

Implementación de una Plataforma de Desarrollo CANSAT Multipropósito

Leonardo Anchino[†], Andrés Torti, Emmanuel Doyis, Emanuel Bernardi
Grupo de Investigación en Robótica y Control (GIROC)
Universidad Tecnológica Nacional,
Facultad Regional San Francisco (UTN-FRSFCO)
San Francisco, Córdoba, Argentina
[†]anchinoleonardo@gmail.com

Resumen—El presente artículo describe los aspectos técnicos y procedimentales involucrados en la implementación de un nanosatélite no orbital tipo CanSat. Este dispositivo integra componentes mecánicos, electrónicos y elementos de software, desarrollados específicamente para fomentar la divulgación científico-tecnológica, y el despertar de vocaciones científicas. Por último, es importante remarcar que la presente plataforma se empleará como instrumento de desarrollo en la enseñanza de tecnologías espaciales.

Keywords—Educación, nanosatélite, electrónica, firmware, tecnologías espaciales

I. INTRODUCCIÓN

En la actualidad, el ingreso y posterior desgranamiento observado en las carreras de grado, con orientación tecnológica, se ha convertido en un importante obstáculo para el desarrollo de las actividades científicas y técnicas del país. Si bien no es tarea sencilla diagnosticar las razones que ocasionan dicha problemática, es pertinente identificar como causas a la falta de vocación, al desconocimiento y al desinterés en afrontar carreras altamente demandantes [1, 2].

Por tal motivo, desde la presente propuesta, enmarcada en el PID-UTN “Desarrollo de una plataforma educativa basada en proyectos CANSAT” (CCUTNSF0005414), que además constituye parte del proyecto final de grado de dos de los autores [3], se pretende exponer y argumentar las técnicas empleadas en el desarrollo e implementación de un nanosatélite no orbital tipo CanSat [4], que se utilizará como instrumento de divulgación de las tecnologías espaciales.

En base al entorno de aplicación del dispositivo, se consideró necesario cumplir con los siguientes requerimientos:

1. construir una plataforma lista para usar, fácil de replicar, y de diseño libre y abierto.
2. contemplar la capacidad de expansión a través de módulos de simple conexión y configuración que le permitan al usuario, con su funcionalidad base, obtener un CanSat completamente funcional.
3. cumplir con la normativa impuesta por la Agencia Espacial Europea (ESA) [5] que determina las dimensiones, el peso y las velocidades terminales.

Desde el punto de vista de su utilización, el objetivo principal de la plataforma es la realización de misiones de forma similar a las de un satélite orbital. Para ello, el CanSat

es elevado, a través de un globo cautivo o un cohete, a una altura máxima de 1000 m (o 150 m desde un cuadricóptero), para luego descender a velocidad controlada por medio de su paracaídas integrado. Durante la misión, ascenso y descenso, el satélite envía datos a tierra, que luego se analizan para cumplir con el objetivo de la misión, previamente propuesto.

Por último, es de destacar que la plataforma, capturada en la Fig. 1, forma parte de la propuesta de UNISEC-Argentina [6] para fomentar la enseñanza de tecnologías espaciales, y se plantea utilizarla en el dictado del CANSAT Workshop 2019.



Figura 1: Satélite CanSat

II. DESCRIPCIÓN DE LAS HERRAMIENTAS

La plataforma presentada consta de tres partes fundamentales, a saber:

- CanSat: dispositivo a ‘lanzar’ que incluye un sistema de protección interno de tensiones y corrientes, dos actuadores (liberación y paracaídas), un módulo de comunicaciones, una baliza electrónica y diversos sensores para la medición de variables físicas, como lo son la temperatura, la humedad, la presión, la aceleración, el campo magnético, entre otras.
- Estación terrena: módulo compuesto de un transceptor de radiofrecuencia que realiza el enlace entre la terminal de telemetría (en tierra) y el satélite CanSat (en vuelo).
- Software de telemetría: administra y visualiza todos los parámetros del satélite permitiendo la realización de misiones y la creación de reportes.

Durante el diseño de la plataforma se consideró importante la implementación a través de herramientas de software libre y abierto, de modo que su utilización facilite a los posibles usuarios replicarla y expandir su funcionamiento.

II-A. Desarrollo de hardware

En el desarrollo de los esquemas y los correspondientes paneles de circuito impreso, tanto del dispositivo CanSat como de la estación terrena, se empleó el software de diseño asistido por computadora *KiCad* [7]. Dicha herramienta, de acceso libre y gratuito, cuenta con una gran cantidad de librerías pre-diseñadas, o sencillamente realizables a través del software gratuito *Library Loader* de la empresa *SamacSys* [8], reduciendo su complejidad y el tiempo de desarrollo involucrado.

II-B. Desarrollo de firmware

La unidad de control del satélite CanSat está compuesta por el microcontrolador ESP32, de la empresa *Espressif* [9], por lo que se utilizó el framework de desarrollo ESP-IDF, provisto por la misma. Este se compone de la totalidad de los drivers de periféricos y provee las herramientas necesarias para compilar y grabar el firmware en el microcontrolador. Además, dicha herramienta se basa en una versión modificada de FreeRTOS [10], que nos permite aprovechar ambos núcleos.

II-C. Herramientas de desarrollo del software de telemetría

En el desarrollo de la propuesta se estableció que la aplicación sea fácil de utilizar, intuitiva, multiplataforma, multi-lenguaje y responsiva. Teniendo esto en cuenta, se utilizaron las siguientes herramientas:

1) *Template Vuestic Admin* [11]: plantilla de administración responsiva creada en *Vue.js* por la empresa *Epicmax*, bajo la licencia MIT [12]. Esta, cuenta con soporte para la inclusión de nuevos idiomas, además del español e inglés.

2) *Electron* [13]: framework de código abierto utilizado para la creación de aplicaciones de escritorio con HTML, JS y CSS. Permite crear ejecutables de la aplicación para los sistemas operativos Mac OS, Windows y Linux.

3) *Vue CLI Plugin Electron Builder* [14]: plugin que permite convertir la aplicación realizada en *Vue.js* a *Electron* manteniendo su funcionalidad.

III. IMPLEMENTACIÓN DEL HARDWARE

La plataforma de desarrollo se divide en dos dispositivos.

III-A. Cansat

Como se observa en la Fig. 2, el satélite CanSat se compone de múltiples PCBs apilados, e interconectados entre sí. Se los designa de arriba hacia abajo, y se los describe:

1) *PCB_A*: en este módulo se alojan los pines de conexión de la batería LiPo, de una celda, que luego se distribuyen en el dispositivo. Además, incluye la conexión de los servos de liberación de globo y paracaídas, el GPS para obtener su localización y altitud, el interruptor de encendido y apagado general del sistema, y el circuito transceptor que establece el enlace de radiofrecuencia con la estación terrena.

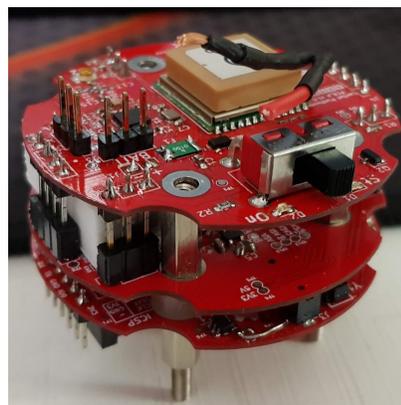


Figura 2: Estructura del CanSat.

2) *PCB_B*: aquí se gestiona la carga y descarga de energía en la batería. Esto es, se construyen las tensiones de alimentación requeridas por el sistema, incluso las tensiones auxiliares que utilizan los circuitos de expansión. Es de destacar que las últimas están protegidas frente a cortocircuitos, sobrecorrientes y sobre-tensiones, de modo que un fallo sobre las mismas no afecte el funcionamiento general, asegurando así la integridad del satélite. Las tensiones generadas son: +5 V para los servos de liberación de globo y paracaídas, +3.3 V para el sistema supervisor, y de manera diferenciada +5 V y +3.3 V para los circuitos de expansión.

3) *PCB_C*: consta principalmente del sistema de control y supervisión del satélite. Lo integra un microcontrolador ESP32, junto a los sensores de presión atmosférica, de temperatura y humedad, e inerciales. Entre sus funciones se destacan la activación del paracaídas, el control de las fuentes de alimentación y mantener la comunicación con la estación terrena. Esto asegura la integridad del satélite, y permite que los módulos de expansión no tengan que cumplir con dichas tareas, simplificando su diseño. Además, en esta se encuentran el conector de carga de la batería y el de expansión. Este último, permite ampliar el funcionamiento base del satélite, de forma segura y sencilla. Para ejemplificar el uso del conector de expansión, se desarrolló un módulo compatible con *Arduino UNO* y otro para prototipos, que contiene un sector perforado para crear circuitos simples rápidamente.

III-B. Estación terrena

Esta se compone de una antena acoplada a un transceptor de radiofrecuencia, con sus elementos de polarización y filtrado, que trabaja a 915 MHz y se conecta al terminal de telemetría para establecer comunicación con el satélite CanSat.

IV. CONCLUSIONES Y TRABAJO FUTURO

Luego de haber ensayado la plataforma a través de múltiples lanzamientos, de hasta 150 m de altura, se ha observado que los resultados alcanzados cumplen con los objetivos inicialmente propuestos. En resumen, se ha construido una plataforma de desarrollo, lista para usar, basada en un satélite CanSat, cuya operación es sencilla, y de diseño libre y abierto.

Las pruebas realizadas expusieron unos pocos aspectos de hardware a mejorar en la siguiente versión, que se encuentra en desarrollo. Por último, se planea ensayar la plataforma a su altura máxima de diseño (1000 m) que no fue alcanzada por falta de medios de elevación.

REFERENCIAS

- [1] UTN-BA. *Esfuerzos en todo el país para promover el ingreso a la UTN*. URL: <https://www.frba.utn.edu.ar/esfuerzos-en-todo-el-pais-para-promover-el-ingreso-a-la-utn/> (visitado 02-05-2019).
- [2] ITBA. *Apostar a las ingenierías, las carreras del futuro*. URL: <https://www.itba.edu.ar/la-universidad/prensa/apostar-a-las-ingenierias-las-carreras-del-futuro/> (visitado 02-05-2019).
- [3] Leonardo Anchino y Andrés Torti. «Desarrollo de satélite CanSat para utilizar como plataforma educativa y fomentar el interés en ingenierías». Proyecto final de carrera. Universidad Tecnológica Nacional, mar. de 2019.
- [4] European Space Agency. *What is a CanSat?* Feb. de 2019. URL: https://www.esa.int/Education/CanSat/What_is_a_CanSat (visitado 02-05-2019).
- [5] European Space Agency. *2019 CanSat Guidelines*. 2019.
- [6] UNISEC Global. *UNISEC Argentina Point of Contact*. URL: <http://www.unisec-global.org/reyna.html> (visitado 02-05-2019).
- [7] *KiCad EDA*. URL: <http://kicad-pcb.org/> (visitado 02-05-2019).
- [8] SamacSys. *Library Loader*. URL: <https://www.samacsys.com/library-loader/> (visitado 02-05-2019).
- [9] Espressif. *ESP32 Overview*. URL: <https://www.espressif.com/en/products/hardware/esp32/overview> (visitado 02-05-2019).
- [10] *FreeRTOS*. URL: <https://www.freertos.org/> (visitado 02-05-2019).
- [11] Epicmax. *Vuestic Admin Dashboard*. 2019. URL: <https://github.com/epicmaxco/vuestic-admin> (visitado 02-05-2019).
- [12] Epicmax. *Vue.js*. URL: <https://vuejs.org/> (visitado 02-05-2019).
- [13] *Electron*. 2019. URL: <https://electronjs.org/> (visitado 02-05-2019).
- [14] Noah Klayman. *Vue CLI Plugin Electron Builder*. 2019. URL: <https://nklayman.github.io/vue-cli-plugin-electron-builder/> (visitado 02-05-2019).