

Diseño e implementación de módulos funcionales (Sensores y Telemetría) de OBSW para el desarrollo de nanosatélites

Jaime García González

4 de julio de 2019

Universidad Carlos III de Madrid

- 1 Introducción
- 2 Análisis y diseño
- 3 Implementación y pruebas
- 4 Plan de proyecto
- 5 Conclusiones y trabajo futuro

1 Introducción

2 Análisis y diseño

3 Implementación y pruebas

4 Plan de proyecto

5 Conclusiones y trabajo futuro

Descripción del proyecto (1/3)

- Cátedra Universidad Carlos III - SENER.
- Desarrollo de un nanosatélite universitario (Proyecto MARTÍN-LARA).
- Tomar fotografías de la Tierra y estudiar los efectos de la radiación en componentes electrónicos.
- Filosofía *NewSpace*: Colonizar el espacio con nanosatélites y ofrecer nuevos servicios.

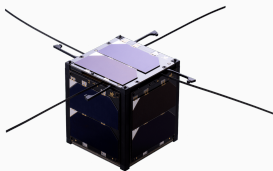


Figura 1: *CubeSat*. [Fuente: <https://satsearch.co/>]

Estándar *CubeSat*:

- Desarrollar nanosatélites en aulas universitarias.
- Restricciones de dimensiones ($10 \times 10 \times 10 \text{ cm}$) y masa ($1,33 \text{ kg}$).
- Múltiples aplicaciones: observación de la Tierra, comunicaciones, IoT.

Descripción del proyecto (3/3)

La mayoría de estas misiones se componen de:

- Onboard Software (OBSW).
- Ground System (GS).

Comunicación permanente mediante *data link*.

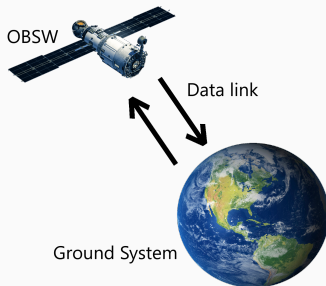


Figura 2: *Data link* entre OBSW y GS

El OBSW de estos nanosatélites se caracteriza por:

- **Modificación del entorno.** Reciben información del entorno utilizando sensores y modifican su estado con actuadores.
- **Sistema crítico.** Ejecutado en un sistema operativo en tiempo real, RTOS. Dos premisas:
 1. Ejecutar acciones correctas.
 2. Garantizar plazos de respuesta.

El incumplimiento de una de estas condiciones implica el **fallo total** de la misión.

1. **Definición de la arquitectura del sistema.** Se participará en el diseño de la arquitectura global de la misión.
2. Diseño e implementación del simulador térmico. Definición, diseño e implementación de los diferentes componentes térmicos del sistema.
3. Diseño e implementación del módulo de telemetría. Definición e implementación de los paquetes transmitidos entre diferentes componentes del sistema.

Objetivos

1. **Definición de la arquitectura del sistema.** Se participará en el diseño de la arquitectura global de la misión.
2. **Diseño e implementación del simulador térmico.** Definición, diseño e implementación de los diferentes componentes térmicos del sistema.
3. **Diseño e implementación del módulo de telemetría.** Definición e implementación de los paquetes transmitidos entre diferentes componentes del sistema.

Objetivos

1. **Definición de la arquitectura del sistema.** Se participará en el diseño de la arquitectura global de la misión.
2. **Diseño e implementación del simulador térmico.** Definición, diseño e implementación de los diferentes componentes térmicos del sistema.
3. **Diseño e implementación del módulo de telemetría.** Definición e implementación de los paquetes transmitidos entre diferentes componentes del sistema.

1 Introducción

2 Análisis y diseño

3 Implementación y pruebas

4 Plan de proyecto

5 Conclusiones y trabajo futuro

Core Flight Executive (cFE)

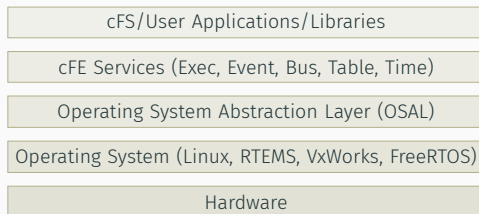


Figura 3: Arquitectura sistema CFS

cFE es un OBSW desarrollado por NASA, que se caracteriza por:

- *Framework* de desarrollo y entorno de ejecución.
- Conjunto de servicios (Software Bus, Time, Events, etc).
- Análisis de rendimiento en tiempo real.
- Arquitectura modular.

Arquitectura de la misión

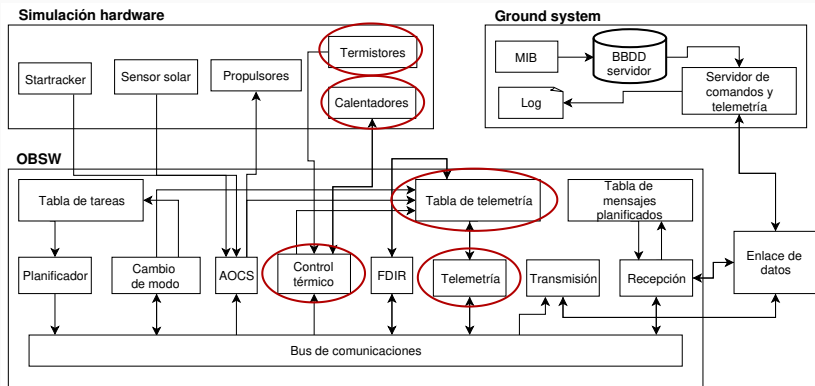


Figura 4: Arquitectura de la misión

El sistema se compone de tres subsistemas: Simulación hardware, OBSW y ground system.

Diseño de la solución

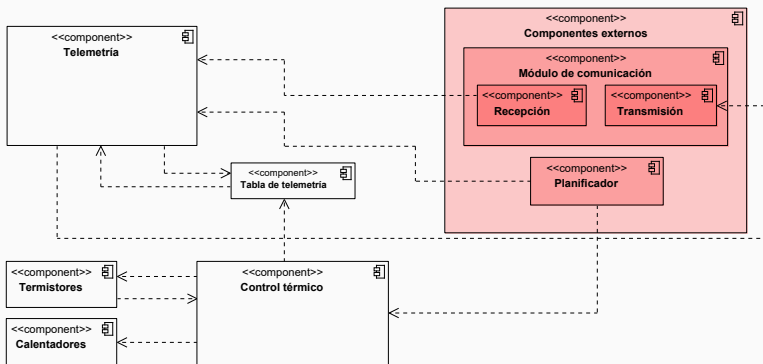


Figura 5: Diagrama de componentes

- 1 Introducción
- 2 Análisis y diseño
- 3 Implementación y pruebas**
- 4 Plan de proyecto
- 5 Conclusiones y trabajo futuro

Los componentes desarrollados se dividen en dos bloques:

- **Módulo de telemetría.** Monitorizar el estado del sistema y enviar los paquetes al módulo de transmisión.
- **Simulador térmico.** Control de la temperatura interna del sistema. Simulación software de componentes hardware.

Los componentes desarrollados se dividen en dos bloques:

- **Módulo de telemetría.** Monitorizar el estado del sistema y enviar los paquetes al módulo de transmisión.
- **Simulador térmico.** Control de la temperatura interna del sistema. Simulación software de componentes hardware.



Figura 6: Estructura de paquete

- **Cabecera.** Identificador de paquete, *checksum* y *timestamp*.
- **Segmento de datos.** Valores de sensores, actuadores y control del sistema.

Principales características:

- Evitar números en coma flotante (errores de precisión).
- Reducir *overhead* en sistemas empujados.
- Optimización de espacio empaquetando indicadores binarios en una variable de 32 bits.

31	30	29	28	...	1	0
----	----	----	----	-----	---	---

Figura 7: Empaquetado de indicadores binarios

Paquete periódico

- Envío periódico (0.2 Hz).
- Toda la información del sistema.
- Incluye *flags* de estado.

Paquete puntual

- Envío \Leftrightarrow solicitud del GS.
- Diferentes tipos de paquetes (térmico, posicional y solar).
- Incluye *flags* de estado.

Simulador térmico (1/2)

Simulación software de los componentes térmicos del nanosatélite y módulo del OBSW.

- **Calentadores.** Simulan un aumento de la temperatura.
- **Termistores.** Simulan una lectura de temperatura.

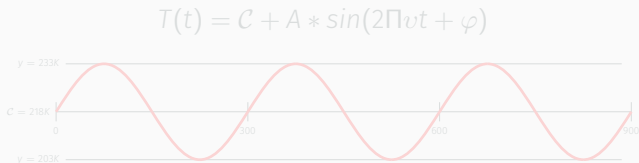


Figura 8: Simulación lectura de temperatura

Simulador térmico (1/2)

Simulación software de los componentes térmicos del nanosatélite y módulo del OBSW.

- **Calentadores.** Simulan un aumento de la temperatura.
- **Termistores.** Simulan una lectura de temperatura.

$$T(t) = C + A * \sin(2\pi vt + \varphi)$$

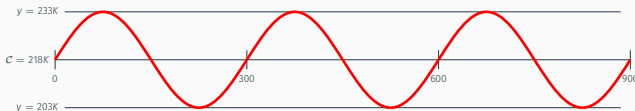


Figura 8: Simulación lectura de temperatura

- **Control térmico.** Módulo de OBSW que controla la temperatura interna del sistema.

Verifica los umbrales de temperatura de cada termistor y enciende/apaga su calentador asociado.

Termistor \rightarrow verificar umbrales \rightarrow calentador

Para lograr una simulación real, se añade un factor de ruido.

$$T_i = \text{Thermistor}_i + \text{rand}(-2, 2) \mid \forall i \in \{1, 2, 3\}$$

Pruebas y evaluación (1/2)

Pruebas de verificación e integración para todos los componentes desarrollados.

Entorno de pruebas

- Intel Core i7 - 2.60 GHz.
- 12 GB RAM DDR3.

Sistema final

- ARM Cortex A8 - 1 GHz.
- 512 MB LPDDR RAM.

Intervalo de confianza para comprobar que se cumplen las restricciones temporales.

$$\mu = \bar{x} \pm z \frac{\sigma}{\sqrt{n}}$$

Pruebas y evaluación (2/2)

Módulo	Cómputo máximo (ms)	Cota inferior (ms)	Cota superior (ms)
Telemetría	31.25	0.0168	0.0186
Control térmico	31.25	0.0162	0.0705

Tabla 1: Intervalo de confianza

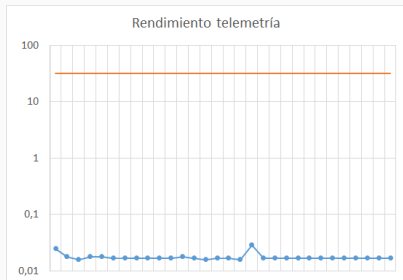


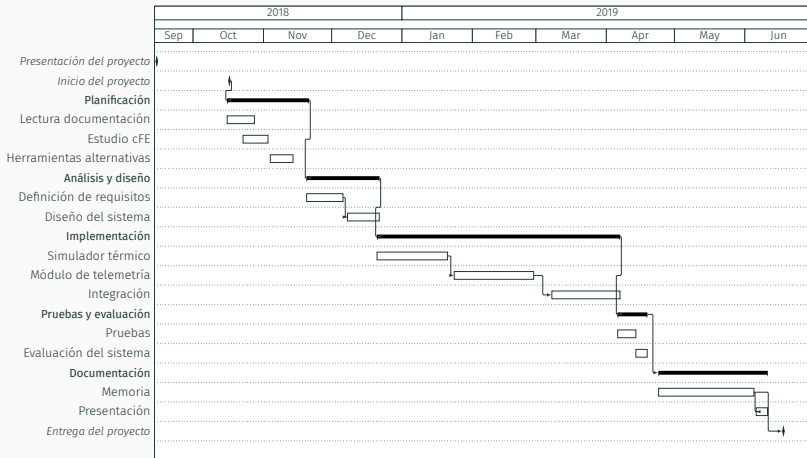
Figura 9: Rendimiento telemetría



Figura 10: Rendimiento c. térmico

- 1 Introducción
- 2 Análisis y diseño
- 3 Implementación y pruebas
- 4 Plan de proyecto**
- 5 Conclusiones y trabajo futuro

Planificación



Presupuesto

El proyecto ha tenido una duración aproximada de **8 meses**. El coste total del mismo asciende a **14931.47 €**.

Descripción	Coste
Costes de personal	8560.00 €
Amortización	132.13 €
Costes indirectos	282.10 €
Margen de imprevistos (10 %)	897.46 €
Margen de beneficio (25 %)	2468.02 €
Total sin I.V.A.	12340.06 €
Total	14931.47 €

Tabla 2: Resumen de costes

El proyecto está sujeto a las siguientes leyes/normativas:

- **Tratado sobre el espacio exterior.** Control de actividades espaciales de los Estados, registro de lanzamientos, etc.
- **CCSDS.** Desarrollo de estándares para misiones espaciales.
- **ECSS.** Desarrollo de estándares para misiones espaciales a nivel europeo.
- **CNAF.** Control de radiofrecuencias en el territorio nacional.

- 1 Introducción
- 2 Análisis y diseño
- 3 Implementación y pruebas
- 4 Plan de proyecto
- 5 Conclusiones y trabajo futuro**

- **Arquitectura del sistema.** Arquitectura modular, componentes independientes, fácil mantenimiento y desarrollo.
- **Simulador térmico.** Simulación de lectura y aumento de temperatura. Control de la temperatura interna del sistema.
- **Módulo de telemetría.** Definición e implementación de paquetes periódicos y puntuales. Optimización de espacio, evitando pérdida de precisión. Adaptado para sistemas empotrados.

- **Arquitectura del sistema.** Arquitectura modular, componentes independientes, fácil mantenimiento y desarrollo.
- **Simulador térmico.** Simulación de lectura y aumento de temperatura. Control de la temperatura interna del sistema.
- **Módulo de telemetría.** Definición e implementación de paquetes periódicos y puntuales. Optimización de espacio, evitando pérdida de precisión. Adaptado para sistemas empotrados.

- **Arquitectura del sistema.** Arquitectura modular, componentes independientes, fácil mantenimiento y desarrollo.
- **Simulador térmico.** Simulación de lectura y aumento de temperatura. Control de la temperatura interna del sistema.
- **Módulo de telemetría.** Definición e implementación de paquetes periódicos y puntuales. Optimización de espacio, evitando pérdida de precisión. Adaptado para sistemas empotrados.

Primer año de la Cátedra UC3M - SENER, trabajo a realizar en siguientes iteraciones:

- **Integración con Simulink.** Simulación con modelos reales (*model-in-the-loop*).
- **Gestión de memoria.** Nuevos paquetes, corrección de errores, nueva funcionalidad (*payloads*).
- **Documentación.** Generar documentación del código con Doxygen.
- **Sensores/actuadores.** Nuevos componentes que complementen la misión.

Diseño e implementación de módulos funcionales (Sensores y Telemetría) de OBSW para el desarrollo de nanosatélites

Jaime García González

4 de julio de 2019

Universidad Carlos III de Madrid