



Universidad de Chile
Facultad de Cs. Físicas y Matemáticas
Departamento de Ingeniería Eléctrica

Diseño e implementación del software de vuelo para un nano-satélite tipo cubesat

Carlos González Cortés

Miembros de la comisión

Dr. Marcos Díaz Quezada

Dr. Claudio Estévez Montero

Ing. Alex Díaz Becerra

Universidad de Chile

Tabla de contenidos

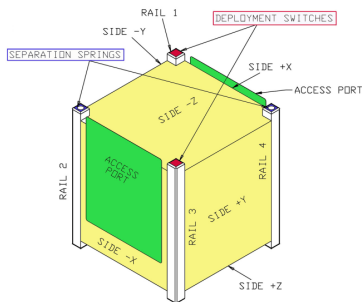
- 1 Introducción
- 2 Marco teórico
 - Sistemas embebidos
 - Sistemas operativos
 - Patrones de diseño
- 3 Diseño
 - Requerimientos operacionales
 - Arquitectura de software
- 4 Implementación
 - Clientes
 - Comandos
 - Procesador de comandos
 - Ejecutor de comandos
- 5 Resultados
 - Control central
 - Energía
 - Comunicaciones
- 6 Conclusiones

Introducción

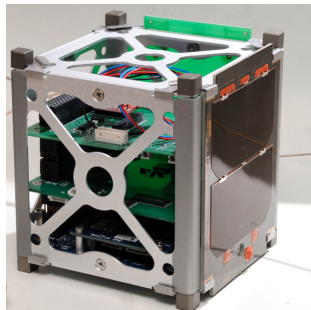
Proyecto SUCHAI

Diseño, construcción, lanzamiento y operación de un nano-satélite, con fines educativos y científicos.

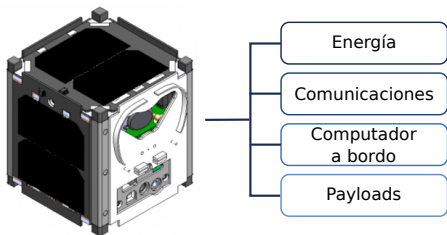
Es el primer proyecto satelital desarrollado por estudiantes en el país.



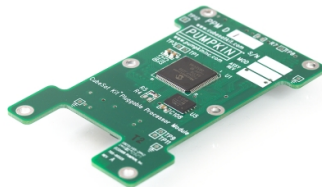
(a) Estandar Cubesat



(b) Cubesat SUCHAI



(c) Subsistemas



(d) OBC

Computador a bordo

Controla todas las operaciones del satélite e integra los diferentes subsistemas. Principales características:

- Microcontrolador PIC24F
- CPU @ 32 MHz
- Memoria RAM de 16 kB
- Memoria FLASH de 256 kB

Objetivos generales del trabajo

Diseñar e implementar el software que controla las operaciones del satélite una vez en órbita

Definir
Requerimientos



Diseñar



Implementar

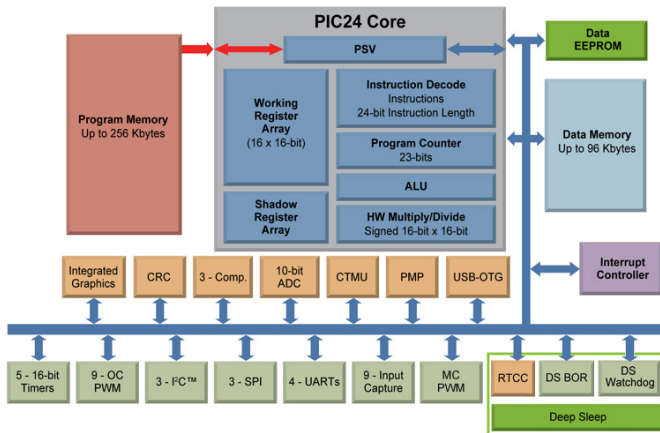


Verificar
Requerimientos



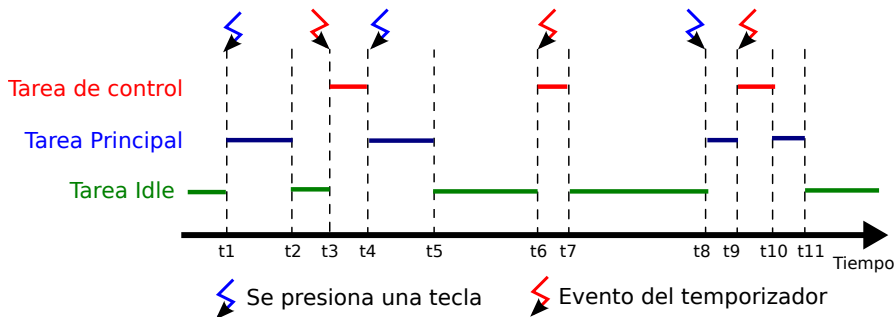
Probar

- Diseñados para cumplir funciones específicas .
- Aplicaciones de tiempo real.
- Alto nivel de integración: CPU y periféricos

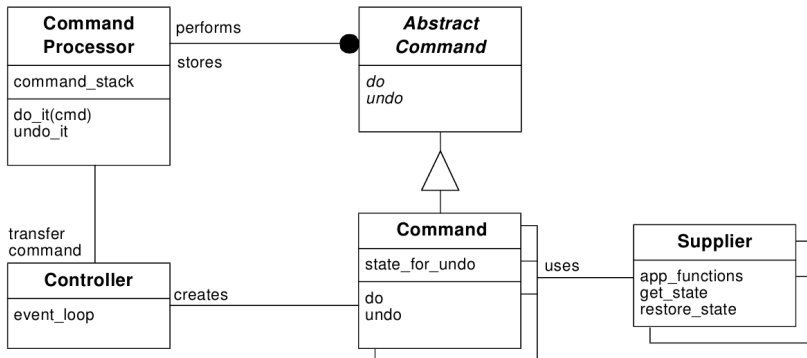


■ Sistemas operativos de tiempo real (RTOS)

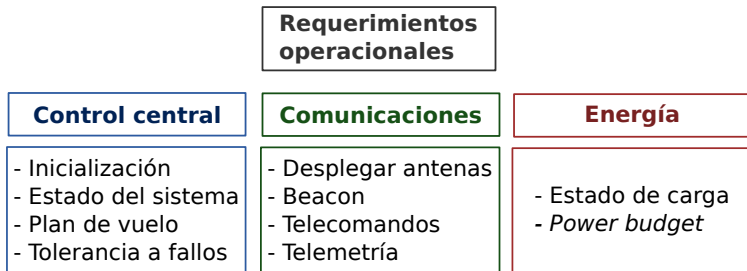
- Capa de abstracción entre la aplicación y el *hardware*
- Funcionan bajo requerimientos de *timing* estrictos.
- Deterministas en la ejecución de tareas.
- Funcionamiento basado en eventos y prioridades.



- Técnica usada en desarrollo de software orientado a objetos.
- Soluciones bien probadas para cierto tipo de problemas.
- Patrón Procesador de Comandos.
 - Separa la solicitud de una acción de su ejecución.
 - Encapsula los requerimientos en comandos.
 - Aplicaciones que soportan una gran cantidad de funciones.

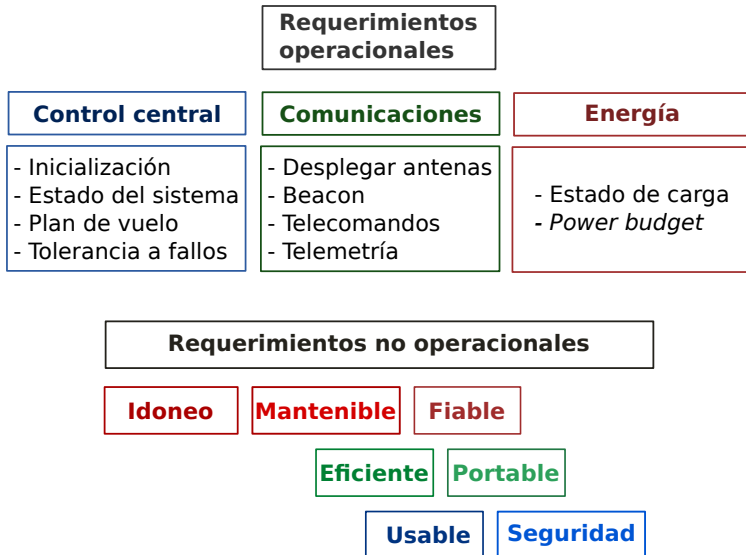


Proceso de diseño



Diseño

Requerimientos operacionales y no operacionales





Aplicación

Sistema Operativo

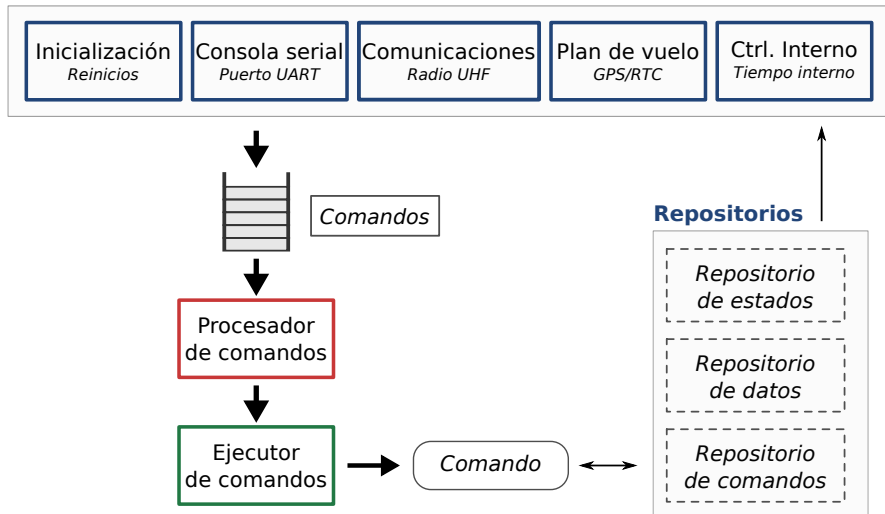
Controladores

MCU

Periféricos

Payloads

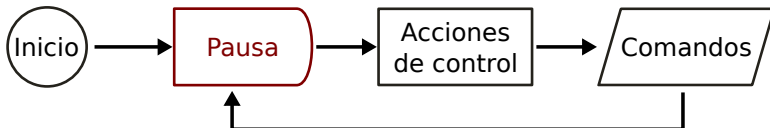
Cientes



Proceso de implementación

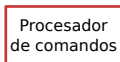
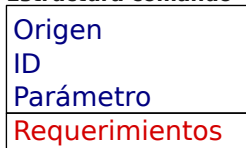
- Implementan la inteligencia del sistema.
- Tareas de FreeRTOS, concurrentes y de baja prioridad.
- Ejecución periódica, *hard-realtime* o *soft-realtime*.

Cientes

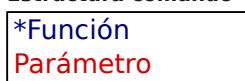


- Representados por estructura de datos.
- Se reconocen por su identificador y metadatos.
- Todo comando es una función.

Estructura comando

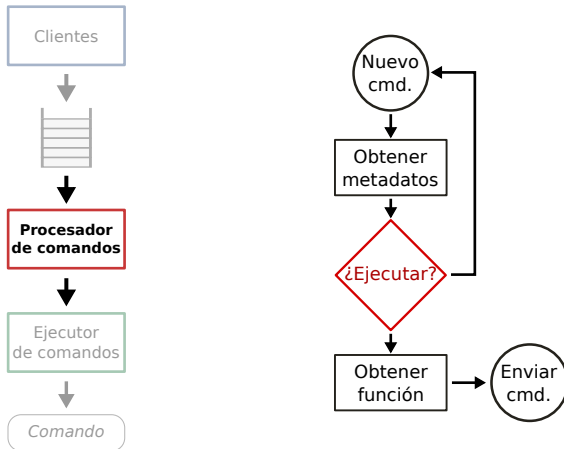


Estructura comando



```
int funcion_comando(void *param)
{
    printf("Ejecutar comando");
    return EXIT_OK;
}
```

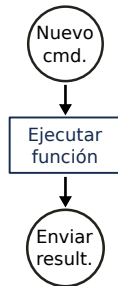
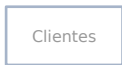

- Operaciones de control sobre comandos.
- Tareas de FreeRTOS, prioridad media.
- Ejecución basada en eventos.



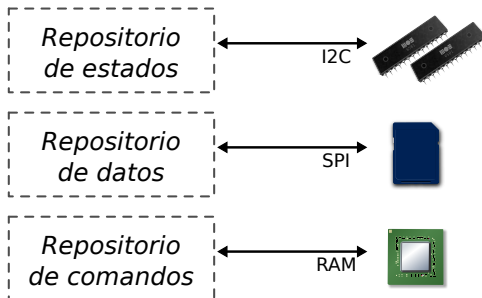
Implementación

Ejecutor de comandos

- Entorno de ejecución para el comando.
- Tareas de FreeRTOS, alta prioridad.
- Ejecución basada en eventos.



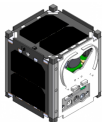
- Librerías para manejo de datos.
- Lectura, escritura y gestión de diferentes tipos de información.
- Implementación requiere *hardware* externo.



Pruebas y resultados

■ *Hardware in the loop simulation*

Sistema embebido



Respuesta del sistema



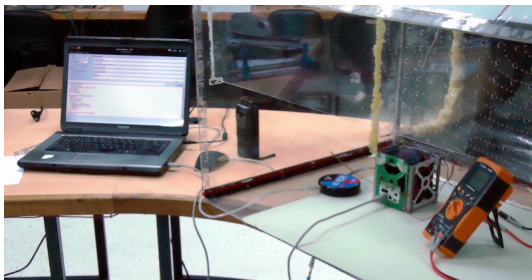
Estimulos simulados



Operación simulada



■ Montaje de la prueba



■ Inicio del sistema

- Correcta inicialización de subsistemas.
- Disponibilidad de consola serial para pruebas.

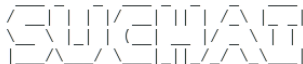
■ Plan de vuelo

- Se ejecuta un plan de vuelo con comandos cada 10 minutos.
- Se verifica la ejecución de los comandos.

■ Variables de estado

- Se registran variable, se generan gráficos para verificar.
- Pruebas de reinicio y verificación de consistencia de variables.

```
>>[Console] Started
```

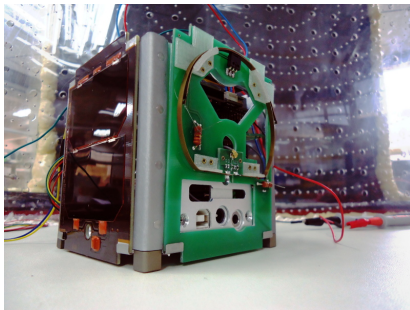


```
>>[Communications] Started  
>>[FlightPlan] Started  
>>[Houskeeping] Started  
>>
```

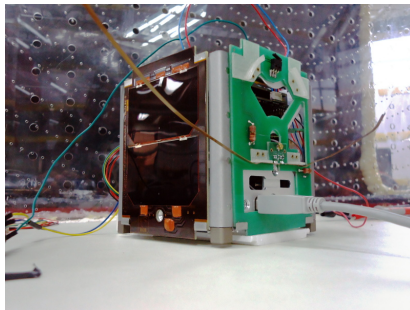
- Estado de carga de baterías
 - Se verifica la lectura de datos desde EPS.
 - Se verifica la carga de la batería.
 - Se verifica la estimación de SOC.
- Presupuesto de energía
 - Implementado en el procesador de comandos.
 - Se simula operación con baja carga en baterías.
- Resultados satisfactorios.

Hora	Comando	SysReq	SOC	Resultado
20:18:33	0x300C	4	4	Ejecutado
20:19:10	0x8000	10	4	Rechazado
20:19:13	0x8000	10	4	Rechazado
20:19:19	0x8002	10	4	Rechazado
20:19:19	0x5000	1	4	Ejecutado
20:19:45	0x8003	10	4	Rechazado
20:20:03	0x8003	10	4	Rechazado

■ Despliegue de antenas.



(e) Previo al despliegue



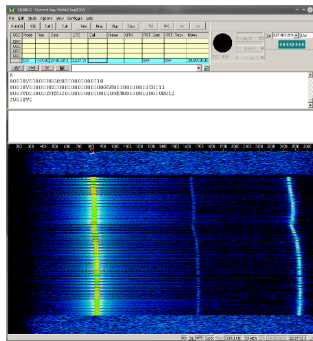
(f) Antenas desplegadas

■ Generación, transmisión y recepción de *beacons*.

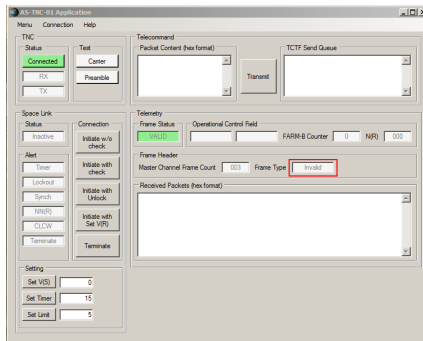
- SUCHAIATINGDOTUCHILEDOTCL-11000017H30761940780001
- 00V00000000XHX020000000000000000GBW00000000DK000024

■ Transmisión y recepción de telemetría y telecomandos.

■ Pruebas insatisfactorias



(g) Recepción beacon



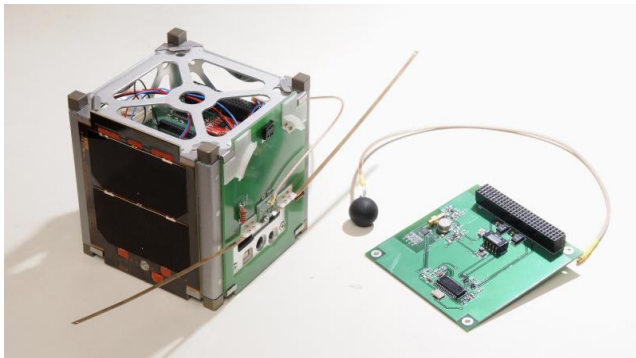
(h) Recepción telemetría

Conclusiones

- Uso generalizado de patrones de diseño, adaptados a lenguaje de programación procedural.
 - Arquitectura de tres capas: divide el problema convenientemente.
 - Procesador de comandos: permite cumplir con requerimientos operacionales y no operacionales.
- FreeRTOS: una solución bien probada y robusta.
- Verificación de requerimientos.
 - Se cumplen los requerimientos operacionales del área de control central, energía y *payloads*
 - No se cumplen los requerimientos del área de comunicaciones debido a fallas en el equipo.
- Desarrollo de un proyecto de ingeniería, como parte del proceso de formación profesional: experiencia y desarrollo de habilidades blandas.

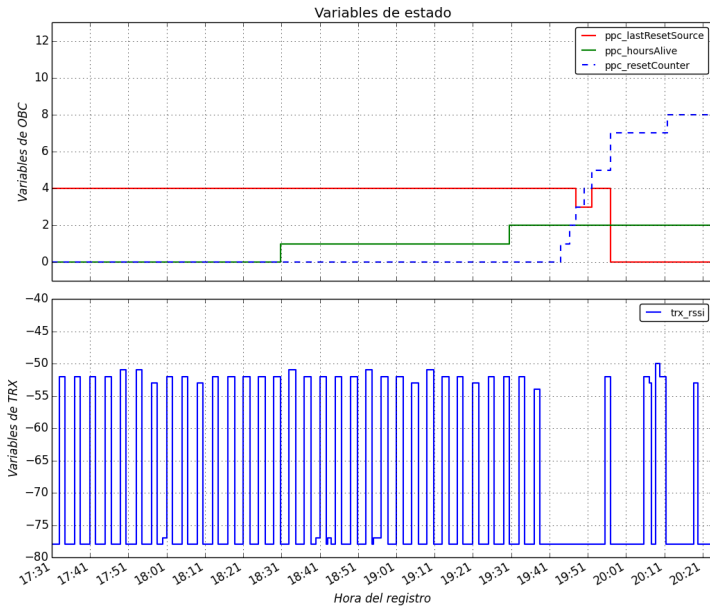
- Mejoras en el área de tolerancia a fallos.
- Agregar múltiples ejecutores de comandos.
- Portar el *software* a diferentes plataformas.
- Integrar y probar un nuevo *transceiver*.

Muchas gracias por su atención
¿Consultas?

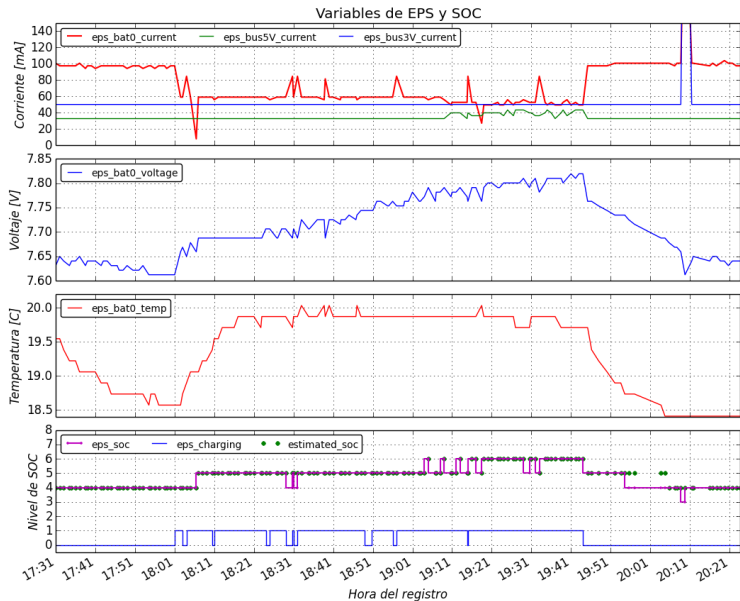


ANEXOS

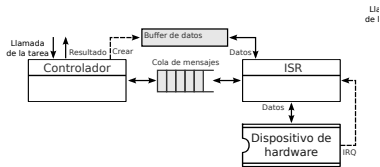
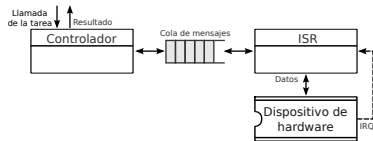
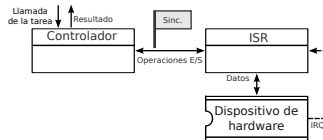
Variables de estado



Variables EPS



Drivers



Calidad de software: ISO-25010

