

# DISPOSITIVO DE ADQUISICIÓN Y REGISTRO DE DATOS PARA COHETES EXPERIMENTALES

LAROSA, Facundo - MIGNONE, Martín - CASTELUCCI VIDAL, Iván - FERNÁNDEZ, Martín  
flarosa@frh.utn.edu.ar - mnmignone@gmail.com - icvidal@frh.utn.edu.ar – martingabrielfernandez@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 un dispositivo capaz de registrar múltiples datos en un vehículo/vector aéreo a fin de poder, en tiempo diferido, determinar la trayectoria y características principales de vuelo a partir de un solo dispositivo integrado, que contenga un conjunto de sensores (giróscopo, acelerómetro, magnetómetro en tres ejes, barómetro, sensor de temperatura y módulo GPS) con el fin de analizar los factores estructurales que influyen en su movimiento y validar los futuros diseños de los mismos, permitiendo generar modelos con mayor precisión utilizando datos reales. Esto permitirá a futuro, tener una realimentación para el mejoramiento continuo de dichos modelos de vuelo.

## 1.- INTRODUCCION

Este dispositivo es una continuación del proyecto acelerómetro ACELERIX [1] con el fin de extender la cantidad de variables físicas registradas (además de la aceleración en tres ejes) durante el despegue, vuelo y aterrizaje de un vehículo/vector aéreo de forma tal de poder hacer, luego del aterrizaje (de forma *offline*) un estudio más completo sobre la dinámica del vuelo realizado. El potencial que tiene este proyecto es muy amplio, principalmente porque no solo se limita al estudio de los vectores aéreos, sino que puede utilizarse en cualquier tipo de aplicación en donde se requiera medir y registrar cualquier objeto en movimiento. A su vez, contribuye a generar una realimentación directa de datos valiosos para la etapa de diseño de este tipo de vectores, ya que ayuda a validar las predicciones realizadas durante dicha etapa, como también, mejorar y hacer más precisos los modelos de vuelo utilizados.

## 2.- METODOLOGIA

El registrador de vuelo se diseñó en forma modular, es decir, que cada variable o conjunto específico de variables físicas corresponde a un módulo con los sensores asociados; todos ellos montados sobre una única placa principal, conectados a un microcontrolador encargado de realizar todas las funciones de adquisición, registro y almacenamiento de los datos provistos por dichos módulos. Éstos facilitan la implementación del sistema completo y acortan tiempos de desarrollo, ya que proveen sus datos a través de una interfaz digital, evitando la necesidad de implementar circuitos de adaptación y acondicionamiento de señal, disminuyendo así, al mínimo la degradación de las mediciones por efecto del muestreo, la digitalización de las señales y el ruido eléctrico.

El microcontrolador utilizado es un LPC 4337 de la firma NXP [2] con la característica de poseer dos núcleos asimétricos, brindando la posibilidad de distribuir las tareas más exigentes en tiempo al núcleo más potente, y dejando las funciones de propósito general al otro. Así, se pueden separar las tareas de atención de los

módulos de entrada y salida de las tareas de procesamiento. Los módulos utilizados comprenden desde una unidad de medición inercial (IMU, por sus siglas en inglés *Inertial Measurement Unit*) [3] capaz de medir aceleración y velocidad angular en tres ejes, un magnetómetro en tres ejes, un receptor GPS, un barómetro y un sensor de temperatura. Las mediciones provenientes de diferentes fuentes son obtenidas por el microcontrolador y grabadas en una memoria SD para su registro y procesamiento.

Asimismo, se ha implementado el firmware con una metodología de diseño orientada a objetos, permitiendo la reutilización de gran parte del código, en caso de ser necesario modificar el firmware o la arquitectura de hardware. Así se promueve al desarrollo de una línea de instrumentos facilitando el agregado de funcionalidades de hardware (sensores adicionales, módulos de telemetría, etc.) o de software (procesamiento en tiempo real, filtros de fusión de datos, etc.) a futuro.

### 3.- DESARROLLO

El diagrama en bloques del registrador se muestra en la Figura 1.

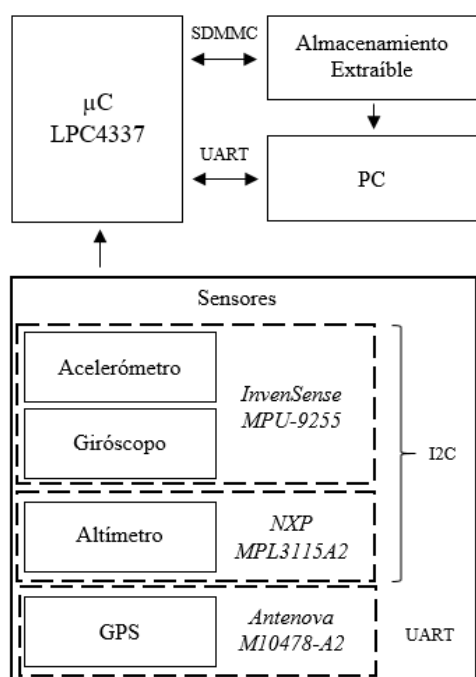


Figura 1. Diagrama en bloques del dispositivo.

Todos los componentes que se aprecian en la Figura 1, fueron montados en un circuito impreso personalizado y diseñado con el fin de obtener un dispositivo compacto capaz de montarse en un espacio reducido, facilitando la instalación del registrador en el vector bajo análisis. Además, se evitan efectos de vibraciones que puedan distorsionar las mediciones de interés, y causar daños a la placa. La versión actual del diseño mide 70 x 90 mm y pesa 80. En la Figura 2 se muestra la vista superior del mismo.

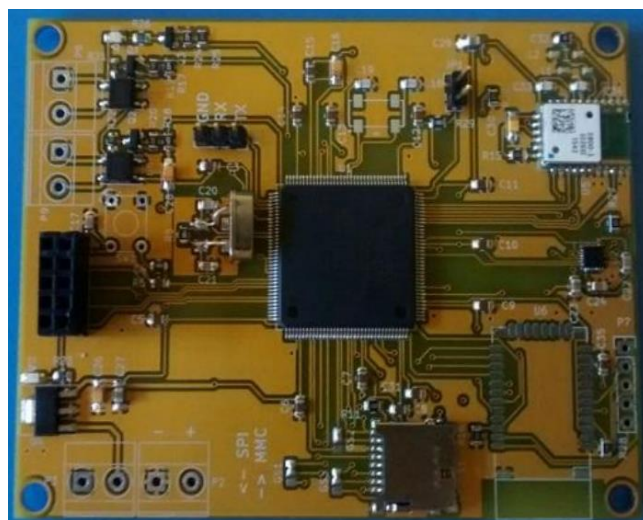


Figura 2. Placa principal del registrador de vuelo.

Todos los componentes fueron seleccionados para alimentarse con un solo nivel de tensión de 3.3 V, con el fin de evitar el uso de distintos reguladores y líneas de alimentación, promoviendo, a su vez, el uso de baterías para dicha función. La alimentación de la placa se realiza a través de un regulador LDO (regulador de baja caída, por sus siglas en inglés, *Low Drop Out*), al cual le provee energía una batería de litio de 3.7 V capaz de suministrar 1200 mAh al sistema. El registrador posee un consumo promedio de 200 mA, por lo cual esta batería es capaz de energizar al sistema por tiempos prolongados, lo cual la hace adecuada para ser utilizada durante las diferentes etapas de pruebas y durante la fase de vuelo.

El microcontrolador utilizado posee dos núcleos asimétricos, un Cortex-M4 [4] y un Cortex-M0 [5]. El primero está provisto con una unidad de punto flotante por hardware e instrucciones para procesamiento de múltiples

datos, por lo que es el encargado de realizar todos los cálculos y procesamiento de tiempo real. Por otro lado, el segundo al ser de propósito general, es utilizado para las tareas de menor exigencia.

Todos los módulos seleccionados poseen la característica de proveer sus medidas en forma digital, facilitando el registro de todos los datos provenientes de todos los sensores, y también el rango de alimentación comprende los 3,3 V utilizados en la placa principal.

El primero de ellos corresponde a una Unidad de Medición Inercial o IMU InvenSense MPU-9255 [6], la cual cuenta con un acelerómetro, giróscopo y magnetómetro de 3 ejes, proveyendo la aceleración, la velocidad angular y la intensidad del campo magnético respectivamente. Luego, remitiéndonos a la Figura 1, podemos apreciar el Altimetro NXP MPL3115A2 [3] el cual es un sensor de presión y temperatura con altímetro incluido compensado en temperatura, que junto, con la IMU proveen los principales datos sobre la dinámica del vuelo del vector. Ambos módulos poseen como interfaz la comunicación I2C.

Por último, el módulo GPS M10478-A2 de Antenova [7], integra una antena interna, lo que significa que no requiere conectores para ella ni el diseño de una línea de transmisión en el circuito impreso, facilitando su integración en la placa principal. La frecuencia de las soluciones de posición es de 10Hz, y la comunicación se realiza a través de una interfaz UART. La configuración del dispositivo se realiza por medio de comandos PMTK [8], brindando las soluciones en formato NMEA [9].

El almacenamiento de todos los datos registrados fue realizado mediante el uso de una tarjeta SD, conectada al microcontrolador a través de un zócalo, y configurada en modo MMC [10] lo cual asegura velocidades de transferencia mayores.

Además, el registrador posee dos salidas de tipo contacto seco por medio de relés de estado sólido para interactuar con componentes del

sistema externos al mismo, adaptables a cada aplicación.

Para que los datos puedan ser registrados, procesados y almacenados sin sufrir pérdidas y/o corrupción de los mismos, se diseñó un software con un esquema como el que se muestra en la Figura 3.

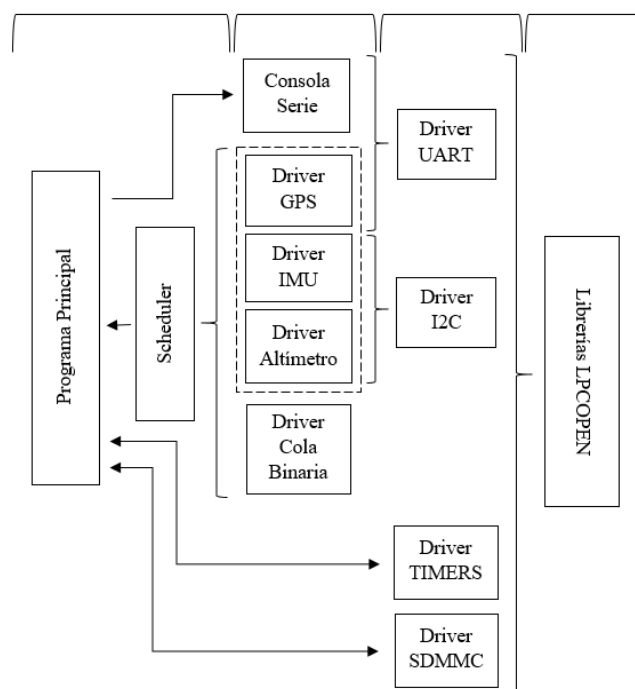


Figura 3. Esquema del software diseñado.

Como puede apreciarse el diseño del código se basó en la implementación de capas de software [11], generando de esta forma, una arquitectura en el cual están claramente divididas las partes dependientes del hardware que está siendo utilizado de las partes del código del registrador en sí mismo, permitiendo, frente a cualquier cambio de los componentes de hardware, que el código existente sea fácilmente adaptable al nuevo diseño. Otro beneficio de implementar el código de esta manera es que permite el desarrollo y prueba de distintos bloques en simultáneo, lo que posibilita distribuir la carga de trabajo entre múltiples programadores. Luego uno de ellos se encarga de unir los distintos bloques en el programa principal.

El diseño de software se compone de cuatro capas distintas: la capa superior es la del programa principal y la del sincronizador de tareas, los cuales permanecerán sin cambios aun si se decidiera optar por algún cambio de hardware. La capa inmediatamente inferior implementa los *drivers* de los respectivos sensores, además de *drivers* específicos de la aplicación. Estos son independientes del microcontrolador que se esté utilizando, y están implementados usando funciones genéricas que se desprenden de los drivers ubicados en la capa inferior. Estos *drivers* son los que interactúan con los periféricos del microcontrolador, y proveen a los programadores con funciones simples para el manejo y configuración de los mismos. A su vez estos últimos utilizan funciones proporcionadas por las librerías del *LPCOpen* [12], lo cual facilitó su implementación.

El registrador posee dos modos de funcionamiento: el modo registrador y el modo administrador.

El modo registrador involucra la adquisición y almacenamiento de los datos en la memoria SD y es el modo de funcionamiento principal del registrador. Las mediciones de los distintos sensores se adquieren de manera periódica partiendo de una base de tiempos generada por interrupción que se dispara con un período de 1 ms. Esto permite atender a los tiempos de la IMU que es el sensor más rápido del sistema sin perder muestras. En cada ingreso a la rutina de interrupción, una máquina de estados (*sensor\_sm*) se encarga de leer los sensores de manera no bloqueante, y almacena los datos obtenidos en una cola binaria para luego volcarlos al almacenamiento externo.

La estructura de datos tiene un tamaño fijo dada por un bloque de mediciones de la IMU (4500 bytes), luego una medición del altímetro (4 bytes), una medición del GPS (16 bytes), y para cerrar el bloque se inserta una suma de verificación (en inglés, llamada *checksum*) de 4 bytes al final. Si algún sensor no reportara datos, esos datos faltantes se reemplazan con un relleno (*stuffing*) de datos nulos. El *checksum* se calcula

como la OR exclusiva (XOR) de palabras de 4 bytes del bloque de datos hasta completarlo.

El muestreo de datos ocurre a elevada velocidad; sin embargo, la memoria SD posee una latencia variable para poder comenzar una operación de escritura, por lo que fue necesario implementar colas de datos para evitar pérdidas. Fueron utilizadas dos colas binarias rotativas; una de ellas, la "activa", en la cual se van almacenando los datos de los sensores a medida que van siendo leídos por el microcontrolador, y la otra "lista", que es una cola que ya ha sido completada y está disponible para ser almacenada en la memoria, tarea que ocurre en simultáneo con la recepción de nuevos datos, los cuales van siendo guardados en la cola "activa". Una vez que la cola "activa" es completada, se intercambian las funciones entre ambas colas, marcando a esta última como "lista", y limpiando la otra y marcándola como "activa". Cada cola cubre un segundo de datos, lo cual es ampliamente superior al tiempo máximo de latencia de la tarjeta SD.

Este modo de operación es utilizado "en vuelo" para luego realizar la descarga de información para el posterior análisis. El modo administrador de operación implementa la descarga de dicha información. Para ello, existe dos alternativas: por un lado, extrayendo la memoria SD del dispositivo para leerla desde cualquier ordenador; y la otra corresponde a la descarga a través de una interfaz serie, provista en la placa principal, por medio de una UART de 3 pines a 115200 baudios, la cual permite la administración de los archivos generados por el sistema.

## CONCLUSIONES

El proyecto se encuentra finalizado, habiendo cumplido con todos los objetivos propuestos, pasando satisfactoriamente todas las pruebas realizadas de cada módulo individual y en conjunto.

Actualmente, estamos aguardando para realizar el ensayo en vuelo del registrador, para

pasar a la siguiente etapa de procesamiento y análisis estadístico de los resultados.

[12] **LPCXpresso IDE User Guide**, *NXP Semiconductors*, Rev. 8.1 — 15 February, 2016, 9-16.

## REFERENCIAS

[1] Larosa, F.S., Mignone, M.N., Castelucci Vidal, I., Guanca, A.A., Giampetruzzi, J.G. (2015). **Módulo de adquisición y registro para un acelerómetro en tres ejes para vectores de alta dinámica**. Universidad Tecnológica Nacional, Facultad Regional Haedo, Haedo, Argentina.

[2] Product Data Sheet. **LPC435x/3x/2x/1x**, *NXP Semiconductors*, Rev. 5.3 — 15 March 2016,

[3] **IMU MPU-9255 Product Specification**, *InvenSense Inc*, Revision: 1.0, Release Date: 09/14/2014.

[4] Technical Reference Manual, **Cortex M4F**, Revision R0P0, *ARM Limited*, 2010.

[5] Technical Reference Manual, **Cortex M0**, Revision R0P0, *ARM Limited*. 2009.

[6] Data sheet: Technical data, **MPL3115A2 I2C precision pressure sensor with altimetry**, *NXP Semiconductors*, Rev. 5.1 — 13 September 2016.

[7] Product Specification, **GPS RADIONOVA® RF Antenna Module, Part No. M10478-A2-1**, Release Date 21 October 2015.

[8] **PMTK command packet**, Rev.A08, *GlobalTop Tech Inc*, 2011.

[9] **NMEA Reference Manual**, Rev.A08, *SiRF Technology, Inc*, 1050-0042 January 2005, Revision 1.3.

[10] **SDCard Datasheet**, V1.2, *Cypress Semiconductor Corporation*. Rev. \*D, 2012, 6-24.

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