



POLITÉCNICA

STRAST

Software embarcado del microsatélite UPMSat-2

Grupo de Sistemas de Tiempo Real y Arquitectura de Servicios Telemáticos
Universidad Politécnica de Madrid

Juan Zamorano, Alejandro Alonso, Emilio Salazar, Ángel Esquinas, Jorge Garrido,
Daniel Brosnan, Peter Bradley, Juan Antonio de la Puente



STRAST

Desarrollo del software

- Para el desarrollo del software embarcado se usará preferentemente un lenguaje de alto nivel que permita analizar las propiedades temporales y de integridad del software. Los lenguajes Ada y SPARK cumplen estas características.
- Se usará preferentemente software libre, siempre que esté disponible, para las herramientas de desarrollo de software y para las bibliotecas externas que se incluyan en el sistema de software.
- Se usarán técnicas de ingeniería de software adecuadas para facilitar la verificación y validación del software, y para asegurar su mantenimiento. Para ello se aplicarán los principios de modularidad, jerarquía, y ocultamiento de información, entre otros.
- Se aplicarán los estándares ECSS-E-ST-40C y ECSS-Q-ST-80C al desarrollo del software.
- Las herramientas necesarias para la compilación y depuración de software deben estar disponibles durante un tiempo suficiente para poder efectuar el mantenimiento del mismo, y al menos hasta el final del año 2016.
- Se usará el sistema internacional de unidades (SI) para todas las unidades de ingeniería del sistema.

STRAST

Software de a bordo: funcionalidad

- Control y determinación de actitud (ADCS).
- Adquisición y procesado de telemetría (TM).
- Descodificación y procesamiento de órdenes remotas (TC).
- Supervisión y control de la plataforma.
- Gestión del tiempo.
- Detección, aislamiento y recuperación de fallos de los subsistemas de la plataforma (FDIR).
- Adquisición de datos de mantenimiento (housekeeping).
- Adquisición, gestión y almacenamiento de datos de la carga de pago.
- Registro de eventos.

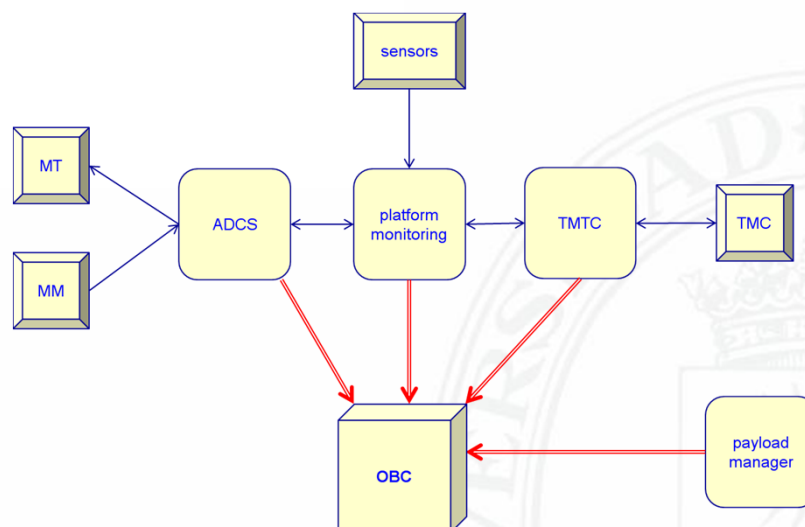
2011-05-12

©2011 grupo STRAST

3

STRAST

Computador de a bordo: contexto



2011-05-12

©2011 grupo STRAST

4

Computador de a bordo: síntesis

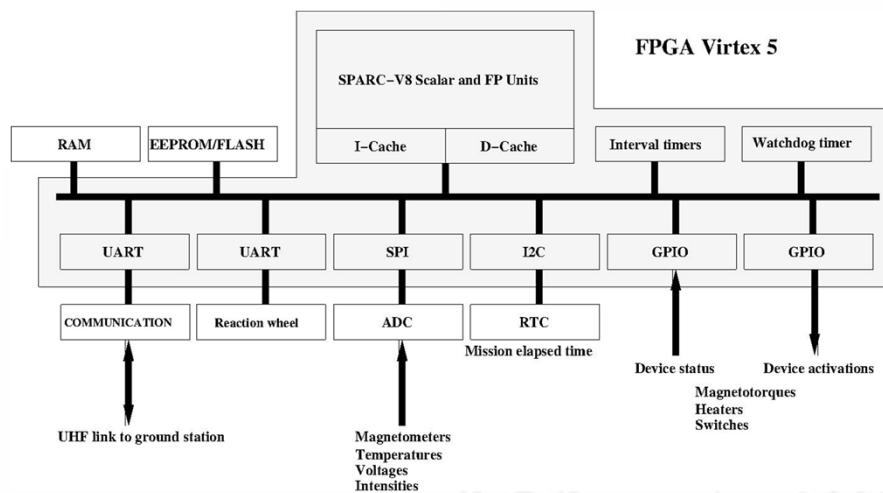
- Desarrollado a partir de la biblioteca de IP cores en VHDL GRLIB de Aeroflex Gaisler.
 - GNU GPL open-source license.
- IP Cores:
 - LEON3
 - Temporizadores
 - Controladores de bus AMBA
 - UART, I2C, SPI, GPIO
- ISE Design Suite de Xilinx para PLDs

2011-05-12

©2011 grupo STRAST

5

Computador de a bordo: componentes



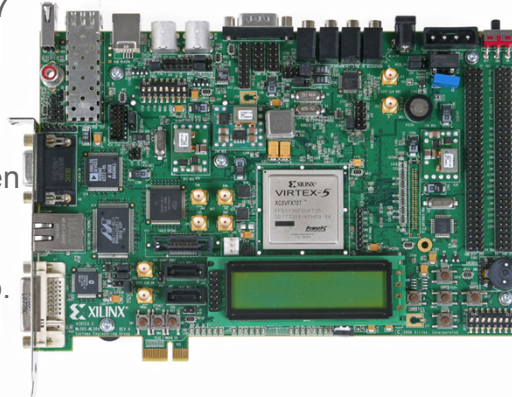
2011-05-12

©2011 grupo STRAST

6

Computador de a bordo: modelos

- Modelo de ingeniería para desarrollo basado en la placa Virtex-5 FXT FPGA ML507 Evaluation Platform.
 - Desarrollado por STRAST/UPM
- Modelo de vuelo basado en una FPGA Virtex-5 con memorias tolerantes a la radiación y diseño robusto.
 - Desarrollado por TECNOBIT



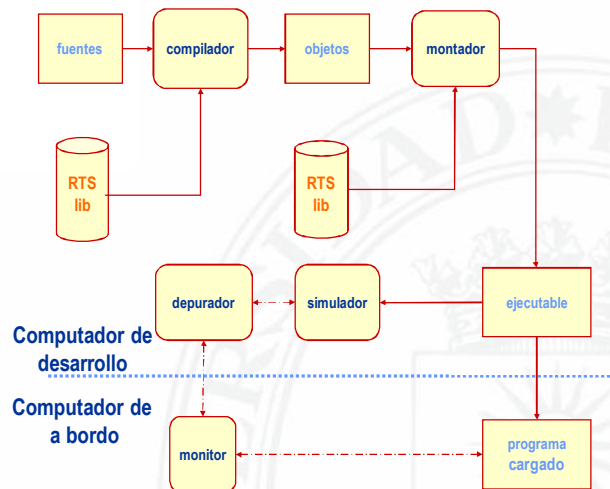
2011-05-12

©2011 grupo STRAST

7

Software de a bordo: desarrollo cruzado

- Herramientas de desarrollo de GNU
 - Compilador GCC/GNAT
 - Depurador GDB
 - Biblioteca newlib
- Simulador TSIM de Aeroflex Gaisler
- Monitor GRMON de Aeroflex Gaisler
- Enlace de depuración JTAG



2011-05-12

©2011 grupo STRAST

8

STRAST

Software de a bordo: software de desarrollo

- Sistema operativo de tiempo real ORK de STRAST/UPM para LEON3.
- Herramienta de generación de código para UML/MARTE de STRAST/UPM.
- Sistema de desarrollo cruzado para Ada generado para SPARC v8 a partir de las herramientas de GNU.
- Desarrollo de manejadores propios para UART, I2C, SPI, memoria FLASH, entradas y salidas digitales, RTC y ADC.
- Sistema de control de versiones subversion y de control de *bugs* mantis.
- Herramientas para medidas de WCET: RapiTime (Rapita systems), Absint Advanced Analyzer (Absint) y Bound-T (Tidorum).
- Herramienta de generación de código para MATLAB/Simulink de MathWorks.

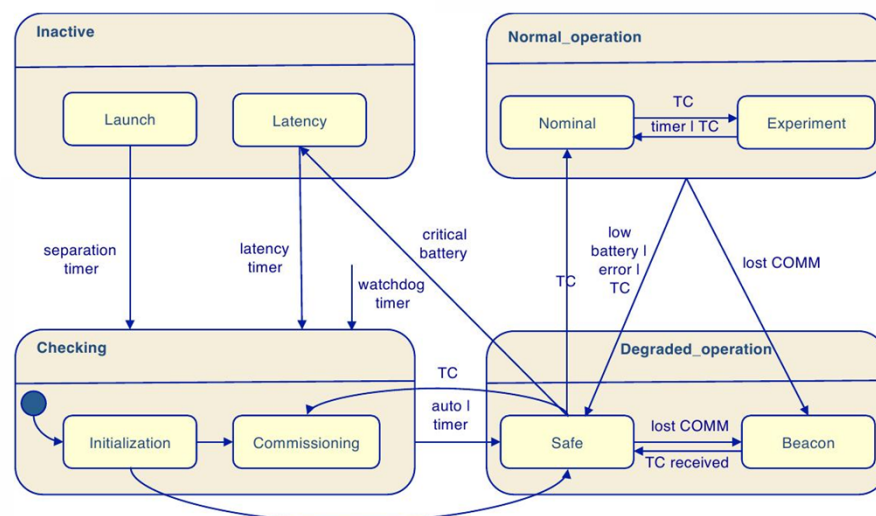
2011-05-12

©2011 grupo STRAST

9

STRAST

Software de a bordo: modos de funcionamiento



2011-05-12

©2011 grupo STRAST

10

Software de a bordo: adquisición de datos

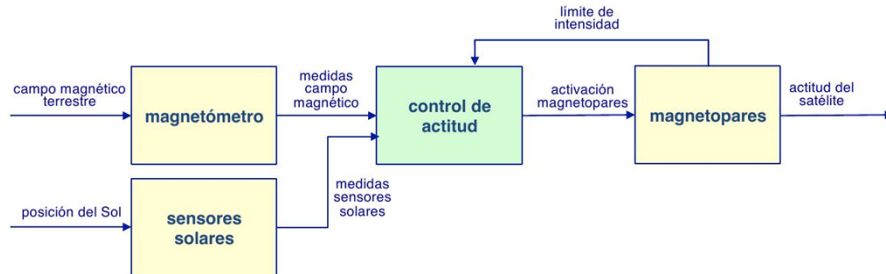
- ▶ El software leerá periódicamente los sensores de datos de mantenimiento de la plataforma (housekeeping) para supervisar su funcionamiento.
 - Temperaturas: caras del satélite, baterías, magnetómetros, OBC, experimentos.
 - Actitud: salida MGT, intensidad en MGT, células solares
 - Energía: tensión en bus, tensión en baterías, intensidades en baterías.
 - Intensidad en paneles solares.

Software de a bordo: comunicaciones

- ▶ Descodificación de los telecomandos para:
 - Cambiar de modo
 - Puesta en marcha de experimentos
- ▶ Almacenamiento, empaquetado y envío de la telemetría cuando el satélite entre en cobertura:
 - 2 veces al día durante 6 minutos.
 - Velocidad 9600 baudios.
 - Datos de *housekeeping* y experimentos.

Software de a bordo: control de actitud

- ADCS: Attitude Determination and Control System
 - Control de la orientación del satélite.



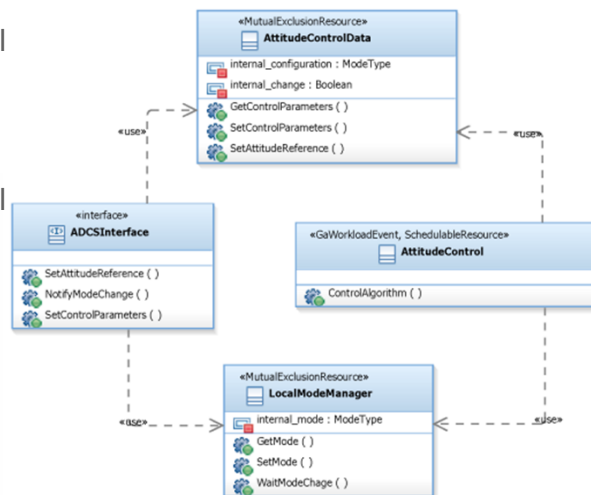
2011-05-12

©2011 grupo STRAST

13

Control de actitud: diagrama de clases

- El código funcional se genera a partir del modelo Simulink.
- El código funcional se incluye en la tarea periódica generada a partir de AttitudeControl por la herramienta MDD.



2011-05-12

©2011 grupo STRAST

14

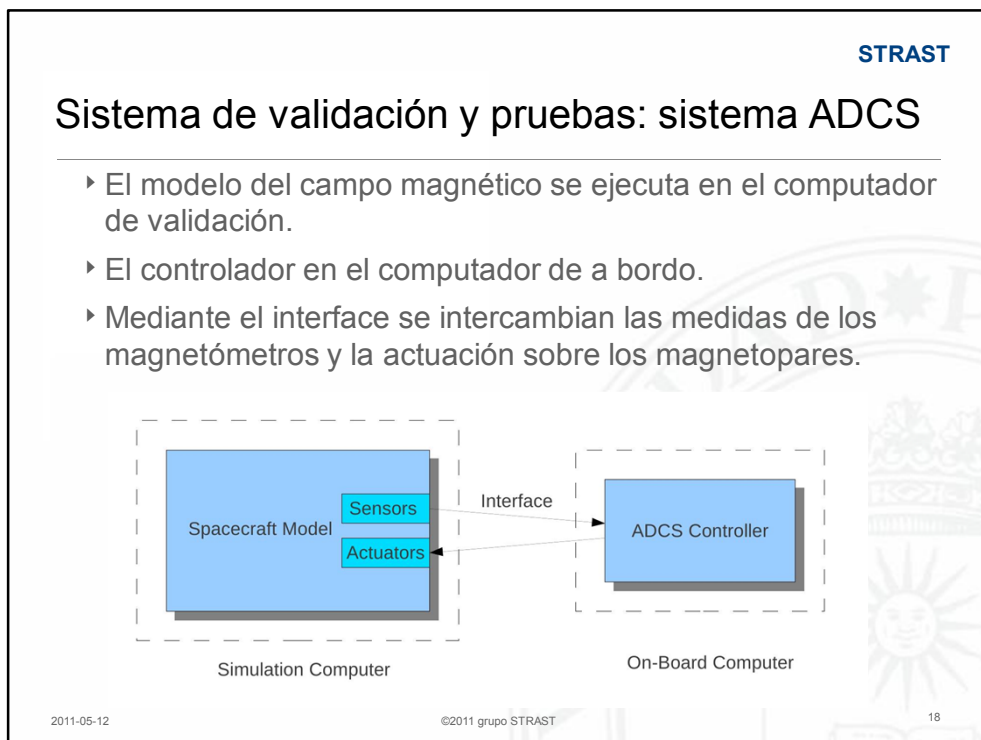
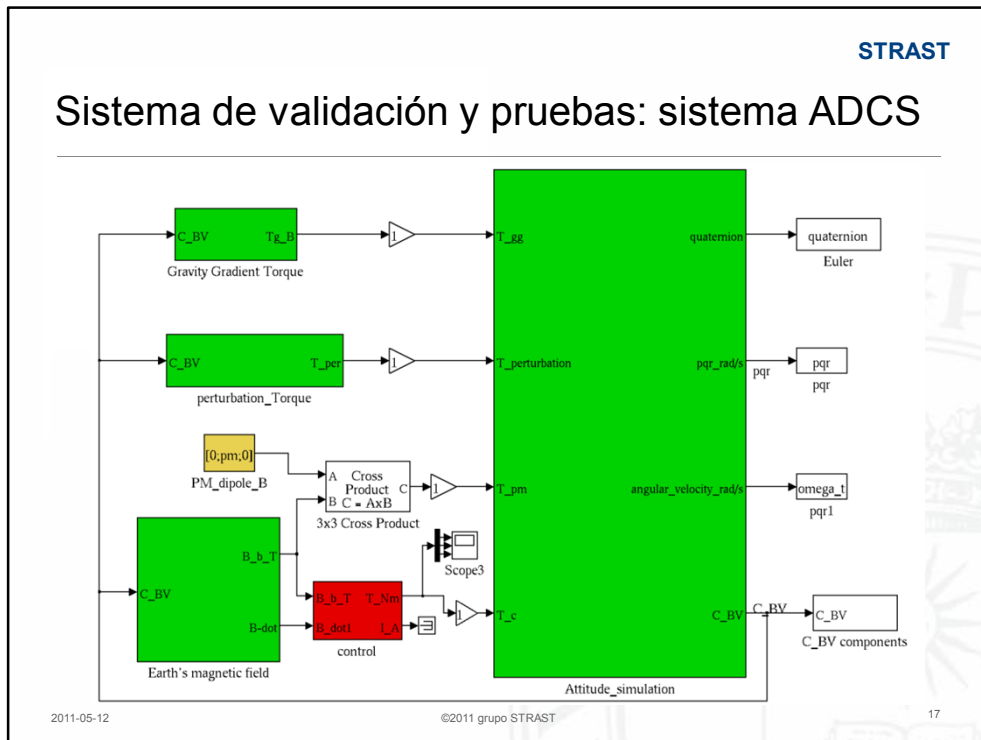
Sistema de validación y pruebas: funcionalidad

- ▶ La función de este sistema es simular el entorno del satélite para las pruebas de los subsistemas del computador de a bordo.
 - Simulación del campo magnético terrestre para el control de actitud mediante modelo.
 - Simulación de la plataforma mediante envío de señales analógicas y digitales.
 - Comprobación de la actuación sobre las salidas digitales
 - Envío de telecomandos y recepción de telemetría mediante línea serie.

Sistema de validación y pruebas: computador

El sistema de validación y pruebas se efectuará desde un computador con las siguientes características:

- ▶ Computador con arquitectura PC/ix86.
- ▶ Sistema operativo MS Windows (32 bits).
- ▶ Interfaz gráfica.
- ▶ Software MATLAB y Simulink con los Toolboxes para Control System y Data Acquisition.
- ▶ Tarjetas con salidas analógicas y digitales: Advantech PCI 1710, PCI 1720 y PCI 1730.



Conclusiones

- ▶ Proyecto multidisciplinar: Ingeniería Aeroespacial, de Telecomunicación e Informática:
 - Ingeniería de software
 - Sistemas empotrados y de tiempo real
 - Estructura de computadores
 - Sistemas operativos
 - Co-diseño hardware/software
- ▶ Demostrador tecnológico de las herramientas y desarrollos del grupo.
- ▶ Enseñar tecnologías usadas en la industria espacial europea.

