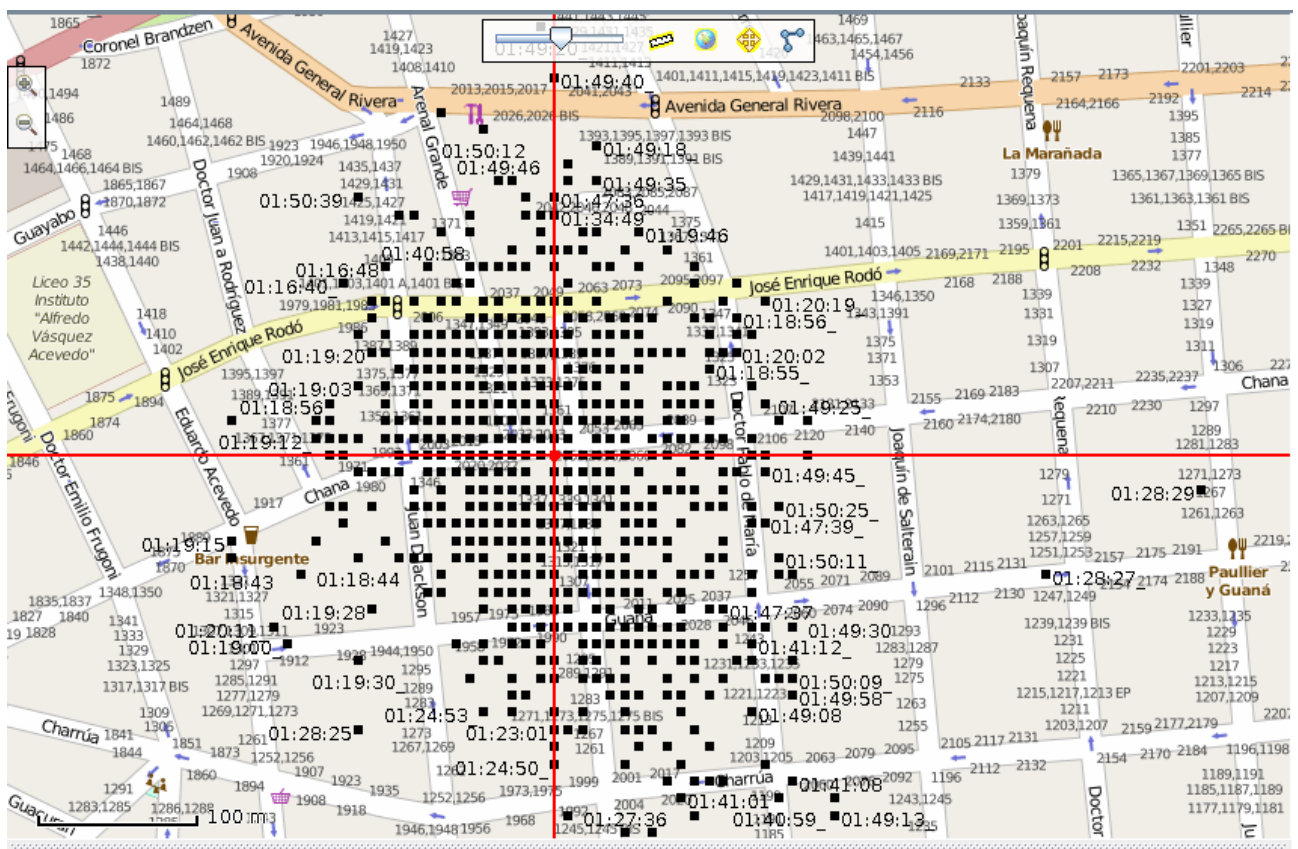
```
#valores para los BW filtros PLL y DLL. Orden
```

```
Tracking_GPS pll_bw_hz=50.0;
```

```
Tracking_GPS.dll_bw_hz=2.0;
```

```
Tracking_GPS.order=3;
```

La siguiente imagen superpone todos los puntos de ubicación que se obtuvieron.



La intersección de las líneas rectas indica el lugar esperado.

Conociendo la posición que se debería obtener y la que se visualiza, se ve que se tiene una precisión de aproximadamente +-150 metros. No es algo excelente, pero esta se podría mejorar, por ejemplo, simplemente promediando los resultados. O agregando un filtro pasabajos a las coordenadas.

Conclusión

La tecnología sdr abre muchas oportunidades que antes se dificultaban debido a la adquisición de hardware adhoc para alguna aplicación RF. Se logro ensamblar un sistema gps con buenos resultados de funcionamiento. No se pudo comprobar el funcionamiento con otros rtl-sdr mas económicos que el USRP Ettus b100, esto fue debido a que la antena y bias T que se adquirió era incompatible en impedancia 75 R vs 50 R. En la web se encontraron experiencias de otros que tuvieron éxito, con lo cual abre la posibilidad de obtener un gps de muy bajo costo.

Comandos del programa

- Comando del programa en que se mantienen persistentes todas las ubicaciones

```
gnss-sdr --config_file=<archivo de configuracion> | java -jar gnss-sdrMap log
```

- Comando del programa en que solo se mantiene la ultima ubicación.

```
gnss-sdr --config_file=<archivo de configuracion> | java -jar gnss-sdrMap
```

el archivo de configuracion usado se llama gps_usrp.conf