CAMPUS
DE EXCELENCIA
INTERNACIONAL

Graduado en Ingeniería Informática

Universidad Politécnica de Madrid

Escuela Técnica Superior de
Ingenieros Informáticos

TRABAJO FIN DE GRADO

Integración y validación del software del computador de a bordo del satélite UPMSat-2.

Autor: David Herrero Sánchez

Director: Juan Zamorano Flores

MADRID, ABRIL DE 2018

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Resumen

Aquí el texto del abstract.
Palabras clave: palabra 1, palabra 2, palabra 3. . .

Abstract

Here goes the abstract text.

Keywords: keyword 1, keyword 2, keyword 3…

1 INTRODUCCIÓN

El satélite UPMSat-2 forma parte del proyecto UPMSat-2 UNION liderado por el Institudo de Microgravedad Ignacio Da Rivas (IDR) junto con la Universidad Politécnica de Madrid (UPM).

Como denota el nombre del satélite, este proyecto nace como sucesión al ya lanzado exitosamente UPM-SAT1, y está siendo desarrollado en colaboración con las distintas escuelas que componen la UPM, cada una aportando sus conocimientos de ingeniería en los que está especializada.

Como detalles técnicos del satélite cabe destacar la información proporcionada en la página web del proyecto[1]:

- Peso: 50 kg.
- Medidas: 0.5m x 0.5m x 0.6m (ancho, largo, alto).
- Volumen útil: 0.4m x 0.4m x 0.25m.
- Carga útil: 15 kg.
- Potencia: 15 W.
- Órbita polar: 600 km de altitud.
- Vida útil: 2 años.

1.1 Objetivos

Este Trabajo de Fin de Grado (TFG) tiene como objetivo integrar el software (SW) de vuelo del computador de a bordo (OBC[1]) del satélite.

Los objetivos concretos son los siguientes:

- Estudio y comprensión del SW de los distintos subsistemas ya desarrollados.
- Análisis en profundidad del documento de requisitos software (SRS [2]).
- Desarrollo de un programa que permita ejecutar una batería de pruebas.

[1] On Board Computer
[2] Software Requirements Specification

- Desarrollo de un programa que permita diseñar las baterías de pruebas de forma gráfica.
- Validación y verificación de los requisitos SW.
- Preparación de parches para arreglar los fallos detectados en la validacion.

1.2 Estructura del documento

Este documento está estructurado de la siguiente forma: en el capítulo **??** se describe a grandes rasgos cuales son las restricciones bajo las que se ha desarrollado el SW del OBC, sus necesidades y cómo son las herramientas de desarrollo de las que se dispone.

A COMPLETAR CON EL RESTO DE CAPÍTULOS.

2 ESTADO DEL ARTE

El software del OBC del UPMSat-2 está desarrollado en el lenguaje Ada usando un compilador cruzado para procesadores LEON3.

Los motivos para emplear este lenguaje de programación son bastantes, pero principalmente destaca por ser un lenguaje orientado al desarrollo de software para sistemas empotrados y de tiempo real como es el UPMSat-2, ya que incorpora de forma innata todos los mecanismos necesarios para el manejo y control de la concurrencia. Además cabe destacar que el lenguaje Ada es uno de los lenguajes oficiales de la Agencia Espacial Europea (ESA[2]) para este tipo de sistemas.

Puesto que cada uno de los subsistemas que componen el OBC tiene sus propias tareas o hilos de ejecución para llevar a cabo su funcionalidad, es necesario controlar de alguna manera sus prioridades y la compartición de recursos. Para ello, se emplea el perfil de Ravenscar[3], que es un subconjunto de las características de Ada que restringe ciertos aspectos como los siguientes:

- Una tarea con prioridad N no puede acceder a recursos compartidos (conocidos como *objetos protegidos o protected objects*) de prioridad inferior.
- Si una tarea con prioridad N accede a un objeto protegido de prioridad M con $M \geq N$, esta tarea adquiere la prioridad M hasta que haya terminado de usar dicho objeto.
- Una tarea no puede llamar a otra directamente.
- Un objeto protegido solo puede tener una *entry* [1].
- Solo puede haber una tarea esperando en la *entry* de un objeto protegido.
- Todos los tiempos de espera (sentencias *delay*) deben ser relativos.

En el caso de que no se cumpla alguna de las normas anteriores se lanzará automáticamente la excepción PROGRAM_ERROR finalizando la ejecución del software.

Gracias al perfil de Ravenscar se consigue controlar de forma mucho más eficiente la concurrencia, evitando entre otras cosas la aparición de *deadlocks*.

[1] Procedimiento en el que una tarea va a quedarse esperando hasta que se dé una condición establecida en el objeto protegido.

3 BIBLIOGRAFÍA

[1] I. de Microgravedad Ignacio Da Rivas. (2018). Página web oficial del proyecto upmsat-2 union, dirección: http://www.idr.upm.es/tec_espacial/upmsat2/01_UPMSAT2.html.

[2] E. S. Agency. (2018). Lenguajes de programación avalados por la ESA, dirección: http://www.esa.int/TEC/Software_engineering_and_standardisation/TECRFBUXBQE_0.html.

[3] A. Burns, «The ravenscar profile», 1999. dirección: http://www.dit.upm.es/~str/proyectos/ork/documents/RP_spec.pdf.