



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
- Arquitectura aplicación

4 Implementación

- Clientes
- Comandos
- Procesador y ejecutor de comandos

5 Trabajos futuros

6 Consultas

Tabla de contenidos

1 Introducción

2 Marco teórico

- Sistemas embebidos
- Sistemas operativos
- Patrones de diseño

3 Diseño

- Requerimientos operacionales
- Arquitectura de software
- Arquitectura aplicación

4 Implementación

- Clientes
- Comandos
- Procesador y ejecutor de comandos

5 Trabajos futuros

6 Consultas

Tabla de contenidos

1 Introducción

2 Marco teórico

- Sistemas embebidos
- Sistemas operativos
- Patrones de diseño

3 Diseño

- Requerimientos operacionales
- Arquitectura de software
- Arquitectura aplicación

4 Implementación

- Clientes
- Comandos
- Procesador y ejecutor de comandos

5 Trabajos futuros

6 Consultas

Tabla de contenidos

1 Introducción

2 Marco teórico

- Sistemas embebidos
- Sistemas operativos
- Patrones de diseño

3 Diseño

- Requerimientos operacionales
- Arquitectura de software
- Arquitectura aplicación

4 Implementación

- Clientes
- Comandos
- Procesador y ejecutor de comandos

5 Trabajos futuros

6 Consultas

Tabla de contenidos

1 Introducción

2 Marco teórico

- Sistemas embebidos
- Sistemas operativos
- Patrones de diseño

3 Diseño

- Requerimientos operacionales
- Arquitectura de software
- Arquitectura aplicación

4 Implementación

- Clientes
- Comandos
- Procesador y ejecutor de comandos

5 Trabajos futuros

6 Consultas

Tabla de contenidos

1 Introducción

2 Marco teórico

- Sistemas embebidos
- Sistemas operativos
- Patrones de diseño

3 Diseño

- Requerimientos operacionales
- Arquitectura de software
- Arquitectura aplicación

4 Implementación

- Clientes
- Comandos
- Procesador y ejecutor de comandos

5 Trabajos futuros

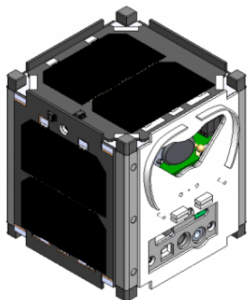
6 Consultas

Introducción

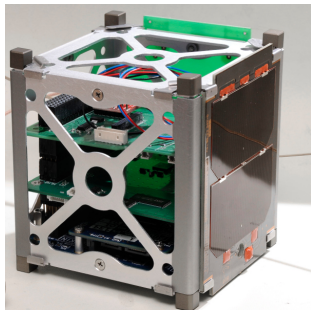
Proyecto SUCHAI

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

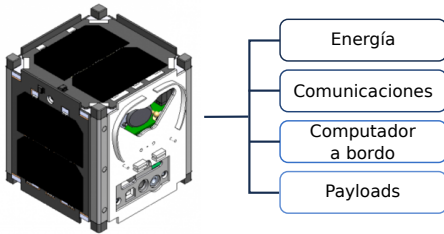
Es el primer proyecto satelital desarrollado por estudiantes en el país.



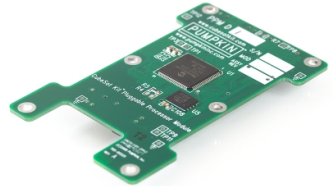
(a) Estandar Cubesat



(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

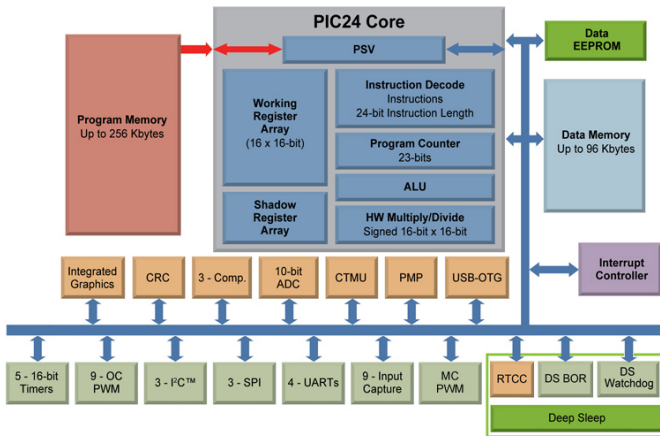
Objetivos generales

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

Definir
Requerimientos

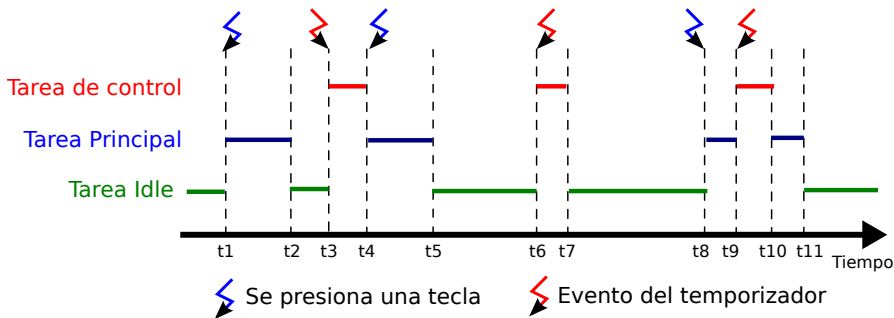


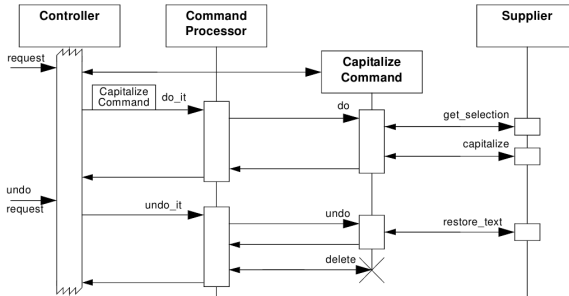
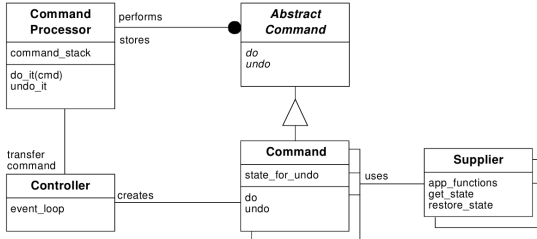
Sistemas computacionales diseñados para cumplir funciones específicas, en aplicaciones de tiempo real. Integran en un mismo chip un microcontrolador y una serie de periféricos.



Los sistemas operativos de tiempo real (RTOS) se caracterizan por:

- Ser una capa de abstracción entre la aplicación y el *hardware*
- Funcionar bajo requerimientos de *timing* estrictos.
- Ser deterministas en la ejecución de tareas.
- Funcionamiento basado en eventos y prioridades.





Área de control central

- Inicialización.
- Estado del sistema.
- Plan de vuelo.
- Tolerancia a fallos.

Área de comunicaciones

- Configuración del TRX.
- Despliegue de antenas.
- Generar y enviar *beacon*.
- Telecomandos y telemetría.

Area de energía

- Estimar nivel de carga de baterías
- Operar según un presupuesto de energía.

Area de *payloads*

- Integrar diferentes *payloads*
- Ejecutar acciones de *payloads*



Aplicación

Sistema Operativo

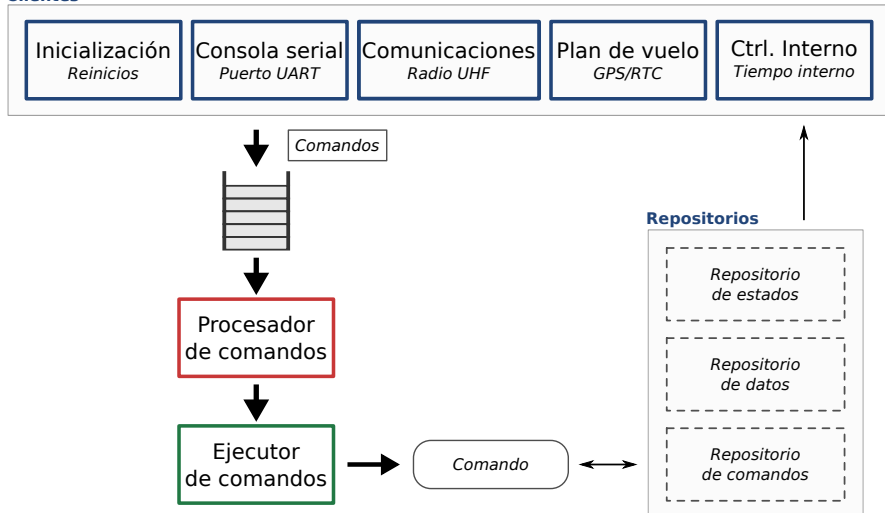
Controladores

MCU

Periféricos

Payloads

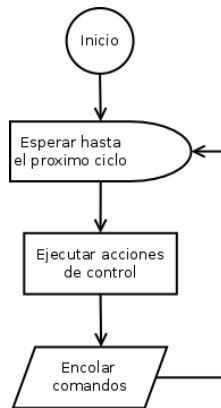
Clientes



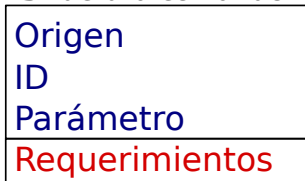
Cientes



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



Estructura comando



Procesador
de comandos



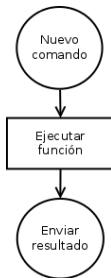
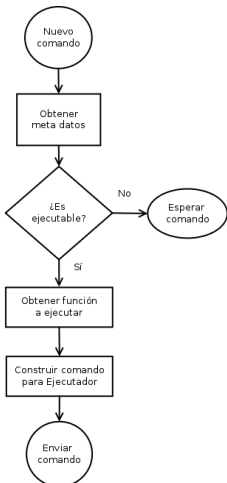
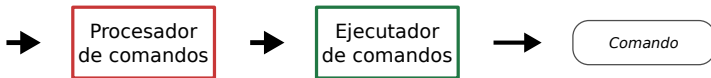
Estructura comando



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

Implementación

Procesador y ejecutor de comandos



- 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?

