

# ARQUITECTURA DE RECEPTOR GPS DEFINIDO POR SOFTWARE

LAROSA, Facundo - MIGNONE, Martín - GHIGNONE, Ramiro - CASTELLUCCI VIDAL, Iván  
GIAMPETRUZZI, Julián – GUANCA, Aníbal  
flarosa@frh.utn.edu.ar - mnmignone@gmail.com – rghignone@frh.utn.edu.ar - icvidal@frh.utn.edu.ar –  
jggiampetruzzi@yahoo.com.ar – anibalguanca@gmail.com  
Universidad Tecnológica Nacional – Facultad Regional Haedo  
Departamento de Ingeniería Electrónica  
París 532 – TE 011-4443-7466 (Int 145)  
Haedo, Buenos Aires, Argentina

## RESUMEN

El objetivo del proyecto es desarrollar una arquitectura de receptor GPS definido por software con características de interés para la modelización de la trayectoria de un vector / vehículo aéreo, tales como: alta tasa de actualización de datos, incorporación de algoritmos de corrección de datos, reducción del tiempo de sincronización de los lazos de seguimiento del receptor, implementación de algoritmos de navegación con acoplamiento débil o fuerte y la integración de sensores inerciales para complementar la navegación.

## 1.- INTRODUCCIÓN

Se diseñó una arquitectura de receptor GPS definido por software conformada por los dos módulos principales: un *front end* y un *back end*. El *front end* se encarga de recibir la señal de radiofrecuencia (RF) proveniente de la antena del receptor, amplificarla, filtrarla, convertirla a frecuencia intermedia y digitalizarla. El *back end* se encarga de recibir los datos del *front end* y realizar las operaciones en tiempo real más demandantes (implementación de módulos de búsqueda de seguimiento) como también implementar las funciones de control general del receptor y los cálculos necesarios para obtener la posición del receptor. Dichos cálculos se implementaron construyendo librerías propias en lenguaje C optimizadas para sistemas embebidos de forma tal que puedan ser integradas en un microcontrolador.

## 2.- METODOLOGIA

La decisión de implementar una arquitectura de receptor GPS definido por software o SDR (por sus siglas en inglés, *Software Defined Radio*) fue debido a las numerosas ventajas que este tipo de sistemas provee; principalmente la

posibilidad de poder reconfigurar por software gran parte de las capas funcionales que integran al receptor, abarcando desde parámetros a nivel de señal de RF, hasta la forma en que los datos pueden ser interpretados. Además, provee la posibilidad de realizar actualizaciones del sistema, agregando nuevas funcionalidades sin tener que modificar el hardware, sino, solo cambiando software [1], facilitando de esta manera las iteraciones sobre el diseño y acelerando el proceso de desarrollo.

Teniendo en cuenta dicha estructura, el proyecto se llevó cabo de forma modular, es decir, que cada parte o subsistema se desarrolló independientemente de los demás permitiendo la posibilidad de realizar pruebas para validar cada módulo por separado, para finalmente realizar la integración final que constituya el sistema completo.

La primer etapa contempló el diseño del *front end*, para lo cual fue necesario el estudio de las etapas de RF y demodulación utilizada por el sistema GPS, el análisis y búsqueda de los componentes disponibles en el mercado, para concluir con el diseño de placas de circuito impreso (del inglés, *printed circuit board*, PCB), la adquisición de componentes, construcción y montaje del prototipo del subsistema.

La segunda y tercer etapas comprenden el diseño del *back end*. En la segunda etapa se realiza la interfaz directa con el *front end*, incluyendo el procesamiento digital requerido para decodificar los datos a partir de las señales GPS, específicamente, los módulos de búsqueda y seguimiento implementados mediante lenguajes descriptores de hardware (en nuestro caso, VHDL) a fin de integrarlos en una FPGA (del inglés, *Field Programmable Gate Array*). La última etapa, concierne al estudio, análisis y diseño del procesamiento digital necesario de los datos de navegación, para realizar todas las operaciones correspondientes para hallar la posición final del receptor en el sistema de referencia utilizado por el GPS, así como también el estudio y diseño de librerías de dichos algoritmos en lenguaje de alto nivel (tipo C) optimizadas para su implementación sobre microcontroladores.

### 3.- DESARROLLO

A continuación, se muestra el diagrama en bloques de la arquitectura diseñada.

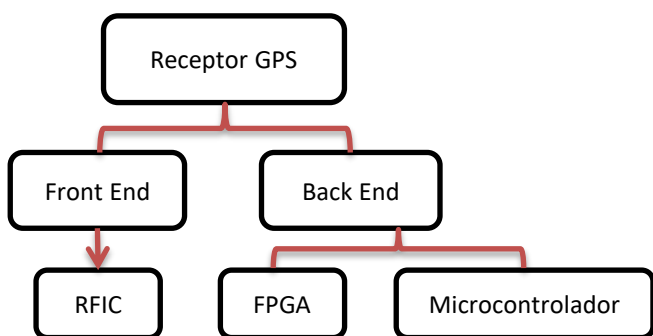


Figura 1. Diagrama en bloques del sistema completo.

Como podemos apreciar en la Figura 1, este diagrama corresponde al esquema general de un SDR [1]. En la arquitectura propuesta, se desarrollaron dos *front end*. El primero se compone principalmente de un RFIC (circuito integrado de radio frecuencia, por sus siglas en inglés *RF Integrated Circuit*) modelo MAX2769 [3], y el otro, modelo SKYWORKS 4150 [4]. Ambos cumplen la función de recibir la señal proveniente de la antena GPS centrada en 1575,42MHz e implementar la etapa de

amplificación (mediante amplificadores de bajo ruido, o LNA por sus siglas en inglés, *Low Noise Amplifier*), filtrado y mezcla en cuadratura (canales I y Q), llevando la portadora de la señal a una frecuencia intermedia FI (método heterodino) que se pueda procesar digitalmente [5]. Luego, dicha señal se muestrea mediante dos ADC (uno para cada canal) a una frecuencia de 16.384 MHz, alrededor de cuatro veces la FI (cumpliendo con margen el criterio de Nyquist) [6], obteniendo a la salida la señal digitalizada.

La diferencia entre ambos modelos, es que el primero tiene la capacidad de poder reprogramar ciertos parámetros de operación (ancho de banda, frecuencia intermedia, nivel de amplificación, resolución de los ADC) dentro de ciertos rangos predefinidos de valores, mediante una interfaz serie. En cambio, el segundo, posee parámetros de configuración predefinidos.

Se diseñaron las placas electrónicas de ambos modelos en base a las recomendaciones del fabricante y los kits de evaluación de los mismos. En ambos diseños se implementó la pista de RF proveniente del conector de la antena como una línea de transmisión de 50  $\Omega$  de topología guía de onda coplanar con plano de masa o GCPW (por sus siglas en inglés, *Grounded Coplanar Waveguide*) [7], debido a que se adecuaban mejor a las especificaciones de construcción de los fabricantes de las placas y también para asegurar que la señal de la antena llegase con la suficiente potencia al integrado. También se utilizaron las técnicas de “*vias fences*” de forma de disminuir al mínimo la propagación de la energía de forma lateral a la pista principal [8].

Ambos modelos fueron diseñados con el fin de obtener la señal de salida centrada en 4MHz, con un ancho de banda de 2MHz con los canales I y Q de forma digital. En la figuras 2 y 3 se muestran imágenes de la vista superior de las placas de circuito impreso.

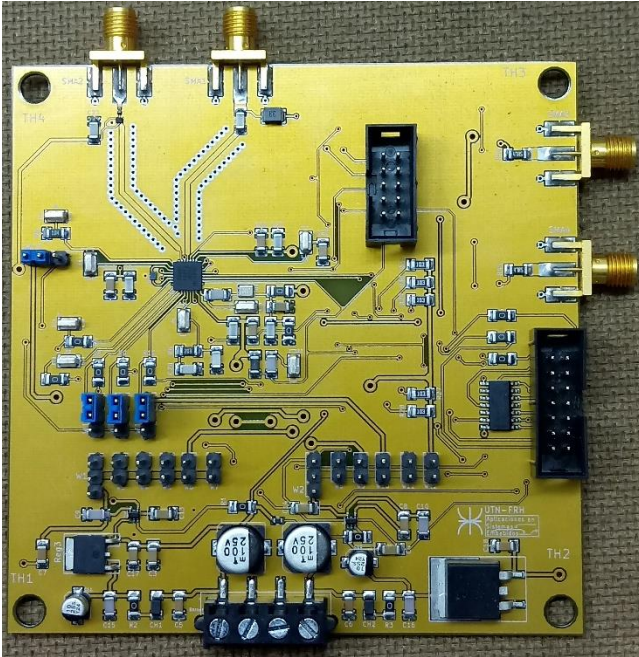


Figura 2. Cara superior de placa PCB del *front end* MAX2769.

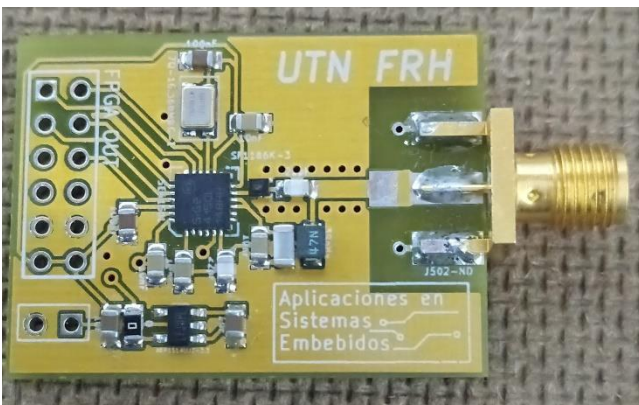


Figura 3. Cara superior de placa PCB del *front end* SKYWORKS 4150.

Hasta aquí se considera la etapa de RF, comprendiendo el *front end* del sistema. Éste se conecta con el siguiente módulo implementado en la FPGA, que comprende la primer etapa del *back end*.

En la FPGA se implementan los módulos de búsqueda y seguimiento. El módulo de búsqueda tiene por función correlacionar la señal digitalizada en frecuencia intermedia (FI) proveniente del *front end* con una réplica local la cual puede configurarse para cualquier satélite y para diferentes frecuencias. De esta manera, se puede detectar si un determinado satélite está en vista y cuál es el corrimiento respecto de la

frecuencia central por efecto Doppler al detectarse un máximo de correlación por encima de determinado umbral.

El módulo de seguimiento (en inglés, *tracking*) tiene por función generar las señales que permitan mantener el sincronismo de la señal proveniente de un satélite determinado, a pesar del corrimiento que se produce en el tiempo de la señal de *spreading* del satélite y de su frecuencia central por efecto Doppler.

Ambos módulos fueron descriptos utilizando el lenguaje de descripción de hardware VHDL. Los módulos se verificaron y validaron a través de la simulación funcional, utilizando como referencia señales GPS digitalizadas en FI generadas sintéticamente. En [9] se puede verificar exhaustivamente el desarrollo realizado.

Para la última etapa del *back end*, La plataforma de cómputo utilizada fue la EduCIAA [10], la cual posee un procesador NXP LPC 4337 basado en ARM Cortex-M4F [11] que se utiliza para procesar los datos de navegación (provistos por la FPGA) provenientes de todos los satélites que estén a la vista y realizar las operaciones necesarias para, primero obtener la posición de los satélites, y en base a eso, finalmente, obtener la posición del receptor.

En una primera instancia fue necesario el estudio de los cálculos necesarios para obtener la posición de un receptor a partir de los datos de navegación y observables (efemérides y pseudorangs) provistos por la constelación de satélites, según la especificación GPS [12].

La determinación de la posición del receptor a partir de los datos de navegación consta de tres partes: cálculo de posición de los satélites, cálculo de la posición del usuario y aplicación de correcciones de pseudorango y posición [13]. Para llevar a cabo estos cálculos, se desarrolló una biblioteca de cabeceras y funciones en C que implementan dichos cálculos, de forma tal que sean capaces de ser fácilmente aplicados en una plataforma móvil (como un microcontrolador) [14].

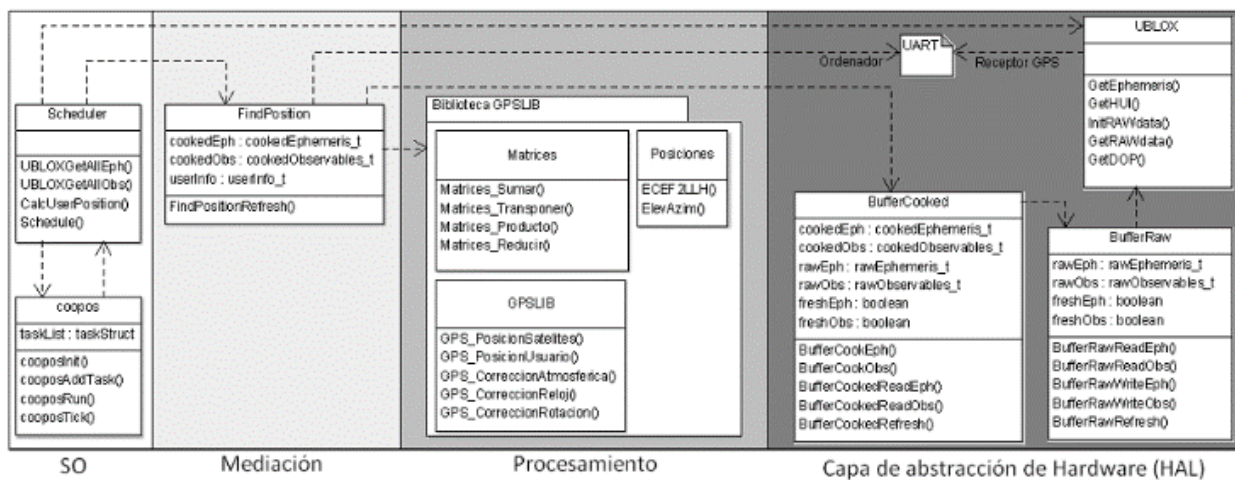


Figura 5. Capas del software desarrollado

La validación de los algoritmos y bibliotecas implementadas se realizó experimentalmente a partir de los datos crudos (posiciones y pseudorángos) de un receptor GPS comercial modelo uBLOX 6 NEO-6M [15]. Dicho receptor se posicionó en un punto geodésico patrón (verificado y validado por el entonces Instituto Geográfico Militar; actualmente, Instituto Geográfico Nacional) y los datos fueron adquiridos a través de una PC, para luego ser procesados por las bibliotecas desarrolladas a fin de hallar la posición de los satélites y la del receptor. Finalmente, se comparó la posición obtenida a través de los datos crudos con la posición de referencia del punto patrón.

La posición patrón está definida como: 34° 33' 19,986'' latitud Sur, 58° 30' 25.769'' longitud Oeste y altura 35.65m (equivalente en coordenadas ECEF (del inglés, *Earth Centered Earth Fixed*) a (2747040,7; -4484020,311; -3597389,777) metros.

Tabla 1. Cálculo de posición del receptor a partir de efemérides y observables.

Nº	X [m]	Y [m]	Z [m]	Error [m]
1	2,747051E+06	-4,484048E+06	-3,597403E+06	32,36
2	2,747042E+06	-4,484037E+06	-3,597397E+06	18,23
3	2,747040E+06	-4,484033E+06	-3,597391E+06	12,77
4	2,747048E+06	-4,484045E+06	-3,597407E+06	30,98
5	2,747047E+06	-4,484045E+06	-3,597410E+06	24,84

En la tabla 1, se observa como resultado un error cuadrático medio promedio de 25,28 metros de la posición calculada respecto de la verdadera, cumpliendo con las especificaciones del sistema GPS que define como cota superior de error medio de una posición en 30m utilizando el sistema de posicionamiento estándar (del inglés, *Standard Positioning System*, SPS) [12].

Una vez validada la librería, se llevó a cabo, la implementación de estos algoritmos en la EduCIAA con vistas a migrar la biblioteca a un sistema embebido para su utilización futura. Asimismo, a fin de mejorar la eficiencia de cálculo y reducir el uso computacional se optimizó la implementación de los algoritmos describiendo las funciones más demandantes en el lenguaje nativo del microprocesador, en este caso, el conjunto de instrucciones Thumb 2 de ARM [11].

En base a la estructura elegida para este receptor, se estructuró el software con vistas a abstraer la operación de la biblioteca del hardware utilizado [16].

Como se puede ver en la Figura 5, el software embebido consta de cuatro capas: el sistema operativo (SO), la capa de mediación, la capa de procesamiento y la capa de abstracción de hardware (HAL, por sus siglas en inglés, *Hardware Abstraction Layer*).

El sistema operativo se encarga de ejecutar las tareas que comandan la toma de datos del receptor GPS y su procesamiento. La capa de

mediación responde a un patrón de software comúnmente llamado “mediador” [17] que permite abstraer al sistema operativo de las interfaces y operaciones involucradas en la toma de datos y el procesamiento, y simplemente presenta la solución de posición de usuario. La biblioteca desarrollada, que hemos denominado “GPSLIB”, recibe los datos de navegación y observables en un formato independiente del hardware utilizado y calcula la posición del usuario. Finalmente, la HAL provee datos del receptor utilizado (en este caso, el UBLOX) a través de un buffer abstrayendo completamente a las funciones de mayor nivel de las particularidades del receptor empleado. Cada módulo está escrito en lenguaje C y comprende un archivo de cabecera .h y un archivo de código .c.

A fin de validar la librería implementada, se midieron el error, la dispersión y los tiempos de ejecución para obtener una estadística con la cual comparar la optimización luego.

En la tabla 2 se muestran los valores calculados de posición y el error respecto del punto patrón conocido. En la figura 6 se aprecia un gráfico que indica la proporción de muestras contenidas dentro de un entorno alrededor de la posición verdadera

Tabla 2. Análisis estadístico de las muestras tomadas.

Coordenadas ECEF	Valor esperado [m]	Valor medio obtenido [m]	Desvío Estándar [m]
X	2696817.56	2696824.61	7.1484
Y	-4512032.80	-4512038.53	9.7384
Z	-3600366.33	-3.600366.04	5.7659

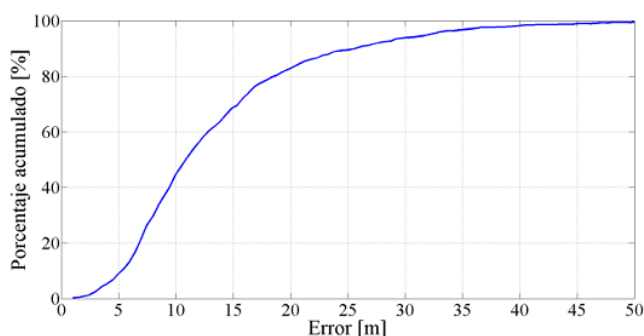


Figura 6. Proporción de las muestras contenidas en función de la distancia a la posición verdadera.

Se puede apreciar que cerca del 95% de las muestras presentan un error menor o igual a los 30 m que especifica el estándar GPS como error máximo.

En una etapa posterior, se procedió a la optimización del código, la cual contempló la implementación de las funciones más demandantes computacionalmente en lenguaje ensamblador, enfocándose en aprovechar los registros de propósito general disponibles para disminuir al mínimo los accesos a memoria. En la siguiente tabla, podemos apreciar los tiempos de ejecución obtenidos con y sin optimización, para cada una de las funciones implementadas.

Tabla 3. Tiempos de ejecución con y sin optimización.

Sección	Sin optimizar [μseg]	Optimizado [μseg]	Mejora
GPS_PosicionSatelite	389,81	21,76	94,42%
GPS_PosicionUsuario (construccion de matriz alfa)	48,01	N/A	N/A
GPS_PosicionUsuario (preinversion)	63,71	21,15	66,8%
GPS_PosicionUsuario (inversion)	118,85	34,39	71,06%
GPS_PosicionUsuario (postinversion)	75,71	25,24	66,66%
GPS_CorreccionAtmosferica	492,47	N/A	N/A

Observamos en las zonas optimizadas correspondientes a operaciones matriciales una reducción del tiempo consumido aproximadamente a un tercio del valor original.

## CONCLUSIONES

El proyecto global se encuentra actualmente en fase de prueba de cada módulo por separado y su integración con el resto de los módulos del sistema.

Ambos *front ends* están siendo testeados exhaustiva e individualmente.

Con respecto al *back end*, los módulos de búsqueda y seguimiento implementados en la FPGA fueron probados utilizando una señal GPS simulada (en frecuencia intermedia), obteniendo resultados satisfactorios. Actualmente, se

encuentra en fase de prueba con señales reales para completar su validación y también la implementación de las interfaces de control para poder, por un lado, recibir los datos provenientes del *front end*, y por el otro, para enviarle los datos resultantes al microcontrolador, completando el receptor.

En cuanto al microcontrolador y los algoritmos implementados, se logró diseñar y construir una biblioteca de funciones capaz de procesar las efemérides y observables de los satélites de la constelación GPS para obtener la posición de usuario de acuerdo a las especificaciones del sistema.

Asimismo, se logró optimizar el tiempo de procesamiento empleado en secciones críticas del mismo aproximadamente a un tercio de su valor original, siendo este resultado promisorio para aplicaciones con altas tasas de muestreo y/o bajo consumo.

Los próximos pasos contemplan la conexión de todos los módulos según el diagrama funcional y las pruebas necesarias para realizar todo el camino de recepción, de forma de validar el sistema completo en todo su conjunto.

Una vez validado el sistema entero, el objetivo final es diseñar el receptor integrando todos los módulos en una sola placa de circuito, de forma tal, de obtener un receptor GPS definido por software integrado, configurable y flexible de acuerdo a la aplicación donde se requiera su utilización.

## REFERENCIAS

[1] JEFFREY H. REED (2002). **Software Radio: A Modern Approach to Radio Engineering**. Prentice Hall PTR Vol. 1, 3-6.

[2] JEFFREY H. REED (2002). **Software Radio: A Modern Approach to Radio Engineering**. Prentice Hall PTR Chapter 1, 4.

[3] Maxim Integrated™(2010). **MAX2769 Universal GPS Receiver**.19-0791; Rev 2; 6/10.

[4] Skyworks Solutions, Inc.™(2012). **SE4150L: GPS Receiver IC.202445A**.

[5] JEFFREY H. REED (2002). **Software Radio: A Modern Approach to Radio Engineering**. Prentice Hall PTR Chapter 2, 14-20.

[6] Leon W. Couch, II (2002) – Séptima Edición. **Sistemas de Comunicación digitales y analógicos**. Prentice Hall Chapter 3, Section 6, 180-188.

[7] David M. Pozar (2012) – Fourth Edition. **Microwave Engineering**. John Wiley & Sons, Inc. Chapter 3, 95-96, 159-160.

[8] Asanee Suntives, Arash Khajooeizadeh, Ramesh Abhari (2009) **Using Via Fences for Crosstalk Reduction in PCB Circuits**. Department of Electrical and Computer Engineering, McGill University, Canada.

[9] Facundo Larosa (2016), **Módulo de búsqueda, seguimiento y decorrelación para un sistema GPS sobre FPGA**, Memoria de Proyecto Final de Especialización en Sistemas Embebidos, Universidad de Buenos Aires

[10] <http://www.proyecto-ciaa.com.ar/>

[11] **Cortex M4F**, Technical Reference Manual, Revision R0P0, ARM Limited.

[12] J. Bao Yen Tsui (2000). **Fundamentals of Global Positioning System Receivers: A software approach**, Second Edition, Wiley & Sons, Chapter 2.

[13] United States Air Force (2008), **The GPS Standard Positioning System**.

[14] Facundo Larosa, Mignone Martín, Castellucci Vidal Iván, Ghignone Ramiro, Giampetruzzi Julián (2016), **Desarrollo y Validación de Algoritmos en C para Cálculo de Posición de Usuario en el Sistema GPS**. Congreso Argentino de Ingeniería (CADI) 2016

[15] **UBLOX 6 Receiver Description**, Version 7.03.

[16] E. Gamma et al, **Patrones de Diseño**, Addison Wesley, 251-259.

[17] R.Ghignone, I. Castellucci Vidal, J.Giampetruzzi, F. Larosa, **Implementación y optimización de una biblioteca embebida para receptor GPS**, *Congreso de Microelectrónica Aplicada (uEA) 2016*