



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE DEPARTAMENTO DE INGENIERÍA
ELÉCTRICA

DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE DE CONTROL PARA EL
COMPUTADOR A BORDO DE UN PICOSATÉLITE

TESIS PARA OPTAR AL TÍTULO DE TÍTULO DE INGENIERO CIVIL
ELECTRICISTA

CARLOS EDUARDO GONZÁLEZ CORTÉS

PROFESOR GUÍA:
MARCOS DÍAZ QUEZADA

MIEMBROS DE LA COMISIÓN:
JORGE LÓPEZ HIDALGO

SANTIAGO DE CHILE
OCTUBRE 2012

Índice general

1. Introducción	1
2. Contextualización	2
2.1. Fundamentación y objetivos generales	2
2.2. Bibliografía y Estado del Arte	3
2.2.1. Satélites	3
2.2.2. Satélites tipo Cubesat	3
2.2.3. Microcontroladores	3
2.2.4. Sistemas operativos	3
2.2.5. Arquitectura de Software	3
2.3. Objetivos específicos	3
2.4. Antecedentes generales	4
2.5. Antecedentes específicos	4
2.6. Hipótesis de trabajo y metodología	4
2.7. Plan de trabajo	4
2.7.1. Carta Gantt	8
2.8. Infraestructura disponible	8
2.8.1. Instalaciones	8
2.8.2. Hardware	8
2.8.3. Software	9
2.9. Condiciones contractuales	9
3. Implementación	10
3.1. Arquitectura de software	10
3.1.1. Arquitectura Global	10
3.1.2. Controladores de hardware	10
3.1.3. Sistema operativo	10
3.1.4. Aplicación	10
4. Pruebas y resultados	12
4.1. Pruebas modulares	12
4.1.1. Pruebas a Console	12
4.1.2. Pruebas a Hauskeeping	12
4.1.3. Pruebas a Dispatcher	12
4.1.4. Pruebas a Executer	12
4.2. Pruebas de arquitectura	12

4.3. Pruebas de integración básica	12
5. Conclusiones	14
5.1. Conclusiones generales	14
5.2. Conclusiones específicas	14
5.3. Trabajo futuro	14

Capítulo 1

Introducción

Este capítulo considera la introducción al trabajo realizado en esta memoria. Considera los siguientes tópicos:

Capítulo 2

Contextualización

2.1. Fundamentación y objetivos generales

Esta memoria se enmarca en el desarrollo del proyecto SUCHAI que consiste en la implementación, lanzamiento y operación de un pico-satélite Cubesat, siendo esta la primera aproximación en esta materia para la universidad y el país. Uno de los componentes fundamentales de un satélite es su computador abordo, sistema encargado de dar inteligencia y operatividad al satélite durante todo su tiempo de vida útil en el espacio. En el caso de un pico-satélite se tiene el desafío de dotar de todas la funcionalidades estándar de un satélite en un sistema computacional de recursos extremadamente limitados, estamos hablando de sistemas embebidos que utilizan microcontroladores de baja potencia y capacidad de cómputo como microcontroladores PIC24 o PIC18.

El objetivo de este trabajo es el diseño, desarrollo e implementación del *software* que gobierna el computador a bordo del satélite. Se requiere diseñar una arquitectura de *software* que abarque desde controladores de hardware hasta la aplicación final para el control de satélite. Esta arquitectura debe cumplir con requerimientos de calidad de *software* como modularidad, expansibilidad y facilidad de mantenimiento estando adaptada en específico a sistemas embebidos que emplean microcontroladores de gama media.

La implementación se llevará a cabo en específico para el satélite SUCHAI y busca proveer la funcionalidad básica de este sistema que incluye la interacción de un computador a bordo, un sistema de control de energía y un sistema de comunicaciones. De esta manera el *software* de control se cuenta como un recurso más que será considerado y adaptado a la necesidades específicas del proyecto en la etapa de integración general de sistemas del satélite.

2.2. Bibliografía y Estado del Arte

2.2.1. Satélites

Pico-satélites

2.2.2. Satélites tipo Cubesat

Estándar

Aplicaciones

2.2.3. Microcontroladores

Microcontroladores PIC

Descripción

Arquitectura

Aplicaciones

2.2.4. Sistemas operativos

FreeRTOS

2.2.5. Arquitectura de Software

Patrones de diseño

Command Pattern

2.3. Objetivos específicos

Los objetivos específicos del proyecto se enumeran a continuación

- Diseñar una arquitectura de *software* para el sistema de control del satélite
- Implementar controladores de hardware para el microcontrolador
- Implementar controladores de periféricos principales (Transceiver, EPS y RTC)
- Integrar un sistema operativo de tiempo real multitarea como sistema embebido
- Implementar el flujo principal de la arquitectura del *software* de control del satélite
- Integrar sistema de comunicaciones al *software* de control
- Integrar sistema de energía al *software* de control
- Pruebas del sistema integrado

El listado de objetivos presenta un orden temporal en la ejecución de estas tareas puesto que objetivo es dependiente del cumplimiento de los objetivos anteriores. El trabajo se puede considerar terminado cuando se ha probado la implementación e integración del *software* con los módulos de comunicaciones y energía obteniendo un sistema satelital con las mínimas funcionalidades.

2.4. Antecedentes generales

Carrera satelital chilena

2.5. Antecedentes específicos

SUCHAI proposal

Investigaciones asociadas

Lagmuir Probe

2.6. Hipótesis de trabajo y metodología

2.7. Plan de trabajo

1. Documentación
 - (a) Programación en C
 - (b) Programación de Microcontroladores PIC24
 - (c) Sistemas operativos de tiempo real
 - (d) Sistema Operativo FreeRTOS

- (e) Gestión del equipo de trabajo
 - i. Carta Gantt área electrónica
 - ii. Planilla de registro de actividades del equipo
- 2. Controladores de hardware. Funcionamiento y pruebas independientes
 - (a) Organización y utilización de repositorio de software SVN
 - (b) Recepción del Cubesat kit. Pruebas de funcionamiento básicas
 - (c) Programación de drivers para microcontrolador PIC24F y periféricos
 - i. ADC (TOo)
 - ii. I2C
 - iii. UART
 - iv. SPI (TOo)
 - v. PWM (TOo)
 - vi. RTCC (FRa)
 - (d) Drivers para el manejo estándar de la MB y el PPM del Cubesat kit
 - i.
 - ii. RTC M41T81S
 - iii. Memoria Flash AT25DF641 (DVa)
 - (e) Drivers para Transceiver
 - i. Pruebas con software ALLSPACE
 - A. Configuración
 - B. Beacon
 - C. Telemetría
 - D. Telecomandos
 - ii. Programación de drivers
 - A. Configuración
 - B. Beacon
 - C. Telemetría
 - D. Telecomandos
 - iii. Pruebas utilizando drivers
 - A. Configuración
 - B. Beacon
 - C. Telemetría
 - D. Telecomandos
 - (f) Drivers de EPS (FRs)
 - i. Pruebas EPS
 - ii. Programación de drivers EPS
 - iii. Pruebas de EPS con drivers
- 3. Sistema Operativo

- (a) Incorporación de un SO de tiempo real preemptive que permite la ejecución de tareas simultáneamente y sincronización
 - (b) Pruebas con Salvo (Insatisfactorias)
 - (c) Pruebas con FreeRTOS (Sistema seleccionado)
4. Aplicaciones (Software SUCHAI)
- (a) Evaluación y diseño de arquitectura de software
 - i. Arquitectura basada en módulos por cada sistema (Desechada)
 - ii. Reunión con colaboradores del DCC
 - iii. Arquitectura basada en patrón de diseño ?Command Pattern?
 - (b) Implementación de sistema de inicialización del satélite (TOo)
 - i. Tarea de inicialización del sistema (TaskDeployment).
 - ii. Comprobación de inactividad en los 30min posteriores al lanzamiento
 - iii. Rutina para el despliegue de antenas
 - iv. Inicializaciones del software SUCHAI
 - v. Creación de Listeners
 - vi. Término de la tarea taskDeployment
 - (c) Implementación de la arquitectura de software basada en comandos.
 - i. Implementar Listeners (Varias tareas simultaneas)
 - A. Console, interprete de consola serial
 - B. Communications, recepción de TC y envío de TM
 - C. FlightPlan, itinerario de activiades
 - D. HouseKeeping, control rutinario interno del sistema
 - E. Generación de comandos
 - F. Inteligencias basada en repositorio de datos o algún dispositivo específico (TRX, EPS o Payloads).
 - ii. Implementar Dispatcher
 - A. Recibe en una cola los comandos provenientes desde diferentes listeners
 - B. Decide la ejecución de cada comando
 - Según estado de energía
 - C. Envía comando a Executer para su final ejecución
 - D. Recibe estatus de ejecución del comando
 - iii. Implementar Executer
 - A. Solo un executer a la vez
 - B. Cada comando posee todo el procesador y recursos
 - C. Máximo 2min de ejecución (WDT)
 - iv. Implementar Repositorio de datos (TOo)
 - A. Actualizado constantemente desde el Listener Housekeeping usando los comandos cmdDRP
 - B. Contiene variables de estado

mPPC_state variables del PIC, PPM y MB mDEP_state variables del
Despliegue de antenas mRTC_state variables con la hora actual mEPS_state
variables de la EPS mTRX_state variables del TRX

C. Contiene el plan de vuelo

D. Necesidad de sincronización

v. Implementar Repositorio de comandos

A. Generar sintaxis estándar de comandos

Estructura de comandos para dispatcher Estructura de comandos para
executer Sintaxis estándar de comandos

B. Almacena la lista de comandos disponibles

Arreglos con comandos para cada área

C. Documentación y control de comandos disponibles (TOo)

Identificador único que incluye área y código Planilla online para referencia
y control

D. Implementación de comandos para cada módulo necesario

cmdConsole, consola serial. cmdPPC, microprocesador. cmdDRP, actu-
alizar repositorio de datos. cmdTRX, funcionalidades de comunicaciones
cmdEPS, funcionalidades de energia cmdRTC, funcionalidades para RTC
cmdPAY, funcionalidades específicas a un payload

5. Integración de Hardware básico

(a) Integración de TRX

i. Conexión con cable coaxial del TRX con la antena

ii. Recepción de telemetría en estación terrena

iii. Envío de telecomandos desde estación terrena

(b) Integración de EPS

i. Medición de consumo energético (si aplica)

ii. Comprobación de que los killswitch (BATT POS) cortan la alimentación de
la EPS hacia el bus 5V_sys Vcc_sys y V_batt

iii. Comprobación de alimentación de buses de energia

iv. Comprobación de carga de baterías

v. Comprobar sistema de estimación de carga

6. Pruebas generales

(a) Pruebas de funcionamiento prolongado

(b) Pruebas de funcionamiento intenso

(c) Pruebas de control de fallos (simulados)

(d) Pruebas de telecomandos

(e) Pruebas de telemetría

2.7.1. Carta Gantt

2.8. Infraestructura disponible

2.8.1. Instalaciones

El proyecto se desarrolla en las dependencias de la facultad, específicamente el laboratorio SPEL (*Spacial and Planetary Exploration Laboratory*), tercer piso del edificio de Electrotecnologías. Este laboratorio alberga proyectos relacionados el desarrollo de radiosondas para monitoreo de condiciones climáticas, radares y proyectos aeroespaciales como el SUCHAI.

2.8.2. Hardware

Cubesat Kit Parte fundamental del hardware disponible corresponde a un Cubesat Kit adquirido a la compañía Pumpkins. El kit incluye la estructura de un Cubesat de una unidad con un microcontrolador PIC24F256GA110; una placa de desarrollo para el mismo sistema; y herramientas de programación y *debug* para el microcontrolador.



Figura 2.1: Cubesat Kit

Transceiver Se cuenta con un sistema de comunicaciones que consta de un *transceiver* satelital FSK para la banda de 430MHz de la empresa ALLSPACE adaptado al estándar Cubesat. Es el sistema encargado de recibir los telecomandos desde tierra y transmitir hacia la estación terrena la telemetría generada por el satélite.

Se cuenta además con una estación terrena dotada con un sistema de radio en la banda de 430MHz, incluyendo antena y control de posición de la misma; un TNC para digitalizar la señal de radio; y un servidor para controlar los equipos de forma remota.

EPS Se adquiere un sistema de control de energía para el satélite que incluye baterías de litio, paneles solares y un sistema de control para medir variables importantes de energía y ajustar su funcionamiento.

Equipos de medición El laboratorio cuenta con un osciloscopios; analizadores de espectro; generadores de señales; cajas limpias, para pruebas que requieren entornos descontaminados; entre otros.

2.8.3. Software

La mayor parte del software utilizado por el proyecto es de código libre o gratuito. Se desataca la amplia utilización de sistemas Linux para el desarrollo de este trabajo.

Entorno de desarrollo

Para el desarrollo de la aplicación para el microcontrolador PIC24F se utiliza el entorno de desarrollo MPLABX de Microchip, que es gratuita; se utiliza el compilador MPLAB C30, en versión lite, gratuito; se tiene un sistema de control de versiones mediante SVN.

Sistema operativo

Para dotar de funcionalidades avanzadas de multitarea al microcontrolador se ha utilizado el sistema operativo de tiempo real FreeRTOS de código libre y gratuito; adaptado especialmente a microcontroladores de gama media como el utilizado en el proyecto.

2.9. Condiciones contractuales

El desarrollo de este trabajo se enmarca dentro del proyecto SUCHAI el cual es patrocinado y financiado por la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile.

Parte del trabajo realizado para esta memoria ha sido considerado en el desarrollo de los cursos Seminario de Diseño e Innovación Tecnológica I, II y III (ME4030, EL5030, EL6030).

Capítulo 3

Implementación

En este capítulo se describe el proceso de diseño e implementación del software de control para el satélite. Considera el diseño de una arquitectura de software a nivel global y sobre todo el detalle de la aplicación final. La aplicación debe considerar requisitos de diseño tales como: ser modular; ser compacta en su ejecución; ser extensible.

3.1. Arquitectura de software

3.1.1. Arquitectura Global

3.1.2. Controladores de hardware

3.1.3. Sistema operativo

3.1.4. Aplicación

A nivel de aplicación se considera la siguiente estructura de software

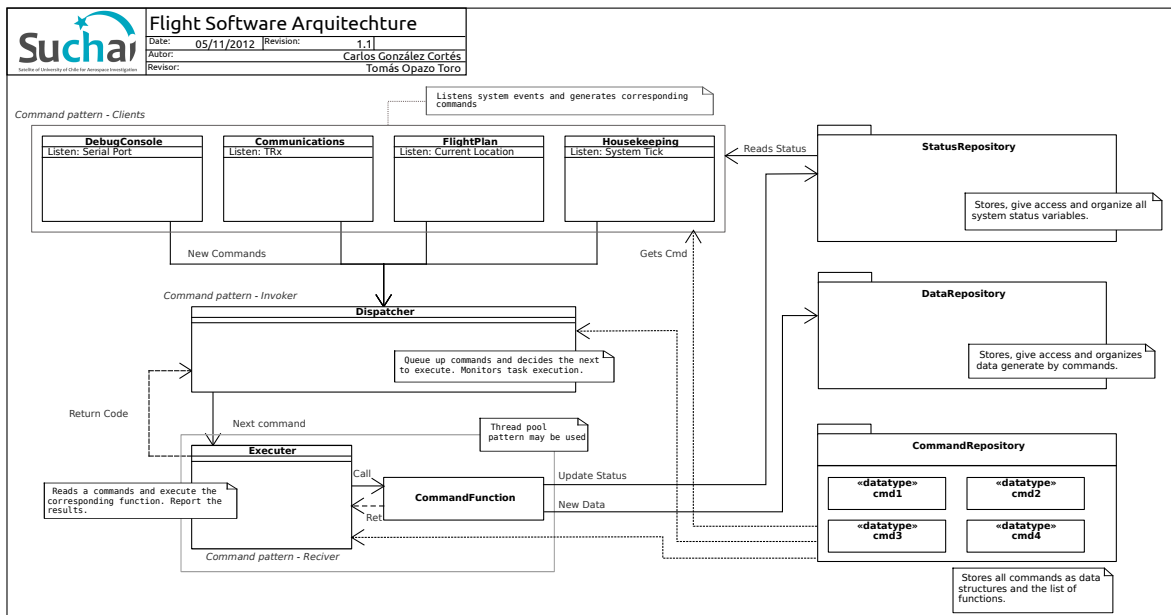


Figura 3.1: Arquitectura de software

Capítulo 4

Pruebas y resultados

4.1. Pruebas modulares

4.1.1. Pruebas a Console

4.1.2. Pruebas a Hauskeeping

4.1.3. Pruebas a Dispatcher

4.1.4. Pruebas a Executer

4.2. Pruebas de arquitectura

Se prueba el funcionamiento del flujo principal de la arquitectura de comandos

4.3. Pruebas de integración básica



Figura 4.1: Pruebas de integración

Capítulo 5

Conclusiones

5.1. Conclusiones generales

5.2. Conclusiones específicas

5.3. Trabajo futuro

Bibliografía