

# Introducción a los receptores GNSS de bajo costo

*Por José Ramón Martínez Batlle*

Desde la navegación a estima (*dead reckoning*), basada en una posición previa conocida, la humanidad ha avanzado enormemente en las técnicas de navegación y posicionamiento. El recurso más eficiente disponible en la actualidad para el posicionamiento global preciso son los sistemas de navegación por satélite. Las versiones con cobertura global de estos sistemas, se denominan sistemas de navegación global por satélite o "GNSS" (siglas de *Global Navigation Satellite Systems*), y la modalidad que provee coordenadas precisas al instante se denomina "navegación cinética satelital en tiempo real" o "GNSS-RTK" o, simplemente "RTK". Tanto la tecnología GNSS en general, como su modalidad RTK, son muy útiles en aplicaciones de geografía física (especialmente en geomorfología), agrimensura, topografía, agricultura de precisión, entre otras disciplinas.

Basada en múltiples segmentos, la tecnología GNSS requiere tanto de hardware especializado (e.g. receptores y procesadores de alta capacidad) como de software eficiente capaz de ejecutar algoritmos complejos. Hasta hace apenas 5 a 10 años, sólo un reducido número de fabricantes manufacturaba (y todavía manufactura) costosos equipos con firmware de código cerrado y bajo licencias privativas, dirigidos principalmente al mercado profesional y corporativo. Este panorama cambió significativamente con la publicación y actualización periódica de la biblioteca RTKLIB (primera versión, 2007), por Tomoji Takasu, con la cual se superó la principal limitación de software para generar soluciones RTK. Igualmente, la popularización en la última década de potentes microprocesadores y las iniciativas de hardware libre, permitieron portar aplicaciones complejas a computadoras de placa reducida (e.g. Raspberry Pi) y a microcontroladores (Arduino, ESP). Finalmente, la irrupción en el mercado de receptores GNSS de bajo costo (e.g. u-blox), impulsada especialmente por la industria automotriz, abarató significativamente esta tecnología.

En este curso te ofreceré los recursos necesarios de software y hardware para que aprendas a usar la tecnología GNSS desde cero o a para que mejores tus capacidades al respecto. Específicamente, verás paquetes software de código abierto para procesamiento GNSS y de representación (RTKLIB, QGIS, GoogleEarth), a los cuales podrás acceder sin costo alguno. Igualmente, te mostraré hardware de bajo costo disponible en el mercado, y te guiaré en la puesta en funcionamiento de una unidad GNSS basada en Raspberry Pi, modalidad "hazlo tú mismo/a" (DIY).

## Básicos del curso

### Carga docente

Modalidad	Presencial y virtual
Número de semanas	1
Número de horas	20

### Contenido, distribución de carga horaria

Tema	Horas
<b>Teoría.</b> Sistemas GNSS (1). Software de código abierto, proyectos DIY, aplicaciones en el mundo real	2
<b>Práctica.</b> Familiarización con los recursos informáticos y acercamiento al hardware libre	2
<b>Teoría.</b> Sistemas GNSS (2). Vistazo general, conceptos, sistemas/plataformas satelitales	2
<b>Práctica.</b> Armando un receptor DIY (1). Puesta a punto de Raspberry Pi, instalación de SO, herramientas.	2
<b>Teoría.</b> Fuentes de error, resolución de errores	2
<b>Práctica.</b> Armando un receptor DIY (2). Configuración del receptor, conexión a Raspberry Pi.	2
<b>Teoría.</b> Procesamiento de datos GNSS (1)	2
<b>Práctica.</b> Colecta de datos de campo, RTK	2
<b>Teoría.</b> Procesamiento de datos GNSS (2)	2
<b>Práctica.</b> Posproceso de datos de campo, PPK	2
<b>TOTAL</b>	<b>20</b>

### Metodología

Las sesiones serán tanto teóricas como prácticas. La "teoría" abarcará los aspectos básicos de los GNSS incluyendo, de manera frecuente, demostración de aplicaciones. Las sesiones teóricas se desarrollarán tanto en el aula asignada como por medio de

reunión virtual (modalidad híbrida). Las sesiones prácticas serán presenciales, y se desarrollarán en el aula y en el terreno. Necesitarás una PC para realizar las prácticas de aula. Te recomiendo instalar en tu PC el software mencionado en la sección "**Software requerido**" antes de que inicie el curso. En las prácticas de terreno, usarás el receptor GNSS que fabricaremos en el aula.

## Resultados esperados

Al finalizar este cursos, estarás en capacidad de:

- Comprender principios básicos de funcionamiento, tipologías y aplicaciones de los sistemas de navegación global por satélite.
- Conocer equipamiento GNSS de bajo costo.
- Conocer computadoras de una sola placa óptimas para el procesamiento de datos GNSS.
- Usar, de forma básica, software de código abierto para el procesamiento y representación de datos GNSS.
- Aplicar distintos métodos para el cálculo de posiciones y resolución de errores en receptores GNSS.
- Construir un receptor GNSS.

Con el último resultado sólo preveo mostrar la integración de los componentes y realizar pruebas sencillas, dejando un receptor completo y funcional al finalizar el curso. Cada estudiante puede llevar componentes e intentar armar uno (o comenzar a armar uno) en tiempo real mientras se realiza el curso.

Los componentes requeridos son muy variables, y los precios también fluctúan mucho en función de las marcas y las prestaciones de los equipos. Sin embargo, esta es una lista mínima (al terminar el curso tendrás más ideas sobre qué comprar y cómo configurar tu propio diseño):

- **(US\$20)**. Computadora monoplaca o SBC (*single board computer*). Recomendada Raspberry Pi Zero W (no es necesaria la Raspberry Pi Zero 2 W, que es más rápida, pero para los fines del receptor la Zero W es suficiente), que ronda los 20

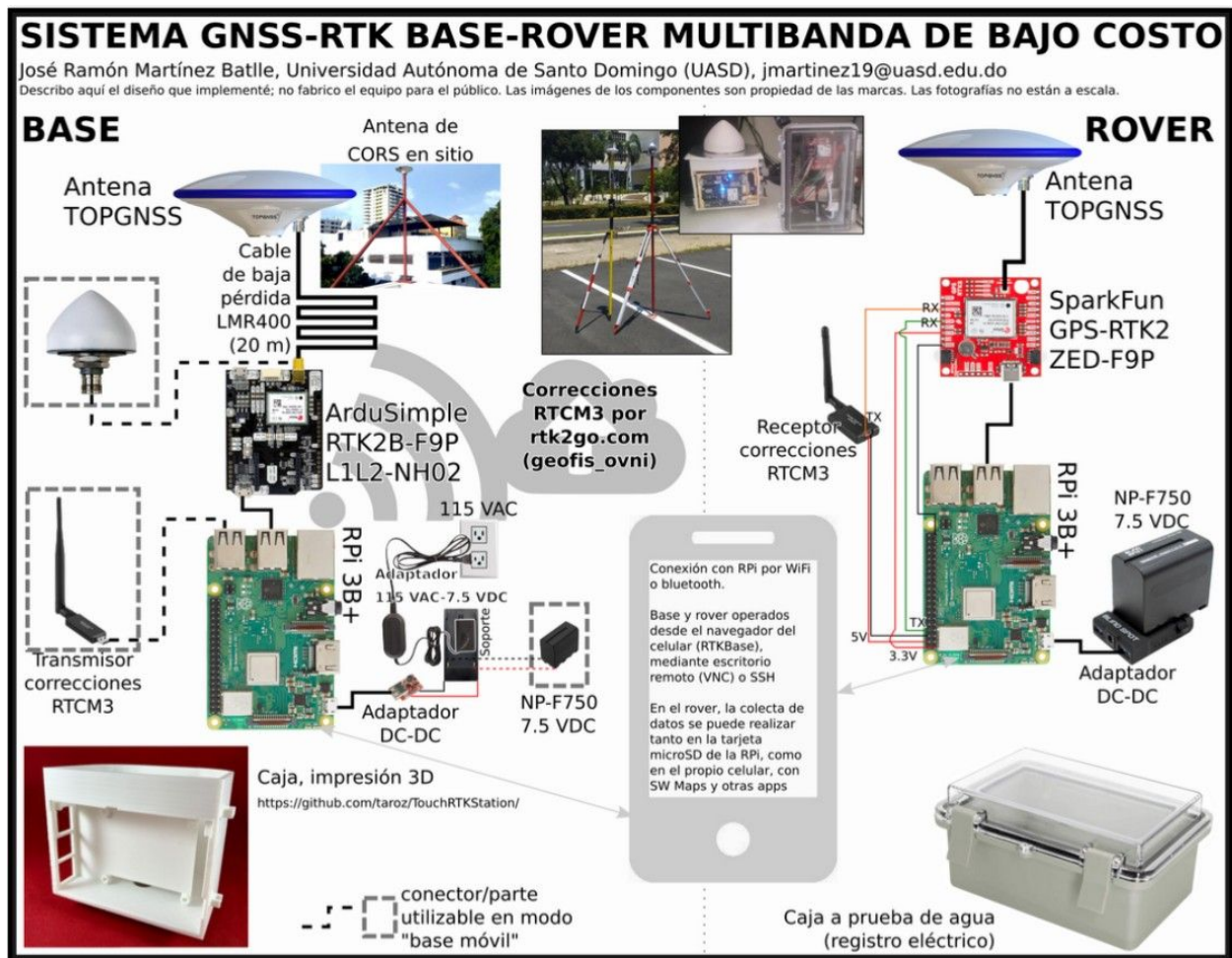
dólares. Hay alternativas, como Banana Pi, Mango Pi, etc., pero la Raspberry ofrece mayor soporte.

- **(US\$100-200)**. Módulo GNSS de doble frecuencia. "Esta es la pieza". Las dos frecuencias ideales son L1/L2, pues aunque L1/L5 es una combinación posible, la precisión obtenida con dicha configuración de frecuencias dependerá mucho del módulo, por lo que mejor no arriesgarse y quedarse con L1/L2. Antes sólo había en el mercado tres marcas de módulos GNSS de doble frecuencia de bajo costo, pero China está cambiando este panorama rápidamente. He probado varios módulos de u-blox (Suiza), y no tengo quejas. En concreto, el ZED-F9P es bastante bueno. Entre las marcas chinas cito también a SinoGNSS, Unicore Communications, SkyTraq (Tw), AllyStar, Quectel, entre otras. Los precios te podrían sorprender, porque algunos módulos de estas marcas (de doble frecuencia y de triple frecuencia) pueden costar menos de US\$100.
- **(US\$50)**. Antena de doble frecuencia (L1/L2 o L1/L5, dependerá del receptor), cable con sus correspondientes terminales. Algunas antenas vienen con su cable ya instalado, otras necesitan uno, normalmente con terminal TNC. El módulo normalmente usa terminal SMA, por lo que es importante fijarse antes de comprar el cable requerido. Una de las marcas más económicas, es TOPGNSS, pero hay otras aún más económicas, por precios incluso cercanos a US\$30.
- **(US\$10)**. Cables USB. Dependerá mucho del módulo GNSS que adquieras y del diseño que quieras implementar. Normalmente, los módulos GNSS montados en placas de desarrollo tienen entradas microUSB o tipo C. La Raspberry Zero viene con microUSB *on the go*, por lo que necesitarás un cable de este tipo ("OTG") para alimentar el módulo GNSS a través de dicho puerto. Es común usar un cable microUSB a USB-A, pero también se puede fabricar un microUSB a microUSB directamente. Para armar receptores sin necesidad de usar el puerto USB, suelen usarse "*jumpers*" (disponibles en el país) y conectar el módulo con la Raspberry a través del GPIO (este diseño suele combinarse con una "*proto*board" o "*bread*board", pero yo prefiero ahora las *perf*board).
- **(US\$0-20)**. Alimentación de energía. Esto varía mucho, pero puede ir desde 0, usando un cargador de celular, hasta *power banks* o sistemas de diseño propio

usando la batería 18650 u otras. Cada sistema tiene sus ventajas y desventajas. El cargador de celular, si tiene la potencia necesaria, funcionará, pero sólo para pruebas en interiores. Un *power bank* es buena solución, pero tienen un costo no menor de US\$10 en RD. Los sistemas propios son la solución ideal, porque decides qué potencia instalar, configuras todo para que ocupe el menor espacio y pese lo menos posible, pero requieren mucha labor manual.

- **(US\$5)**. "*Breadboard*", placa de pruebas, placa de prototipos o, en el lenguaje de las tiendas de la 30 de Marzo "*protoboard*". Ya no las suelo usar, pero son cómodas para comenzar a "puyar" electrónica.
- **(US\$10)**. Caja, solución de intemperización. En el mundo DIY, lo más usado son las cajas de registro eléctrico de tapa transparente, pues ofrecen la protección contra humedad necesaria. Sin embargo, no son ergonómicas, tampoco son a prueba de caídas. Otras opciones son intemperizar con fundas a prueba de agua, o cajas diseñadas al efecto con impresoras 3D.
- **(US\$5)**. Opcional: puedes ensayar con Arduino y ESP, que son placas con microcontroladores muy eficientes. Sin embargo, la programación de un receptor con estas placas es más desafiante, pero no dejes de intentarlo.
- Aquí puedes consultar un ejemplo de sistema base-rover completo:  
<https://geofis.github.io/rtk-para-todos/#/section-3>





## Perfil de estudiante

Dirigido a estudiantes y profesionales de cualquier área con interés en el uso y desarrollo de tecnología GNSS desde el segmento usuario. En la Escuela de Ciencias Geográficas queremos fomentar la participación de estudiantes y profesionales de la carrera de geografía, así como de agrimensura. También nos interesa recibir participantes de carreras afines, como ingeniería civil, arquitectura, agronomía y geología.

Es deseable que dispongas de computadora portátil o una computadora de una sola placa, por ejemplo, Raspberry Pi. También son bien valorados los conocimientos básicos de software libre y de código abierto (e.g. núcleo Linux), de lenguajes de programación (Python, C, entre otros), y de aplicaciones de representación espacial (e.g. QGIS, GoogleEarth).

# Software requerido

Preinstala, en tu propia PC, las siguientes aplicaciones:

- RTKLIB: La aplicación principal que usarás será RTKLIB. Esta corre tanto en Windows como en Linux (imagino que MacOS también). En Windows, RTKLIB se usa corriendo ejecutables precompilados. En Linux, dichos ejecutables se pueden correr mediante Wine. Aunque la versión original de RTKLIB es la de Tomoji Takasu, nosotros usaremos el fork RTKBLIB demo5 de Tim Everett.

- Web oficial de descarga de RTKLIB demo5:

[https://github.com/rtklibexplorer/RTKLIB/releases/download/b34d/demo5\\_b34d.zip](https://github.com/rtklibexplorer/RTKLIB/releases/download/b34d/demo5_b34d.zip)



- Descomprime el contenido del ZIP en una carpeta; cuando quieras ejecutar un programa, harás clic sobre su ejecutable ".exe"
- QGIS. Te servirá para ver resultados sobre un mapa/imagen.
- Web oficial de descarga de QGIS: <https://qgis.org/es/site/forusers/download.html>



- Tutorial de instalación:

[https://www.youtube.com/watch?v=FFf3\\_P6-4Ag](https://www.youtube.com/watch?v=FFf3_P6-4Ag)



- GoogleEarth. Visualizarás archivos KML con este visor.

- Web oficial de descarga:

<https://support.google.com/earth/answer/21955?hl=es>



- Git. Se usa para control de versiones y descargar repos fácilmente. No es imprescindible, pero sí útil:

- Web oficial de descarga de Git:

<https://git-scm.com/>



- Tutorial de instalación:

<https://www.youtube.com/watch?v=dbEqN7EY-Mg>

