

Lista de tareas pendientes

■ todo marco teorico 1	2
■ todo marco teorico 2	2



UNIVERSIDAD DE CHILE

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

DEPARTAMENTO DE INGENIERÍA CIVIL ELÉCTRICA

DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA PARA EXPERIMENTOS EN
MICROGRAVEDAD Y DE ELECTRÓNICA EN AMBIENTE HOSTIL CON
NANO-SATÉLITES

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

JOSÉ ALBERTO OGALDE ORTIZ

PROFESOR GUÍA:

MARCOS DÍAZ QUEZADA

MIEMBROS DE LA COMISIÓN:

CLAUDIO FALCÓN BEAS

TOMÁS OPAZO TORO

SANTIAGO DE CHILE
SEPTIEMBRE 2016

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: JOSÉ ALBERTO OGALDE ORTIZ
FECHA: SEPTIEMBRE 2016
PROF. GUÍA: SR. MARCOS DÍAZ QUEZADA

DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA PARA EXPERIMENTOS EN MICROGRAVEDAD Y DE ELECTRÓNICA EN AMBIENTE HOSTIL CON NANO-SATÉLITES

Típicamente un satélite es diseñado de manera completa entorno a sus payloads a bordo. Este diseño *payload-céntrico* tiene algunas desventajas, como lo son grandes *budget* y un ciclo de desarrollo más largo. Una forma de resolver este problema es establecer una interfaz estandarizada entre el satélite y los payloads.

Una interface como tal permite separar las etapas de diseño y operación entre el bus del satélite con el de los payloads. Esto puede facilitar la integración de un mismo payload en múltiples satélites, así como reducir el ciclo de desarrollo total de la misión.

Para evitar estos problemas se plantea una plataforma estandarizada entre payloads y el bus de un satélite, la cual está diseñada para operar en los nano-satélites tipo CubeSat que son los más utilizados en la comunidad académica.

SUCHPAY es una plataforma que implementa tal interfaz entre el bus y un payload a bordo. La plataforma provee herramientas de software para el desarrollador de payload con el fin de facilitar la integración del payload hacia el bus del satélite. También se provee una API de comandos para el bus, con la cual se estandariza el tipo de comunicación entre un payload y el bus del satélite.

A modo de validación de la plataforma, se integran dos payload experimentales que consisten en el estudio de la potencia inyectada en un circuito RC y el estudio del coeficiente de disipación de energía en microgravedad y ambientes hostiles. Ambos payload se integran en el bus de SUCHAI, un CubeSat de 1U desarrollado por la Universidad de Chile.

Gracias al pulento por su incondicional sabiduría en todo momento

Agradecimientos

Muchas gracias al spaghetti volador por su infinita sabiduría.

JOSÉ ALBERTO OGALDE ORTIZ

Índice general

Agradecimientos	III
1. Introducción	1
1.1. Objetivos Generales	1
1.2. Objetivos Específicos	1
1.3. Estructura	1
2. Marco Teórico	2
2.1. Software Architecture	2
2.1.1. Component	2
2.1.2. Sistemas embebidos	2
2.1.3. Microcontrolador	3
2.1.4. Nano-Satélites	4
2.1.5. Bus	5
2.1.6. Sensores transductores	5
2.1.7. Actuador	5
2.1.8. Controlador de hardware	6
2.1.9. Microgravedad	6
2.1.10. Sistema fuera de equilibrio	6
2.1.11. Circuito pasivo	6
3. Diseño de la plataforma	8
3.1. Requerimientos	8
3.1.1. Requerimientos Operacionales	8
3.1.2. Requerimientos No-Operacionales	8
3.1.3. Requerimientos mínimos	8
3.2. Hardware	8
3.3. Software	8
3.3.1. Arquitectura del Software	8

4. Implementación	9
Capítulo Adicional que no es apéndice	10
Apéndices	11
A . Tweets	11
Referencias	12

Índice de tablas

2.1. Resumen de familias de Microchip	3
---	---

Índice de figuras

2.1. Circuito RC pasa-bajos	7
---------------------------------------	---

Capítulo 1

Introducción

La etapa de diseño consiste en explicitar los requerimientos del sistema y generar una arquitectura de software que representa de manera conceptual la solución a implementar, en esta etapa se hace un uso extensivo del concepto de patrones de diseño para conseguir la solución al problema planteado. Durante la implementación se toma la solución conceptual y se programa el software utilizando las herramientas adecuadas a la plataforma consiguiendo una plataforma base que consta de controladores de hardware, sistema operativo y la aplicación específica a la misión. Las pruebas se desarrollan sobre el sistema integrado en sus tres subsistemas básicos mediante la puesta en funcionamiento del satélite por un tiempo prolongado usando la técnica denominada hardware on the loop simulation adecuada para probar sistemas embebidos complejos.

1.1. Objetivos Generales

El principal objetivo de este trabajo es diseñar e implementar una plataforma abierta que sirva como interfaz entre un satelites y sus payloads. En este contexto, abierta significa que los usuarios tendrán acceso completo a las especificaciones de diseño y que pueden construir, modificar y extender los diseños a su propio gusto. El hecho de definir una interfaz estándar entre el bus y los payloads, permite acelerar los tiempos de desarrollo para múltiples satélites así como también focalizar el desarrollo de payloads.

1.2. Objetivos Específicos

1.3. Estructura

Capítulo 2

Marco Teórico

todo marco teórico 1

Ver cuales son los documentos que hay eliminar/agregar

todo marco teórico 2

Ver cuales son los documentos que hay que editar.

2.1. Software Architecture

A software architecture is a description of the subsystems and components of a software system and the relationships between them. Subsystems and components are typically specified in different views to show the relevant functional and non-functional properties of a software system. The software architecture of a system is an artifact. It is the result of the software design activity.

2.1.1. Component

A component is a n encapsulated part of a software system. A component has a n interface. Components serve as the building blocks for the structure of a system. At a programming-language level, components may be represented as modules, classes, objects or a set of related functions.

2.1.2. Relationship

A relationship denotes a connection between components. A relationship may be static or dynamic. Static relationships show directly in source code. They deal with the placement of components within a n architecture. Dynamic relationships deal with temporal connections and dynamic Interaction between components. They may not be easily visible from the static structure of source code.

2.1.3. Sistemas embebidos

Los computadores personales son utilizados por gran parte de la población dado que se caracterizan por tener capacidad para atender una cantidad importante de procesos dispuestos por un

sistema operativo en orden para atender diferentes aplicaciones del usuario final. Un microcontrolador también es un tipo de computador salvo que sus recursos normalmente son más limitados que los de un computador personal. Además, un microcontrolador normalmente tiene más flexibilidad para el uso directo de periféricos y poder comunicarse a nivel de máquina con otro dispositivo electrónico.

Un sistema embebido está compuesto por uno más microcontroladores en conjunto con dispositivos electrónicos con los cuales comunicarse para realizar alguna tarea. El hardware que compone a un microcontrolador está fijo, por lo que el desarrollador de sistemas embebidos solamente tiene acceso a la programación de éste en conjunto con la correcta interconexión con otros dispositivos electrónicos. Debe considerarse que a diferencia de un computador personal, el programa que provee la funcionalidad del microcontrolador se denomina *firmware*, ya que en general es específico para la plataforma de hardware utilizada.

Microcontroladores PIC

Este trabajo se concentra en el uso de los microcontroladores PIC desarrollados por la compañía Microchip. La familia de microcontroladores PIC es bastante amplia adaptándose a un amplio rango de necesidades, la tabla 2.1 resume las principales características de los diferentes modelos y puede ser utilizada como una guía para determinar el dispositivo adecuado según la aplicación:

Tabla 2.1: Resumen de familias de Microchip

Familia	Datos	Memoria Programa	Memoria RAM	Velocidad
PIC10	8 bit	512 Words	64 Bytes	16MHz
PIC12	8 bit	4 Kwords	256 Bytes	32MHz
PIC16	8 bit	16 Kwords	2 Kbytes	48MHz
PIC18	8 bit	64 Kwords	4 Kbytes	64MHz
PIC24	16 bit	512 Kbytes	96 Kbytes	70 MIPS
dsPIC	16 bit	512 Kbytes	54 Kbytes	70 MIPS
PIC32	32 bit	512 Kbytes	128 Kbytes	80 MHz

2.1.4. Microcontrolador

Un microcontrolador es un computador pequeño dentro de un solo circuito integrado, que posee un solo núcleo de proceso, memoria y periféricos de entrada/salida. Los microcontroladores están diseñados para los sistemas embebidos, en contraste a los microprocesadores utilizados en computadores personales u otras aplicaciones de propósito general.

Los microcontroladores son utilizados en dispositivos y productos automáticamente controlados como por ejemplo, controles remotos, sistema de control de automóviles, etc.

2.1.5. Nano-Satélites

Corresponden a los satélites artificiales con una masa entre 1 y 10 kg. Con los continuos avances en la minaturización y capacidad de la electrónica en conjunto al uso de constelaciones de satélites, los nano-satélites están crecientemente capaces de desempeñarse bien en misiones comerciales. Un caso destacable son los nano-satélites que siguen el protocolo CubeSat, cuyo primer lanzamiento fue el 2003 y en el año 2012 se lanzaron 75 satélites al espacio.

CubeSat

Es un tipo de nano-satélite hecho de múltiples unidades cúbicas de 10x10x10 cm, no tiene más de 1.3 kg de masa por unidad y a menudo utiliza componentes *commercial off-the-shelf* (COTS) para su diseño electrónico y estructura.

Los usos típicos involucran experimentos que pueden ser miniaturizados o experimentos para servir a propósitos como la observación del planeta o comunicación de radio amateur.

Uno de los atractivos de los CubeSats es su bajo costo en comparación a otras tecnologías satelitales. Esto lo convierte en buen candidato para probar nuevas tecnologías a bordo de estas plataformas dado que su costo puede justificar misiones riesgosas.

Proyecto SUCHAI

Satellite of the University of CHile for Aerospace Investigation (SUCHAI) corresponde al primer satélite artificial diseñado y desarrollado localmente en Chile por un conjunto de estudiantes de pregrado y profesores del Departamento de Ingeniería Eléctrica de la Universidad de Chile. SUCHAI es un nano-satélite de tipo CubeSat de una unidad (aproximadamente 1 kg), cuyo objetivo principal es educacional y científico. La misión de SUCHAI se considera en los siguientes *payloads*

- **Langmuir probe:** Estudio de la ionósfera en sincronización con el *incoherent scatter radar* (ISR) tomando medidas simultáneas de la variación en la densidad de electrones en la ionósfera.

- **Eletrónica fuera del equilibrio:** Un circuito RC para estudiar las fluctuaciones fuera de equilibrio en la potencia inyectada al estar en un ambiente hostil como el espacio.
- **Experimento termal:** Estudia la disipación de calor en dispositivos electrónicos en un ambiente vacío.
- **Cámara:** Una cámara digital para analizar la factibilidad de observar la tierra desde un cubesat.
- **Receptor GPS:** Un módulo receptor GPS para obtener la posición del satélite y aprender cómo utilizarlo en futuras misiones.

Payload

Un *payload* es la combinación de hardware y software de una nave espacial que interactúa con un *sujeto* (porción del mundo exterior fuera de la nave espacial), de forma de completar los objetivos de la Misión. Los payloads típicamente son únicos y específicos para cada misión, dado que son la razón fundamental por la que la nave espacial es lanzada. El propósito del resto de la nave espacial es mantener al payload sano, feliz y apuntando en la dirección correcta.

2.1.6. Bus

Es un dispositivo que actúa como sistema de comunicación entre los componentes de un sistema. En este caso, un bus interconecta todos los módulos de nano-satélite integrando las partes individuales en un sistema completo.

PC-104

Es una familia de estándares de sistemas embebidos que define los factores y forma de un bus. El PC-104 está pensado para ambientes especializados donde se necesita un computador pequeño y robusto. El estándar permite diseño modulares y además, posibilita poner placas una de encima de otra en forma de pila.

2.1.7. Sensores transductores

Es un dispositivo que convierte un tipo de energía hacia otra. Usualmente un transductor convierte una señal de un tipo hacia una señal eléctrica. Los transductores son fundamentales en los sistemas de automatización, de medidas y de control, donde se generan señales eléctricas a partir de otras variables físicas (e.g. energía, fuerza, torque, movimiento, posición, etc). Según su uso de potencia, existen transductores activos y pasivos.

2.1.8. Actuador

Es un dispositivo que es responsable de accionar algún tipo de mecanismo sobre un sistema. El tipo de energía utilizado por el actuador es variable (e.g. eléctrico, mecánico, químico, etc).

Conversor ADC

Un conversor Análogo-a-Digital es un dispositivo que convierte una señal analógica hacia un número digital representado por una señal digital.

Conversor DAC

Un conversor Digital-a-Análogo es un dispositivo que convierte un número digital hacia una señal analógica.

2.1.9. Controlador de hardware

Corresponde a un programa/software que manipula o controla un dispositivo de hardware específico. Un driver provee una interfaz de software de manera que el sistema operativo u otro programa pueda acceder a las capacidades del dispositivo de hardware sin necesidad de conocer detalles muy precisos sobre éste.

2.1.10. Microgravedad

Corresponde a la situación física donde la *fuerza-g* dada por la ley de gravitación son muy pequeñas, prácticamente nulas. También se le conoce como *zero-G* o μg . Esta condición se produce por ubicarse a gran distancia de la Tierra o bien encontrarse orbitando la Tierra en caída libre.

2.1.11. Sistema fuera de equilibrio

La termodinámica fuera del equilibrio es una rama de la física que trata con sistemas y procesos que en general no están en estado de equilibrio termodinámico, pero pueden ser adecuadamente descrito en términos de variables que son relacionadas con variables de estados termodinámicas. Un sistema fuera de equilibrio es un sistema que cumple con estos requisitos. Sin embargo, es de particular interés para este trabajo de título, los sistemas que se encuentran en estado estacionario fuera del equilibrio o NESS (NonEquilibrium Steady State).

2.1.12. Circuito pasivo

Corresponde a un circuito eléctrico compuesto por elementos discretos que en conjunto consumen energía en vez generarla.

Circuito RC

Corresponde a un circuito compuesto por un resistor (R) junto a un capacitor (C). En el desarrollo de este trabajo se refiere a un circuito RC con el resistor y capacitor conectados en serie de forma que forman un filtro pasa bajos.

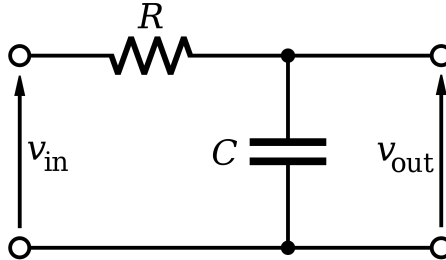


Figura 2.1: Circuito RC pasa-bajos

Filtro pasa bajos

Corresponde a un dispositivo cuya característica es bloquear el paso de señales de *alta* frecuencia a la salida del dispositivo, por lo que solamente permite el paso de señales de *baja* frecuencia. Si la señal es una combinación de múltiples señales de diferentes componentes de frecuencias, solamente las componentes de baja frecuencia logran pasar a través del dispositivo.

Potencia inyectada

Corresponde a la potencia instantánea consumida por un circuito eléctrico.

Capítulo 3

Diseño de la plataforma

3.1. Requerimientos

3.1.1. Requerimientos Operacionales

Se refieren a las funcionalidades que se espera que la plataforma a bordo del satélite debe realizar. Son los requisitos básicos que el sistema debe cumplir para considerar que se cuenta con una plataforma capaz de llevar a cabo la ejecución y representación del payload. La lista de requerimientos proviene de una serie de reuniones sostenidas con los integrantes de los diferentes grupos de trabajo, todo de acuerdo con los lineamientos del jefe de proyecto.

3.1.2. Requerimientos No-Operacionales

Por requerimientos no-operacionales, se entienden aquellos atributos asociados a la calidad de la plataforma, o bien criterios que permiten determinar cómo debería ser la interfaz que se está diseñando. A diferencia de los requerimientos operacionales que explican lo que la plataforma debería hacer, los no funcionales explican las cualidades y restricciones que guiarán el proceso de diseño.

3.1.3. Requerimientos mínimos

3.2. Hardware

3.3. Software

3.3.1. Arquitectura del Software

API de comunicación

SDK para Payloads

Capítulo 4

Implementación

Capítulo Adicional que no es apéndice

Apéndices

A . Tweets

Referencias