



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
ELÉCTRICA – TELECOMUNICACIONES

DISEÑO, IMPLANTACIÓN Y VALIDACIÓN DE PROTOCOLOS Y SOFTWARE DE
COMUNICACIONES DIGITALES PARA TRANSMITIR IMÁGENES DE SATÉLITE
PEQUEÑO A ESTACIÓN TERRENA.

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
DEL MORAL PEREA BERNARDO

TUTOR PRINCIPAL
DR. GUTIÉRREZ CASTREJÓN RAMÓN, IINGEN-UNAM

MÉXICO, D. F. OCTUBRE 2016

JURADO ASIGNADO:

Presidente: Dra. Moumtadi Fátima

Secretario: Dr. Martínez López José Ismael

Vocal: Dr. Gutiérrez Castrejón Ramón

1er Suplente: Dr. Pérez Alcázar Pablo Roberto

2do Suplente: Dr. Matías Maruri José María

Lugar donde se realizó la tesis: Instituto de Ingeniería, UNAM

TUTOR DE TESIS:

DR. GUTIÉRREZ CASTREJÓN RAMÓN

FIRMA

En memoria de
Dr. Esaú Vicente Vivas

Un excelente maestro, pero sobre todo un gran ser humano. Nunca olvidaré tus últimas palabras y consejos.

Agradecimientos

Gracias Dios por seguir llenándome de bendiciones pero sobre todo por darme fuerza y alumbrar mi camino en los momentos más oscuros.

A las personas más importantes en mi vida, mi padre Juan y mi madre Elisa. Estoy completamente agradecido con ustedes por sus grandes enseñanzas, por su sacrificio incondicional, sus palabras de aliento, su cariño y toda la confianza depositada en mí.

A mi hermano Miguel Angel por compartir conmigo grandes momentos de mi infancia.

A mi novia Mitzi por apoyarme y confiar en mí todo este tiempo.

A mis amigos y compañeros de trabajo del Instituto de Ingeniería Emilio Jiménez, Pedro Ramírez, Jorge González y Ulises Bravo.

Al Dr. Esaú Vicente Vivas (†) por sus enseñanzas y apoyo durante mis estudios de licenciatura y gran parte de mis estudios de maestría. Siempre te estaré agradecido.

Al Dr. Ramón Gutiérrez Castrejón y al Ing. Enrique Ramón Gómez Rosas por brindarnos todo su apoyo para poder finalizar este proyecto.

Al CONACYT (beca N° 631219) por el apoyo económico brindado durante mis estudios de maestría.

A los profesores del posgrado que me apoyaron dentro de mi formación académica así como a mi segundo hogar la Universidad Nacional Autónoma de México.

Índice general

| | |
|---|-----------|
| Lista de acrónimos | IV |
| Índice de Figuras | VI |
| Índice de Tablas | VII |
| Resumen | VIII |
| Abstract | IX |
| 1. Introducción | 1 |
| 1.1. Filosofía de diseño de satélites pequeños | 1 |
| 1.2. Estándar Cubesat | 2 |
| 1.3. Protocolos de comunicación en cubesats | 4 |
| 1.4. Retos en un ambiente espacial | 5 |
| 1.5. Objetivos de la tesis | 6 |
| 1.6. Organización de la tesis | 6 |
| 2. Arquitectura Satelital y Hardware usado | 8 |
| 2.1. Subsistema de Potencia Eléctrica | 9 |
| 2.2. Subsistema de Comunicaciones | 10 |
| 2.3. Subsistema de Control y Orientación | 10 |
| 2.4. Computadora de Vuelo | 11 |
| 2.4.1. Hardware de la Computadora de Vuelo | 11 |
| 2.4.1.1. Microcontrolador TMS570LS1227 | 11 |
| 2.4.1.2. Memoria | 12 |
| 2.4.1.3. Reloj en tiempo real | 12 |
| 2.4.2. Protocolos de Comunicación | 12 |
| 2.5. Cámara Satelital | 14 |
| 3. Diseño e Implementación de Software | 16 |
| 3.1. Tareas y manejo de datos | 16 |
| 3.1.1. Protocolo I2C | 17 |
| 3.1.1.1. Estructura de comandos internos | 19 |
| 3.1.2. Comunicación con banda S | 19 |
| 3.1.3. Obtención de temperatura | 20 |
| 3.1.3.1. Configuración de sensores de temperatura | 21 |
| 3.1.3.2. Descripción de bus 1-wire e implementación de protocolo | 21 |
| 3.1.4. Uso de reloj en tiempo real | 23 |
| 3.1.5. Magnetómetro triaxial | 26 |
| 3.1.6. Uso y organización de memoria flash | 27 |
| 3.1.6.1. Memoria flash tipo NOR para almacenado de telemetría | 27 |
| 3.1.6.2. Memoria flash interna de MCU para control de tareas y/o comandos | 29 |
| 3.1.7. Control de Cámara | 30 |
| 3.1.7.1. Interfaz de comandos | 31 |
| 3.1.7.2. Captura de imágenes | 32 |

| | | |
|-----------|--|-----------|
| 3.1.7.3. | Almacenamiento de imágenes | 35 |
| 3.2. | Comandos operativos | 37 |
| 3.2.1. | Telemetría básica | 38 |
| 3.2.2. | Telemetría intermedia | 39 |
| 3.2.3. | Telemetría avanzada | 40 |
| 3.2.4. | Telemetría de propagación orbital | 41 |
| 3.2.5. | Apuntamiento satelital para captura y descarga de imágenes | 42 |
| 3.2.6. | Descarga de imágenes satelitales | 43 |
| 3.2.7. | Contacto satelital | 44 |
| 3.3. | Gestión interna satelital | 44 |
| 3.3.1. | Diagnóstico satelital | 44 |
| 3.4. | Implementación de software | 46 |
| 3.4.1. | Code Composer Studio | 46 |
| 3.4.2. | HalCoGen | 48 |
| 4. | Pruebas de validación | 49 |
| 4.1. | Validación de computadora de vuelo | 49 |
| 4.1.1. | Validación de protocolo I2C | 49 |
| 4.1.2. | Validación de sensores de temperatura | 51 |
| 4.1.3. | Validación de magnetómetro triaxial | 53 |
| 4.1.4. | Validación de memorias flash externas | 53 |
| 4.2. | Interfaz de Estación Terrena | 54 |
| 4.3. | Pruebas de calificación espacial | 64 |
| 4.3.1. | Pruebas de vibración | 64 |
| 5. | Conclusiones y Trabajo futuro | 67 |
| 5.1. | Trabajo futuro | 68 |
| A. | Diagrama a bloques de computadora de vuelo | 69 |
| B. | Diagrama a bloques de reloj en tiempo real | 70 |
| C. | Documentos generados a partir de pruebas de vibración | 71 |

Lista de acrónimos

| | |
|-------|--|
| ADC | Analogic digital converter. |
| ADCS | Attitud determination and control subsystem. |
| CAN | Controller Area Network. |
| CMOS | Complementary metal-oxide-semiconductor. |
| COTS | Commercial of the shelf. |
| CV | Computadora de vuelo. |
| DC | Direct current. |
| ECC | Error correcting code. |
| GPIO | General-purpose input output. |
| IMU | Inertial measurement unit. |
| I2C | (ó IIC) Inter integrated circuit. |
| LEO | Low earth orbit. |
| LIN | Local interconnect Network. |
| MCU | Microcontroller. |
| OBC | On board computer. |
| PCB | Printed circuit board. |
| P-POD | Poly picosatellite orbital deployer. |
| SCI | Serial comunication interface |
| SEL | Single event latchup. |
| SEU | Single event upset. |
| SPI | Serial peripheral interface. |
| SSTP | Small spacecrafft technology program. |
| TI | Texas Instruments. |
| UART | Universal asynchronous receiver transmitter. |
| USB | Universal serial bus. |
| UHF | Ultra high frequency. |
| VHF | Very high frequency. |

Índice de figuras

| | |
|--|----|
| 1.1. Clasificación de satélites | 1 |
| 1.2. Filosofía de satélites convencionales vs satélites pequeños | 2 |
| 1.3. Estructura cubesat 1U y P-POD | 2 |
| 1.4. Altura vs temperatura | 5 |
| 2.1. Estructura y ensamble final del satélite cubesat | 8 |
| 2.2. Diagrama a bloques del subsistema de potencia | 9 |
| 2.3. Tarjeta electrónica de potencia eléctrica | 9 |
| 2.4. Astrodev He-100 y Microhard n2420 | 10 |
| 2.5. Tarjeta electrónica y módulo de actuadores de control y orientación | 11 |
| 2.6. Protocolos de Comunicación | 13 |
| 2.7. Tarjeta electrónica de computadora de vuelo | 14 |
| 2.8. Cámara IDS UI-1220SE y tarjetas electrónicas | 15 |
| 3.1. Funciones por subsistema satelital | 16 |
| 3.2. Transferencia de datos en bus I2C | 17 |
| 3.3. Modo de operación de bus I2C para evitar errores | 18 |
| 3.4. Subsistemas conectados al bus I2C | 19 |
| 3.5. Transferencia de datos en bus SCI | 20 |
| 3.6. Configuración de sensores de temperatura | 21 |
| 3.7. Operaciones básicas de bus 1-wire | 22 |
| 3.8. Diagrama de flujo para lectura de temperaturas | 23 |
| 3.9. Diagrama de flujo para configuración de reloj en tiempo real | 25 |
| 3.10. Diagrama de flujo para lectura de campos magnéticos | 27 |
| 3.11. Uso de segmentos en memoria flash | 28 |
| 3.12. Dispositivo maestro y memorias flash en bus SPI | 29 |
| 3.13. Diagrama de cámara satelital | 30 |
| 3.14. Conexiones eléctricas de cámara satelital | 31 |
| 3.15. Diagrama de flujo para operación con cámara satelital | 33 |
| 3.16. Imagen correctamente expuesta | 34 |
| 3.17. Imagen sub-expuesta | 34 |
| 3.18. Imagen sobre-expuesta | 35 |
| 3.19. Diagrama de flujo para almacenamiento de imágenes satelitales | 36 |
| 3.20. Diagrama de flujo para recepción de comandos operativos | 37 |
| 3.21. Diagrama de flujo de telemetría básica | 38 |
| 3.22. Diagrama de flujo de telemetría intermedia | 39 |
| 3.23. Diagrama de flujo de telemetría avanzada | 40 |
| 3.24. Diagrama de flujo para descarga de propagación orbital | 41 |
| 3.25. Ejecución de algoritmos de control y apuntamiento satelital | 42 |
| 3.26. Diagrama de flujo para descarga de imágenes | 43 |
| 3.27. Diagrama de flujo de diagnóstico satelital | 45 |
| 3.28. Ventana de editor de code composer studio | 46 |
| 3.29. Ventana para depuración de código de code composer studio | 47 |
| 3.30. Hardware XDSv100 para programación de software | 47 |
| 3.31. Ventana principal de plataforma halcogen | 48 |

| | |
|--|----|
| 4.1. Validación de protocolo I2C | 50 |
| 4.2. ID de 3 sensores de temperatura | 51 |
| 4.3. Lectura de temperatura de los 20 sensores | 52 |
| 4.4. Lectura de registros de magnetómetro | 53 |
| 4.5. ID de memorias flash | 54 |
| 4.6. Radio ICOM IC-9100 y TNC Kantronics 9612 | 55 |
| 4.7. Diagrama de conexión entre PC y radios de comunicación | 55 |
| 4.8. Pantalla principal de interfaz de estación terrena | 56 |
| 4.9. Pestaña para configuración y manejo de cámara satelital | 57 |
| 4.10. Proceso de descarga de imagen satelital | 57 |
| 4.11. Proceso de descarga de mosaico de imágenes | 58 |
| 4.12. Proceso de prueba en subsistema de potencia eléctrica | 59 |
| 4.13. Pestaña de telemetría instantánea | 59 |
| 4.14. Gráficas en software de estación terrena de voltajes y corrientes en panel | 60 |
| 4.15. Gráfica en Matlab de voltajes y corrientes en panel | 60 |
| 4.16. Gráficas en software de estación terrena de voltajes, corrientes y temperaturas en C.V. | 61 |
| 4.17. Gráficas en Matlab de voltajes, corrientes y temperaturas en la C.V. | 61 |
| 4.18. Gráficas en software de estación terrena de voltajes, corrientes y porcentaje de carga en baterías | 62 |
| 4.19. Gráficas en Matlab de voltajes, corrientes y porcentaje de carga en baterías | 62 |
| 4.20. Gráficas en software de estación terrena de giroscopio y magnetómetro | 63 |
| 4.21. Gráficas en Matlab de giroscopio y magnetómetro | 63 |
| 4.22. Máquina vibratoria y test-POD | 65 |
| 4.23. Vibración de cubesat en eje X y Z | 66 |
| | |
| A.1. Diagrama de computadora de vuelo | 69 |
| | |
| B.1. Diagrama a bloques de reloj en tiempo real | 70 |
| | |
| C.1. Prueba de vibración en eje X. Hoja 1. | 72 |
| C.2. Prueba de vibración en eje X. Hoja 2. | 73 |
| C.3. Prueba de vibración en eje X. Hoja 3. | 74 |
| C.4. Prueba de vibración en eje X. Hoja 4. | 75 |
| C.5. Prueba de vibración en eje Y. Hoja 1. | 76 |
| C.6. Prueba de vibración en eje Y. Hoja 2. | 77 |
| C.7. Prueba de vibración en eje Y. Hoja 3. | 78 |
| C.8. Prueba de vibración en eje Y. Hoja 4. | 79 |
| C.9. Prueba de vibración en eje Z. Hoja 1. | 80 |
| C.10. Prueba de vibración en eje Z. Hoja 2. | 81 |
| C.11. Prueba de vibración en eje Z. Hoja 3. | 82 |
| C.12. Prueba de vibración en eje Z. Hoja 4. | 83 |

Índice de tablas

| | |
|---|----|
| 1.1. Cubesats y protocolos de comunicación | 4 |
| 2.1. Tipo de memorias usadas en cubesats | 12 |
| 2.2. Especificaciones de la computadora de vuelo | 14 |
| 3.1. Estructura de comandos internos entre subsistemas | 19 |
| 3.2. Ubicación de sensores de temperatura | 21 |
| 3.3. Tabla de registros de reloj en tiempo real | 24 |
| 3.4. Tabla de registros de magnetómetro Honeywell | 26 |
| 3.5. Mapa de memoria externa | 28 |
| 3.6. Mapa de memoria flash de microcontrolador Hércules | 30 |
| 3.7. Estructura de comandos entre cámara y computadora de vuelo | 31 |
| 3.8. Estructura de paquete imagen | 32 |
| 3.9. Estructura de comando entre estación terrena y satélite | 37 |
| 4.1. Formato ID sensores de temperatura | 51 |
| 4.2. Registro para lectura y configuración de sensores 1-wire | 52 |
| 4.3. Formato ID de memorias flash externas | 53 |

Resumen

El objetivo principal del presente trabajo de tesis fue diseñar software operativo y hacer uso de diversos protocolos de comunicación para poder obtener y transmitir datos de telemetría y descarga de imágenes de un satélite a una estación terrena. Todo el software fue implantado dentro de la computadora de vuelo de un satélite cuyas dimensiones, peso y tamaño están especificados dentro del estándar cubesat. La computadora de vuelo está basada en un microcontrolador de 32 bits de Texas Instruments y cuenta con un magnetómetro triaxial, un reloj en tiempo real, diversos sensores de temperatura, protecciones de sobrecorriente contra efecto latch up y dos memorias flash externas.

El software generado a partir de los objetivos planteados contempla los siguientes aspectos: manejo y control de una cámara con sensor CMOS monocromático, uso del bus I2C como principal protocolo de comunicación entre subsistemas satelitales, uso del protocolo UART o puerto serie para el envío de imágenes por medio de un radio en banda S, uso del protocolo SPI para la interacción con las memorias flash y reloj en tiempo real y uso del protocolo 1-wire para obtener lectura de los sensores de temperatura dispersos en el satélite. Todo el software desarrollado permite a la computadora de vuelo interactuar con los demás subsistemas satelitales de manera que ésta pueda realizar un manejo, obtención, almacenamiento y envío de telemetría satelital e imágenes a una estación terrena. Adicionalmente se desarrollaron comandos operativos que permiten al satélite tener autonomía para poder operar y recuperarse ante un problema presente en un ambiente espacial.

El método empleado para validar el subsistema de computadora de vuelo consistió en generar software básico que permitiera leer valores fijos dentro de los registros de control en cada dispositivo. Con esto se asegura que el microcontrolador se comunica correctamente con cada uno de los dispositivos por medio del protocolo de comunicación empleado. También, se procedió a validar los comandos operativos que permiten el manejo, almacenamiento y envío de telemetría e imágenes. Para lograrlo se usó un software de estación terrena el cual muestra la información recibida de manera gráfica y/o de forma numérica. La interpretación de datos y el correcto despliegue de imágenes por parte del software de estación terrena nos permite validar que la computadora de vuelo recolecta, almacena y envía telemetría de forma exitosa mediante una comunicación en la frecuencia UHF/VHF y banda S.

Abstract

The main objective of this thesis proposal was to design operating software and make use of several communication protocols in order to collect and transmit telemetry data and downloading images from the satellite to a ground station. All the software was implanted into the on board computer of a satellite whose dimensions, weight and size are specified within the cubesat standard. The on board computer is based on a 32 bit microcontroller of Texas Instruments, a triaxial magnetometer, a real time clock, temperature sensors, overcurrent protection against latch up effect and two external flash memories.

The software generated from the goals includes the following points: management and control of a camera with monochromatic CMOS sensor, use of I2C bus as a main communication protocol between satellite subsystems, use of UART protocol or serial port to transmit images with a S band radio, use of SPI protocol for interaction with flash memories and real time clock and finally use of 1-wire protocol to get data of all temperature sensors within the satellite. All the software developed allows the on board computer interacts with others subsystems of the satellite so that the on board computer does management, obtain, store and transmit telemetry and images to a ground station. Additionally operational commands that allow the satellite to have autonomy to operate and recover from a problem present in a space environment were developed.

The method used to validate the on board computer was generate basic software to read fixed values in the control registers on each device. This ensures that the microcontroller is successfully communicating with each device via the communication protocol used. Operational commands were also validated with the use of a ground station software which displays the received information graphically and numerically. Data interpretation and proper display of images by the ground station software allow us to validate that the on board computer collects, stores and send telemetry successfully by a communication in the UHF/VHF frequency and S band.

Capítulo 1

Introducción

La necesidad cada vez más grande de mayor capacidad y mayor complejidad en satélites [1], condujo a un crecimiento natural en la masa de los satélites. Esta tendencia fue limitada en un inicio por la capacidad de los lanzadores disponibles y después por todos los requerimientos de infraestructura y gran inversión por parte de las agencias espaciales. Como resultado de esto, se ha provocado una revolución espacial que ha dado como resultado la aparición de satélites pequeños.

1.1. Filosofía de diseño de satélites pequeños

El programa de tecnología de satélites pequeños (SSTP, por sus siglas en inglés) de la NASA [2] clasifica a los satélites pequeños como aquellos satélites que cuentan con una masa inferior a 180 kg. Dentro de esta clasificación, los satélites pequeños a su vez son agrupados de acuerdo a su peso, teniendo así: minisatélites con una masa de 100-500 kg, microsateélites con una masa de 10-100 kg, nanosatélites con una masa de 1-10 kg y picosatélites con una masa por debajo de 1 kg. Esto se ilustra gráficamente en la figura 1.1.

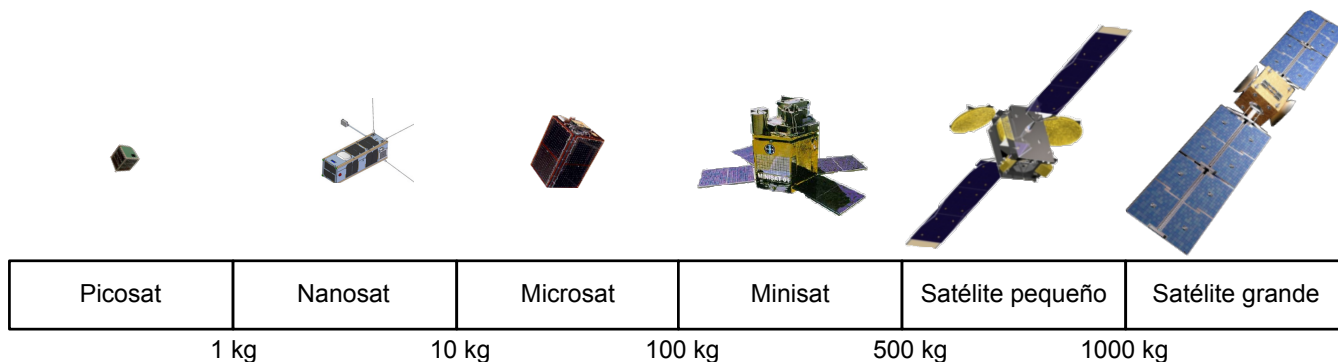


Figura 1.1: Clasificación de satélites

Los satélites pequeños son construidos en tiempos de desarrollo cortos [1] (generalmente entre 12 y 18 meses) a un costo relativamente bajo y haciendo un uso máximo de componentes comerciales (COTS, por sus siglas en inglés). Los satélites están pensados, específicamente, para lograr sus objetivos de misión en el tiempo de vida al cual fueron diseñados. Los objetivos de la misión deben ser claros de tal forma que se alcance la mayoría de éstos con un costo mínimo, por lo general con el objetivo de alcanzar 80% de los objetivos de la misión con un 20% del costo. Las principales diferencias entre la filosofía de diseño de satélites pequeños y satélites convencionales son mostradas en la figura 1.2.

Hoy en día, el rápido desarrollo de dispositivos electrónicos permite que los satélites pequeños hagan uso de tecnología nueva, sin uso previo en el espacio, generando la incertidumbre de si el dispositivo será capaz de funcionar correctamente, puesto que no fue diseñado para uso espacial.

En misiones satelitales convencionales, se tiene suficiente tiempo y recursos para llevar a cabo un programa exhaustivo de pruebas en tierra antes del vuelo, pero esto es problemático si existe un bajo costo y tiempo de diseño corto como es el caso de los satélites pequeños.

Para reducir el riesgo asociado al uso de componentes COTS, se debe confiar en la experiencia y pruebas a nivel de sistemas en lugar de realizar comprobación en dispositivos individuales. Sin embargo, también se requiere que los ingenieros de diseño tengan un conocimiento del entorno al que será sometido su sistema y el uso de técnicas apropiadas de diseño. Algunas de las medidas empleadas podrían ser las siguientes:

- Minimizar la variedad de dispositivos o materiales.
- Evitar el uso de sustancias tóxicas, volátiles o sustancias explosivas.
- Mantener interfaces simples.
- Minimizar partes móviles.
- Evitar bucles en software.

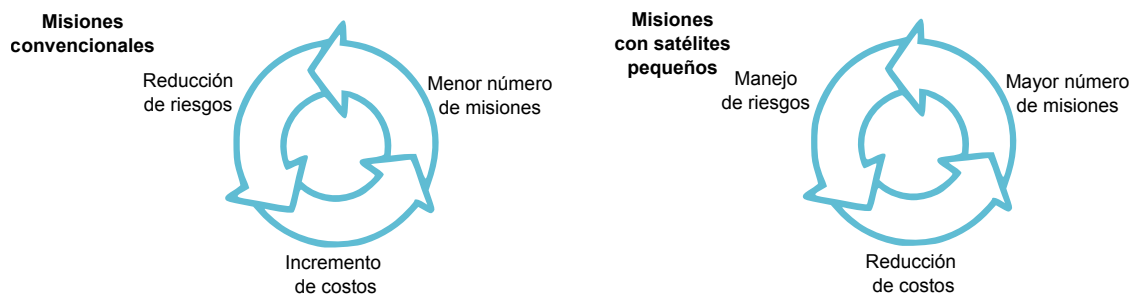


Figura 1.2: Filosofía de satélites convencionales vs satélites pequeños

1.2. Estándar Cubesat

Iniciado en 1999, el proyecto cubesat [3] empezó como una colaboración de esfuerzo entre el profesor Jordi Puig Suari de la California Polytechnic State University (Cal Poly, por sus siglas en inglés), San Luis Obispo, y el profesor Bob Twiggs de Stanford University’s Space Systems Development Laboratory (SSDL, por sus siglas en inglés). Siguiendo con la filosofía de diseño de un satélite pequeño, el propósito del proyecto es proporcionar un estándar de diseño de nanosatélites y picosatélites para reducir costos y tiempo de desarrollo, incrementar accesibilidad al espacio y generar lanzamientos constantes. En este momento, el proyecto cubesat es una colaboración internacional con alrededor de 100 universidades, preparatorias y dependencias privadas desarrollando nanosatélites con experimentos científicos. Un cubesat está basado en un satélite cúbico con dimensiones de 10 x 10 x 10 cm (1U), sin embargo, también puede ser compuesto por combinaciones como 10 x 10 x 20 cm (2U) o 10 x 10 x 30 cm(3U). La figura 1.3 muestra, del lado izquierdo, la estructura de un cubesat 1U y, en el centro y derecha, la interfaz de lanzamiento o P-POD.

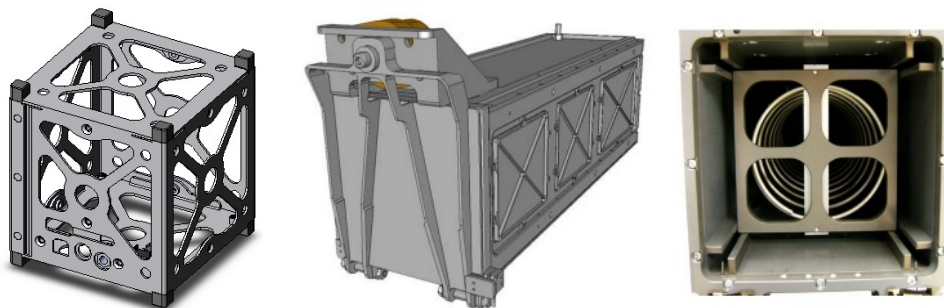


Figura 1.3: Estructura cubesat 1U y P-POD

Puesto que la principal misión del programa cubesat es proporcionar acceso al espacio a satélites pequeños, Cal Poly desarrolló el Poly Picosatellite Orbital Deployer (P-POD, por sus siglas en inglés) como método de protección al satélite cubesat y al vehículo lanzador. Una vez considerado lo anterior, los desarrolladores de cubesats deben asegurar el éxito de su satélite con la implementación de pruebas y verificación de sus subsistemas. Algunas de estas pruebas son: vibración, ciclado térmico, prueba de impacto e inspección visual. La figura 1.3 muestra una estructura correspondiente a un cubesat 1U y la interfaz de expulsión o P-POD.

Al igual que un satélite convencional grande, un cubesat cuenta con diversos subsistemas [5] y cada uno de ellos tiene la capacidad de operar en forma independiente. A continuación se presentan los subsistemas satelitales y se describen sus principales objetivos.

- 1.- **Subsistema de control y orientación.** El subsistema de control y orientación tiene como objetivo estabilizar y orientar un satélite con respecto a un punto de referencia. Dependiendo de los objetivos de la misión, el satélite tendrá diferentes requisitos de apuntamiento. En algunas ocasiones el satélite puede mantenerse girando pero en otras es necesario que señale una ubicación en específico. Para lograr esto, el subsistema debe contar con dispositivos de referencia como son: sensores de horizonte de tierra, sensores de sol, rastreador de estrellas o magnetómetros que le permiten medir el campo magnético terrestre. En cuanto a dispositivos de control puede contar con: propulsores, ruedas inerciales, bobinas de torque magnético y gradientes de gravedad.
- 2.- **Subsistema de potencia.** El subsistema de potencia eléctrica se encarga de suministrar, almacenar, regular y distribuir la energía a todo el satélite. Este subsistema puede hacer uso de tres fuentes diferentes: Solar con el uso de celdas solares, nuclear con el uso de reactores y en forma química empleando baterías. Debido a su tiempo de vida, un cubesat emplea únicamente dos de estos métodos, solar y química.
- 3.- **Subsistema de control térmico.** Éste subsistema se encarga de mantener la temperatura del satélite dentro de los rangos de temperatura especificados, evitando así daños en algunos componentes electrónicos o baterías.
- 4.- **Propulsión.** El subsistema de propulsión permite al satélite modificar su órbita (si es necesario), mantener su órbita debido a perturbaciones y, a veces, como un complemento para el subsistema de control y apuntamiento satelital. Para lograr esto, se hace uso de propulsores de diversos tamaños dependiendo de la capacidad del satélite.
- 5.- **Subsistema de manejo de datos.** La mayoría de los satélites se dedican a la recolección y transmisión de cierto tipo de información, incluso si éste satélite no fue diseñado para uso en telecomunicaciones. Este subsistema se encarga de recolectar datos de la carga útil y enviarlos a la estación terrena. Los datos recolectados pueden ser transmitidos a medida que se obtienen o pueden ser almacenados para su posterior transmisión. El satélite puede contar con diferentes métodos de enlace de bajada y uso de frecuencias como: VHF, UHF, banda S (2-4 GHz), banda X (8-10 GHz), banda K (13-15 GHz), etc.
- 6.- **Control de telemetría y rastreo.** La telemetría es un aspecto muy importante y con la que la mayoría de los satélites cuenta. La telemetría contiene información relacionada al estado del satélite y sus subsistemas, de tal forma que la estación terrena pueda modificar mediante uso de comandos futuras acciones. Hoy en día, los satélites tienen la capacidad de almacenar instrucciones y realizarlas después de manera autónoma. Cabe destacar que en algunas ocasiones, el subsistema de manejo de datos y control de telemetría y rastreo, son asociados en un mismo subsistema permitiendo conservar peso y espacio en el satélite.
- 7.- **Computadora de vuelo.** El subsistema de computadora de vuelo interactúa con los diferentes subsistemas satelitales y se encarga de realizar tareas o funciones especificadas por el diseñador como pueden ser:
 - Control de altitud
 - Control de navegación
 - Procesamiento de comandos
 - Procesamiento de telemetría
 - Manejo de carga útil satelital
 - Manejo de potencia eléctrica
 - Recuperación ante una falla
 - Manejo de datos.

Las funciones que realiza la computadora de vuelo pueden ser complejas o simples dependiendo el grado de autonomía con que el satélite fue diseñado. Una gran autonomía corresponde a mayor capacidad computacional, de tal manera que el sistema pueda operar y tomar decisiones ante una falla sin necesidad de interactuar con alguna estación terrena durante meses.

- 8.- **Estructura.** La estructura satelital contiene y protege toda la electrónica del satélite. Ésta debe ser capaz de sobrevivir el violento proceso de lanzamiento y posicionamiento, ser capaz de desplegar partes mecánicas móviles y finalmente mantener la integridad del satélite durante maniobras y cambios de orientación y posicionamiento. La estructura satelital generalmente corresponde al 20% de la masa total del satélite, sin embargo, este peso se debe en gran medida al soporte de la carga útil satelital.
- 9.- **Estación terrena.** Aunque la estación terrena no es un subsistema contenido en el satélite, ésta no deja de ser importante debido a que es parte de un sistema espacial completo. La estación terrena se encarga de facilitar el control del satélite monitoreando sus tareas, transmitiendo y recibiendo información recolectada y ordenándola en una manera simple para el usuario.

1.3. Protocolos de comunicación en cubesats

Un punto muy importante a tratar cuando se habla del diseño de un cubesat, es lo referente a el o los protocolos de comunicación a usar. Hoy en día, son diversos los protocolos de comunicación usados por los satélites cubesat y cada uno es empleado de acuerdo a los requerimientos de cada misión. Mientras que el bus USB (Universal Serial Bus) y CAN (Controller Area Network) son usados esporádicamente, el protocolo de comunicación I2C [6] es el estándar más popular para satélites pequeños. Algunas de las características del bus I2C son las siguientes:

- Únicamente se requieren dos líneas; línea de datos (SDA) y línea de reloj (SCL).
- Cada dispositivo conectado al bus cuenta con una dirección única y diferente que le permite mantener una comunicación maestro-esclavo.
- Su comunicación es bidireccional con palabra de datos de 8 bits a una velocidad de 100 kbit/s en modo estándar, 400 kbits/s en modo rapido y hasta 3.4 Mbit/s en modo de alta velocidad.
- Consumo de corriente extremadamente bajo.
- Gran inmunidad al ruido.
- Amplio rango de temperatura de operación.
- Integrado en gran parte de circuitos integrados.

La siguiente tabla muestra algunos protocolos de comunicación usados por cubesats y fue obtenida a partir de una búsqueda [4] por las principales características de cada misión satelital:

| Satélite | Protocolos de comunicación | Lugar de desarrollo |
|------------|----------------------------|---|
| Aalto-1 | I2C,SPI,UART | Aalto University, Finlandia |
| All-Star | I2C,SPI | Colorado Space Grant Consortium, EUA |
| Cute-I | SCI | TITech y SRTL, Tokyo |
| Delfi-C3 | I2C | Delft University of Technology, Holanda |
| Delfi-n3Xt | I2C | Delft University of Technology, Holanda |
| GOLIAT | I2C,SPI | Romanian Space Agency, Rumania |
| KySat-1 | UART,I2C,SPI | Kentucky Space Consortium, EUA |
| M3/M-Cubed | I2C,SPI,USART,USB | University of Michigan, EUA |
| RAX | I2C | University of Michigan, EUA |
| ROBUSTA | CAN,SPI | University of Montpellier, Francia |
| Xatcobeo | I2C | University of Vigo e INTA, España |

Tabla 1.1: Cubesats y protocolos de comunicación

Como se puede observar en la tabla, existen diversas universidades y organizaciones de diferentes partes del mundo que ya tienen una misión satelital exitosa y a medida que el tiempo pasa, más y más universidades de diversos países se han empezado a interesar en el desarrollo de satélites cubesat. No es extraño que cada día observemos distintas misiones satelitales probando diversos tipos de experimentos, circuitos electrónicos, diversos protocolos de comunicación y nuevos algoritmos que han permitido mejorar el diseño y aumentar la probabilidad de éxito en misiones futuras.

1.4. Retos en un ambiente espacial

Cuando hablamos del diseño de un satélite, siempre debemos tomar en cuenta que éste será sometido a un ambiente totalmente diferente al que experimentaría estando en tierra. En muchos sentidos, el ambiente espacial es hostil con dispositivos electrónicos y demás hardware contenido dentro del satélite por lo que, el satélite debe ser pensado para trabajar en este ambiente.

A continuación se presentan algunos efectos que padece un satélite en el espacio, debido en gran medida, a la presencia del sol.

Temperatura: Un satélite en órbita [5] puede modificar su temperatura debido a distintas fuentes de calor, como son la exposición directa a la radiación solar, emisiones infrarrojas de la tierra y calor generado por su propio sistema. En la mayoría de los casos, las fuentes de calor pueden tener ciclos ya que el satélite pasará de la luz a la obscuridad y viceversa, generando cambios de temperatura rápidamente. Por todo lo anterior, el satélite debe ser capaz de operar y, si es posible, tener un manejo de temperatura para evitar que dispositivos electrónicos, baterías, etc., pueden resultar dañados. La imagen 1.4 muestra como la temperatura [5] puede variar con respecto a la altura de la superficie de la tierra.

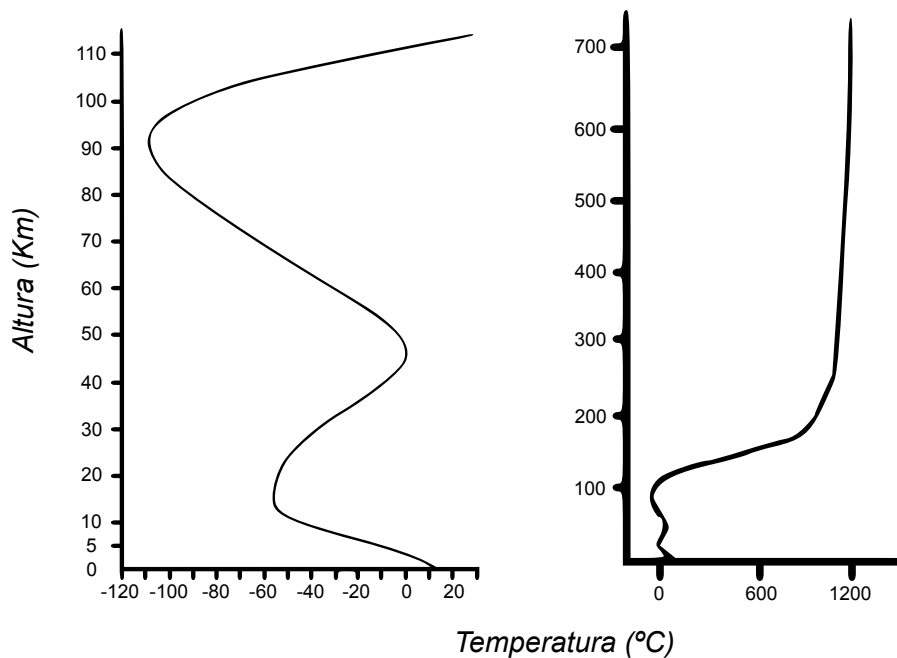


Figura 1.4: Altura vs temperatura

Radiación: La radiación espacial [7] es generada por partículas emitidas por diversas fuentes dentro y más allá del sistema solar. Las partículas de alta energía pueden llegar a depositar una carga eléctrica dentro de componentes electrónicos e inclusive cambiar la estructura física de los materiales. Para el caso de una órbita terrestre baja (LEO, por sus siglas en inglés) entre 200 y 2000 km de la superficie de la tierra, la exposición a rayos cósmicos y partículas de erupciones solares, aumenta a medida que se aumenta la altitud y la inclinación del satélite con respecto al ecuador. Sin embargo, en la órbita LEO, el campo geomagnético terrestre puede llegar a proporcionar cierta protección ante la radiación presente. Los efectos que sufre un satélite pueden ser divididos en dos áreas: dosis total ionizante (TID, por sus siglas en inglés) y efectos de evento único (SEE, por sus siglas en inglés).

TID se manifiesta como fallas funcionales, modificación de voltajes, entre otros y es debido a acumulación de energía depositada en el material.

SEE ocurre cuando un ion golpea el material depositando suficiente energía en el dispositivo. Dentro de los SEE, existen dos categorías: "soft errors" y "hard errors".

Un caso de soft errors es el single event upset (SEU), que afecta principalmente memorias o registros de datos y se manifiesta cambiando valores de bits (bit-flip).

En cuanto a hard errors existe el single event latchup (SEL) en donde el dispositivo excede la máxima corriente especificada y, en caso de que el dispositivo no se desenergice, éste será destruido.

1.5. Objetivos de la tesis

El Instituto de Ingeniería de la UNAM en conjunto con algunas dependencias del Instituto Politécnico Nacional y en coordinación por parte de la Agencia Espacial Mexicana, están desarrollando un proyecto satelital con el principal objetivo de poner en órbita un nanosatélite basado en el estándar cubesat. Este proyecto, a su vez, permitirá que un equipo de estudiantes obtenga gran experiencia en el diseño y realización de un sistema satelital real.

El cubesat está diseñado para operar en la órbita baja de la tierra (LEO, por sus siglas en inglés) a una altura entre 400 y 600 km de la superficie terrestre. El satélite cubesat cuenta con subsistema de potencia, computadora de vuelo, subsistema de control y orientación, subsistema de comunicaciones en UHF/VHF y banda S y una cámara con sensor CMOS monocromático como carga útil con el objetivo de capturar imágenes.

Este trabajo de tesis tiene como objetivo principal el diseño de software y uso de diversos protocolos de comunicación para obtener y transmitir datos de telemetría y descargar imágenes a una estación terrena. El software estará contenido dentro del subsistema de computadora de vuelo y se hará uso del bus I2C como principal protocolo de comunicación entre subsistemas.

Además de su objetivo principal, el software diseñado permitirá a la computadora de vuelo realizar tareas tales como:

- Recolección de telemetría para conocer el estado del satélite en línea de vista y/o capacidad de almacenamiento para su posterior descarga.
- Manejo de la cámara satelital permitiendo captura y almacenamiento de imágenes en dos tamaños distintos.
- Rutinas para mantener la integridad del satélite ante fallas de temperatura y radiación que se presenten.
- Programación de acciones a realizar como captura de imágenes y captura de datos orbitales.
- Rutinas iniciales cuando el satélite es expulsado del P-POD.

1.6. Organización de la tesis

El contenido de esta tesis está organizado de la siguiente manera:

El capítulo 1 describe qué es un satélite pequeño y compara la filosofía de diseño entre un satélite convencional y un satélite pequeño. Dentro de los satélites pequeños se describe el estándar cubesat y el ambiente espacial al cual será expuesto. Se nombran algunos protocolos de comunicación usados por cubesats, dando principal énfasis en el bus I2C, y finalmente se presentan los objetivos del presente trabajo de tesis.

En el capítulo 2 se presenta la arquitectura satelital propuesta por el equipo de trabajo del Instituto de Ingeniería. Se da una breve descripción de cada subsistema y se dará principal énfasis a los subsistemas encargados de la implantación de software y adquisición de telemetría e imágenes.

El capítulo 3 describe principalmente la fase de diseño de software y los protocolos de comunicación empleados para realizar las tareas relacionadas a la telemetría. Se describen también los comandos operativos y las tareas a realizar de forma autónoma por el satélite para evitar o reaccionar ante las adversidades de operar en un ambiente espacial.

El capítulo 4 presenta los resultados experimentales obtenidos a partir del software desarrollado. Los resultados incluyen, control de la cámara para captura y descarga de imágenes, manejo y obtención de telemetría satelital, etc. Toda esta información es interpretada mediante el uso de un software de estación terrena implantado en una PC y dicho software será descrito en este capítulo.

Finalmente, el capítulo 5 presenta los objetivos alcanzados, define mejoras al diseño de software y se dan algunas recomendaciones para trabajo futuro.

Capítulo 2

Arquitectura Satelital y Hardware usado

Debido a que la computadora de vuelo interactúa con los diferentes subsistemas satelitales, este capítulo tiene el objetivo de presentar la arquitectura, el hardware y demás subsistemas que contendrá el cubesat de manera que el lector pueda ver claramente los alcances de la misión.

El cubesat diseñado cuenta con una configuración 3U (10cm x 10cm x 30cm) y un peso aproximado de 4kg. El cubesat contará con todos los requerimientos necesarios para poder ser transportado y posteriormente expulsado desde el P-POD o interfaz de lanzamiento. La estructura diseñada está hecha de aluminio 7075-t6 con un término anodizado el cual evitará una prematura corrosión del material. Por las dimensiones de la estructura, cada tarjeta electrónica debió ser diseñada para contener sus componentes electrónicos en una área de 9.5 x 9.5 cm. La figura 2.1 presenta la estructura satelital y el ensamble final del satélite sin antenas.



Figura 2.1: Estructura y ensamble final del satélite cubesat

2.1. Subsistema de Potencia Eléctrica

El subsistema de potencia eléctrica proveerá, almacenará y distribuirá la energía eléctrica a cada subsistema y componente del cubesat. Este subsistema consiste de 4 paneles solares distribuidos sobre la parte exterior del satélite, un conjunto de baterías de polímero de litio, y hardware diverso para generar la distribución de voltajes y corrientes. El control de éste subsistema se logra con el uso de un microcontrolador de la familia MSP430 de Texas Instruments y, adicionalmente, diversos convertidores analógico-digital (ADC, por sus siglas en inglés), con el fin de conocer el valor real de los voltajes y corrientes generados. La figura 2.2 es una vista general del funcionamiento del subsistema de potencia eléctrica y la forma en cómo se distribuye la energía.

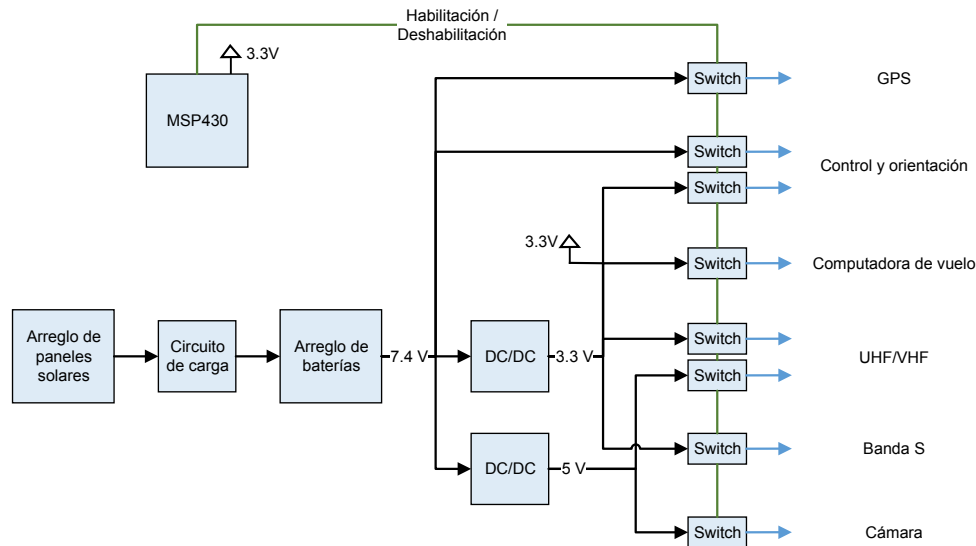


Figura 2.2: Diagrama a bloques del subsistema de potencia

Como muestra la imagen, el arreglo de paneles deberá ser capaz de almacenar la suficiente energía en las baterías para que el satélite pueda operar en modo eclipse o a la sombra. Debido a los diversos voltajes que manejan los subsistemas satelitales, fue necesario el uso de fuentes conmutadas o fuentes DC/DC que permiten la reducción de los niveles de voltaje y, en todos los casos, el microcontrolador esperará la señal de mando desde la computadora de vuelo para poder encender o apagar algún subsistema. Cabe mencionar que el subsistema de potencia eléctrica cuenta con circuitos de protección, permitiendo apagar cualquier subsistema en caso de presentarse un efecto de sobrecorriente sin necesidad de esperar la orden desde la computadora de vuelo.

La figura 2.3 muestra la tarjeta electrónica del subsistema de potencia eléctrica fabricada por el Ing. Jorge Gonzalez Vergara como parte del proyecto y que forma parte del equipo de trabajo del Instituto de Ingeniería UNAM.

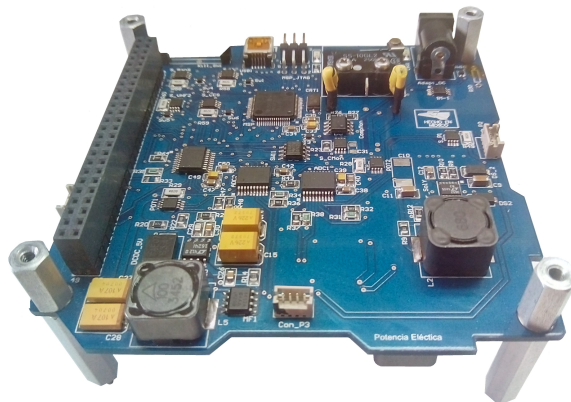


Figura 2.3: Tarjeta electrónica de potencia eléctrica

2.2. Subsistema de Comunicaciones

El subsistema de comunicaciones consta de dos módulos comerciales que permitirán generar el enlace de comunicación entre el cubesat y la estación terrena para el intercambio de datos y telemetría. Los módulos son el He-100 para comunicaciones en UHF/VHF y el radio de comunicaciones en banda S, Microhard n2420.

El He-100 [8] es un sistema de comunicación compatible con el estándar cubesat y se caracteriza por tener selección de frecuencia y potencia de salida variable. Además, tiene la capacidad de comunicarse con estaciones terrenas en las frecuencias de radio aficionados a una velocidad de 1200 bps, 9600 bps o grandes velocidades de transmisión con modulación GMSK. El radio transmite a una frecuencia de 440 MHz y recibe a una frecuencia de 140 MHz contando con la opción de modificar dichas frecuencias por software. La comunicación entre radios se logra usando el protocolo de comunicación AX.25 o con el uso de un protocolo binario especificado.

Adicionalmente, cuenta con una señal de beacon la cual corresponde a un mensaje enviado cada determinado tiempo permitiéndole al cubesat ser reconocido por cualquier estación terrena en diferentes partes del planeta.

El radio de comunicaciones en banda S, Microhard n2420 [9], logra un enlace a una velocidad de hasta 230 kbps y cuenta con una etapa de filtrado que le proporciona rechazo al ruido e interferencia. El radio opera a una frecuencia de 2.4 GHz y una potencia de 1 Watt. El objetivo del radio será únicamente la descarga de imágenes satelitales, esto debido a que permite la descarga de imágenes de manera más rápida que con el uso del radio UHF/VHF. La figura 2.4 presenta, del lado izquierdo, el módulo Astrodev He-101 y, del lado derecho, el módulo Microhard n2420.



Figura 2.4: Astrodev He-100 y Microhard n2420

2.3. Subsistema de Control y Orientación

El subsistema de control y orientación proporcionará un control activo en tres ejes para el satélite. Para la determinación de altitud, el subsistema hace uso de una unidad de medición inercial (IMU) que incluye magnetómetros y giróscopos así como una conexión directa con un GPS para obtener datos de latitud, longitud y altitud. La orientación satelital se logra con el uso de bobinas de torque magnético y el uso de ruedas inerciales. Para el manejo de sensores, uso de actuadores e implantación de software y algoritmos de control, el subsistema de control y orientación hace uso de un microcontrolador de la familia TMS570 de Texas Instruments. El subsistema de control y orientación deberá proporcionar los siguientes modos de operación:

- **Detumble:** Este modo es activado después de la expulsión permitiendo reducir el giro del cubesat.
- **Posicionamiento:** En este modo, la antena directiva de banda S debe ser apuntada hacia la estación terrena para lograr un enlace satelital. Del mismo modo, la cámara satelital deberá apuntar hacia un objetivo para lograr la captura de imágenes.

Finalmente, las ruedas inerciales cuentan con una masa o disco metálico que le permite generar mayor momento de inercia y están contenidas junto con las bobinas de torque magnético dentro de un contenedor hecho de nylamid que provee características como: resistencia al impacto, reducción de peso y es un material dieléctrico.

La figura 2.5 muestra, del lado izquierdo, la tarjeta electrónica del subsistema de control y orientación. En la parte derecha se encuentra el módulo que contiene las ruedas inerciales y bobinas de torque magnético. Este subsistema fue desarrollado por el M.I. Emilio Augusto Jiménez Madrigal como parte de este proyecto y también forma parte del equipo de trabajo del Instituto de Ingeniería UNAM.

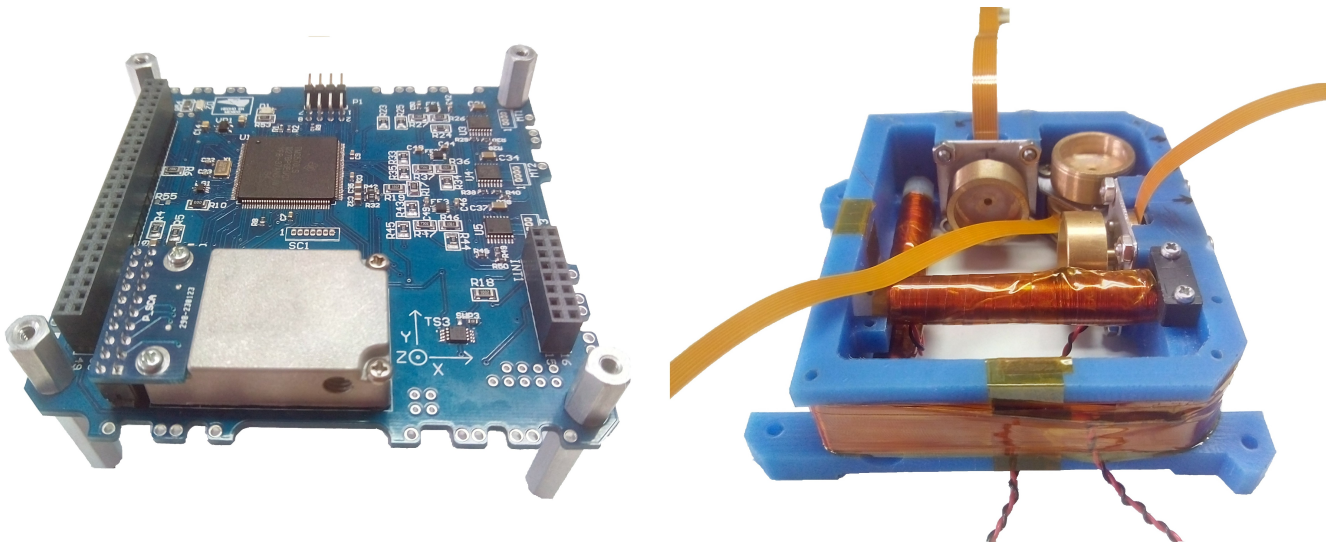


Figura 2.5: Tarjeta electrónica y módulo de actuadores de control y orientación

2.4. Computadora de Vuelo

La computadora de vuelo del cubesat será responsable de tareas como: manejo de datos y uso de cámara satelital, obtención y almacenado de telemetría, interoperabilidad con los demás subsistemas y ejecución de comandos.

La principal característica de este subsistema es su gran confiabilidad ya que corresponde a la parte central del satélite. Algunas de las medidas empleadas para lograr lo anterior son: uso de componentes de grado automotriz, uso de componentes resistentes a cierta dosis de radiación y el uso de un microcontrolador diseñado para su uso en sistemas seguros.

La Computadora de vuelo es de diseño propio por lo cual su ensamble y validación son presentados como parte de este trabajo de tesis. A continuación se presenta la arquitectura y características principales del subsistema.

2.4.1. Hardware de la Computadora de Vuelo

La computadora de vuelo de un satélite debe contar principalmente con una unidad de procesamiento, como puede ser un microcontrolador o un FPGA. Además de su unidad de procesamiento [2], sus principales componentes incluyen memoria, reloj e interfaces para comunicación con otros subsistemas.

La computadora de vuelo usada por el cubesat tiene como procesador principal un microcontrolador de 32 bits de la familia TMS570 de Texas Instruments, 2 memorias flash tipo NOR para almacenado de telemetría, un magnetómetro triaxial Honeywell, un reloj en tiempo real como temporizador para diversas tareas, sensores de temperatura y varios circuitos protectores para evitar efectos de sobrecorriente. La mayoría de los circuitos electrónicos fueron elegidos para trabajar en un intervalo de temperatura de entre -40 a 80°C y que estuvieran disponibles en el mercado. A continuación se describirán los componentes principales y el por qué fueron elegidos.

2.4.1.1. Microcontrolador TMS570LS1227

El TMS570LS1227 [10] pertenece a la familia de microcontroladores de 32 bits conocidos como Hércules. Es un dispositivo de alto rendimiento y grado automotriz diseñado para simplificar el desarrollo de sistemas funcionalmente seguros. Cuenta con protección por hardware en energía, reloj, reset y funciones de procesamiento básico. Algunas de sus características clave son:

- Doble núcleo de procesamiento en lockstep en un CPU ARM Cortex-R4F para detección de fallas. Distribución de reloj, energía y reset son implementados para mitigar fallas comunes en el CPU.
- Código corrector de errores (ECC) en SRAM y memoria flash.
- Cuenta con un oscilador interno de respaldo para cuando se presentan fallas en su PLL y oscilador externo.

- Retroalimentación de entradas y salidas analógicas y digitales.
- Pin dedicado al monitoreo del MCU en caso de falla

El dispositivo cuenta con múltiples interfaces de comunicación que son: tres MibSPI, hasta dos SPI, un LIN, un SCI, tres DCANs, un I2C, un Ethernet y un controlador FlexRay.

Por sus características de seguridad integradas y sus periféricos de comunicación incluidos, el TMS570LS1227 fue ideal para su uso como principal unidad de procesamiento dentro del cubesat.

2.4.1.2. Memoria

El uso de memorias para almacenamiento de datos es parte primordial de la computadora de vuelo ya que permite obtener y almacenar información mientras el satélite no está en línea de vista con una estación terrena. Hoy en día existen diferentes tipos de memorias, generalmente partiendo con una capacidad de 32 kB e incrementando con toda la tecnología disponible. A continuación se presenta una tabla comparativa entre los diferentes tipos de memoria y sus características presentada en un reporte de la NASA[2].

| Característica | SRAM | DRAM | Flash | MRAM | FERAM |
|----------------------|-----------------------------|------------------------------|----------------------------|---------------------------------------|------------------------------------|
| Descripción | Static random Access Memory | Dynamic Random Access Memory | Memoria derivada de EEPROM | Magnetoresistive Random Access Memory | Ferroelectric Random Access Memory |
| Volátil | No | No | Si | Si | Si |
| Voltaje de Operación | 3.3 - 5 V | 3.3 V | 3.3 V | 5 V | 3.3 V |
| Retención (@70°C) | N/A | N/A | > 10 años | > 10 años | > 10 años |
| Radiación | 1Mrad | < 50 krad | < 30 krad | 1 Mrad | 1 Mrad |
| Rango SEU | bajo-nulo | Alto | bajo-medio | Nulo | Nulo |
| Rango Temperatura | Alto | Industrial | Comercial | Alto | Alto |
| Potencia | 500 mW | 300 mW | 30 mW | 900 mW | 270 mW |

Tabla 2.1: Tipo de memorias usadas en cubesats

El cubesat cuenta con 2 memorias flash tipo NOR de 1GBit, cada una para almacenamiento de telemetría. Las cualidades por las cuales estas memorias fueron elegidas son: un rango de tolerancia a SEU aceptable, bajo consumo de energía, es memoria no volátil y están disponibles en el mercado.

2.4.1.3. Reloj en tiempo real

La computadora de vuelo cuenta con un reloj en tiempo real (RTC, por sus siglas en inglés) de grado automotriz y bajo consumo de potencia [11]. Este dispositivo tiene la finalidad de temporizar tareas como son el almacenado de telemetría o realizar la ejecución de un comando en un día y hora especificado. El RTC hace uso de tres capacitores en paralelo como fuente de respaldo para cuando existe perdida de energía, permitiéndole mantener la fecha sin modificaciones o retrasos.

2.4.2. Protocolos de Comunicación

Como ya se mencionó anteriormente, el cubesat usará como protocolo principal de comunicación entre subsistemas el bus I2C a una frecuencia de 400 KHz. Además de este protocolo, la computadora de vuelo hará uso de los siguientes protocolos de comunicación:

SPI: Creado por Motorola, el Serial Peripheral Interface [12] [13] es un protocolo de comunicación serie síncrono de alta velocidad. SPI requiere únicamente una línea de reloj (SCK), una línea seleccionadora por cada dispositivo conectado (CS) y dos líneas de datos. Entre sus características se encuentran: comunicación hasta 66 MHz, modificación de tamaño de palabra (2 a 16 bits), programación de fase y polaridad del reloj, etc.

SCI: Serial Communication Interface [14] es un puerto serie asíncrono que consta únicamente de dos líneas (transmisión y recepción) y es comúnmente conocido como UART. El módulo SCI soporta comunicaciones digitales entre dispositivos y usa un formato de NRZ o non-return-to-zero. Algunas de sus características son: baud rate programable, detección de errores, operaciones half o full duplex, etc.

1-Wire: Creado por Maxim/Dallas Semiconductor, el protocolo 1-wire [15] permite la interacción con diversos dispositivos como medidores de baterías, memorias, diversos sensores, etc. El protocolo 1-wire fue diseñado para mantener una interfaz mínima debido a que opera con una única línea de datos referenciada a tierra. Con este protocolo un maestro (microcontrolador) puede obtener información de los diversos dispositivos o esclavos conectados al bus y, aunque el protocolo 1-wire no está embebido dentro de gran parte de los microcontroladores, este protocolo puede ser fácilmente desarrollado en cualquier pin de propósito general.

La figura 2.6 es una representación a bloques de los subsistemas satelitales y la manera en cómo interactúan con la computadora de vuelo. Como se puede observar, cada subsistema satelital cuenta con una unidad de procesamiento que le permite mantener comunicación con la computadora de vuelo mediante el bus I2C. Así mismo, existe un bus en común dedicado al protocolo de comunicación 1-wire, éste protocolo es usado por 20 sensores de temperatura dispersos en los subsistemas de computadora de vuelo, control y orientación, comunicaciones en UHF/VHF y potencia eléctrica. Los sensores de temperatura fueron colocados en componentes específicos y el microcontrolador de la computadora de vuelo se encargará de recolectar toda la información de éstos.

El puerto serie SCI o UART está dedicado únicamente a su uso con el módulo de banda S y finalmente el protocolo de comunicación SPI es usado internamente en la computadora de vuelo para su uso con las 2 memorias flash y el RTC.

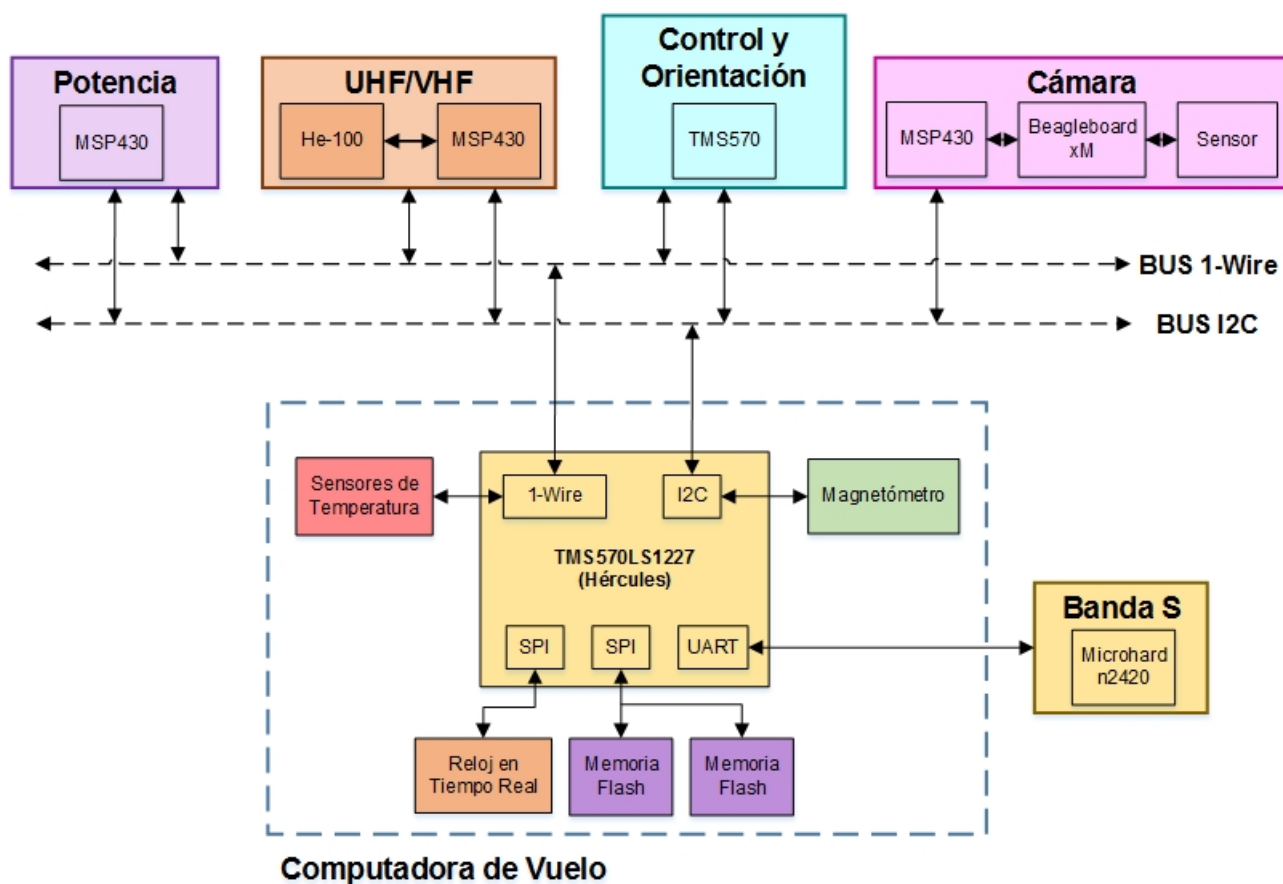


Figura 2.6: Protocolos de Comunicación

Para una mejor comprensión del subsistema de computadora de vuelo, se agregó en el apéndice A del presente trabajo de tesis un diagrama a bloques que contiene componentes principales, conexiones eléctricas y el conector

que permite interoperabilidad con los demás subsistemas. Así mismo, la tabla 2.2 describe las especificaciones de la computadora de vuelo y la figura 2.7 muestra el ensamble final de la tarjeta electrónica.

| | |
|----------------------------|----------------------------------|
| Material y Dimensiones | PC/104, 93 x 95 cm |
| Temperatura de operación | entre -40°C y 80°C |
| Interfaz de programación | JTAG |
| Memoria | 2 GBits Flash NOR |
| Protección | Protección contra sobrecorriente |
| Frec. de operación | 20 MHz |
| Voltaje de nominal | 3.3 Volts |
| Corriente en standby | 80 mA |
| Protocolos de comunicación | I2C,SPI,UART y 1-Wire |

Tabla 2.2: Especificaciones de la computadora de vuelo

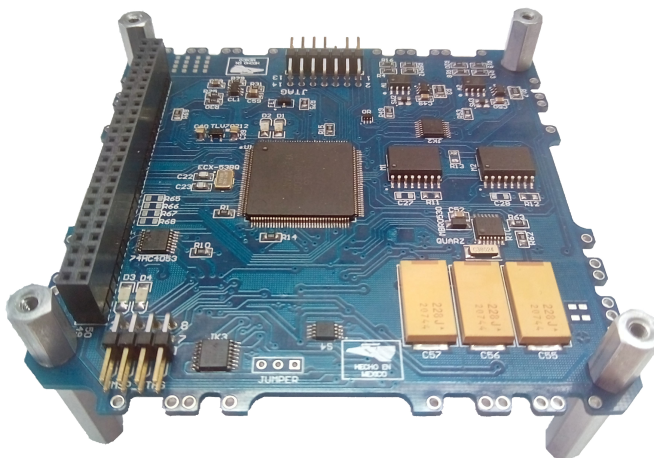


Figura 2.7: Tarjeta electrónica de computadora de vuelo

2.5. Cámara Satelital

La carga útil del cubesat corresponde a una cámara CMOS IDS UI-1220SE [16] y su objetivo principal es la captura de imágenes de la Tierra. La cámara cuenta con un sensor CMOS monocromático, con una resolución de 752 x 480 pixeles (0.36 Mpx), y un filtro infrarrojo en su lente Xenoplan, con una longitud focal de 34.9 mm. La cámara UI-1220SE cuenta con una interfaz USB 2.0 y por tal motivo cuenta con dos tarjetas electrónicas, primero una tarjeta comercial Beagleboard-XM obtiene los datos de la cámara vía USB, después estos datos pasan a la segunda tarjeta que contiene un MSP430 vía UART y finalmente los datos son enviados a la computadora de vuelo del satélite por medio del bus I2C.

La figura 2.8 muestra, del lado izquierdo, la cámara IDS UI-1220SE con sus respectivos soportes mecánicos y, del lado derecho, la tarjeta comercial Beagleboard-XM conectada con la tarjeta electrónica que contiene el microcontrolador MSP430.

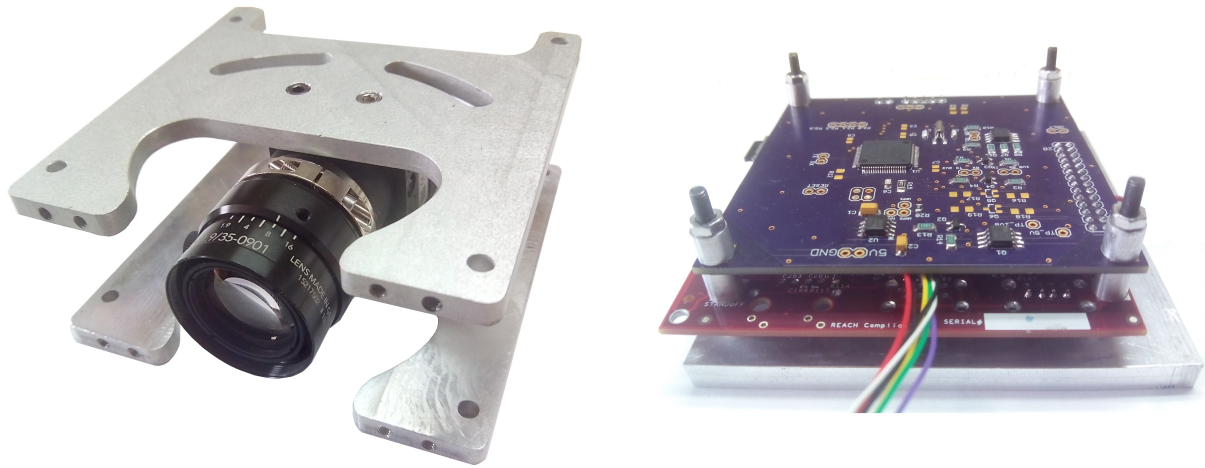


Figura 2.8: Cámara IDS UI-1220SE y tarjetas electrónicas

Capítulo 3

Diseño e Implementación de Software

El primer paso en el análisis y diseño de una misión espacial es definir los objetivos de la misión, esto es, definir los objetivos generales que el sistema debe lograr para ser productivo [17]. El equipo de trabajo del Instituto de Ingeniería UNAM se encargó del diseño del subsistema de potencia eléctrica, computadora de vuelo y subsistema de control y orientación. Como se mencionó anteriormente, cada subsistema cuenta con su propio procesador, permitiendo generar tareas específicas por cada subsistema, pero siendo comandados por la computadora de vuelo. La imagen 3.1 muestra las tareas específicas que realiza cada subsistema satelital.

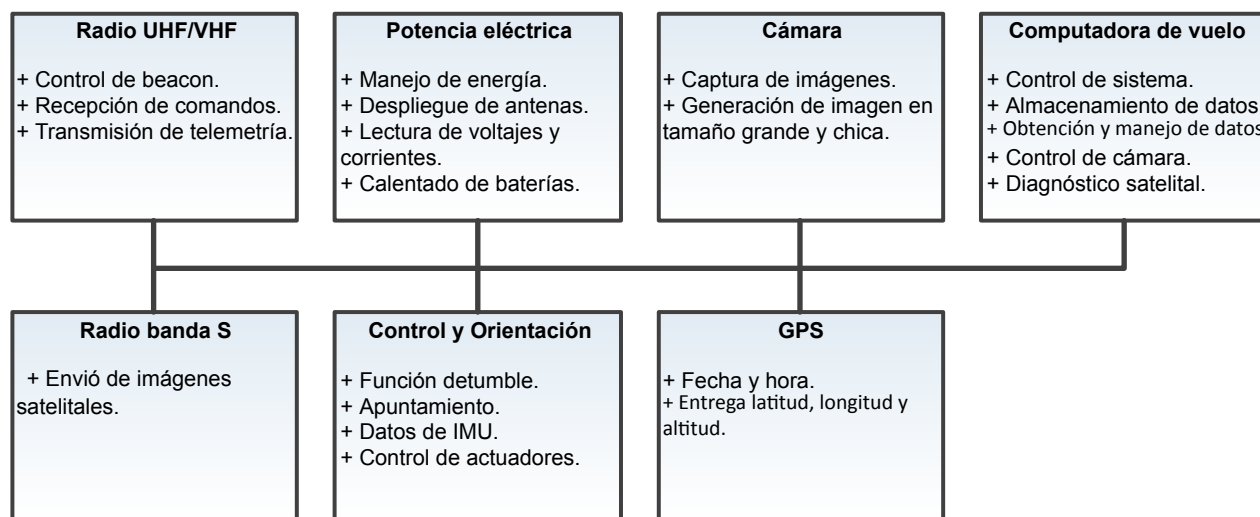


Figura 3.1: Funciones por subsistema satelital

La computadora de vuelo es responsable de todas las actividades generadas por el cubesat y debido a que operará en un ambiente espacial, es de suma importancia que cuente con gran confiabilidad y robustez. Aunque cada subsistema satelital genera datos de telemetría, la computadora de vuelo será responsable de coleccionar y almacenar toda la información para su posterior descarga en línea de vista con la estación terrena. El diseño y los requerimientos de software de la computadora de vuelo, fueron divididos en tres grupos que son: tareas y manejo de datos, comandos operativos y diagnóstico satelital. Este capítulo contiene la descripción de cada grupo y finalmente presenta la plataforma de desarrollo de software y lenguaje de programación usados para su implementación.

3.1. Tareas y manejo de datos

Esta sección tiene por objetivo describir a detalle cada una de las tareas específicas a realizar por la computadora de vuelo. Dentro de las tareas se encuentran uso y manejo del bus I2C, descripción de la estructura de comandos internos entre subsistemas, obtención de temperatura satelital, uso del reloj en tiempo real, uso de magnetómetro triaxial, descripción de comunicación con el módulo de banda S, uso y organización de memorias flash para almacenamiento de telemetría y el control de la cámara.

3.1.1. Protocolo I2C

Cuando un cubesat hace uso del bus I2C, generalmente la computadora de vuelo es el dispositivo maestro y tiene el control sobre los demás subsistemas (esclavos). En un sistema [18], donde el dispositivo está corriendo múltiples tareas, es importante asegurar que solo una tarea o comando acceden al bus. Un nuevo dato enviado antes que el dato previo haya sido completado, puede resultar en información corrupta. Si el dato corrupto no es detectado, puede crear un mal funcionamiento en el sistema dejando al satélite en un estado desconocido.

Para poder comprender los posibles errores y la manera de mitigarlos, es necesario primero entender la manera en cómo opera el bus I2C, los términos usados y sus principales señales[19]. La figura 3.2 representa una transferencia completa de datos en el bus I2C.

Maestro: El maestro es el dispositivo que inicia el mensaje y define la dirección del bus. El maestro es también el responsable de generar la señal de reloj (SCL).

Esclavo: Dispositivo con una dirección única dentro del bus.

Start: Condición del bus en la cual la línea SCL está en alto y la línea SDA hace una transición de un estado alto a bajo. Este es el inicio de cualquier operación y siempre es seguida por una dirección de esclavo. Con el siguiente bit, el maestro determina si se trata de lectura o escritura de datos.

Stop: Condición opuesta a Start en la cual la línea SCL está en alto mientras la línea SDA hace una transición de un estado bajo a alto.

Tamaño de palabra: El tamaño de palabra o dato es siempre de 8 bits.

Longitud de mensaje: No existe una longitud máxima de mensaje.

Acknowledge: "ACK" es la condición en la cual el maestro genera un 9^a pulso de reloj en la línea SCL mientras el dispositivo receptor pone la línea SDA en un estado bajo, significando que el último byte ha sido recibido. Un NACK es generado únicamente por el dispositivo maestro y señala que ningún dato adicional necesita ser enviado.

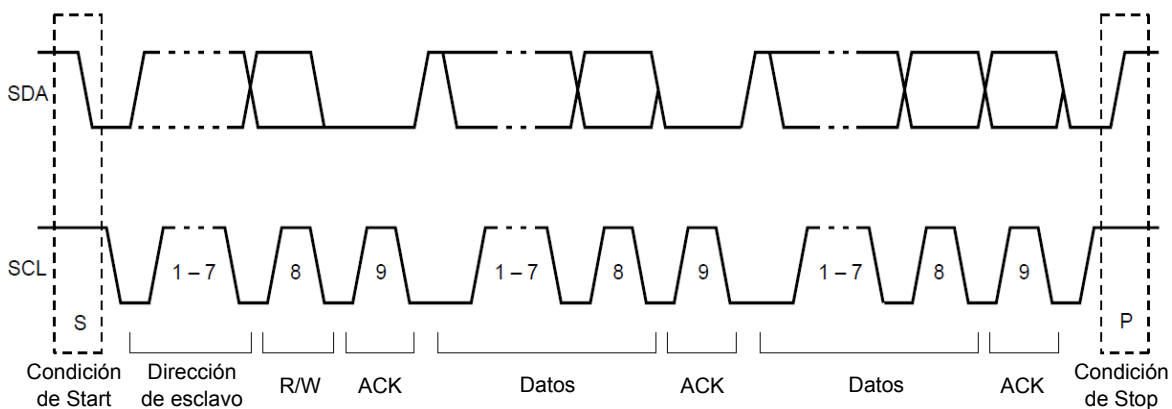


Figura 3.2: Transferencia de datos en bus I2C

Hay ocasiones en que el dispositivo maestro es interrumpido en medio de una comunicación con un esclavo por el bus I2C, generando un estado donde el bus queda atascado. Inicialmente se pensaría que se debe a un problema en el dispositivo pero no lo es. Esto se debe a que existen casos donde el dispositivo maestro (microcontrolador) sufre un extraño reset y es interrumpido en medio de una comunicación con el esclavo. Cuando esto ocurre, el esclavo se mantiene esperando la petición de un dato nuevo pero el maestro olvida donde se encontraba después del reset. Hasta este punto, el dispositivo esclavo pone el siguiente valor de bit en SDA y espera la siguiente señal de reloj en SCL para así poder mandar un bit nuevo. Por supuesto, el maestro no envía la señal de reloj y el esclavo se

quede esperando y esperando. Si el bit que puso el esclavo en SDA es igual a 0, el maestro interpreta que el bus está ocupado o en un modo no operativo.

Para mitigar este error o evitar que el bus I2C entre en un modo no operativo, la figura 3.3 muestra el modo de operación usado por la computadora de vuelo. La imagen muestra un ejemplo donde la computadora de vuelo enviará 10 bytes a cualquier subsistema dentro del bus I2C. En este caso, el microcontrolador continuamente verifica los estados y señales del bus I2C. Paralelo a esto, el MCU corre un timer o reloj interno definido a 80 milisegundos y espera que se cumpla la condición de buffer vacío (acknowledge), bus desocupado y señal de stop. Cuando alguna de estas condiciones no se da dentro de la transmisión, el MCU es interrumpido por la señal del timer e identifica que existió un problema. Como puede observarse, al inicio de cada transferencia de datos, el MCU pone el periférico en un estado de reset, por si es que el bus se encontraba en un modo no operativo.

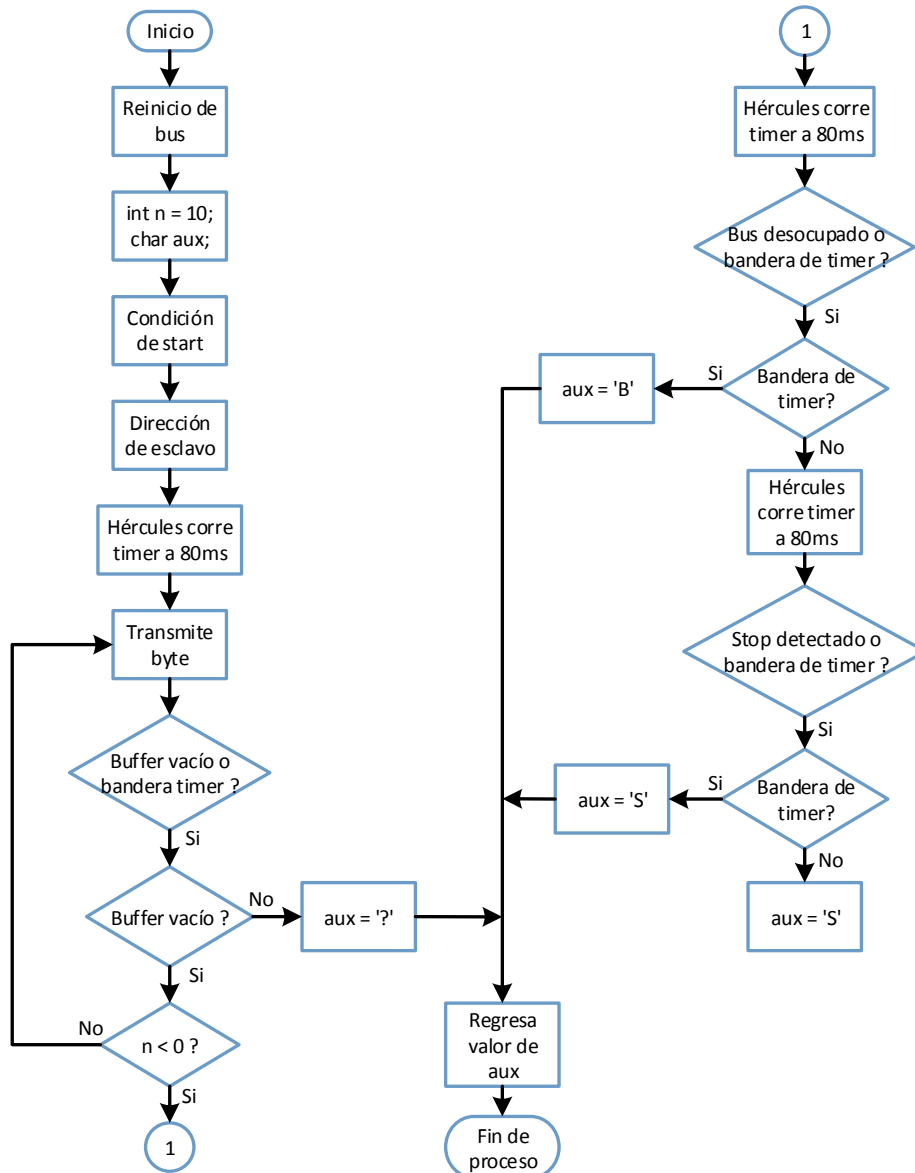


Figura 3.3: Modo de operación de bus I2C para evitar errores

3.1.1.1. Estructura de comandos internos

Todos los subsistemas contenidos dentro del cubesat, a excepción de banda S, cuentan con un microcontrolador conectado al bus I2C. Como se describió anteriormente, el bus I2C opera con un maestro y varios esclavos conectados al bus. La velocidad de transmisión fue fijada a 400 kbits por segundo y cada subsistema cuenta con una dirección de esclavo única, permitiendo a la computadora de vuelo direccionar e iniciar una transferencia de datos en cualquier momento. La figura 3.4 representa las conexiones eléctricas entre los subsistemas conectados al bus I2C.

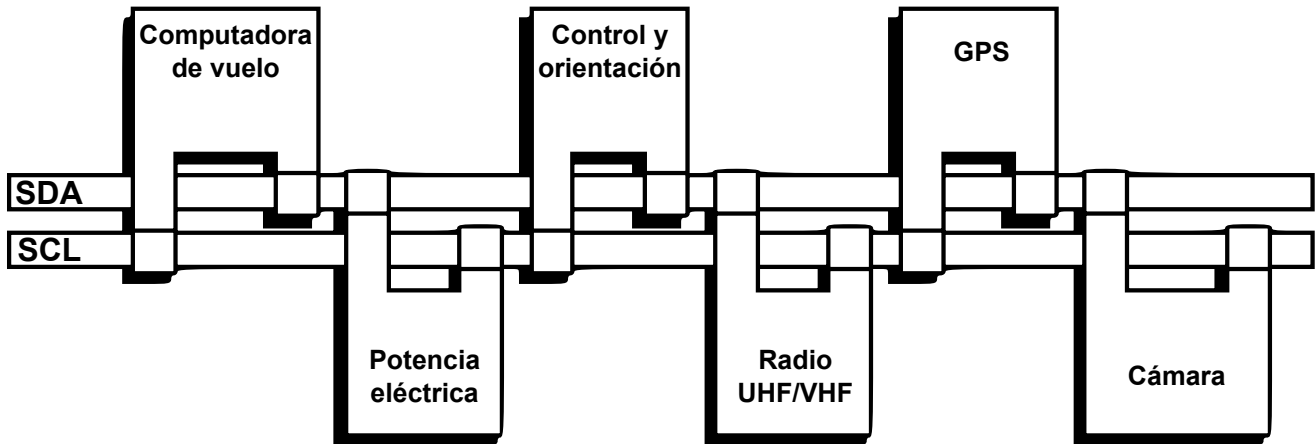


Figura 3.4: Subsistemas conectados al bus I2C

Con el objetivo de garantizar la transferencia de datos entre subsistemas, el cubesat cuenta con una estructura de comandos internos definido. La trama es de longitud variable y contiene un campo de encabezado, número de datos, un campo para número de comando, un campo de datos y un campo para verificación por redundancia cíclica(CRC). La tabla 3.1 muestra la estructura de los comandos internos.

| Campo | Encabezado | Núm. de datos | Núm. de comando | Datos | CRC |
|---------------|------------|---------------|-----------------|---------|-----|
| Núm. de bytes | 2 | 1 | 1 | 1 - 255 | 2 |

Tabla 3.1: Estructura de comandos internos entre subsistemas

Encabezado: El encabezado es el inicio de la trama y corresponde a dos bytes identificadores en formato ASCII.

Número de datos: Indica el tamaño total de la trama o mensaje. De esta manera, el dispositivo receptor sabe el número exacto de datos a recibir y a su vez es necesario para el cálculo del CRC.

Número de comando: Debido a que cada subsistema es capaz de ejecutar tareas o generar datos de telemetría, este campo define a los subsistemas a los que el comando debe ser ejecutado.

Datos: Este campo contiene los datos de la trama o el argumento del mensaje. Para este campo no existe un valor máximo de datos, sin embargo se estableció un máximo de 255 bytes.

CRC: El campo de verificación por redundancia cíclica es un código detector de errores. El CRC es generado y agregado a la trama por el dispositivo que envía información, después el dispositivo receptor obtiene el mensaje y vuelve a generar el CRC para ser comparado con el CRC recibido. Cuando no existen coincidencias entre los dos CRC, el dispositivo receptor detecta información errónea y descarta el mensaje.

3.1.2. Comunicación con banda S

El protocolo de comunicación usado entre el módulo de banda S y la computadora de vuelo es la interfaz de comunicación serie o SCI [20]. El puerto de comunicación SCI se caracteriza por ser un protocolo muy fácil de imple-

mentar y debido a que corresponde a una comunicación serial asíncrona, no necesita de una línea de reloj externa. Este protocolo de comunicación hace uso de dos líneas que son entrada y salida de datos y la comunicación se logra fijando el mismo valor de baud rate entre los dispositivos a comunicarse. La imagen 3.5 muestra la recepción de datos con el uso del bus SCI. Algunos de los términos usados en el protocolo de comunicación SCI son los siguientes:

Start bit: Indica el inicio de una palabra o dato.

Stop bit: Indica el final de una palabra o dato.

Parity bit: Agregado para detección de error (Opcional).

Data bits: Dato o palabra de 8 bits a ser transmitido.

Baud rate: Tasa de transferencia del puerto serial.

Throughput: Datos reales transmitidos por segundo.

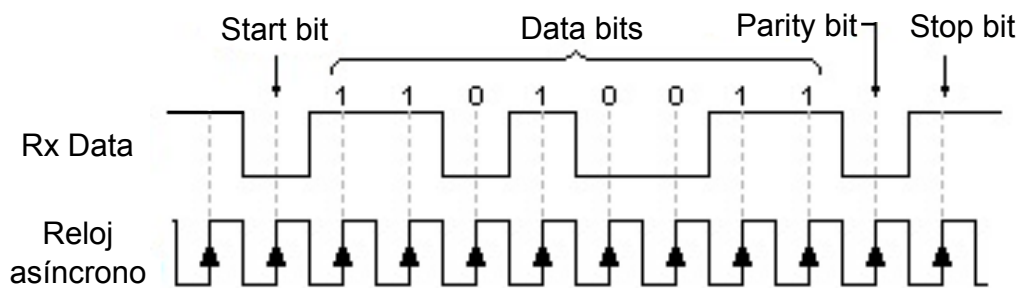


Figura 3.5: Transferencia de datos en bus SCI

La configuración usada por el módulo de banda S corresponde a la siguiente:

- Velocidad de interfaz serial: 57600 bps.
- Tasa de transmisión inalámbrica para enlace con estación terrena: 115200.
- Potencia de transmisión: 30 dBm.
- Tamaño máximo de paquetes: 250 bytes
- Bits de stop: 1.
- Uso de bit de paridad: No

Una vez definidos los parámetros del módulo de banda S, el periférico SCI del microcontrolador de la computadora de vuelo fue programado con los mismos parámetros. Ya que este protocolo de comunicación es más simple de implementar y que la comunicación serial solo se da entre dos dispositivos, el microcontrolador puede detectar fácilmente errores presentes sobre el bus. En este caso no fue necesario implementar un modo de operación para evitar errores como en el caso del bus I2C. El objetivo del radio de banda S es únicamente el envío de imágenes a la tierra por lo que no habrá recepción de datos en el módulo. La estructura de datos enviados al módulo de banda S serán descritos en la sección de comandos operativos.

3.1.3. Obtención de temperatura

Una parte importante dentro de la telemetría satelital, es conocer la temperatura de ciertos componentes y del satélite en general. Para este objetivo, se hizo uso de los sensores de temperatura DS18B20[21] por Maxim Semiconductors los cuales hacen uso del protocolo 1-wire.

El termómetro digital DS18B20 proporciona mediciones de temperatura en grados Celsius con una resolución de 9

a 12 bits y su rango de operación es de $-55\text{ }^{\circ}\text{C}$ a $125\text{ }^{\circ}\text{C}$ con una precisión de $\pm 0.5\text{ }^{\circ}\text{C}$ sobre rangos de $-10\text{ }^{\circ}\text{C}$ a $85\text{ }^{\circ}\text{C}$. A continuación se presenta la configuración de los sensores, descripción del bus 1-wire e implementación del protocolo.

3.1.3.1. Configuración de sensores de temperatura

Una de las características del sensor de temperatura DS18B20 es que puede operar sin la necesidad de una fuente de alimentación externa en modo "parasite power". Esta habilidad es de gran utilidad para aplicaciones que requieren sensado de temperatura remoto o donde existe espacio reducido. La energía necesaria para que el sensor opere es obtenida de la línea de datos. Cuando la línea se encuentra en un estado alto, parte de la carga es almacenada en un capacitor interno que provee energía en caso que el bus esté en un estado bajo. Para asegurar que existe suficiente corriente para todos los sensores conectados al bus, es necesario el uso de un MOSFET. La figura 3.6 muestra la conexión eléctrica usada por los sensores de temperatura.

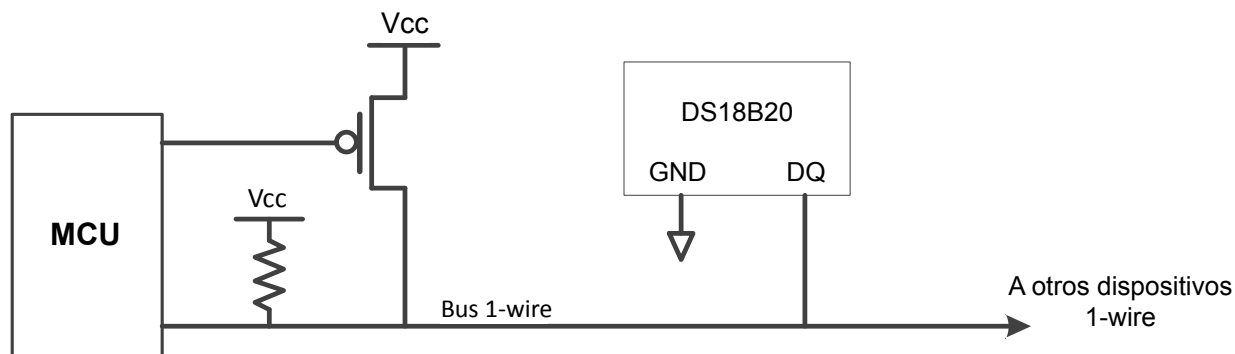


Figura 3.6: Configuración de sensores de temperatura

La configuración antes mostrada, permite a la computadora de vuelo obtener la temperatura de todos los sensores conectados al bus 1-wire aunque éstos se encuentren en un subsistema apagado. Esto se logra ya que los sensores son energizados desde la línea de datos y no localmente.

El cubesat cuenta con 20 sensores de temperatura, todos colocados dentro de los subsistemas diseñados por el equipo de trabajo del Instituto de Ingeniería UNAM. Cada sensor DS18B20 contiene un código único de 64 bits para poder ser identificado dentro del bus 1-wire, permitiéndole al microcontrolador obtener la temperatura de un sensor en específico. La tabla 3.2 presenta el número de sensores colocados por subsistema y su ubicación.

| Subsistema | Núm. sensores | Ubicación |
|-----------------------|---------------|--|
| Potencia eléctrica | 6 | Baterías y microcontrolador |
| Computadora de vuelo | 4 | Microcontrolador y memorias flash |
| Control y orientación | 8 | Microcontrolador, IMU y driver de actuadores |
| Comunicaciones | 2 | Microcontrolador |

Tabla 3.2: Ubicación de sensores de temperatura

3.1.3.2. Descripción de bus 1-wire e implementación de protocolo

Como se mencionó en el capítulo dos, hasta el momento no existe un microcontrolador con el bus 1-wire embebido dentro de su arquitectura, pero el protocolo puede ser fácilmente implementado siempre y cuando el sistema sea capaz de generar pulsos de $1\mu\text{s}$ de forma repetida y precisa. El sensor de temperatura hace uso de cuatro operaciones básicas que son: reset, escribe bit 1, escribe bit 0 y lee bit. A su vez, la función byte es una construcción basada en la operación de escribir bits 1 y 0. La figura 3.7 muestra las formas para cada operación básica y los tiempos necesarios que debe generar el sistema o microcontrolador para poder mantener una comunicación con los sensores conectados al bus.

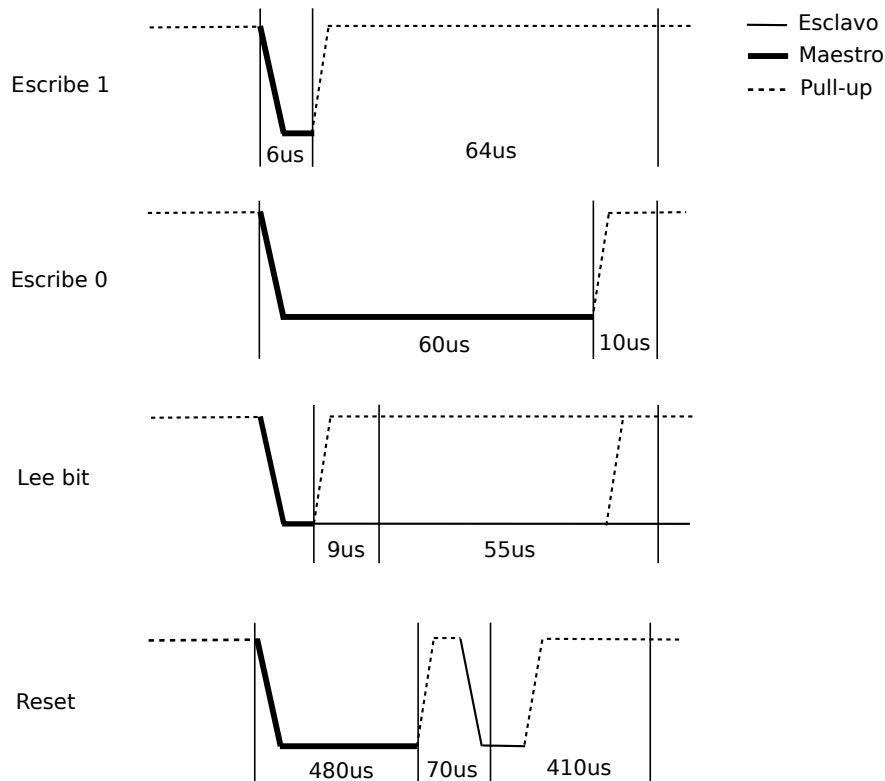


Figura 3.7: Operaciones básicas de bus 1-wire

Una vez que el microcontrolador es capaz de generar las operaciones básicas, existe una serie de comandos que permite conocer la temperatura de los sensores o modificar su configuración. A continuación se describen los comandos usados y una breve descripción de éstos.

Inicialización: Cualquier transferencia de datos con los dispositivos 1-wire empiezan con una secuencia de inicio. La secuencia de inicio consiste en un pulso de reset transmitido por el maestro seguido de un pulso de presencia transmitido por el esclavo. El pulso de presencia avisa al dispositivo maestro que existe un dispositivo en el bus y está listo para operar.

Match rom[55h]: El comando match rom seguido de un código único de 64 bits permite al dispositivo maestro direccionar a un esclavo en específico. Únicamente el esclavo que coincida con el identificador enviado por el maestro responderá al siguiente comando emitido. Los demás dispositivos permanecerán a la espera de un nuevo pulso de reset.

Skip rom[CCh]: El maestro puede usar este comando para direccionar todos los dispositivos conectados al bus simultáneamente sin la necesidad de enviar un código único de 64 bits. Por ejemplo, el maestro puede hacer que todos los sensores realicen simultáneamente una conversión de temperatura usando el comando skip rom seguido del comando de conversión.

Convert t[44h]: Este comando inicia una conversión de temperatura única. Seguido de este comando, el sensor de temperatura almacenará el resultado en un registro de 2 bytes en su memoria y después regresará a un estado de reposo o bajo consumo. Después de emitido el comando, el dispositivo maestro debe activar el MOSFET por al menos $10\mu s$ para lograr una conversión de temperatura correcta.

Lectura de scratchpad[BEh]: Este comando permite al dispositivo maestro leer el contenido del scratchpad. Este registro contiene la última lectura de temperatura registrada por el sensor.

La imagen 3.8 muestra el diagrama de flujo o secuencia de pasos, seguidos por el microcontrolador para obtener la lectura de temperaturas de todos los sensores dispersos en el satélite. El diagrama de flujo no describe el proceso de obtención de ID's de los sensores. La función y algoritmo usado es descrita en el documento "1-wire search algorithm"[22] y hace uso de las funciones básicas previamente descritas.

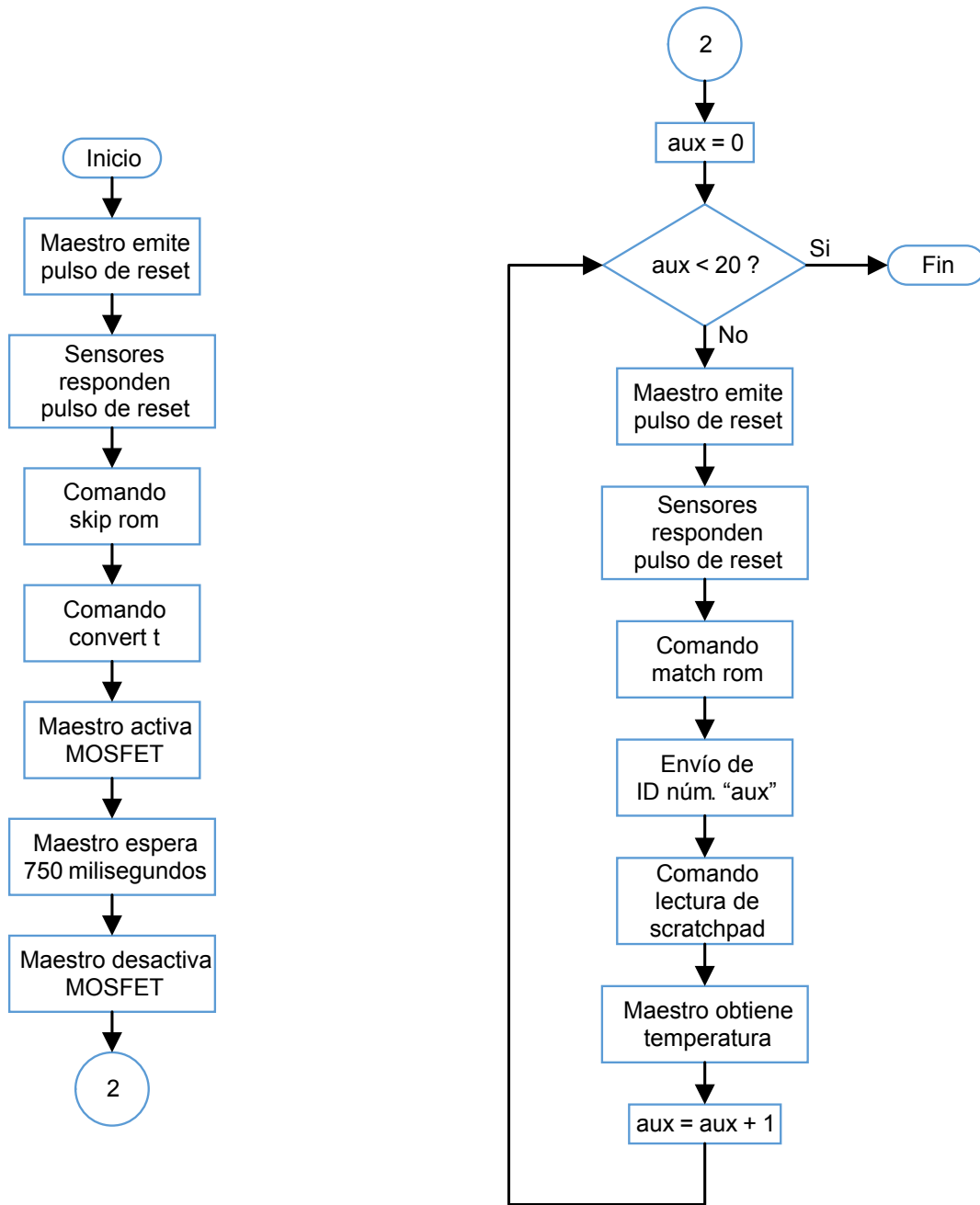


Figura 3.8: Diagrama de flujo para lectura de temperaturas

3.1.4. Uso de reloj en tiempo real

El PCA21125 [23] es un dispositivo CMOS correspondiente a un reloj en tiempo real y calendario optimizado para un bajo consumo de potencia y un rango de temperatura de hasta 125°C. La transferencia de datos se logra mediante la interfaz serial SPI con una tasa máxima de 6 Mbits/s. A su vez cuenta con funciones de alarma y un

timer con la posibilidad de generar una interrupción en un pin dedicado. Dentro de las características y beneficios del circuito integrado se tienen:

- Dispositivo de grado automotriz.
- Proporciona datos de año, mes, día, día de la semana, hora, minutos y segundos basado en su cristal de cuarzo a 32.768 Hz.
- Bajo consumo de corriente: valor típico a 0.8 μ A.
- Tiempo, fecha y alarma en formato decimal codificado en binario (BCD, por sus siglas en inglés).

El apéndice B del presente trabajo de tesis presenta el diagrama a bloques del reloj en tiempo real. El reloj en tiempo real será usado por la computadora de vuelo como dispositivo temporizador de tarea y ejecución de comandos. La tabla 3.3 muestra los 16 registros de control del reloj en tiempo real configurables por la computadora de vuelo. Los 16 registros fueron diseñados en palabras de 8 bits y su descripción es la siguiente:

- Los dos primeros registros 00h y 01h son usados como registros de control y estatus del dispositivo.
- Registros con dirección del 02h al 08h (segundos, minutos, día, día de la semana, mes y año) son usados como contadores de la función de reloj. Todos los registros están codificados en formato BCD.
- Registros con dirección del 09h al 0Ch definen las condiciones de alarma.
- El registro 0Dh define el modo de salida del reloj interno.
- Registros con dirección del 0Eh al 0Fh son usados para la función del conteo de timer. La función de timer cuenta con 4 fuentes de reloj seleccionables permitiendo cuatro periodos de conteo en un rango menor que 1 milisegundo a más de cuatro horas.

| Registro | Nombre |
|--|----------------------------|
| Registros de control y status | |
| 00h | Control 1 |
| 01h | Control 2 |
| Registros de tiempo y fecha | |
| 02h | Segundos |
| 03h | Minutos |
| 04h | Horas |
| 05h | Días |
| 06h | Día de la semana |
| 07h | Mes |
| 08h | Año |
| Registros de alarma | |
| 09h | Alarma de minuto |
| 0Ah | Alarma de hora |
| 0Bh | Alarma de día |
| 0Ch | Alarma de día de la semana |
| Registro de control del reloj de salida | |
| 0Dh | Control de reloj |
| Registros de timer | |
| 0Eh | Control de timer |
| 0Fh | Conteo descendente |

Tabla 3.3: Tabla de registros de reloj en tiempo real

La imagen 3.9 muestra el diagrama de flujo para configurar la fecha, hora, alarma y timer del reloj en tiempo real. Al inicio de cualquier configuración, el microcontrolador debe seleccionar el dispositivo mediante la línea "chip select" del bus SPI. Una vez seleccionado, se tiene 3 posibles opciones:

En caso de seleccionar actualización de fecha y hora, la computadora de vuelo escribirá datos de fecha y hora

obtenidos directamente del GPS en un formato UTC.

En caso de seleccionarse ajuste de alarma, el microcontrolador debe definir mes, día, hora y minuto para que el reloj en tiempo real genere una señal de interrupción en ese momento.

En caso de seleccionar ajuste de timer, el microcontrolador seleccionara la fuente de reloj a 1/60 Hz, permitiendo un intervalo de tiempo entre 1 minuto y hasta 4 horas. Enseguida se definirá el valor de conteo descendente en el cual el dispositivo generará una señal de interrupción que será leída por el microcontrolador.

Finalmente, para cualquiera de los tres casos, el microcontrolador deshabilitará la escritura de registros y configuración del dispositivo mediante la línea "chip select" del bus SPI.

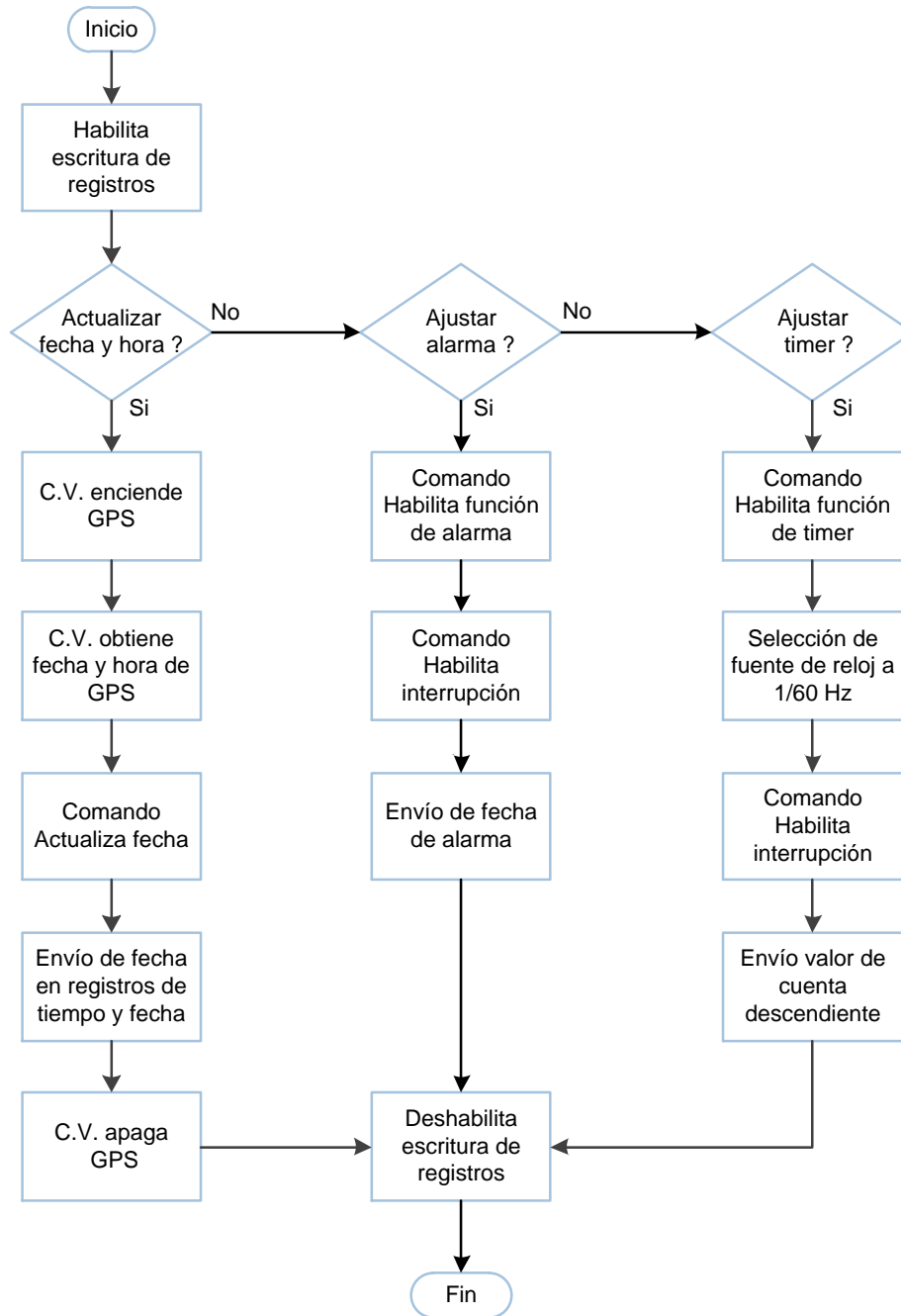


Figura 3.9: Diagrama de flujo para configuración de reloj en tiempo real

3.1.5. Magnetómetro triaxial

La computadora de vuelo cuenta con un magnetómetro triaxial Honeywell HMC5883 [24] para el sentido de campos magnéticos. El sensor fue diseñado para medir dirección y magnitud de los campos magnéticos de la tierra con un intervalo de miligauss a 8 gauss gracias a su construcción en estado sólido. Dentro de sus características se tienen:

- Convertidor analógico digital con resolución de 12 bits y bajo ruido.
- Bajo consumo de corriente. Valor típico a 100 μ A.
- Interfaz I2C.
- Amplio rango de campo magnético.

La tabla 3.4 muestra los 13 registros de control del magnetómetro, configurables por la computadora de vuelo y en palabras de 8 bits.

| Registro | Nombre | Acceso |
|----------|------------------------------|-------------------|
| 00 | Configuración de registro A | Lectura/Escritura |
| 01 | Configuración de registro B | Lectura/Escritura |
| 02 | Registro de modo | Lectura |
| 03 | Registro de salida MSB eje X | Lectura |
| 04 | Registro de salida LSB eje X | Lectura |
| 05 | Registro de salida MSB eje Y | Lectura |
| 06 | Registro de salida LSB eje Y | Lectura |
| 07 | Registro de salida MSB eje Z | Lectura |
| 08 | Registro de salida LSB eje Z | Lectura |
| 09 | Registro de estatus | Lectura |
| 10 | Identificador A (ASCII 'H') | Lectura |
| 11 | Identificador B (ASCII '4') | Lectura |
| 12 | Identificador C (ASCII '3') | Lectura |

Tabla 3.4: Tabla de registros de magnetómetro Honeywell

El magnetómetro cuenta con tres modos de operación: modo medición continua, modo medición única y modo inactivo. La escritura en el registro de modo (02), seleccionará entre los tres diferentes modos de operación. La computadora de vuelo hará uso del modo medición única.

Los registros de configuración A y B definen parámetros como: definición de tasa de muestreo para el modo medición continua, número de muestras promedio por cada medición y configuración de ganancia. La modificación de estos parámetros permitirán a la computadora de vuelo realizar lecturas de campos magnéticos en un intervalo de ± 1.3 gauss.

Los registros con dirección del 03 al 08, pertenecen a registros de solo lectura y corresponden a los valores generados por los tres ejes. Los datos de salida son dos registros de 8 bits. Para cada eje, el registro MSB contiene el valor de los bits más significativos y el registro LSB contiene el valor de los bits menos significativos. La unión de estos registros corresponden a palabras de 16 bits en complemento a dos.

El registro de estatus contiene un bit que es usado para indicar cuando una medición fue completada y finalmente, los registros de identificación, corresponden a tres valores en formato ASCII ('H43') que permiten verificar una correcta comunicación entre el dispositivo maestro y esclavo sobre el bus I2C.

La imagen 3.10 muestra el procedimiento que sigue la computadora de vuelo para obtener lecturas de campos magnéticos con el uso de magnetómetro HMC5883. En un principio, el microcontrolador debe configurar dentro del registro A y B el modo de medición normal, seleccionar 8 muestras promedio por medición y definir el valor de ganancia. En el registro de modo, el microcontrolador debe definir un modo de medición único y una vez hecho esto,

debe esperar 6 milisegundos a que se realice la medición para posteriormente, obtener el valor de todos registros correspondientes al campo magnético en el eje X, Y y Z. Para realizar una nueva lectura, es necesario que el microcontrolador configure nuevamente los 3 registros de control antes mencionados.

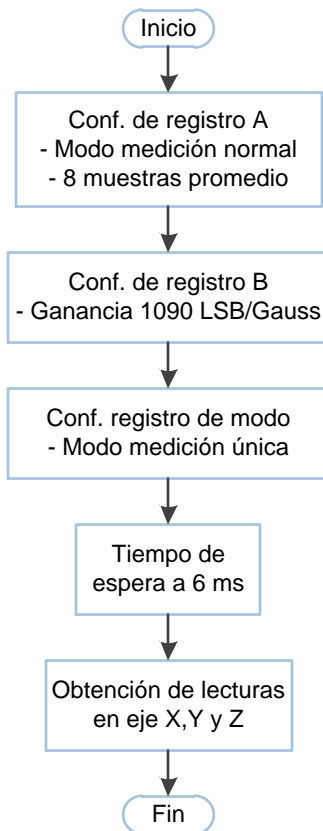


Figura 3.10: Diagrama de flujo para lectura de campos magnéticos

3.1.6. Uso y organización de memoria flash

La computadora de vuelo cuenta con memoria flash no volátil para almacenamiento de telemetría, control de tareas y ejecución de comandos. Para el almacenamiento de telemetría cuenta con dos memorias flash tipo NOR con capacidad de 1 Gbit o 125 MBytes cada una. Para el control de tareas y ejecución de comandos hará uso de 64 kB de memoria flash interna contenida dentro de su microcontrolador. A continuación se dará una breve descripción de los dos tipos de memoria usadas y se detallará la forma en cómo fueron organizadas para el almacenado de datos.

3.1.6.1. Memoria flash tipo NOR para almacenado de telemetría

La memoria flash tipo NOR N25Q00AA [25] cuenta con una interfaz serial SPI con una velocidad máxima de transferencia de datos de 108 MHz. La memoria puede estar protegida en la escritura por software mediante su característica de protección volátil y no volátil.

La memoria es un dispositivo compuesto por 4 chips de 256 Mbits. Cada chip es a su vez particionado internamente en dos segmentos de 128 Mbits. Cada página de memoria puede ser programada individualmente. El dispositivo puede ser borrado por subsector, sector o un único chip de 256 Mbits. La memoria está configurada como 134,217,728 bytes; 2048 sectores (64kB por sector); 32,768 subsectores (4kB por subsector); y 524,288 páginas (256 bytes por página). La tabla 3.5 muestra el mapa de memoria del dispositivo flash, donde cada sector corresponde a 16 subsectores y un subsector corresponde a 16 páginas.

Dentro de las características de las memorias flash se encuentran dos modos de direccionamiento. El modo de direccionamiento a 3 y 4 bytes. La computadora de vuelo implementará el modo a 3 bytes en una memoria para

almacenado de telemetría y el modo a 4 bytes para almacenado de imágenes grandes.

| Sector | Subsector | Dirección inicio | Dirección fin |
|--------|-----------|------------------|---------------|
| 2047 | 32767 | 07FFF000h | 07FFFFFFh |
| | : | : | : |
| | 32750 | 07FF0000h | 07FF0FFFh |
| : | : | : | : |
| 1023 | 16383 | 03FF F000h | 03FF FFFFh |
| | : | : | : |
| | 16368 | 03FF 0000h | 03FF 0FFFh |
| : | : | : | : |
| 511 | 8191 | 01FF F000h | 01FF FFFFh |
| | : | : | : |
| | 8176 | 01FF 0000h | 01FF 0FFFh |
| : | : | : | : |
| 255 | 4095 | 00FF F000h | 00FF FFFFh |
| | : | : | : |
| | 4080 | 00FF 0000h | 00FF 0FFFh |
| : | : | : | : |
| 127 | 2047 | 007F F000h | 007F FFFFh |
| | : | : | : |
| | 2032 | 007F 0000h | 007F 0FFFh |
| : | : | : | : |
| 63 | 1023 | 003F F000h | 003F FFFFh |
| | : | : | : |
| | 1008 | 003F 0000h | 003F 0FFFh |
| : | : | : | : |
| 0 | 15 | 0000 F000h | 0000 FFFFh |
| | : | : | : |
| | 0 | 0000 0000h | 0000 0FFFh |

Tabla 3.5: Mapa de memoria externa

El modo de direccionamiento a 3 bytes permite lectura, escritura y borrado de datos, únicamente en un segmento de 128 Mbits de los ocho disponibles. Con esto, la computadora de vuelo logrará mantener la memoria dividida en ocho segmentos iguales, haciendo uso de un segmento diferente para los diversos tipos de telemetría que necesita almacenar. El modo de direccionamiento a 4 bytes permite funciones de lectura, escritura y borrado más allá de un segmento de 128 Mbits. Este modo permitirá que la computadora de vuelo opere la memoria flash como un único segmento de 1024 Mbits. La imagen 3.11 muestra los 4 segmentos usados en modo de direccionamiento a 3 bytes dentro de la memoria flash 2.

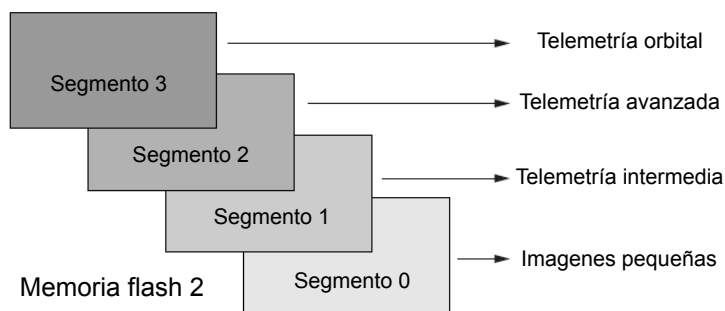


Figura 3.11: Uso de segmentos en memoria flash

Como se mencionó en el capítulo dos, el bus SPI es un protocolo de comunicación serie síncrono y requiere de una línea de reloj (SCK), una línea seleccionadora por cada dispositivo (CS) y dos líneas de datos. A continuación

se describen las señales eléctricas del bus SPI usadas por las dos memorias flash.

SCK: Señal de reloj que proporciona el tiempo de la interfaz serial.

CS: Cuando la línea de CS es puesta en un estado bajo, el dispositivo es seleccionado, permitiendo al dispositivo maestro hacer una transferencia de datos. Para un estado alto, la línea de datos DQ1 es puesta en alta impedancia y el dispositivo hará caso omiso a toda la información enviada sobre el bus SPI.

DQ0: Línea usada para la entrada de datos del dispositivo esclavo y salida de datos para el dispositivo maestro.

DQ1: Línea usada para la salida de datos del dispositivo esclavo y entrada de datos para el dispositivo maestro.

Vcc: Fuente de voltaje para energizar el dispositivo.

Vss: Tierra o señal de referencia para la fuente de voltaje Vcc.

W#: Protección de escritura, W# puede ser usada como un control de protección de escritura de datos.

HOLD#: Señal que permite pausar cualquier comunicación serie con el dispositivo sin ser deseleccionado.

La figura 3.12 muestra las conexiones eléctricas del bus SPI entre el microcontrolador y las dos memorias flash.

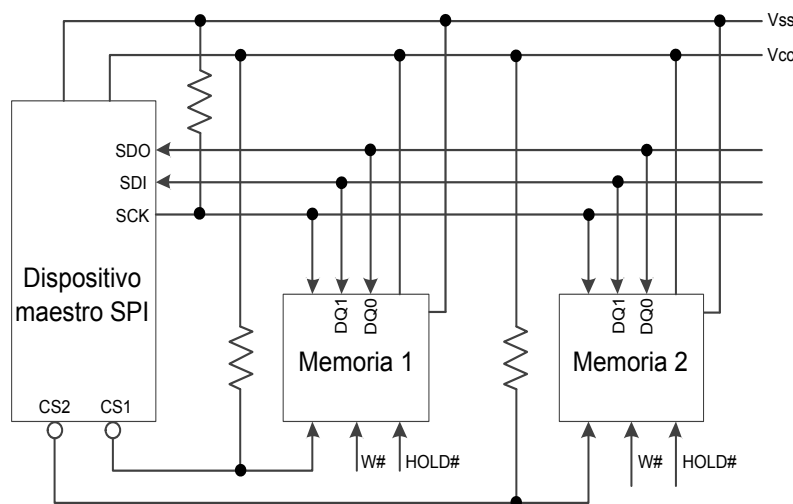


Figura 3.12: Dispositivo maestro y memorias flash en bus SPI

3.1.6.2. Memoria flash interna de MCU para control de tareas y/o comandos

La memoria flash en los microcontroladores de la serie Hércules [26] es memoria no volátil programable y borrrable eléctricamente (EEPROM, por sus siglas en inglés) para ejecución de instrucciones o almacenado de datos. Los microcontroladores Hércules contienen un módulo digital que maneja el acceso a la memoria flash y el microcontrolador puede soportar velocidades de reloj de hasta 180 MHz.

La arquitectura Hércules [27] incluye un banco de memoria flash separado con capacidad de 64K bytes para ser usado como emulación de flash EEPROM (FEE, por sus siglas en inglés). FEE es usado únicamente para el almacenado de datos y no puede ser usado como memoria para instrucciones del CPU. Dentro de las características de la memoria flash [28] se encuentran:

- Lectura, programación y borrado con una única fuente de 3.3V.
- Corrección de error único y detección de error doble (SECEDED, por sus siglas en inglés).
- Código corrector de errores (ECC, por sus siglas en inglés) para el banco de memoria principal.

- Proporciona diferentes modos de lectura para optimizar el rendimiento y verificar la integridad del contenido en la memoria.
- Integra una máquina de estado programación/borrado que permite simplificar algoritmos, lecturas simultaneas mientras se realiza escritura en otro banco de memoria, tiempos de programación y borrados rápidos, etc.

La tabla 3.6 muestra el mapa de memoria flash interna del microcontrolador Hércules.

| Banco de memoria | Sector No. | Segmento |
|--------------------|------------|------------|
| Banco de memoria 0 | 0 | 16k Bytes |
| | 1 | 16k Bytes |
| | 2 | 16k Bytes |
| | 3 | 16k Bytes |
| | 4 | 16k Bytes |
| | 5 | 16k Bytes |
| | 6 | 32k Bytes |
| | 7 | 128k Bytes |
| | 8 | 128k Bytes |
| | 9 | 128k Bytes |
| | 10 | 128k Bytes |
| | 11 | 128k Bytes |
| | 12 | 128k Bytes |
| | 13 | 128k Bytes |
| | 14 | 128k Bytes |
| | 15 | 128k Bytes |
| Banco de memoria 7 | 0 | 16k Bytes |
| | 1 | 16k Bytes |
| | 2 | 16k Bytes |
| | 3 | 16k Bytes |

Tabla 3.6: Mapa de memoria flash de microcontrolador Hércules

Como se puede observar, el banco de memoria 7, consta de cuatro segmentos de 16k bytes cada uno. Por simplicidad, este banco de memoria puede ser configurado para soportar hasta 16 bloques. Cada bloque puede ser considerado como un arreglo y cuando se realiza la configuración de la memoria, el usuario puede decidir el número de bloques requeridos. El mínimo número de bloques es uno y el máximo número dependerá del tamaño de bytes reservado para los bloques anteriores.

Este banco de memoria será usado por el microcontrolador Hércules únicamente para fines de control de tareas y ejecución de comandos. Básicamente, el microcontrolador configurará y dividirá su memoria flash en 6 bloques usados para ejecutar comandos operativos descritos en la sección 3.2.

3.1.7. Control de Cámara

La carga útil del satélite experimental fue diseñada por una dependencia del Instituto Politécnico Nacional y corresponde a una cámara CMOS IDS UI-1220SE [16] de uso industrial. Además de la cámara, el subsistema cuenta con una tarjeta comercial beagleboard-XM y una tarjeta electrónica que contiene un MSP430 de Texas Instruments. La cámara está conectada directamente a la tarjeta beagleboard-XM mediante una conexión USB 2.0 para tareas de control y configuración del sensor de imagen. El módulo completo está conectado al bus I2C mediante el microcontrolador MSP430, sin embargo, se desconoce la conexión eléctrica entre el MSP430 y la tarjeta beagleboard-XM. La imagen 3.13 muestra el diagrama a bloques entre las dos tarjetas electrónicas y la cámara.

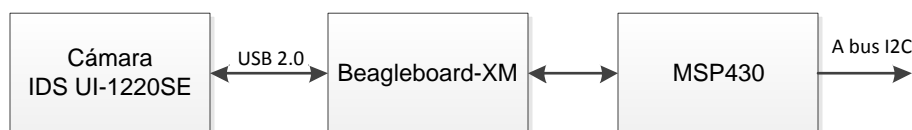


Figura 3.13: Diagrama de cámara satelital

La cámara satelital cuenta con 5 señales eléctricas que le permiten comunicarse con la computadora de vuelo. Dentro de las señales se encuentran 2 líneas de datos correspondientes al bus I2C, una línea de interrupción generada por la cámara cuando un dato se encuentra listo, una señal de reset y una señal de referencia (GND). La imagen 3.14 muestra las señales eléctricas utilizadas para la comunicación entre la computadora de vuelo y el módulo de la cámara.

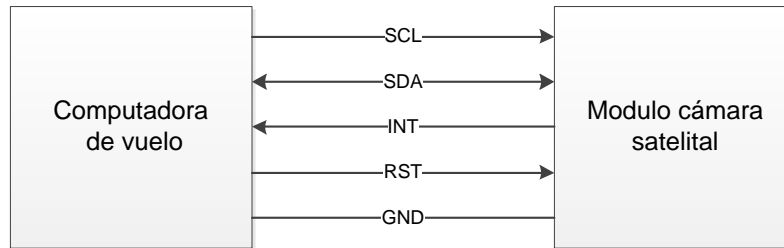


Figura 3.14: Conexiones eléctricas de cámara satelital

3.1.7.1. Interfaz de comandos

La cámara cuenta con un conjunto de registros de 16 bits de lectura y escritura que pueden ser modificados por la computadora de vuelo para tener control sobre la cámara. Para lograr la modificación de estos registros, la computadora de vuelo debe seguir una estructura de comandos, los cuales son paquetes de 12 bytes de longitud fija. La tabla 3.7 presenta la estructura de comandos y a continuación se describe cada uno de los campos.

| Campo | Encabezado | Código de mando | Registro | Reservado | CRC 1 | Argumento | CRC 2 |
|------------|------------|-----------------|----------|-----------|-------|-----------|-------|
| Núm. bytes | 2 | 1 | 1 | 2 | 2 | 2 | 2 |

Tabla 3.7: Estructura de comandos entre cámara y computadora de vuelo

Encabezado: Marcan el inicio del mensaje y corresponden a dos caracteres de sincronía.

Código de mando: Este byte representa si se trata de lectura o escritura de registro.

Registro: Registro a leer o escribir.

Reservado: Bytes reservados para uso futuro.

CRC1: CRC correspondiente a los seis primeros bytes antes mencionados.

Argumento: El argumento consta de 2 bytes y es el valor a escribir dentro del registro seleccionado. En caso de lectura, estos bytes devolverán el valor actual configurado en el registro seleccionado.

CRC2: CRC correspondiente a todos los bytes del comando.

Los registros de lectura/escritura que permiten la configuración de la cámara son los siguientes:

- **Frecuencia de operación:** Frecuencia de operación a la que se encuentra operando la cámara.
- **Tasa de captura:** Este registro contiene el número de capturas que realizará la cámara.
- **Tiempo de exposición:** Valor de tiempo de exposición al momento de capturar una imagen.
- **Captura:** Registro donde se ordena a la cámara realizar captura de imagen. En modo lectura, este registro indica si la cámara ya capturó o se encuentra en proceso de capturar una imagen.

- **Modo de operación:** Este registro indica si la cámara está en modo de inicialización, modo encendido o modo apagado.
- **Selección de imagen:** Selecciona número y tamaño de imagen.
- **Lectura de paquete imagen:** Selecciona número de paquete imagen a leer.
- **Memoria:** Lee número de contenedores de imágenes utilizados en la memoria local del módulo de cámara.
- **Estado de cámara:** Valor de 16 bits que resume estado de la cámara.
- **Error de cámara:** Valor de 16 bits que resume errores presentados en la cámara.

3.1.7.2. Captura de imágenes

La cámara satelital fue diseñada con el principal objetivo de capturar imágenes de la tierra. La cámara cuenta con la opción de entregar la imagen en dos tamaños diferentes y en un formato .bmp o mapa de bits. La imagen grande (752 x 480 pixeles) corresponde a 1426 paquetes, mientras que la imagen pequeña (188 x 120 pixeles) corresponde a 92 paquetes. Cada paquete imagen es un arreglo de 264 bytes y una vez que la computadora de vuelo genera el comando de captura de imagen, ésta debe ser capaz de solicitar todos los paquetes imagen para posteriormente ser almacenados y enviados a la estación terrena. La estructura de cada paquete imagen es mostrada en la tabla 3.8 y a continuación se describe cada uno de los campos.

| Campo | Encabezado | ID. Imagen | Índice de paquete | CRC 1 | Carga útil | CRC 2 |
|------------|------------|------------|-------------------|-------|------------|-------|
| Núm. bytes | 2 | 2 | 2 | 2 | 254 | 2 |

Tabla 3.8: Estructura de paquete imagen

Encabezado: Marcan el inicio del mensaje y corresponden a dos bytes de sincronía.

ID. Imagen: Identifican el número de imagen a la que pertenece el paquete.

Índice de paquete: Contiene el índice o número del paquete dentro de la imagen.

CRC 1: CRC correspondiente a los seis primeros bytes antes mencionados.

Carga útil: Contiene los datos imagen a transferir y consta de 254 bytes.

CRC 2: CRC correspondiente a todos los bytes del paquete imagen.

Para poder obtener la imagen capturada por el satélite, la estación terrena deberá obtener toda la información contenida dentro del campo "carga útil" de todos los paquetes imagen recibidos. El software de estación terrena se encargará de la reconstrucción de la imagen y será presentado en el cuarto capítulo, sin embargo, para fines demostrativos, más adelante se presentarán algunas de las imágenes capturadas por el satélite cubesat.

La figura 3.15 muestra el diagrama de flujo seguido por la computadora de vuelo para operar la cámara satelital. En un inicio la computadora de vuelo emite el comando a potencia para energizar el módulo cámara y espera 1 minuto para que ésta sea inicializada. Pasado el minuto, la computadora de vuelo espera que la cámara se encuentre en modo encendido y en espera para poder ejecutar un comando. Debido a que la frecuencia de operación y el tiempo de exposición están relacionados, se decidió fijar la frecuencia de operación a 10 Hz y únicamente ajustar el valor de tiempo de exposición. Una vez modificados los parámetros, se realiza la captura de la imagen y se decide si será enviada a estación terrena o será almacenada dentro de las memorias flash de la computadora de vuelo. Finalmente, cuando ya no existe otro comando más que ejecutar, la computadora de vuelo emite el comando para desenergizar el módulo y da fin a la función para operar la cámara. El proceso de almacenamiento de imagen será descrito en la subsección 3.1.7.3.

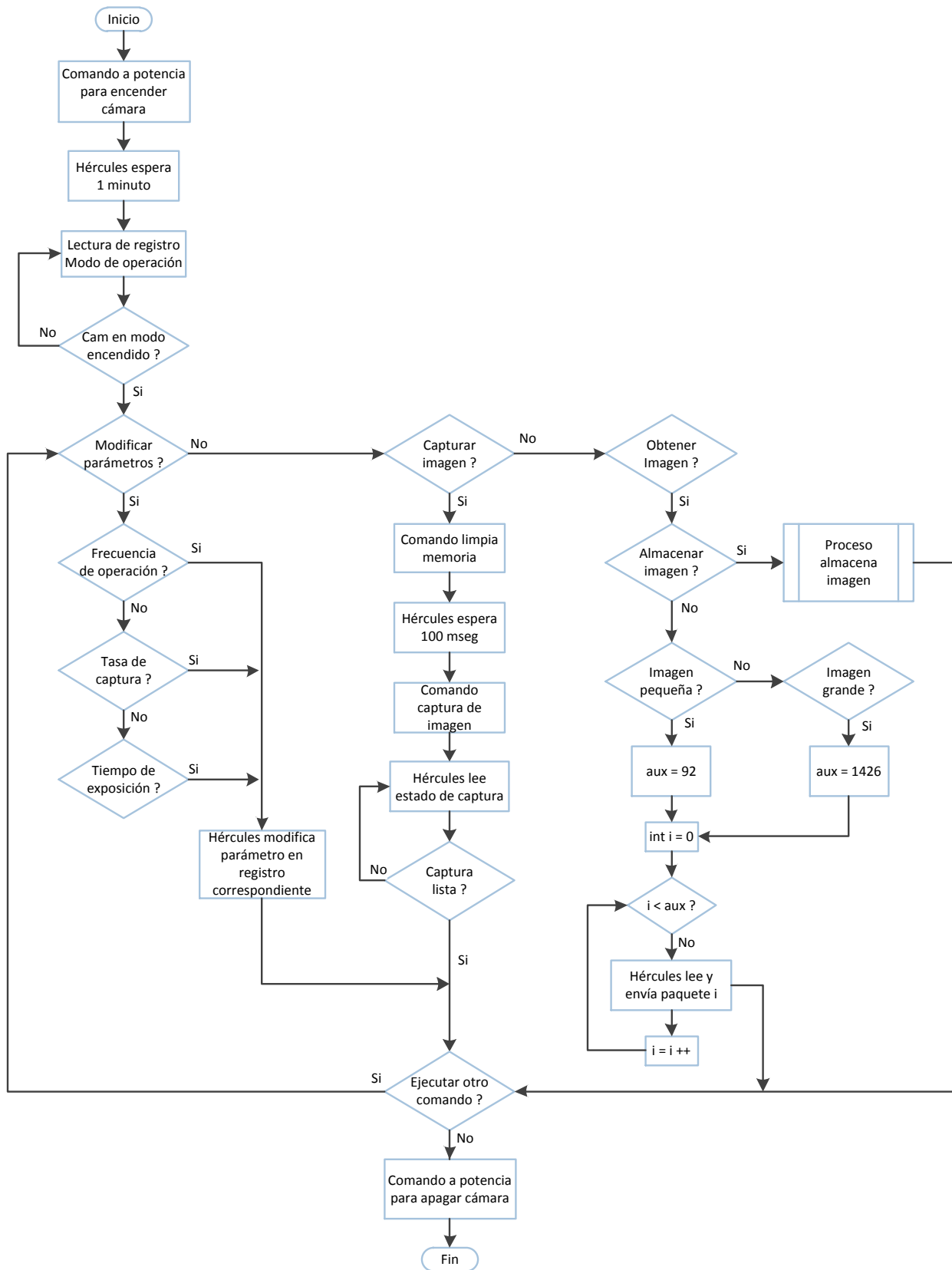


Figura 3.15: Diagrama de flujo para operación con cámara satelital

Aunque el software diseñado para la operación de la cámara permite a la computadora de vuelo manipular todos los parámetros posibles del sensor, cabe señalar que la cámara no cuenta con un autoajuste de parámetros al momento de adquirir una imagen. Como ya se mencionó, la frecuencia de operación fue fijada a 10 Hz, por lo que el operador de la estación terrena solo deberá modificar el tiempo de exposición basándose en la luz que pueda incidir sobre el sensor de imagen.

A continuación se muestran algunas imágenes en sus dos tamaños, capturadas desde el edificio 12 del Instituto de Ingeniería UNAM. En los tres casos se enfocó el mismo objetivo y únicamente se modificó el valor de tiempo de exposición. La imagen 3.16 muestra una captura de imagen con un correcto valor de tiempo de exposición. La imagen 3.17 muestra una imagen sub-expuesta y la imagen 3.18 muestra una imagen sobre-expuesta.



Figura 3.16: Imagen correctamente expuesta

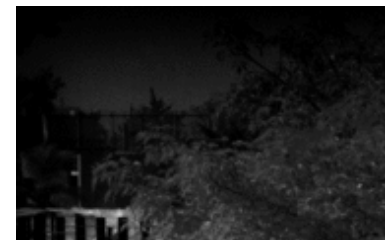


Figura 3.17: Imagen sub-expuesta



Figura 3.18: Imagen sobre-expuesta

3.1.7.3. Almacenamiento de imágenes

La computadora de vuelo cuenta con dos memorias flash con el propósito principal de capturar y almacenar imágenes de cualquier parte de la tierra. En un principio, el satélite recibirá datos para capturar un objetivo en tierra. Posteriormente la computadora de vuelo capturará la imagen y la almacenará en ambos tamaños, grande y pequeña. Cuando el satélite se encuentre nuevamente en línea de vista, la estación terrena descargará la imagen pequeña y evaluará si vale la pena descargar la imagen grande.

Las dos memorias flash dentro de la computadora de vuelo permitirán almacenar 341 imágenes en ambos tamaños, grande y pequeño. Cada paquete imagen recibido de la cámara será guardado en una página de la memoria flash, por lo que, una imagen grande corresponderá a 1426 páginas y una imagen pequeña a 92 páginas.

Para el caso de las imágenes grandes, la memoria flash 1 será direccionada en modo de 4 bytes y cada imagen será contenida en 6 sectores. En modo de direccionamiento a 4 bytes se puede tener un máximo de 2047 sectores, por lo que la división entre 6 sectores da como resultado un máximo de 341 imágenes.

Para el caso de las imágenes pequeñas, la memoria flash 2 será direccionada en modo de 3 bytes y se usará el segmento 0 para su almacenamiento, donde cada imagen pequeña será contenida en 6 subsectores. En modo de direccionamiento a 3 bytes se puede tener un máximo de 4096 subsectores por segmento, por lo que este segmento será capaz de almacenar también 341 imágenes pequeñas.

El almacenar una imagen grande y una pequeña en 6 sectores y 6 subsectores respectivamente, representa dejar páginas en blanco o páginas sin usar entre una imagen y otra, sin embargo, debido a que la memoria solamente puede ser borrada por sector, subsector y chip de 256 Mbits, esta decisión disminuye el desarrollo de software y permite tener un mejor control de las imágenes a la hora de ser almacenadas.

La imagen 3.19 representa el diagrama de flujo seguido por la computadora de vuelo para almacenar la misma imagen en tamaño grande y pequeña dentro de las memorias flash. La función define la variable entera " i " para ser usada como contador del número de paquete imagen y número de página guardada. El microcontrolador Hércules almacenará en su memoria flash interna un valor que corresponde al contador de imágenes ya almacenadas y posteriormente usará este valor para obtener la dirección inicio de una nueva imagen a almacenar.

Recordando que las imágenes grandes y pequeñas serán almacenadas en dos distintas memorias haciendo uso del protocolo SPI, el microcontrolador debe habilitar la respectiva memoria flash con la línea de chip select, antes y después del almacenado de una imagen. Una vez seleccionada la correcta memoria, se deberá configurar el modo de direccionamiento, esto debido a que los registros de configuración de las memorias se encuentran en memoria volátil. Posterior a esto, el Hércules generará el comando para petición del paquete imagen, igual al valor que tenga la variable " i " en ese momento, una vez pedido y almacenado el paquete imagen, incrementará la variable " i " y repetirá este proceso 1426 veces para el caso de las imágenes grandes y 92 veces para el caso de las imágenes pequeñas.

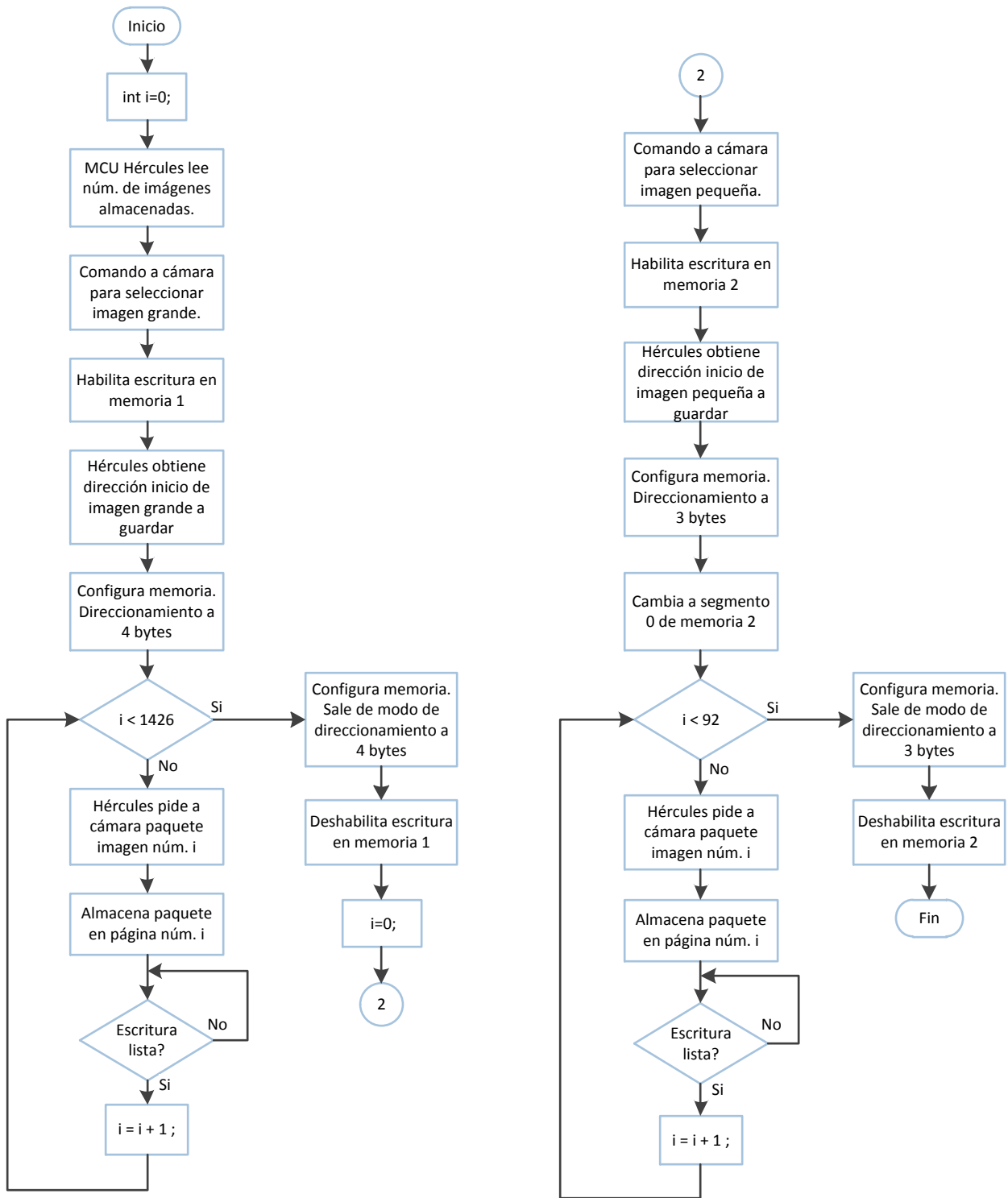


Figura 3.19: Diagrama de flujo para almacenado de imágenes satelitales

3.2. Comandos operativos

Una parte importante del proyecto satelital, es la definición de los comandos operativos para el intercambio de información con la estación terrena. El cubesat cuenta con 8 comandos operativos para la descarga de telemetría, descarga de imágenes y programación de tareas. Dentro de los comandos operativos, algunos corresponden a telemetría obtenida en línea de vista con la estación terrena y otros, a descarga de telemetría previamente almacenada.

Todos los comandos enviados por la estación terrena serán recibidos por el satélite en la frecuencia VHF, mientras que la información enviada por el satélite a la estación terrena será enviada en la frecuencia UHF. Para el caso de las imágenes capturadas, las imágenes grandes serán enviadas por el módulo de banda S y las pequeñas podrán ser enviadas por ambos radios.

Los comandos operativos enviados por la estación terrena al cubesat siguen el formato mostrado en la tabla 3.9.

| Campo | Encabezado | Núm bytes | Núm. de comando | Argumento | CRC |
|------------|------------|-----------|-----------------|-----------|-----|
| Núm. bytes | 2 | 2 | 1 | 1 - 30 | 2 |

Tabla 3.9: Estructura de comando entre estación terrena y satélite

Encabezado: El encabezado corresponde a 2 bytes en formato ASCII y son usados como identificadores de comando.

Núm bytes: Número total de bytes de la trama.

Núm de comando: Especifica el número de comando a ejecutar.

Argumento: Datos necesarios para ejecutar el comando (Necesario solo para algunos comandos).

CRC: CRC correspondiente a todos los bytes del comando.

La imagen 3.20 representa el diagrama de flujo seguido por el satélite para ejecutar un comando operativo recibido desde estación terrena. En todos los casos, el microcontrolador del radio UHF/VHF recibirá el comando y generará el CRC a partir de la información recibida, si el CRC recibido coincide con el CRC generado, el comando se da por válido y es enviado a la computadora de vuelo para su posterior ejecución.

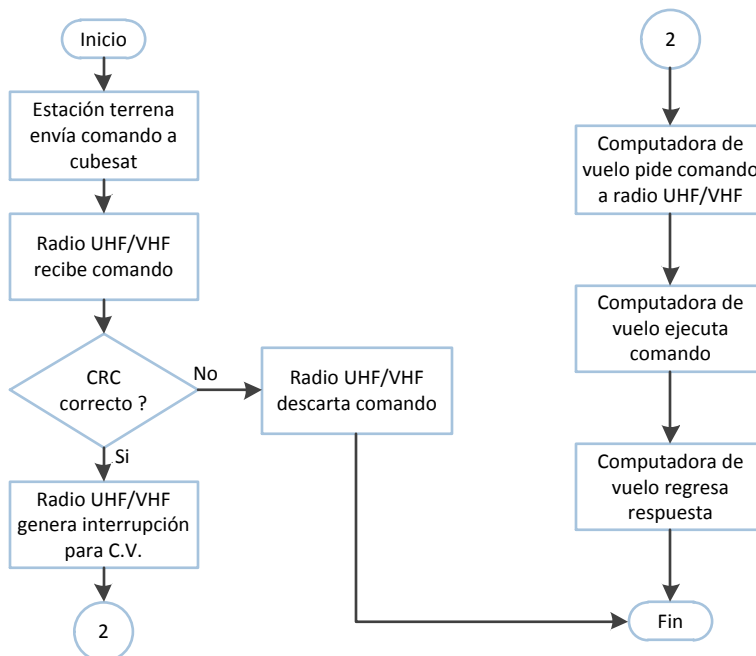


Figura 3.20: Diagrama de flujo para recepción de comandos operativos

3.2.1. Telemetría básica

Este comando tiene el objetivo de conocer un estado básico e instantáneo del satélite. Una vez que el comando es recibido por el satélite, la computadora de vuelo obtendrá y enviará a la estación terrena la siguiente telemetría: nombre del satélite en formato ASCII, porcentaje de carga de las baterías, voltajes y corrientes de sólo tres subsistemas normalmente encendidos (computadora de vuelo, potencia y comunicaciones), siete temperaturas de las veinte disponibles y el número de imágenes almacenadas en sus memorias flash.

La imagen 3.21 muestra el diagrama de flujo seguido por la computadora de vuelo para obtener todos los valores antes mencionados. Como puede observarse, la computadora de vuelo solicita la información secuencialmente, una vez obtenida toda la telemetría necesaria, genera el CRC a partir de los datos obtenidos y envía la respuesta al subsistema de comunicaciones para posteriormente ser enviada a la estación terrena.

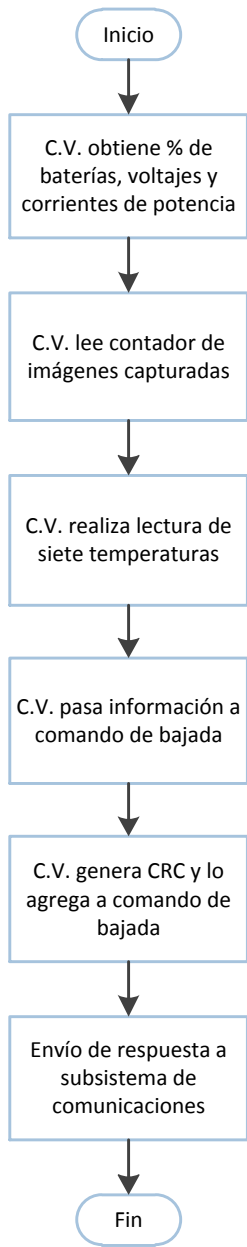


Figura 3.21: Diagrama de flujo de telemetría básica

3.2.2. Telemetría intermedia

Dentro de las tareas autónomas que la computadora de vuelo debe realizar, se encuentra la captura y almacenamiento de telemetría cada 20 min. Para tal fin, la telemetría será almacenada en el segmento 1 de la memoria flash 2. La computadora de vuelo tiene la capacidad de guardar 400 muestras y serán almacenadas en un buffer circular, por lo que, si el satélite deja de ver estación terrena por algún tiempo sin haber descargado la telemetría, la información será sobrescrita con nuevos datos obtenidos. La telemetría capturada tendrá la siguiente información: porcentaje de carga de las baterías, voltajes y corrientes de todos los subsistemas, temperatura de todos los sensores, fecha y hora obtenida del reloj en tiempo real y valor de campos magnéticos obtenidos del magnetómetro de la computadora de vuelo. El objetivo de este comando es la descarga de las 400 muestras almacenadas. La imagen 3.22 muestra el diagrama de flujo seguido por la computadora de vuelo para obtener y almacenar la telemetría intermedia.

Como puede observarse, una vez que la computadora de vuelo obtiene toda la información requerida, primero leerá el último número de telemetría almacenada y en caso de existir 400 muestras, se reiniciará el contador, generando así el buffer circular. Recordando que la memoria sólo puede ser borrada por subsector (correspondiente a 16 páginas) y no por página, el microcontrolador verificará si el número de telemetría a almacenar es múltiplo de 16 y en caso de ser afirmativo, primero se eliminará ese subsector y enseguida se escribirá la nueva información.

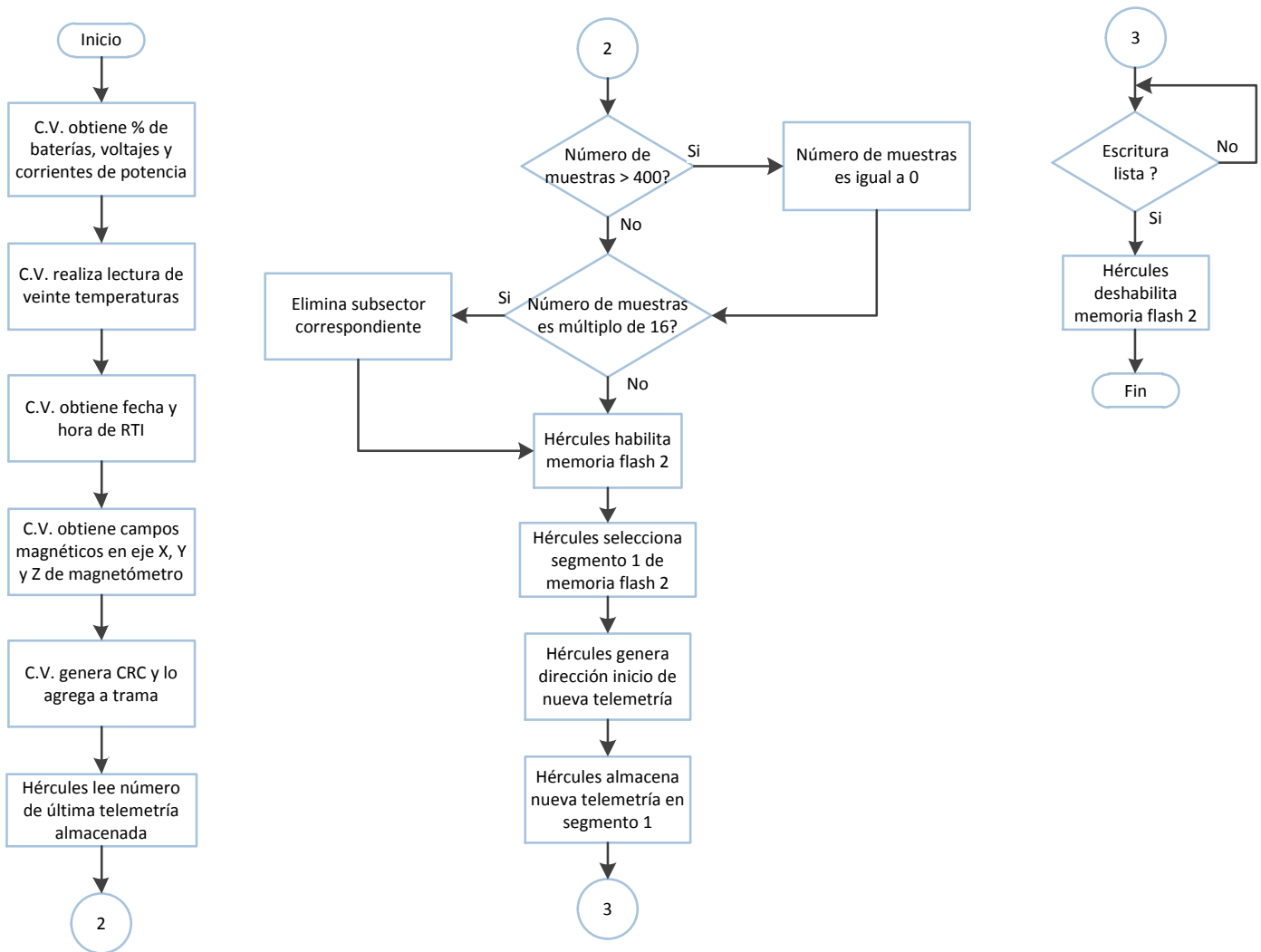


Figura 3.22: Diagrama de flujo de telemetría intermedia

3.2.3. Telemetría avanzada

La telemetría avanzada servirá para conocer el estado de navegación inercial del satélite. La telemetría avanzada cuenta con la siguiente información: datos de latitud, longitud y altitud generados por el GPS y valores de campos magnéticos y aceleraciones por cada eje, generados por la IMU del subsistema de control y orientación. Este comando puede enviar información de dos formas distintas, 50 muestras instantáneas o 800 muestras capturadas y almacenadas en el segmento 2 de la memoria flash 2. Las 800 muestras primero serán capturadas por el subsistema de control y orientación en una maniobra de apuntamiento y después enviadas a la computadora de vuelo para su almacenamiento. La imagen 3.23 muestra el diagrama de flujo seguido por la computadora de vuelo para generar las 50 muestras instantáneas o descargar la telemetría avanzada previamente almacenada. Para el caso de las muestras instantáneas, la computadora de vuelo debe encender el GPS y el subsistema de control y orientación ya que estos estarán apagados, una vez obtenidas y enviadas todas las muestras, la computadora de vuelo apagará nuevamente los subsistemas. Para el caso de las muestras almacenadas, únicamente se enviará la información desde las memorias sin la necesidad de encender los dos subsistemas.

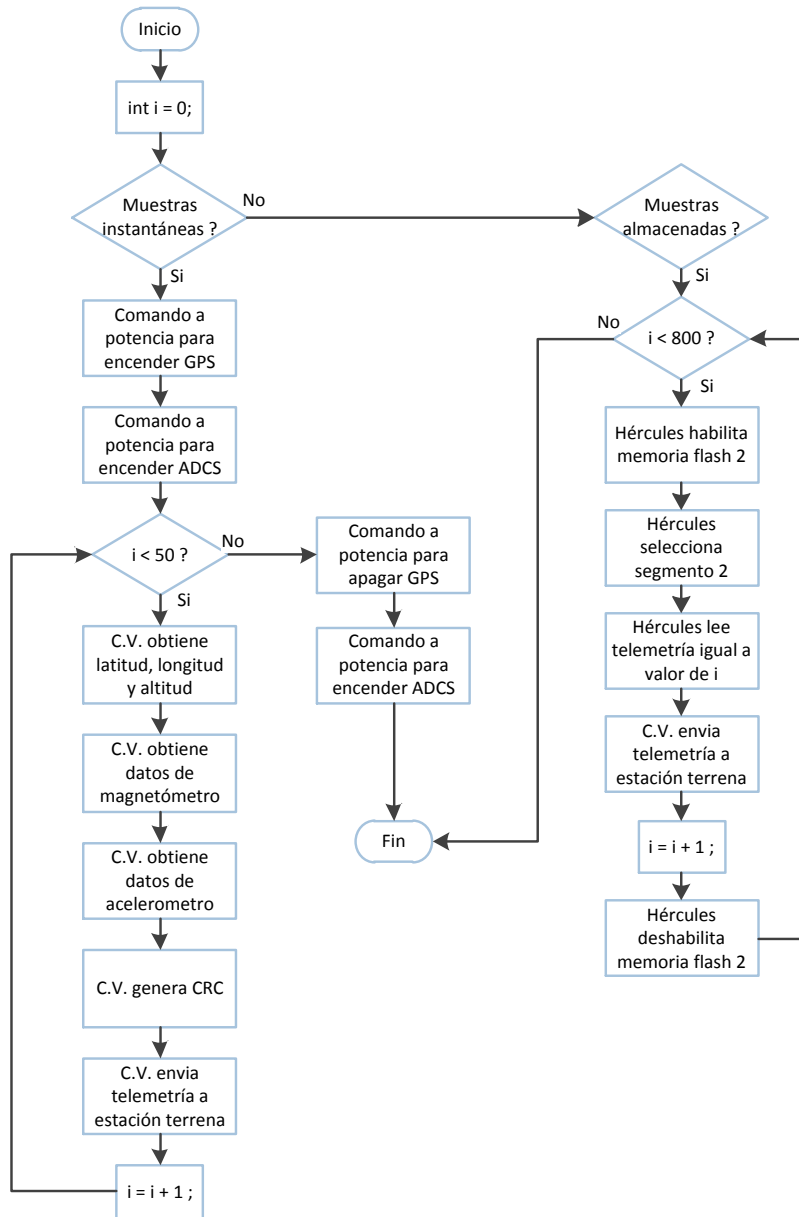


Figura 3.23: Diagrama de flujo de telemetría avanzada

3.2.4. Telemetría de propagación orbital

Esta telemetría permitirá conocer a la estación terrena, la órbita por la cual el satélite se propaga. La telemetría de propagación orbital cuenta con la siguiente información: datos de latitud, longitud, altitud, fecha y hora, todos generados por el GPS. La telemetría será almacenada en el segmento 3 de la memoria 2 y será necesario que en un primer comando, estación terrena defina el número de muestras (máximo de 250 muestras) y tiempo de muestreo (mínimo de 1 minuto). Con este comando, el satélite enviará todas las muestras previamente capturadas. La imagen 3.24 presenta el diagrama de flujo seguido por la computadora de vuelo para enviar toda la información almacenada. Como se puede observar, la computadora de vuelo primero envía a estación terrena el número de muestras almacenadas mientras que el tiempo de muestreo puede ser obtenido con la fecha dentro de cada paquete enviado. Una vez realizado esto, la computadora de vuelo entra en un ciclo donde va leyendo cada muestra almacenada y enseguida la envía a la estación terrena.

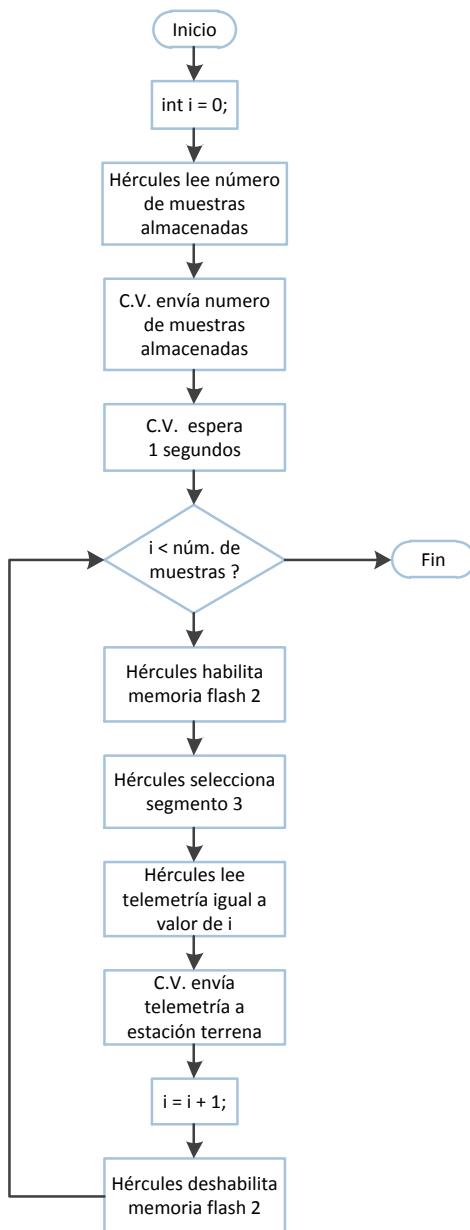


Figura 3.24: Diagrama de flujo para descarga de propagación orbital

3.2.5. Apuntamiento satelital para captura y descarga de imágenes

El uso de este comando es solo para fines de programar o agendar la ejecución del apuntamiento satelital a un objetivo en tierra haciendo uso del subsistema de control y orientación. En un principio, la computadora de vuelo recibirá datos de latitud, longitud y altitud del objetivo a apuntar, fecha y hora inicio del apuntamiento y tarea a realizar (captura o descarga de imagen). En seguida, la computadora de vuelo almacenará los datos de latitud, longitud y altitud recibidos en la memoria flash de su microcontrolador y después programará la función de alarma de su reloj en tiempo real con la fecha y hora establecidos. Debido a que el apuntamiento satelital podrá lograrse en un intervalo de entre 12 y 15 minutos, la computadora de vuelo emitirá el comando de control y apuntamiento satelital 15 min antes de llegar a su objetivo. Una vez estando apuntando al objetivo, el satélite podrá capturar una imagen o realizar el enlace de comunicación con la antena direccional de banda S.

La imagen 3.25 muestra la propuesta del algoritmo de control y estimación de altitud ejecutado por el subsistema de control y orientación. En un principio, el satélite se mantendrá ejecutando tareas autónomas hasta que se genere la interrupción de alarma por el reloj en tiempo real, enseguida la computadora de vuelo encenderá el GPS y el subsistema de control y orientación para poder emitir la ejecución del algoritmo de control con los datos del objetivo en tierra. La línea roja en la imagen, muestra el tiempo en que el subsistema de control y orientación se encuentra haciendo maniobras de apuntamiento hacia el objetivo en tierra, para que después, la computadora de vuelo ejecute la captura de una imagen o el enlace con el radio de banda S.

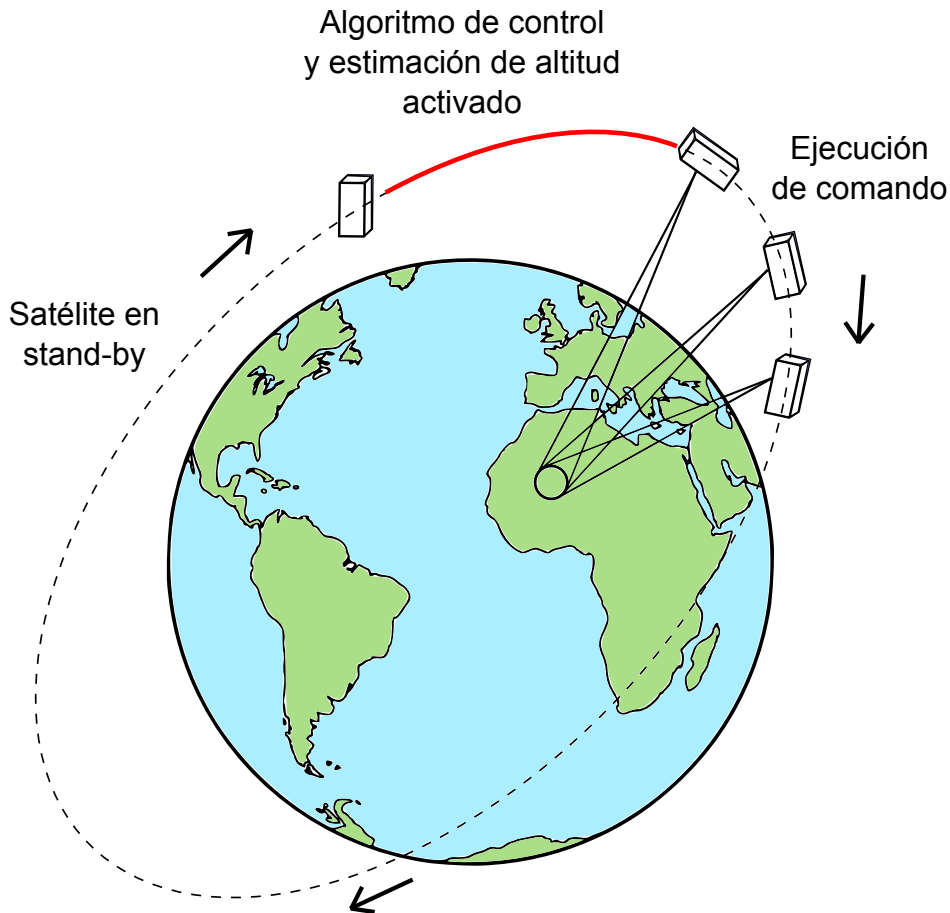


Figura 3.25: Ejecución de algoritmos de control y apuntamiento satelital

3.2.6. Descarga de imágenes satelitales

Este comando tiene la finalidad de descargar imágenes grandes y pequeñas previamente almacenadas por el satélite. En el comando de subida, estación terrena debe definir tamaño, número de imagen y radio de comunicación por el cual se efectuará la descarga. Aunque el satélite permite la descarga de imágenes grandes con el radio UHF/VHF, siempre será preferible hacer uso de banda S debido a su velocidad de transferencia de datos. Cabe señalar, que la transferencia de datos por banda S será usado, siempre y cuando, el enlace con estación terrena se haya logrado. La imagen 3.26 representa el diagrama de flujo seguido por la computadora de vuelo para descargar un único paquete de imagen o una imagen completa mediante el radio UHF/VHF o banda S.

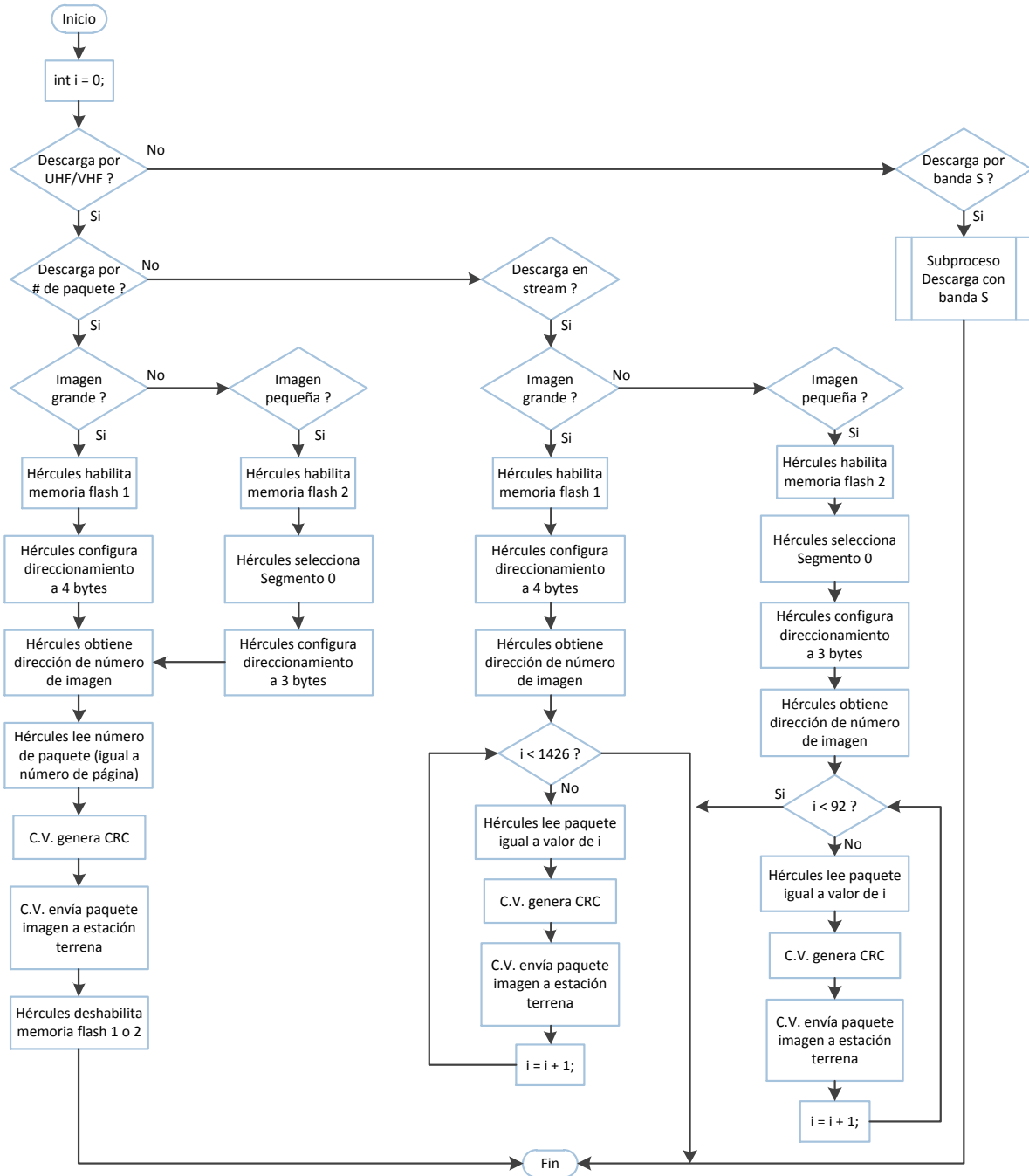


Figura 3.26: Diagrama de flujo para descarga de imágenes

La descarga de imagen con banda S corresponde al mismo procedimiento de descarga como en el caso de UHF/VHF, por lo cual fue omitido.

Como se puede observar, estación terrena cuenta con diversos modos de descarga de imagen. En un principio se debe elegir el radio de comunicación para efectuar la descarga, enseguida se debe elegir descarga por número de paquete o en stream. Cuando se elige número de paquete, estación terrena debe generar un comando de subida por cada paquete requerido, permitiendo reconstruir una imagen cuando ésta no pudo ser descargada completamente. Cuando se elige en stream, el cubesat sólo recibirá un comando de subida e inmediatamente empezará a enviar todos los paquetes de la imagen sin necesidad de recibir una confirmación por cada paquete enviado, generando así, la posibilidad que la imagen no llegue completa. Por último, estación terrena deberá elegir entre los dos tamaños de imagen disponibles a enviar.

3.2.7. Contacto satelital

Este comando tiene el único objetivo de verificar un enlace de comunicaciones entre satélite y estación terrena. La estación terrena enviará un comando de contacto y el satélite deberá contestar un eco devolviendo el mismo mensaje recibido, de esta manera, estación terrena podrá saber si existe comunicación con el satélite.

3.3. Gestión interna satelital

Los retos de operar en un ambiente espacial presentados en la sección 1.4 hacen que la gestión interna satelital [29] sea una tarea importante y necesaria en cualquier satélite. La gestión interna satelital se encarga de coleccionar información de sensores para poder iniciar y ejecutar funciones de recuperación ante un problema presente. Adicionalmente, el satélite puede obtener información, almacenarla y posteriormente descargarla con la finalidad de que la estación terrena haga uso de los datos y conozca el estado del satélite dentro de su órbita.

3.3.1. Diagnóstico satelital

El diagnóstico satelital consta de dos partes: la primera consiste en obtener y almacenar información de sensores (telemetría intermedia, previamente descrita) para que estación terrena pueda conocer el estado del satélite y la segunda parte consiste en ejecutar tareas preventivas a partir de información obtenida de sensores. Los sensores de temperatura colocados en las baterías y el porcentaje de su carga serán leídos cada 5 minutos por la computadora de vuelo, evitando así, una temperatura por debajo de los 10 grados Celsius y un estado de carga por debajo del 25 %.

La imagen 3.27 muestra el diagrama de flujo seguido por la computadora de vuelo para el diagnóstico satelital. Para ejecutar el diagnóstico satelital, la computadora de vuelo usa el timer del reloj en tiempo real y el timer interno de su microcontrolador. En un principio el microcontrolador Hércules configura y ajusta el timer del reloj en tiempo real a 20 minutos y el reloj de su timer interno a 5 min. Una vez configurados, la computadora de vuelo entra en un bucle o ciclo infinito, del cual saldrá únicamente si se genera una interrupción para ejecutar un comando recibido desde estación terrena. En el ciclo infinito, se estará a la espera de una señal de cualquiera de los dos timers y en cuanto se genere la señal, la computadora de vuelo verificará que señal fue la que se generó.

En caso de generarse la señal del reloj en tiempo real, primero se limpiará la interrupción y enseguida se ejecutará el subproceso de almacenado de telemetría intermedia, descrito en la subsección 3.2.2.

En caso de generarse la bandera del timer interno del microcontrolador, la computadora de vuelo primero detendrá el timer, después se leerán los sensores de temperatura en las baterías y su porcentaje de carga (valor generado por el subsistema de potencia). Si existe una temperatura menor a 10°C, se emitirá un comando a potencia para iniciar función de calefacción de las baterías y así evitar daño en ellas. Si existe un porcentaje de carga de baterías menor a 25 %, se emitirá un comando a potencia para que ponga el satélite en un estado de carga con ayuda de las celdas solares.

Finalmente, en cualquiera de todos los casos, la computadora de vuelo reajusta sus timers y regresa a su estado inicial, a la espera de una nueva señal de ambos timers o una interrupción desde estación terrena.

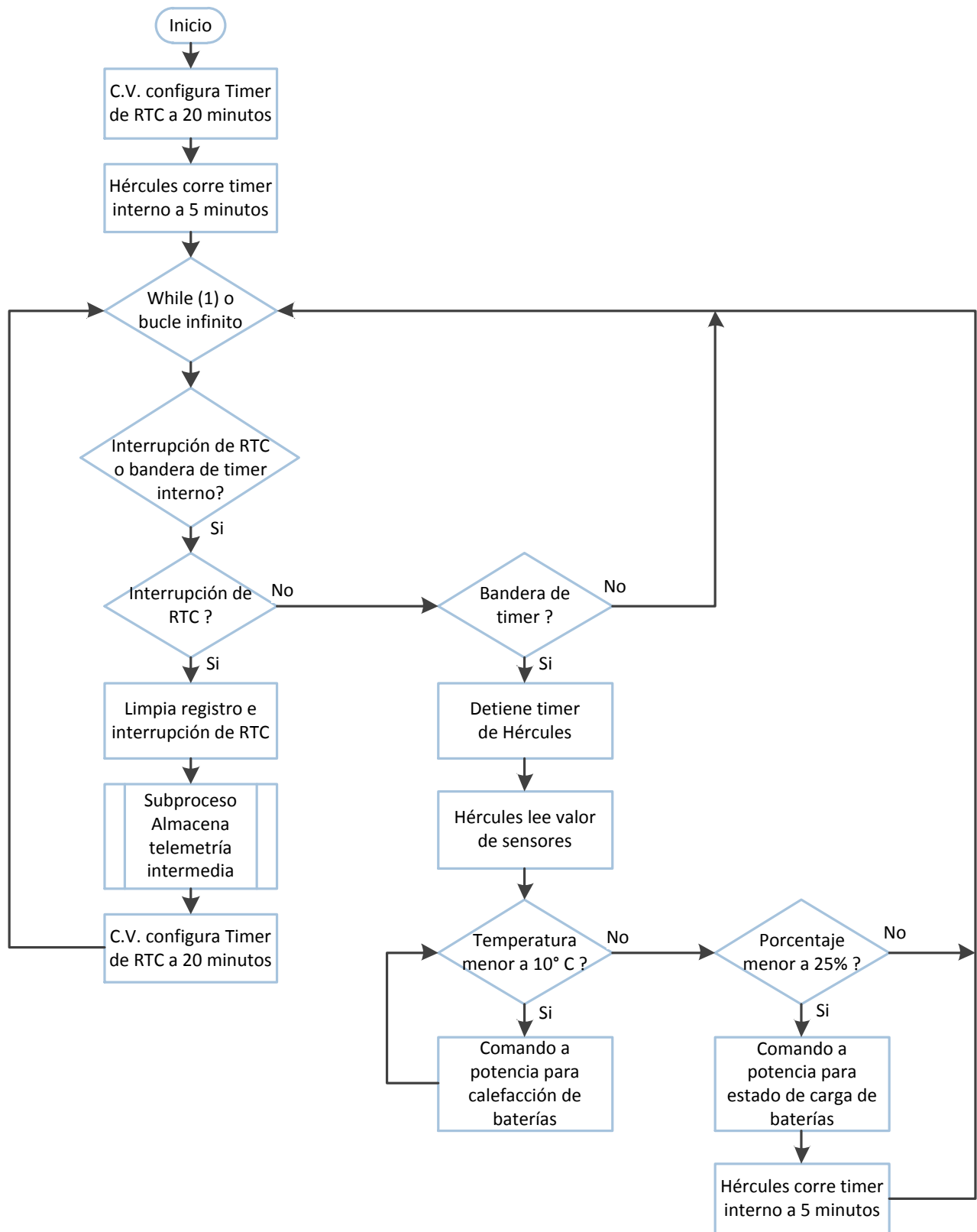


Figura 3.27: Diagrama de flujo de diagnóstico satelital

3.4. Implementación de software

Para el desarrollo e implementación del software de operación, fue necesario el uso de dos ambientes de desarrollo de software, proporcionados por Texas Instruments. Toda la programación diseñada para el microcontrolador Hércules fue generada bajo el lenguaje de programación C y un ambiente gráfico para configuración de periféricos. A continuación se describen las dos plataformas de desarrollo utilizadas y sus principales características.

3.4.1. Code Composer Studio

Code composer studio (CCS, por sus siglas en inglés) es un ambiente de desarrollo integrado de Texas Instruments (TI) [30] para su familia de procesadores embebidos. CCS comprende un conjunto de herramientas utilizadas para desarrollar y depurar aplicaciones embebidas. Este incluye compiladores para cada familia de dispositivos, editor de código fuente, ambiente de generación de proyectos, depurador, simuladores, sistemas operando en tiempo real y otras características. Las herramientas e interfaz de familias de microcontroladores, permiten a los usuarios iniciar más rápido que antes y agregar funcionalidad a sus aplicaciones, gracias a sus herramientas de productividad sofisticadas.

CCS está basado en la plataforma eclipse. El software eclipse fue originalmente desarrollado para la creación de herramientas de desarrollo. Eclipse ofrece una estructura de software excelente para ambientes de desarrollo de generación de software y se está convirtiendo en un marco estándar utilizado por muchos vendedores de software embebido. CCS combina las ventajas de la plataforma de software eclipse con capacidades de depuración avanzadas. La imagen 3.28 muestra la ventana de editor de código y la imagen 3.29 muestra la ventana de depuración de código del software code composer studio.

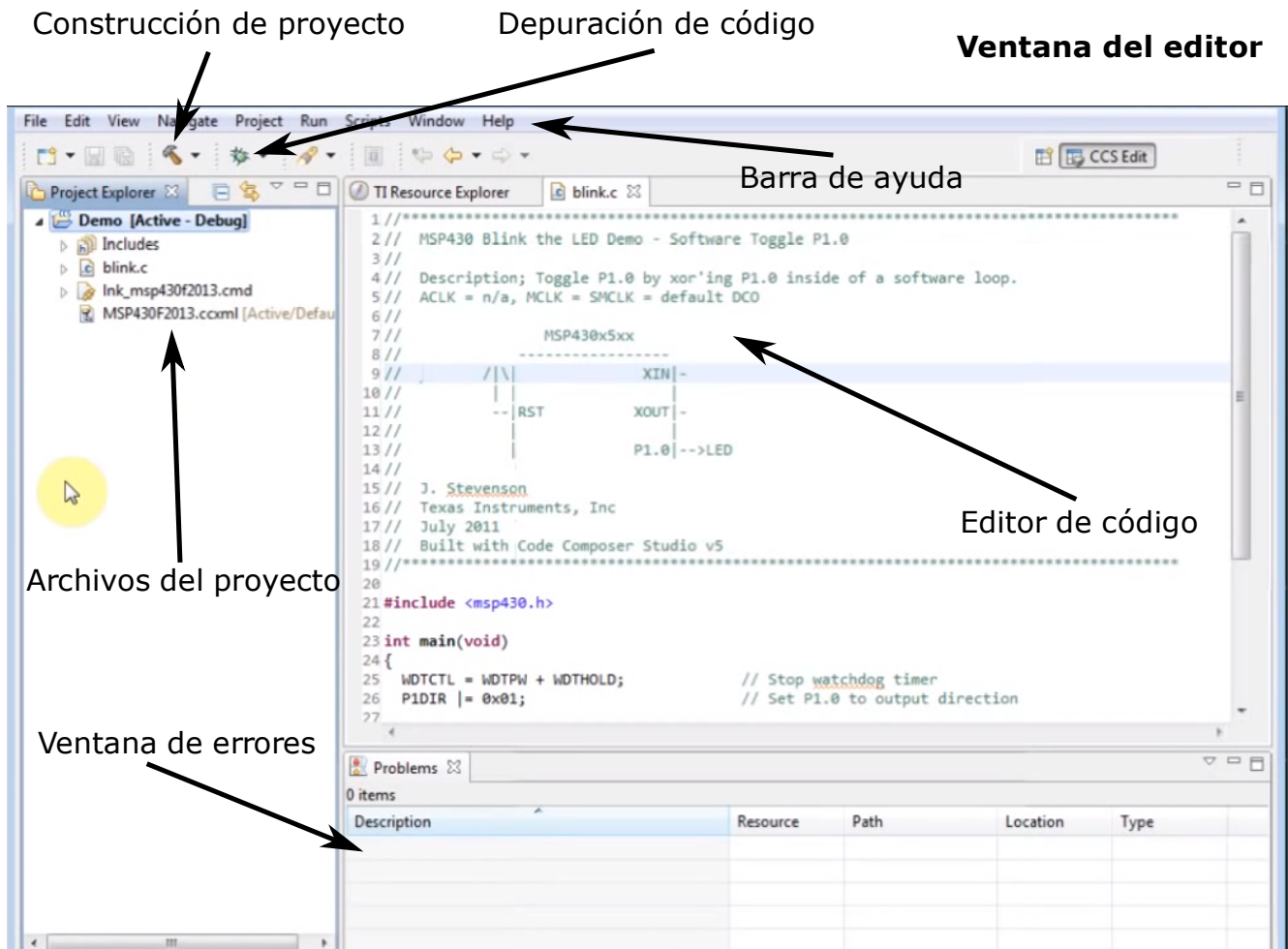


Figura 3.28: Ventana de editor de code composer studio

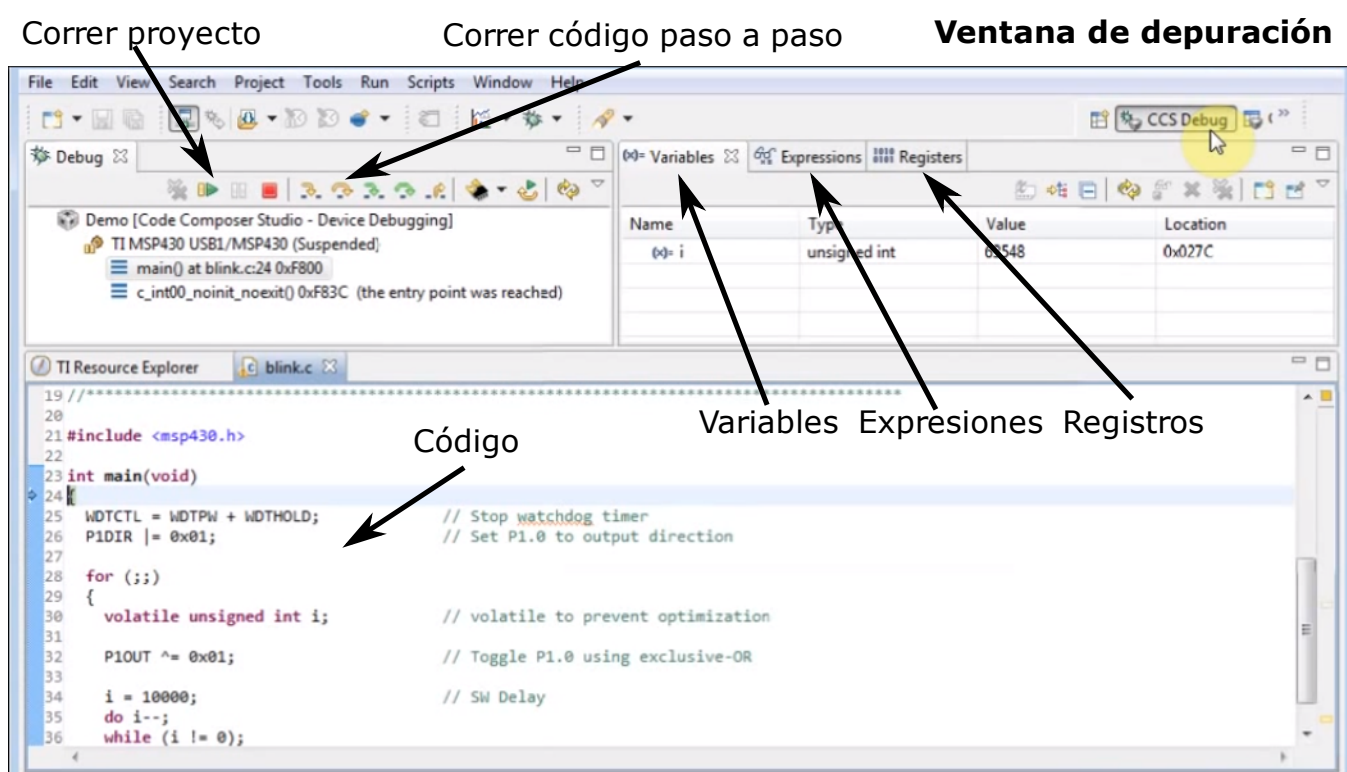


Figura 3.29: Ventana para depuración de código de code composer studio

Para el cargado de software diseñado en CCS o programación del microcontrolador Hércules, se hizo uso del hardware XDS100v2 [31]. Spectrum digital XDS100v2 es un dispositivo de prueba y cargado de software para procesadores de Texas Instruments. Se caracteriza por ser un dispositivo de bajo costo mientras soporta el estándar JTAG (IEEE 1149.1). La imagen 3.30 muestra el dispositivo XDSv100.

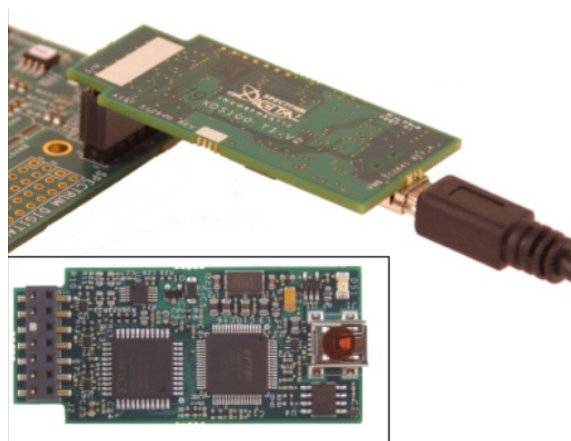


Figura 3.30: Hardware XDSv100 para programación de software

3.4.2. HalCoGen

Hardware Abstraction Layer Code Generation (HalCoGen, por sus siglas en inglés) contiene módulos de software con acceso directo al MCU y es responsable de la inicialización del sistema,[32]. HalCoGen permite a los usuarios generar drivers o funciones del dispositivo para la familia de microcontroladores Hércules. Halcogen es una guía simple que ayuda a configurar y generar capas abstractas del microcontrolador incluyendo funciones seguras. Básicamente, halcogen proporciona una interfaz gráfica de usuario que permite configurar periféricos, interrupciones, relojes y otros parámetros del MCU.

La imagen 3.31 muestra la pantalla principal del software halcogen la cual consta de diversas pestañas donde el usuario puede seleccionar y configurar cualquier módulo o periférico con el que cuenta el microcontrolador de la familia Hércules seleccionado.

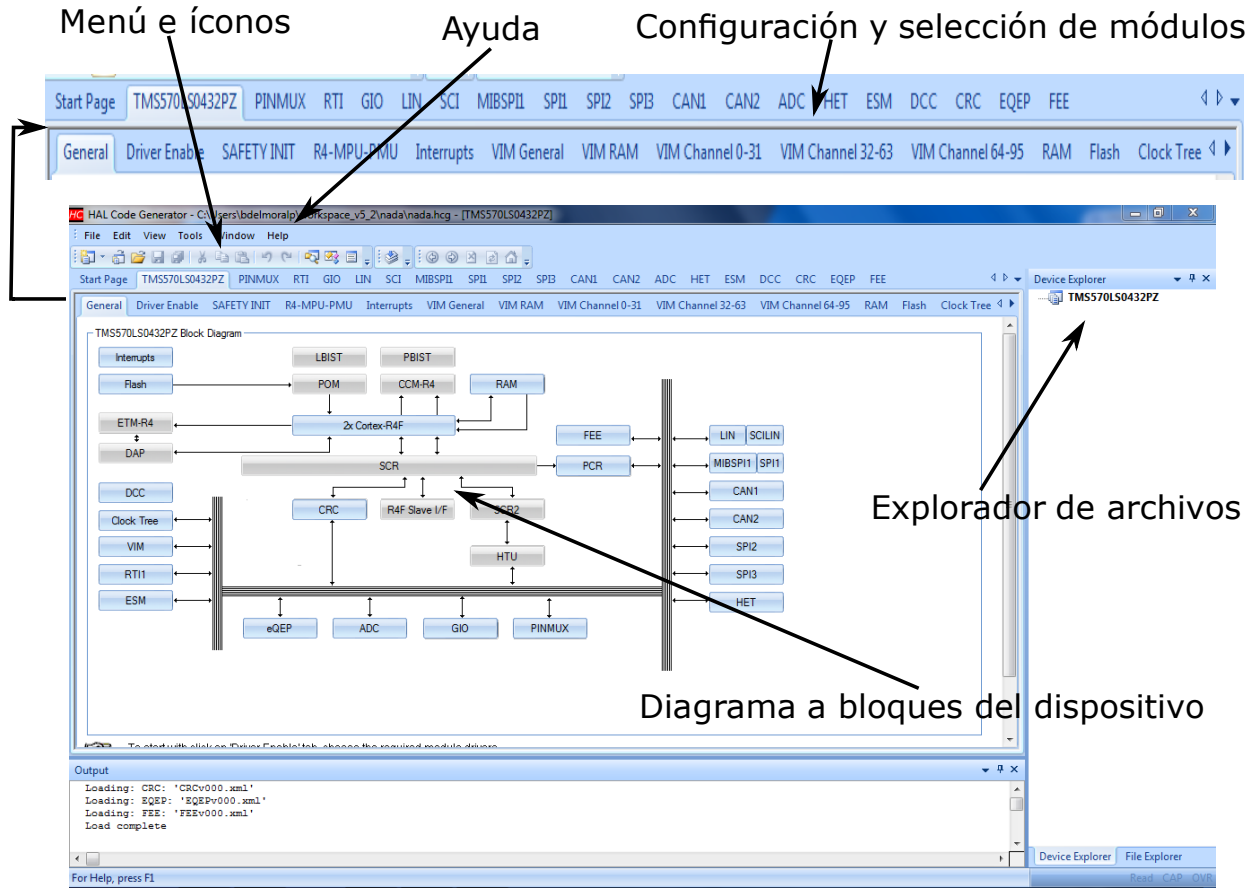


Figura 3.31: Ventana principal de plataforma halcogen

Capítulo 4

Pruebas de validación

En los capítulos anteriores de este trabajo de tesis, se describió la arquitectura satelital del cubesat propuesta por todo el equipo de trabajo. Además de esto, se describió el software desarrollado e implementado dentro del subsistema de computadora de vuelo, de tal forma que el satélite permita intercambio de datos con estación terrena como descarga de imágenes y telemetría satelital. Este capítulo detalla los métodos empleados para evaluar la operación y funcionamiento en software y hardware de la computadora de vuelo individualmente e interactuando con los demás subsistemas.

La sección 4.1 presenta cómo fue validado el hardware dentro de la computadora de vuelo. La sección 4.2 presenta el software de estación terrena usado para validar un enlace inalámbrico y validar que los datos enviados por el satélite son correctamente interpretados. La sección 4.3 detalla brevemente algunas pruebas de calificación espacial y presenta la prueba que fue realizada para validar todo el satélite, con lo cual quedará también validado el subsistema de computadora de vuelo.

4.1. Validación de computadora de vuelo

A continuación se presentan los métodos empleados para validar el hardware y software interno de la computadora de vuelo. En algunos casos se usó un osciloscopio para verificar las señales eléctricas y en otros, bastó con leer un ID o identificador dentro de los registros de lectura de los circuitos electrónicos.

4.1.1. Validación de protocolo I2C

Como ya se mencionó, el bus I2C será el principal protocolo de comunicación entre subsistemas. Aquí la computadora de vuelo fungirá como dispositivo maestro y será la encargada de iniciar cualquier transferencia de datos. La validación de este protocolo consistió en enviar un mensaje desde la computadora de vuelo a otro dispositivo o microcontrolador del satélite, si el mensaje es contestado por el dispositivo o recibido por el otro microcontrolador, entonces la comunicación es correcta. A manera de demostración del correcto funcionamiento del bus I2C, se hizo uso de un osciloscopio que tiene la capacidad de representar qué datos están siendo enviados o recibidos dentro del bus.

A continuación se presenta el proceso que sigue el satélite cuando éste recibe la ejecución de un comando desde estación terrena:

- Primero, el satélite recibe un comando desde estación terrena, seguido de esto, el subsistema de comunicaciones emite una señal de aviso o interrupción a la computadora de vuelo y es aquí donde la computadora de vuelo le responde a comunicaciones por I2C la confirmación del aviso.
- Una vez emitido el aviso, la computadora de vuelo pide por I2C a comunicaciones, los datos recibidos desde estación terrena.
- En este caso, estación terrena pide voltajes y corrientes satelitales por lo que computadora de vuelo pide la información al subsistema de potencia eléctrica.
- Finalmente, computadora de vuelo procesa la información recibida y responde los datos a comunicaciones para su posterior descarga a estación terrena.

La imagen 4.1 presenta de forma descendente los cuatro pasos antes descritos. En el inicio se tiene la confirmación del aviso a comunicaciones, enseguida la petición del comando a comunicaciones, después la petición de datos a potencia eléctrica y al final se tiene la respuesta del mensaje para ser transmitido a estación terrena. Al observar que las señales eléctricas eran correctas y que hubo un correcto manejo del mensaje, se comprueba que el protocolo de comunicación I2C entre subsistemas opera exitosamente.

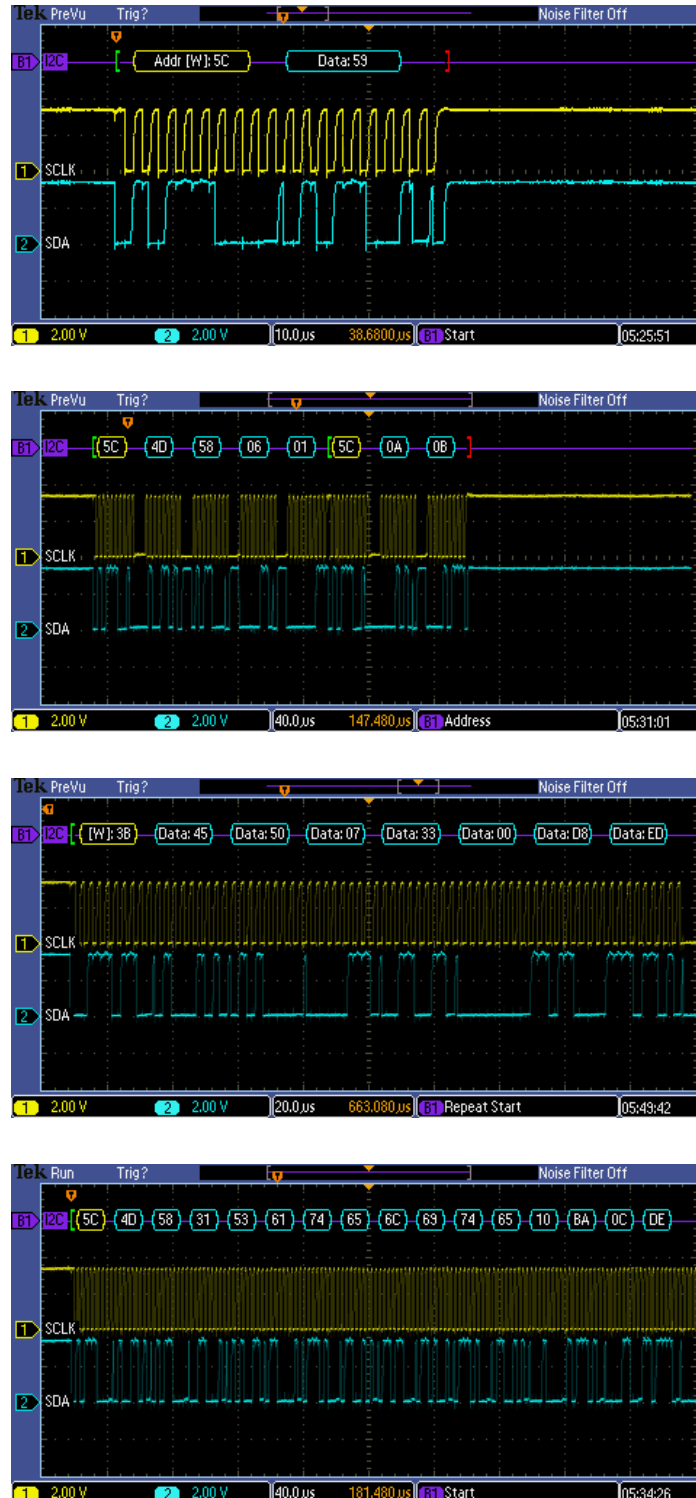


Figura 4.1: Validación de protocolo I2C

4.1.2. Validación de sensores de temperatura

Como recordaremos, los sensores de temperatura operan bajo el protocolo de comunicación 1-wire. El osciloscopio previamente usado no es capaz de procesar dicho protocolo, sin embargo, el método empleado para validar su funcionamiento consistió en leer el ID o identificador de todos los sensores. La tabla 4.1 fue obtenida de la hoja de datos del fabricante [21] y como se puede observar, al inicio del ID se tienen 8 bits del código de familia (en el caso de los sensores de temperatura este valor es 28 hexadecimal), 48 bits únicos de número serial y 8 bits de CRC.

| | | |
|--------------------------------|-----------------------|------------|
| 8-bits código de familia (28h) | 48-bits número serial | 8-bits CRC |
|--------------------------------|-----------------------|------------|

Tabla 4.1: Formato ID sensores de temperatura

Con la tabla antes descrita, se realizó una lectura de los ID's y se verificó que todos contaran con el código de familia y que el CRC generado coincidiera con el CRC leído. La imagen 4.2 presenta una captura de imagen del ambiente code composer studio mostrando las expresiones del microcontrolador donde se tiene 3 de los 20 ID's de los sensores de temperatura. En rojo se marcan los valores que corresponden a los 8 bits del código de familia.

| Expression | Type | Value |
|--------------|----------------------|------------------|
| ROM_sensores | unsigned char[20][8] | 0x080018F4 (Hex) |
| [0] | unsigned char[8] | 0x080018F4 (Hex) |
| (x)= [0] | unsigned char | 0x28 (Hex) |
| (x)= [1] | unsigned char | 0x80 (Hex) |
| (x)= [2] | unsigned char | 0xBF (Hex) |
| (x)= [3] | unsigned char | 0xBE (Hex) |
| (x)= [4] | unsigned char | 0x05 (Hex) |
| (x)= [5] | unsigned char | 0x00 (Hex) |
| (x)= [6] | unsigned char | 0x00 (Hex) |
| (x)= [7] | unsigned char | 0x65 (Hex) |
| [1] | unsigned char[8] | 0x080018FC (Hex) |
| (x)= [0] | unsigned char | 0x28 (Hex) |
| (x)= [1] | unsigned char | 0xE0 (Hex) |
| (x)= [2] | unsigned char | 0xC8 (Hex) |
| (x)= [3] | unsigned char | 0xBE (Hex) |
| (x)= [4] | unsigned char | 0x05 (Hex) |
| (x)= [5] | unsigned char | 0x00 (Hex) |
| (x)= [6] | unsigned char | 0x00 (Hex) |
| (x)= [7] | unsigned char | 0x9A (Hex) |
| [2] | unsigned char[8] | 0x08001904 (Hex) |
| (x)= [0] | unsigned char | 0x28 (Hex) |
| (x)= [1] | unsigned char | 0x58 (Hex) |
| (x)= [2] | unsigned char | 0xAF (Hex) |
| (x)= [3] | unsigned char | 0xBE (Hex) |
| (x)= [4] | unsigned char | 0x05 (Hex) |
| (x)= [5] | unsigned char | 0x00 (Hex) |
| (x)= [6] | unsigned char | 0x00 (Hex) |
| (x)= [7] | unsigned char | 0x7C (Hex) |
| [3] | unsigned char[8] | 0x0800190C (Hex) |
| [4] | unsigned char[8] | 0x08001914 (Hex) |
| [5] | unsigned char[8] | 0x0800191C (Hex) |
| [6] | unsigned char[8] | 0x08001924 (Hex) |
| [7] | unsigned char[8] | 0x0800192C (Hex) |

Figura 4.2: ID de 3 sensores de temperatura

Una vez obtenidos los ID´s de los sensores, el siguiente paso fue diseñar una función que adquiriera la temperatura de todos los sensores. Al igual que los identificadores, los sensores cuentan con un registro de 9 bytes para la lectura de temperatura y configuración de registros. La tabla 4.2 presenta el registro para lectura y configuración de los sensores 1-wire.

| | |
|--------|--|
| Byte 0 | temperatura, bits menos significativos |
| Byte 1 | temperatura, bits más significativos |
| Byte 2 | Registro Th |
| Byte 3 | Registro Tl |
| Byte 4 | Registro de configuración |
| Byte 5 | Reservado |
| Byte 6 | Reservado |
| Byte 7 | Reservado |
| Byte 8 | CRC |

Tabla 4.2: Registro para lectura y configuración de sensores 1-wire

Para validar que la lectura de todos los sensores fuera correcta, la computadora de vuelo primero obtiene los datos del registro y después genera el CRC correspondiente, si el CRC leído corresponde con el CRC generado, entonces la lectura se da por válida. La imagen 4.3 presenta una captura del ambiente code composer studio donde se muestran las 20 lecturas de los sensores de temperatura dispersos en todo el satélite. El valor decimal presentado ya corresponde al valor real en grados Celsius.

| Expression | Type | Value |
|----------------------|----------|----------------------|
| T_Sat | char[20] | 0x08000FE8 (Decimal) |
| (x)= [0] | char | 25 (Decimal) |
| (x)= [1] | char | 24 (Decimal) |
| (x)= [2] | char | 26 (Decimal) |
| (x)= [3] | char | 28 (Decimal) |
| (x)= [4] | char | 24 (Decimal) |
| (x)= [5] | char | 23 (Decimal) |
| (x)= [6] | char | 23 (Decimal) |
| (x)= [7] | char | 24 (Decimal) |
| (x)= [8] | char | 24 (Decimal) |
| (x)= [9] | char | 24 (Decimal) |
| (x)= [10] | char | 24 (Decimal) |
| (x)= [11] | char | 23 (Decimal) |
| (x)= [12] | char | 24 (Decimal) |
| (x)= [13] | char | 24 (Decimal) |
| (x)= [14] | char | 24 (Decimal) |
| (x)= [15] | char | 24 (Decimal) |
| (x)= [16] | char | 23 (Decimal) |
| (x)= [17] | char | 23 (Decimal) |
| (x)= [18] | char | 24 (Decimal) |
| (x)= [19] | char | 24 (Decimal) |
| + Add new expression | | |

Figura 4.3: Lectura de temperatura de los 20 sensores

4.1.3. Validación de magnetómetro triaxial

Dentro de los registros de configuración del magnetómetro triaxial se cuenta con 3 valores fijos de solo lectura. Con estos tres valores, el diseñador puede validar que el microcontrolador tiene una correcta comunicación con el sensor mediante el uso del bus I2C. La tabla 3.4 presentada en el capítulo 3, describe a que número de registro corresponden los valores fijos. Para este caso, se procedió a realizar una lectura de todos los registros del magnetómetro y se verificó que los 3 identificadores correspondieran a los valores fijos descritos dentro de la hoja de datos del fabricante. La imagen 4.4 es una captura del ambiente code composer studio presentando la lectura de todos los registros del magnetómetro. En rojo se marcan los identificadores del dispositivo con lo cual se verifica que el dispositivo funciona correctamente y de la misma forma, se asegura una correcta comunicación entre dispositivo y microcontrolador a través del bus I2C.

| Expression | Type | Value |
|------------|-------------------|------------|
| magn | unsigned char[13] | 0x08000FE8 |
| (x)= [0] | unsigned char | 0x10 (Hex) |
| (x)= [1] | unsigned char | 0x20 (Hex) |
| (x)= [2] | unsigned char | 0x03 (Hex) |
| (x)= [3] | unsigned char | 0xFF (Hex) |
| (x)= [4] | unsigned char | 0x85 (Hex) |
| (x)= [5] | unsigned char | 0x00 (Hex) |
| (x)= [6] | unsigned char | 0xA3 (Hex) |
| (x)= [7] | unsigned char | 0xFE (Hex) |
| (x)= [8] | unsigned char | 0x58 (Hex) |
| (x)= [9] | unsigned char | 0x03 (Hex) |
| (x)= [10] | unsigned char | H |
| (x)= [11] | unsigned char | 4 |
| (x)= [12] | unsigned char | 3 |

Figura 4.4: Lectura de registros de magnetómetro

4.1.4. Validación de memorias flash externas

Así como los sensores de temperatura y el magnetómetro cuentan con identificadores dentro de sus registros de configuración para validar una correcta comunicación, las memorias flash operando sobre el protocolo SPI pueden ser validadas de igual forma. La tabla 4.3 fue obtenida de la hoja de datos del fabricante y presenta el formato de ID de las memorias flash externas.

| Núm. bytes | Nombre | Valor |
|------------|-------------------------------|------------------------|
| 1 | ID de manufactura | 20 hexadecimal |
| 2 | ID de dispositivo | |
| | Tipo de memoria | BA hexadecimal |
| | Capacidad | 21 hexadecimal (1Gbit) |
| 17 | ID único | |
| | 1 Byte Longitud de datos | 10 hexadecimal |
| | 2 Byte Información de conf. | Información única |
| | 14 Bytes Datos personalizados | Opcional |

Tabla 4.3: Formato ID de memorias flash externas

Para validar que el microcontrolador tenía una correcta comunicación con las memorias flash, se procedió a hacer la lectura del registro correspondiente en las dos memorias y comparar que los datos leídos coincidieran con los datos mostrados en la tabla 4.3. La imagen 4.5 corresponde a una captura del ambiente code composer studio y presenta la lectura de los identificadores en las memorias flash. En rojo se marca que los 4 primeros bytes en ambas memorias coinciden con el formato de ID previamente descrito. Con esto se verifica que los dos dispositivos funcionan correctamente y que el microcontrolador se comunica exitosamente con ambas memorias por medio del bus serial SPI.

| Expression | Type | Value |
|------------|--------------------|------------------|
| Mem1 | unsigned short[20] | 0x08000FA8 (Hex) |
| (*) [0] | unsigned short | 0x0020 (Hex) |
| (*) [1] | unsigned short | 0x00BA (Hex) |
| (*) [2] | unsigned short | 0x0021 (Hex) |
| (*) [3] | unsigned short | 0x0010 (Hex) |
| (*) [4] | unsigned short | 0x0000 (Hex) |
| (*) [5] | unsigned short | 0x0000 (Hex) |
| (*) [6] | unsigned short | 0x0023 (Hex) |
| (*) [7] | unsigned short | 0x0024 (Hex) |
| (*) [8] | unsigned short | 0x0081 (Hex) |
| (*) [9] | unsigned short | 0x0017 (Hex) |
| (*) [10] | unsigned short | 0x0010 (Hex) |
| (*) [11] | unsigned short | 0x0000 (Hex) |
| (*) [12] | unsigned short | 0x0035 (Hex) |
| (*) [13] | unsigned short | 0x0000 (Hex) |
| (*) [14] | unsigned short | 0x0020 (Hex) |
| (*) [15] | unsigned short | 0x0010 (Hex) |
| (*) [16] | unsigned short | 0x0003 (Hex) |
| (*) [17] | unsigned short | 0x0013 (Hex) |
| (*) [18] | unsigned short | 0x0024 (Hex) |
| (*) [19] | unsigned short | 0x0001 (Hex) |
| Mem2 | unsigned short[20] | 0x08000FD0 (Hex) |
| (*) [0] | unsigned short | 0x0020 (Hex) |
| (*) [1] | unsigned short | 0x00BA (Hex) |
| (*) [2] | unsigned short | 0x0021 (Hex) |
| (*) [3] | unsigned short | 0x0010 (Hex) |
| (*) [4] | unsigned short | 0x0000 (Hex) |
| (*) [5] | unsigned short | 0x0000 (Hex) |
| (*) [6] | unsigned short | 0x0023 (Hex) |
| (*) [7] | unsigned short | 0x0024 (Hex) |
| (*) [8] | unsigned short | 0x0081 (Hex) |
| (*) [9] | unsigned short | 0x0017 (Hex) |
| (*) [10] | unsigned short | 0x0010 (Hex) |
| (*) [11] | unsigned short | 0x0000 (Hex) |
| (*) [12] | unsigned short | 0x0031 (Hex) |
| (*) [13] | unsigned short | 0x0000 (Hex) |
| (*) [14] | unsigned short | 0x0014 (Hex) |
| (*) [15] | unsigned short | 0x0010 (Hex) |
| (*) [16] | unsigned short | 0x0003 (Hex) |
| (*) [17] | unsigned short | 0x0013 (Hex) |
| (*) [18] | unsigned short | 0x007C (Hex) |
| (*) [19] | unsigned short | 0x007D (Hex) |

Figura 4.5: ID de memorias flash

4.2. Interfaz de Estación Terrena

Después de haber sido validado el software y hardware dentro de la computadora de vuelo, se procedió a validar su funcionamiento interactuando con los demás subsistemas. Para lograr esto, se contó con un software de estación terrena el cual nos ayudará a validar un enlace de comunicación inalámbrico entre estación terrena y satélite en las frecuencias UHF/VHF y banda S. De igual forma, con el uso del software de estación terrena se podrá validar que la computadora de vuelo es capaz de realizar el manejo de imágenes e información de los demás subsistemas.

El software de estación terrena cuenta con dos objetivos. El primer objetivo y principal función es generar una interfaz gráfica de fácil empleo para el usuario, permitiendo la interpretación de los datos recibidos y la comunicación entre estación terrena y satélite. El segundo objetivo será validar todas y cada una de las funciones que los subsistemas satelitales realizan individualmente. Para ello, adicional a los comandos operativos previamente mencionados, la computadora de vuelo es capaz de enviar información única por cada subsistema para poder ser validada con niveles de referencia conocidos.

En pocas palabras, con el software de estación terrena se puede verificar que la computadora de vuelo realiza un correcto manejo, almacenado y envío de telemetría e imágenes a estación terrena. Para el caso de las imágenes satelitales, el correcto despliegue de la imagen en el software, valida que la computadora de vuelo realiza exitosamente la captura, almacenado y envío de éstas. En el caso de la telemetría satelital, bastará con almacenar dentro de las memorias externas un patrón de datos conocidos. Posteriormente, éstos datos son enviados a estación terrena y una vez que son interpretados y graficados por el software, se podrá comparar la información almacenada contra

la recibida para así poder validar que la computadora de vuelo puede retener, almacenar, procesar y enviar correctamente toda la telemetría.

El software de estación terrena fue diseñado bajo el ambiente de desarrollo Labview por el Ing. Enrique Ramón Gómez Rosas. Esta aplicación será implantada en un PC y permitirá el intercambio de información con el satélite, haciendo uso de dos puertos de comunicación seriales. Un puerto destinado para la frecuencia UHF/VHF y otro puerto para banda S. Además del ambiente de desarrollo LabView y la PC, también se contó con un radio comercial ICOM IC-9100 que corresponde al radio de transmisión y recepción de datos usado en la estación terrena en las frecuencias UHF y VHF. También se contó con un controlador de nodo terminal (TNC, por sus siglas en inglés) Kantronics 9612 Plus que realiza empaquetamiento de datos mediante el protocolo de comunicación AX.25 y finalmente, se usó otro módulo Microhard n2420, idéntico al presentado en la arquitectura satelital, usado para recepción de datos con banda S.

La imagen 4.6 presenta, en la parte inferior el radio ICOM IC-9100 en modo satélite, la cual permite enviar información en la frecuencia VHF y recibir información por UHF. En la parte superior se muestra el módulo Kantronics 9612, encargado del empaquetamiento del protocolo AX.25.



Figura 4.6: Radio ICOM IC-9100 y TNC Kantronics 9612

La figura 4.7 presenta el uso de los dos puertos seriales en la PC y la interacción o conexiones eléctricas entre el módulo Microhard n2420 para un enlace en banda S y la conexión entre el módulo Kantronics 9612 y radio ICOM-9100 para un enlace en las frecuencias UHF/VHF.

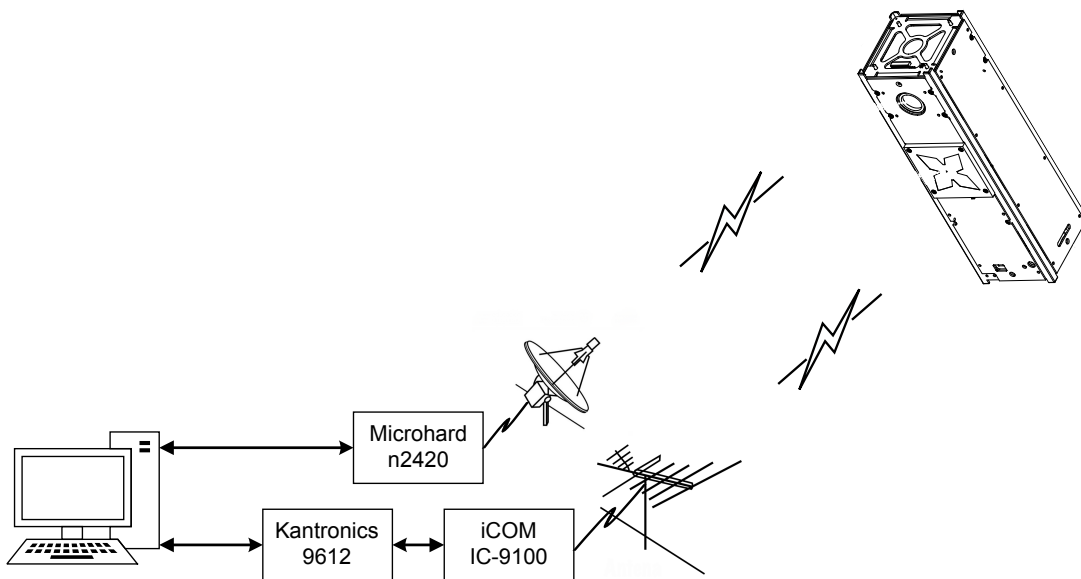


Figura 4.7: Pantalla principal de interfaz de estación terrena

La interfaz de estación terrena está dividida en tres partes, la primera parte es un panel de configuración de puertos serie usados para la frecuencia UHF/VHF y banda S. Aquí el usuario puede configurar el baudrate, número de puerto, etc. La segunda parte permite al usuario ver la información enviada y recibida al satélite en formato hexadecimal. Finalmente, la tercera parte está dividida en 9 pestañas donde cada una permite emitir diferentes comandos al satélite. Toda la interfaz fue diseñada para presentar la telemetría recibida de forma gráfica o en forma numérica. La imagen 4.8 presenta la pantalla principal de la interfaz de estación terrena.

Configuración de puertos serie Datos enviados y recibidos

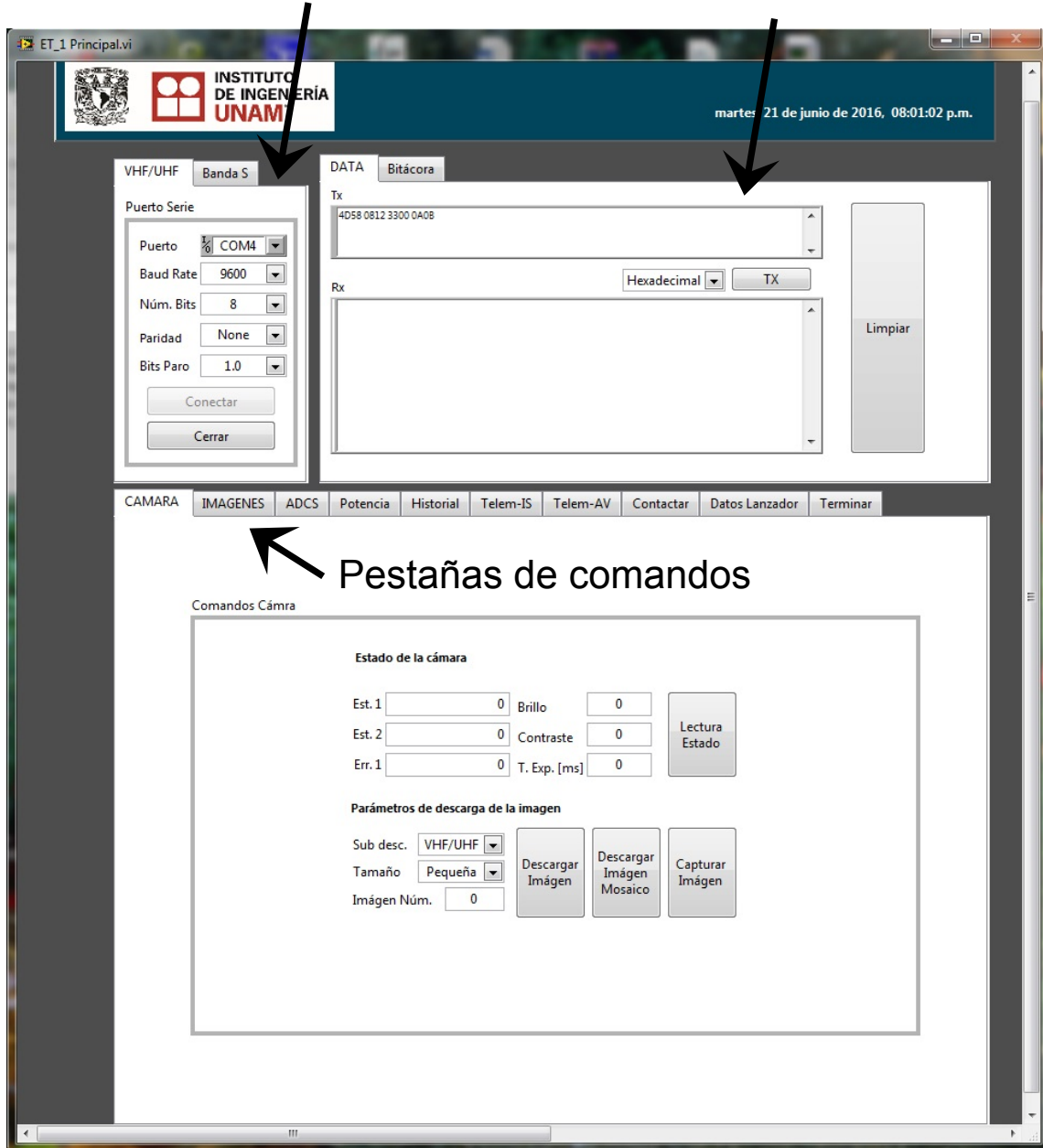


Figura 4.8: Pantalla principal de interfaz de estación terrena

A continuación se describen algunas pestañas, se muestra un ejemplo de la información recibida y la manera en como es interpretada:

Cámara: Esta pestaña esta dedicada a la configuración y manejo de la cámara satelital. Aquí el usuario puede conocer el estado de la cámara, emitir un comando para la captura de imagen definiendo parámetros como tiempo de

exposición, seleccionar y descargar un mosaico de 4 imágenes y finalmente, descargar una imagen en tamaño grande o pequeña por cualquiera de los dos radios, UHF/VHF o banda S. La imagen 4.9 presenta la pestaña dedicada a la configuración y manejo de la cámara satelital.

Comandos Cámara

| | | | | | | |
|--|--------------------------------------|--------------------------------|------------------|--------------------------------|-----------------|--------------------------------|
| Estado de la cámara | Est. 1 | <input type="text" value="0"/> | Brillo | <input type="text" value="0"/> | Lectura Estado | |
| | Est. 2 | <input type="text" value="0"/> | | Contraste | | <input type="text" value="0"/> |
| | Err. 1 | <input type="text" value="0"/> | | T. Exp. [ms] | | <input type="text" value="0"/> |
| Parámetros de descarga de la imagen | | | | | | |
| Sub desc. | <input type="text" value="VHF/UHF"/> | | Descargar Imágen | Descargar Imágen Mosaico | Capturar Imágen | |
| Tamaño | <input type="text" value="Pequeña"/> | | | | | |
| Imágen Núm. | <input type="text" value="0"/> | | | | | |

Figura 4.9: Pestaña para configuración y manejo de cámara satelital

La imagen 4.10 muestra el proceso de descarga de una imagen satelital. A medida que el software de estación terrena recibe cada paquete imagen, ésta despliega los datos contenidos dentro del paquete imagen para ir formando la imagen. Como se puede observar, con esta pestaña se verifica que la computadora de vuelo realiza de manera exitosa el manejo, almacenamiento y envío de imágenes a estación terrena.

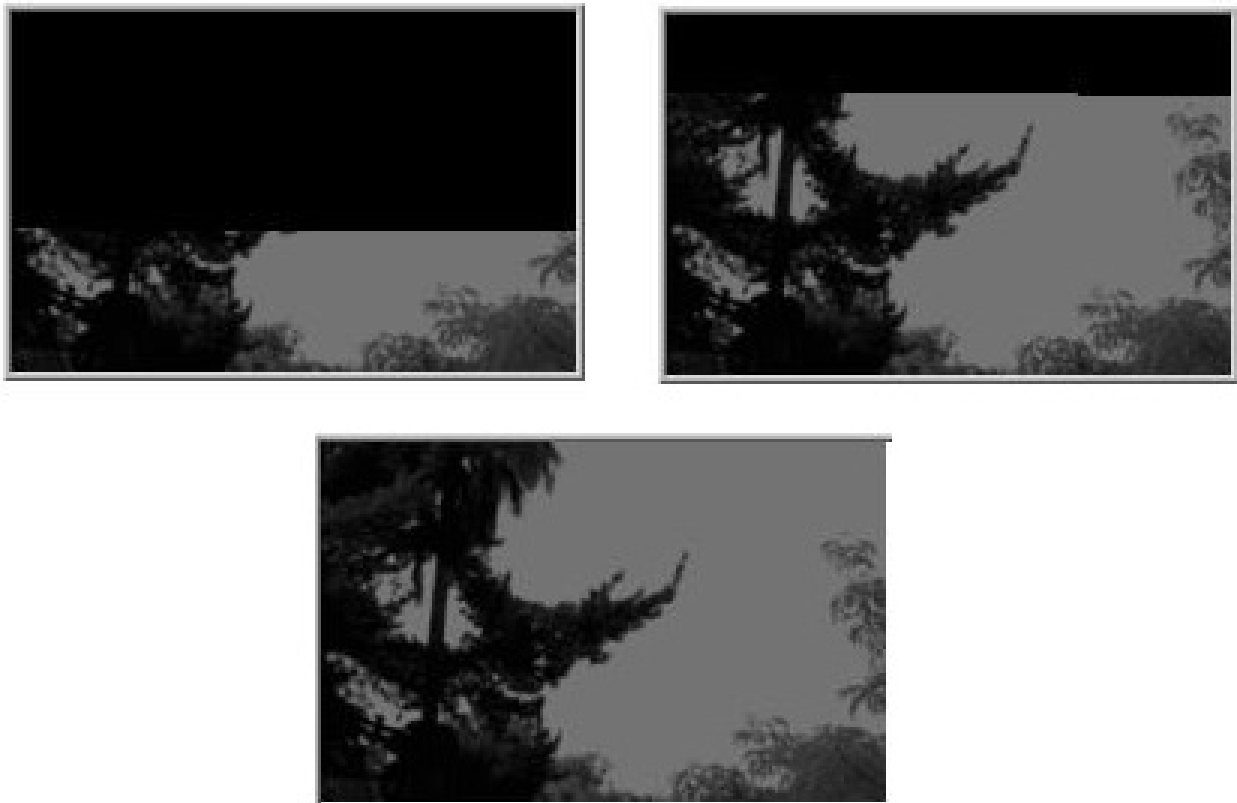


Figura 4.10: Proceso de descarga de imagen satelital

La opción "Descargar Imagen Mosaico" consta de 4 imágenes pequeñas descargadas únicamente por el radio UHF/VHF. Aquí el usuario puede seleccionar 4 imágenes distintas y el objetivo es conocer si el satélite capturó correctamente dichas imágenes. Una vez descargadas las imágenes pequeñas, el usuario evaluará si vale la pena descargar cualquier imagen en tamaño grande basándose en la calidad de ésta y como se describió en el capítulo 3, cuando el satélite captura una imagen, la computadora de vuelo se encarga de almacenar la misma imagen en tamaño grande y pequeño. La imagen 4.11 presenta la exitosa descarga del mosaico de imágenes con lo cual se valida que la computadora de vuelo puede almacenar diversas imágenes, realizar su manejo y envío a la estación terrena. En caso contrario, cualquiera de las imágenes sería desplegada de forma incorrecta.

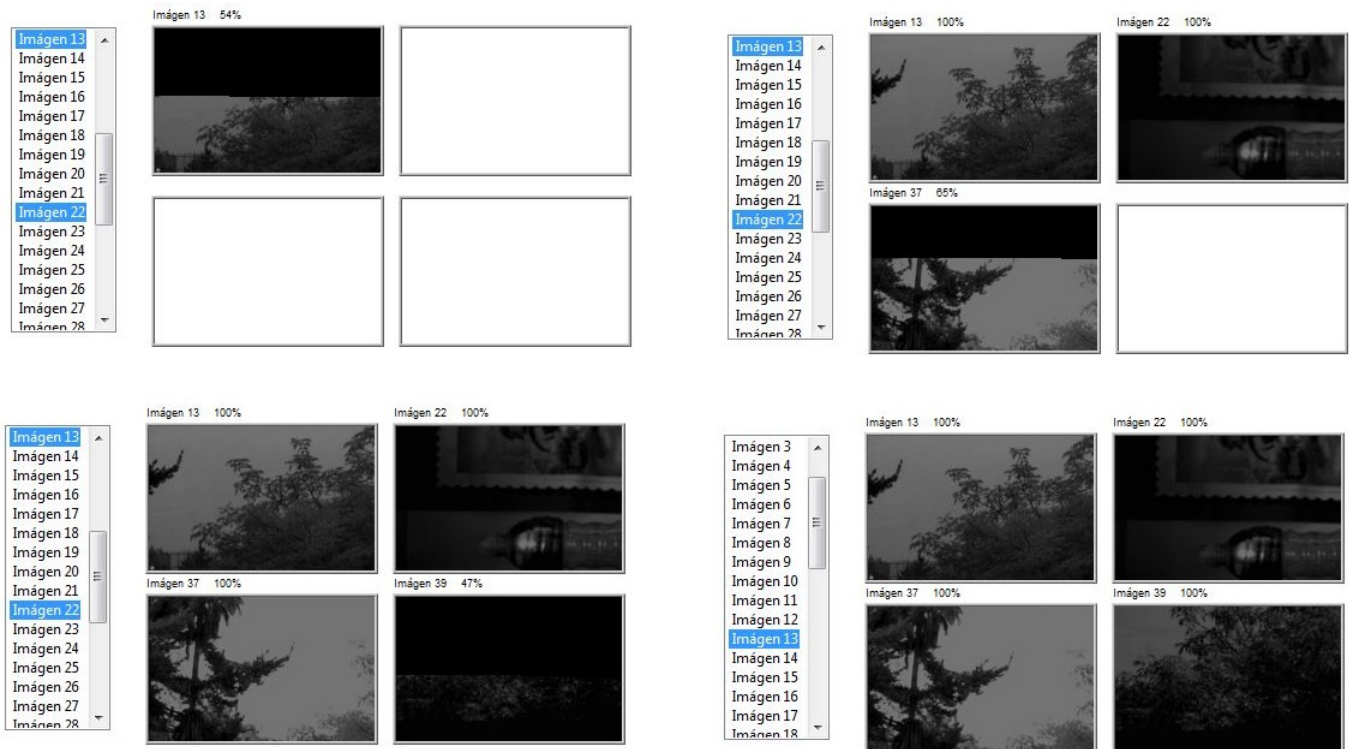


Figura 4.11: Proceso de descarga de mosaico de imágenes

Imágenes: La pestaña de "Imágenes" esta destinada a mostrar la imagen descargada final, permitiendo al usuario guardar la imagen en una carpeta seleccionada.

ADCS: Esta pestaña tiene la finalidad de validar el funcionamiento del subsistema de control y orientación donde el usuario interactúa directamente con el subsistema. El comando es recibido por el satélite, enseguida llega a la computadora de vuelo, la cual verifica el número de comando para después ser reenviado al subsistema de control y orientación. Con ésta pestaña, el usuario puede obtener lecturas directamente de la IMU, girar cualquiera de las tres ruedas inerciales a cierta velocidad y dirección, leer la corriente consumida por cada bobina o rueda inercial y finalmente activar cualquiera de sus bobinas de torque magnético.

Potencia: La pestaña "Potencia" fue diseñada para interactuar directamente con el subsistema de potencia eléctrica. El objetivo es que el equipo de trabajo valide el funcionamiento del subsistema. Esto se logra encendiendo o apagando cualquier subsistema o módulo dentro del satélite y posteriormente realizando una lectura de los voltajes y corrientes. Como se describió en el capítulo 2, el subsistema de potencia eléctrica cuenta con diversas fuentes para la distribución de voltajes y corrientes, por lo que, el equipo de trabajo conoce el valor aproximado para cada subsistema o módulo. La imagen 4.12 muestra una lectura real de voltajes y corrientes de todos los subsistemas y elementos dentro del cubesat. Debido a que las lecturas obtenidas corresponden al valor esperado, se da por exitosa la prueba y con esto se valida que la computadora de vuelo tiene una correcta comunicación con el subsistema de potencia eléctrica así como un correcto manejo de información.

Subsistemas

| | V [V] | I [a] | | V [V] | I [a] |
|----------|-------|---------|----------|-------|---------|
| Panel X+ | 1.386 | 0.007 | COM 3.3 | 3.279 | 0.037 |
| Panel X- | 1.380 | 0.001 | COM 5.0 | 5.312 | 0.057 |
| Panel Y+ | 3.948 | 0.012 | GPS 3.3 | 3.279 | 0.091 |
| Panel Y- | 1.314 | 0.001 | GPS 7.4 | 7.372 | 0.087 |
| C. Vuelo | 3.279 | 0.086 | CAM 5.0 | 5.286 | 0.512 |
| EPS 3.3 | 3.302 | 0.012 | B.S. 3.3 | 3.281 | 0.306 |
| EPS 5.0 | 5.292 | 0.026 | BATT | 7.430 | 0.338 |
| ADS 3.3 | 3.260 | 0.316 | | | |
| ADS 7.4 | 7.392 | 0.051 | | | |

Subsistemas

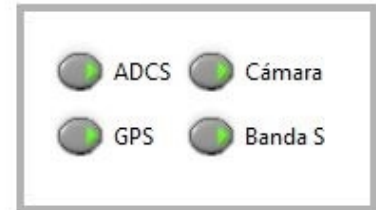


Figura 4.12: Proceso de prueba en subsistema de potencia eléctrica

Telem-IS: La pestaña "Telem-IS" está dedicada al comando operativo de telemetría instantánea, permitiendo conocer el estado de salud del satélite en línea de vista. La imagen 4.13 presenta la pestaña destinada a la telemetría instantánea. En la imagen se muestra una lectura real de éste comando operativo donde la información corresponde a voltajes y corrientes de algunos subsistemas, siete temperaturas, el porcentaje de carga de las baterías y número de imágenes almacenadas. Ya que los valores leídos corresponden con los valores esperados, la operación se da por exitosa validando que la computadora de vuelo se comunica correctamente con potencia eléctrica y de igual forma, los valores de temperatura leídos tienen coherencia con la temperatura ambiente en el momento de la lectura.

| | V [V] | I [a] | | Temperaturas [°C] |
|----------|--------|---------|----------------------|-------------------|
| C. Vuelo | 3.286 | 0.133 | C. Vuelo | 32.000 |
| EPS 3.3 | 3.311 | 0.003 | EPS | 26.000 |
| EPS 5.0 | 5.262 | 0.014 | BAT 1 | 25.000 |
| COM 3.3 | 3.311 | 0.054 | BAT 2 | 25.000 |
| COM 5.0 | 5.288 | 0.053 | COM | 25.000 |
| BATT | 7.447 | 0.074 | SCO 1 | 25.000 |
| | | | SCO 2 | 25.000 |
| % Carga | 17.504 | | Imágenes almacenadas | 60 |



Figura 4.13: Pestaña de telemetría instantánea

Historial: La pestaña de "Historial" está dedicada al comando operativo de telemetría intermedia. Esta pestaña presenta las 400 muestras capturadas y almacenadas por la computadora de vuelo conteniendo información de todos los subsistemas. Para validar que los datos obtenidos y almacenados por el satélite sean correctamente interpretados por el software de estación terrena, se generaron niveles de voltajes, corrientes, temperaturas, etc. con el uso de MatLab versión 2012. Después, los datos generados fueron almacenados en el segmento de memoria flash correspondiente a la telemetría intermedia. Una vez realizado esto, se emitió el comando para descargar las 400 muestras almacenadas y finalmente se compararon las gráficas generadas por el software de estación terrena contra las gráficas generadas con MatLab.

La imagen 4.14 presenta las gráficas generadas por el software de estación terrena a partir de los datos generados con Matlab previamente almacenados en la computadora de vuelo. La imagen 4.15 presenta las gráficas originales generadas en Matlab. Como se puede observar, las gráficas son idénticas con lo cual se verifica que los datos de voltaje y corriente para el panel, interpretados por el software de estación terrena, son correctos.

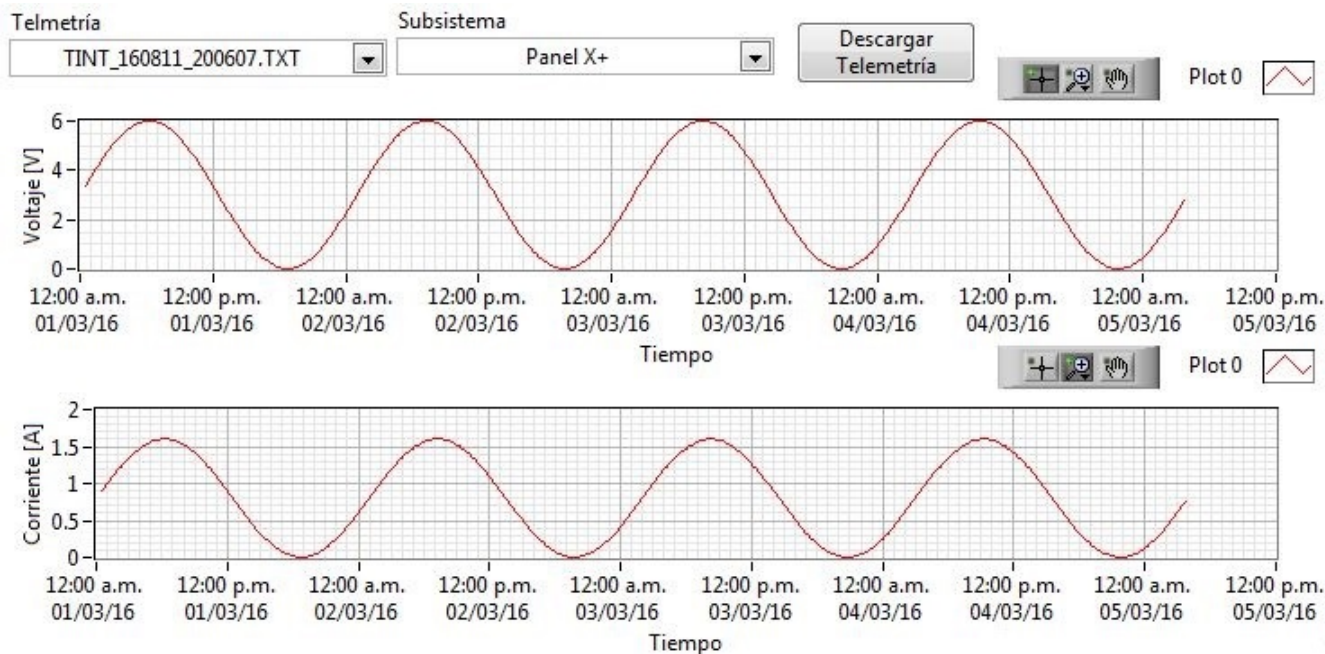


Figura 4.14: Gráficas en software de estación terrena de voltajes y corrientes en panel

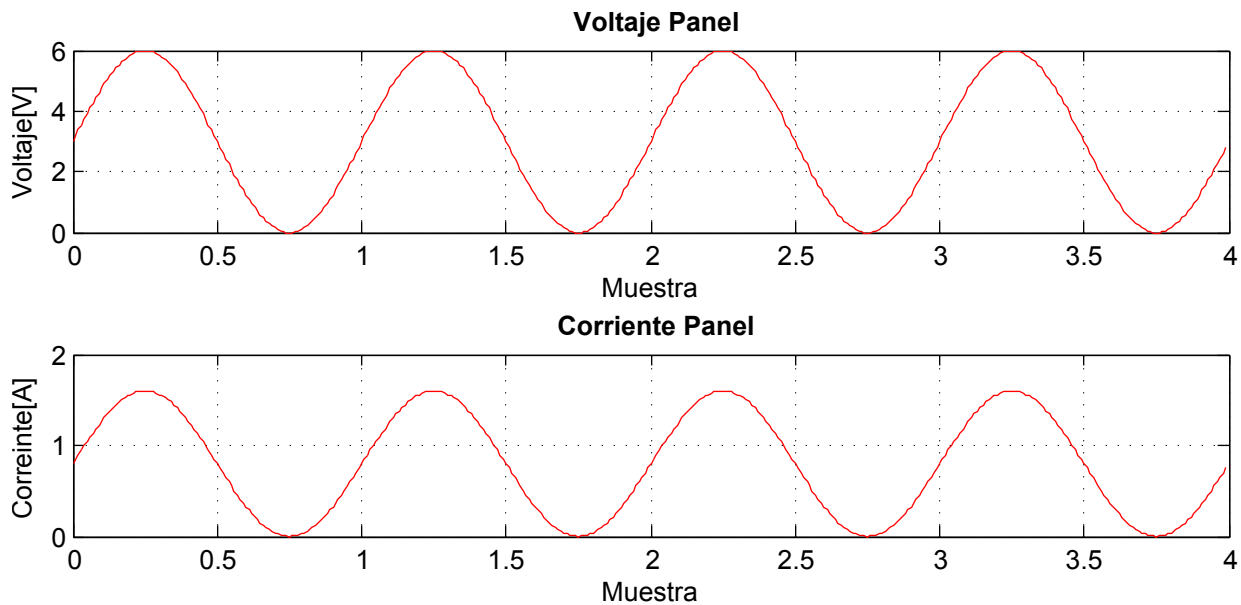


Figura 4.15: Gráfica en Matlab de voltajes y corrientes en panel

La imagen 4.16 presenta las gráficas de voltaje, corriente y cuatro temperaturas generadas por el software de estación terrena a partir de los datos generados con Matlab previamente almacenados en la computadora de vuelo. La imagen 4.17 presenta las gráficas originales generadas con Matlab.

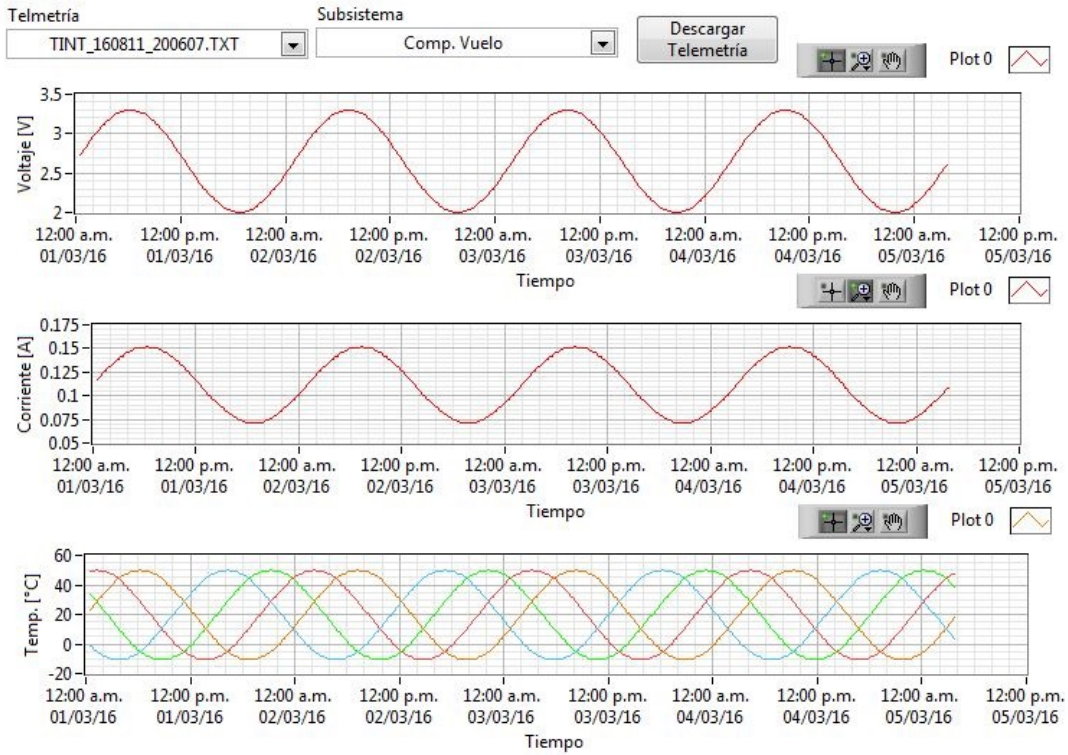


Figura 4.16: Gráficas en software de estación terrena de voltajes, corrientes y temperaturas en C.V.

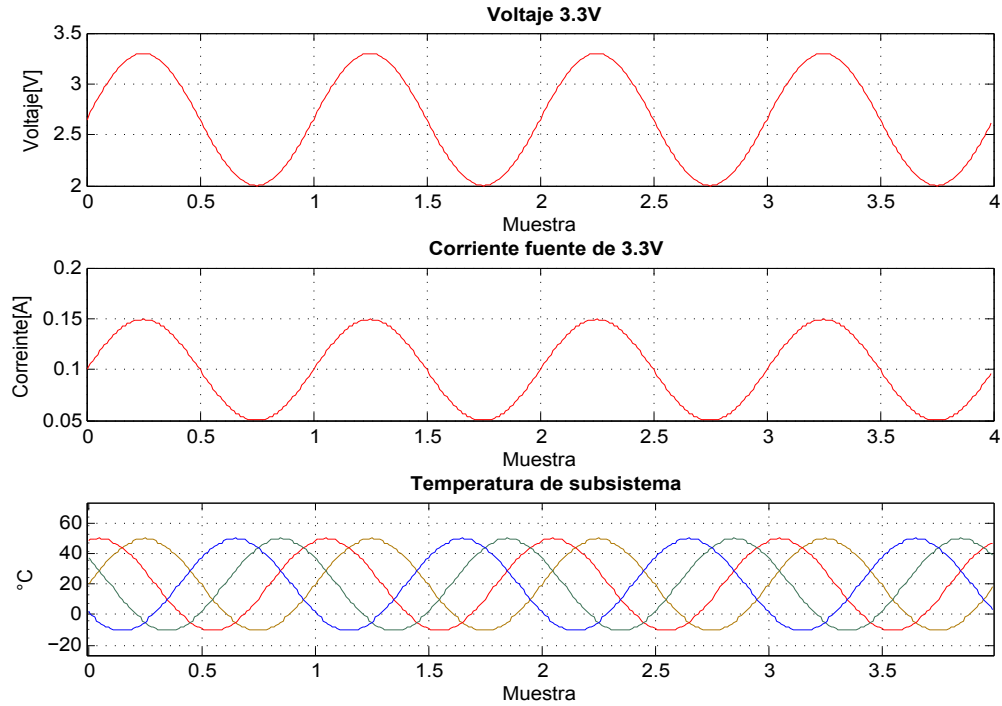


Figura 4.17: Gráficas en Matlab de voltajes, corrientes y temperaturas en la C.V.

La imagen 4.18 presenta las gráficas de voltaje, corriente y porcentaje de carga en las baterías generadas por el software de estación terrena a partir de los datos generados con Matlab. La imagen 4.19 presenta las gráficas originales generadas con Matlab.

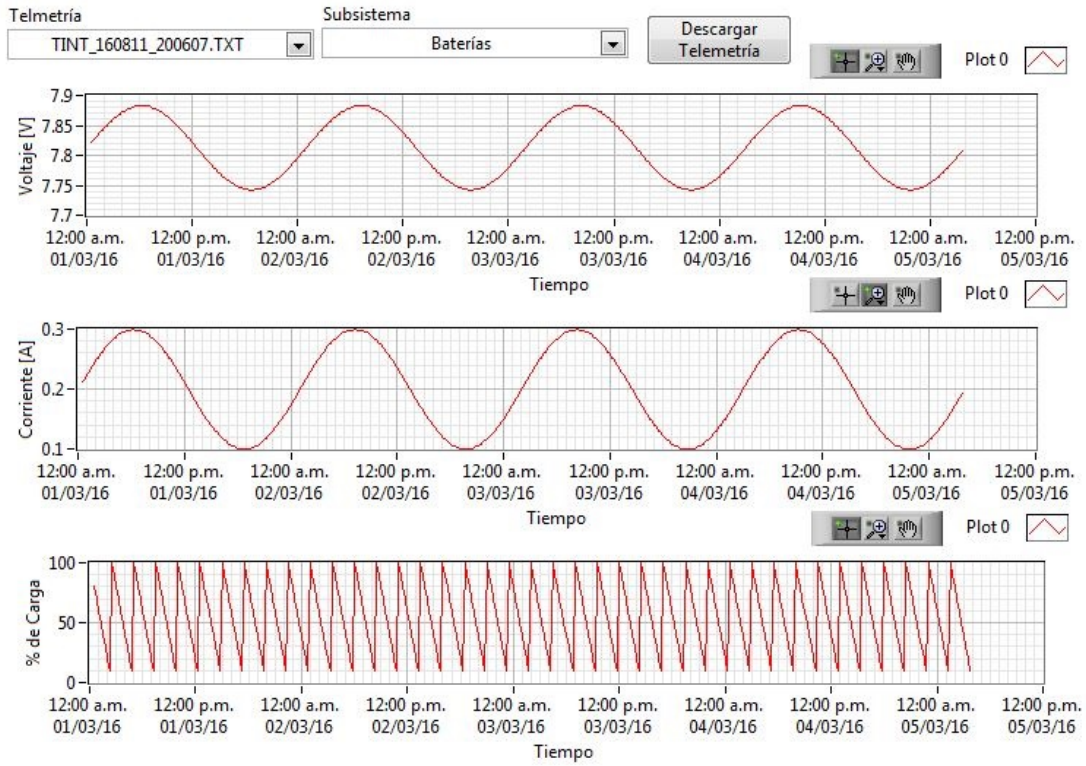


Figura 4.18: Gráficas en software de estación terrena de voltajes, corrientes y porcentaje de carga en baterías

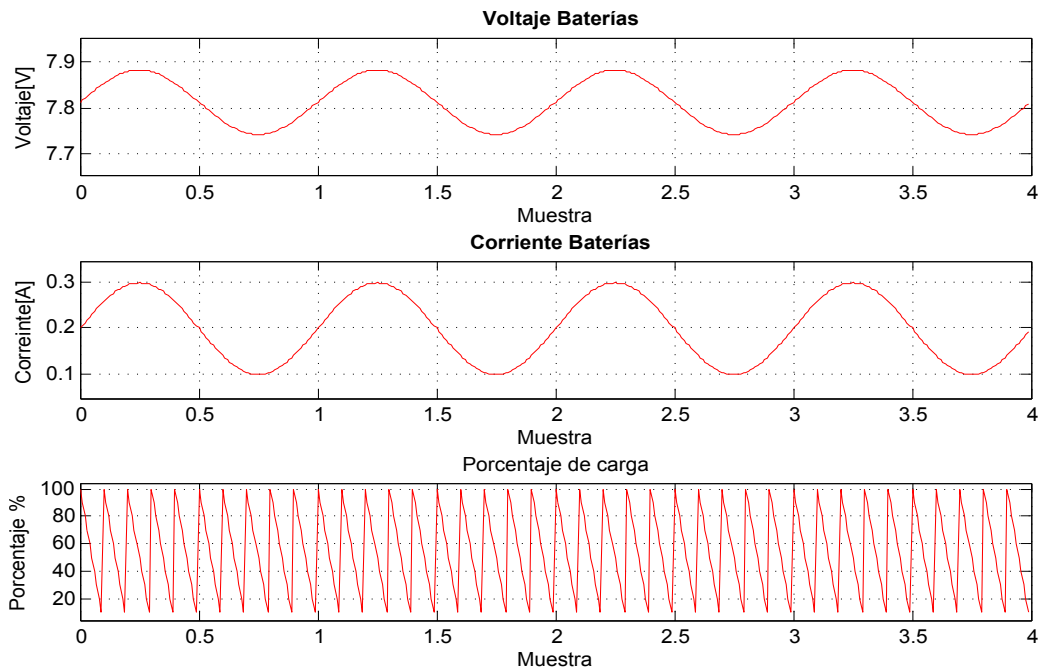


Figura 4.19: Gráficas en Matlab de voltajes, corrientes y porcentaje de carga en baterías

Debido a que la comparación de todas las gráficas antes mostradas son correctas, se da por exitosa la prueba del comando operativo correspondiente a la telemetría intermedia. Con esta prueba se confirma que la computadora de vuelo almacena datos en la sección de memoria correcta y que realiza un buen manejo y envío de datos a la estación terrena. En caso contrario, se podría apreciar en el software de estación terrena que las formas de onda generadas

a partir de la información enviada, no tiene continuidad o son incorrectas.

Telem-AV: La pestaña "Telem-AV" está dedicada al comando operativo de telemetría avanzada. Esta pestaña nos permitirá validar que la computadora de vuelo puede almacenar telemetría con datos del magnetómetro y giroscopio generados por el subsistema de control y orientación. También contiene datos de latitud, longitud y altitud generados por el GPS. La imagen 4.20 presenta de forma gráfica, los valores de magnetómetro y giroscopio generados por el software de estación terrena a partir de datos generados con Matlab. La imagen 4.21 presenta las gráficas originales generadas con Matlab. Como se puede observar, la comparación de las gráficas son correctas y por lo tanto se da por exitosa la prueba correspondiente al comando operativo de telemetría avanzada. En caso que exista un mal manejo y envío de información por parte de la computadora de vuelo, las gráficas no tendrían continuidad o la comparación entre ellas sería incorrecta.

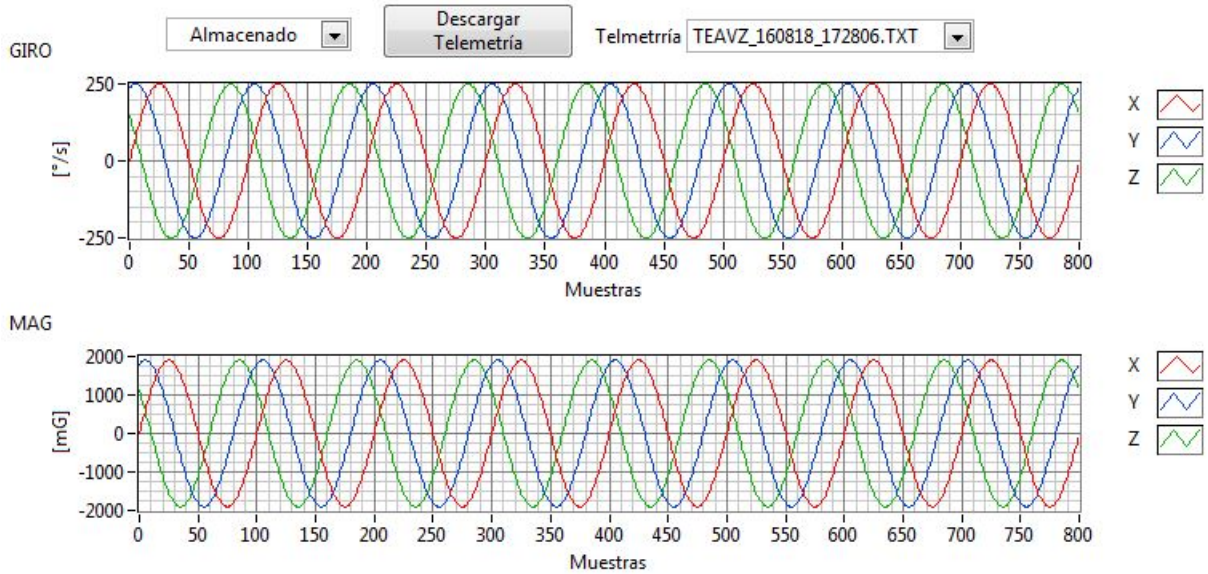


Figura 4.20: Gráficas en software de estación terrena de giroscopio y magnetómetro

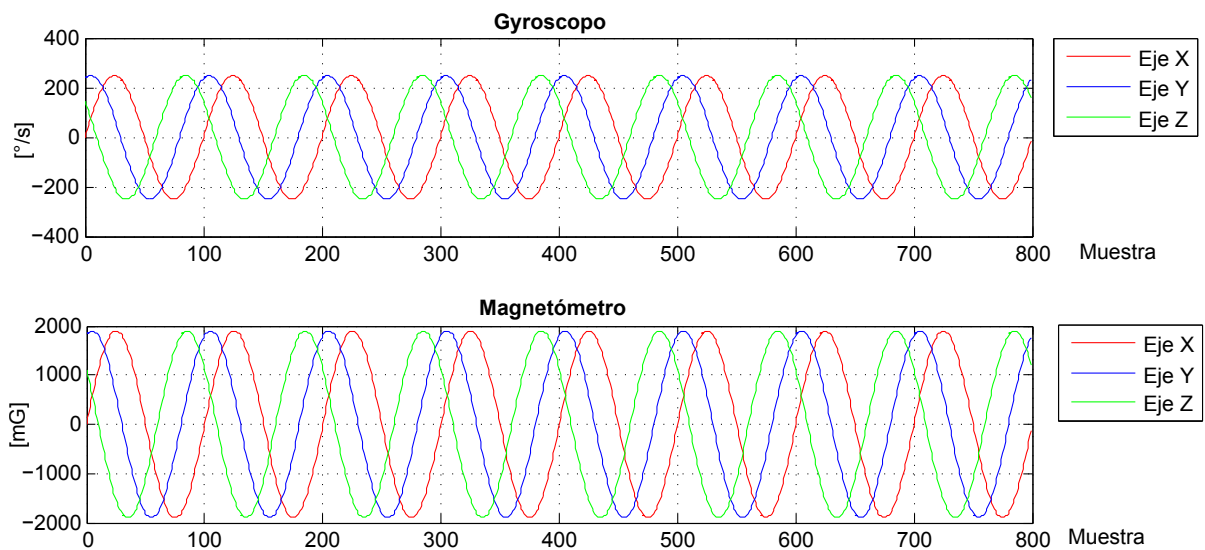


Figura 4.21: Gráficas en Matlab de giroscopio y magnetómetro

4.3. Pruebas de calificación espacial

El proceso de calificación espacial [33] es la fase final en el desarrollo de un cubesat. Todo el hardware satelital debe ser probado y completamente funcional antes de la integración y calificación espacial. Una vez que el cubesat ha pasado exitosamente las pruebas de validación, éste ya no puede ser modificado. En caso de ser modificado, el cubesat deberá completar nuevamente todas las pruebas de validación.

El estándar GSFC-STD-7000 [34] desarrollado por NASA, establece la verificación ambiental de una carga útil, un subsistema o componentes espaciales, sin embargo, en la mayoría de los casos la empresa lanzadora únicamente requiere conocer si el cubesat sobrevivirá al lanzamiento y expulsión con el P-POD, por lo que sólo la prueba de vibración es requerida. La funcionalidad del satélite después de la expulsión es algo que solo debe preocupar a los desarrolladores del satélite. Algunas de las pruebas empleadas por los desarrolladores para validar el correcto funcionamiento de su satélite son: prueba de vibración, prueba de termo vacío o ciclado térmico, prueba de impacto, prueba de interferencia electromagnética y prueba de gasificación o breakout. A continuación se describen algunas de las pruebas antes mencionadas.

Prueba de vibración: La prueba de vibración [33] generalmente se hace antes de la prueba de vacío térmico, permitiendo al ciclado térmico estresar cualquier falla o ruptura creada por la prueba vibratoria. Para esta prueba, el cubesat es colocado en un P-POD o Test-POD y es vibrado en sus tres ejes. Antes y después de cada prueba de vibración aleatoria, el cubesat es vibrado en forma sinusoidal para poder determinar el perfil de respuesta y verificar la funcionalidad del cubesat. Los resultados de la vibración sinusoidal son comparados en base al perfil generado y así se puede determinar si la frecuencia es diferente, indicando que algo está roto o fue quebrado durante la prueba de vibración aleatoria. El perfil de vibración aleatoria debe ser proporcionado por la empresa lanzadora y en caso de no contar con dicho perfil, el cubesat se puede apegar al estándar GSFC-STC-7000.

Vacío térmico: Mediante la prueba de vacío térmico [33] se puede comprobar que el cubesat funcionará en un vacío, que no se sobrecalentará y podrá sobrevivir al lanzamiento. La prueba de vacío térmico inicia colocando al satélite en una presión de al menos 10^{-4} [torr] y realizando una prueba de verificación funcional. Es entonces cuando inicia el ciclado térmico. Para esta prueba, se sugiere generar cinco ciclos de temperatura, manteniendo al satélite por al menos una hora en cada extremo del intervalo de temperatura propuesto. Después de que el satélite alcance el equilibrio de temperatura, se debe generar una prueba funcional del sistema. El ciclado térmico debe ser seleccionado para simular las condiciones del satélite en órbita y en caso de no conocerse el intervalo de temperaturas exacto, se recomienda generar un intervalo de entre -10°C y 40°C .

Prueba de impacto: Un impacto mecánico [35] puede ser descrito como una carga transitoria o un impulso violento en un sistema mecánico. El impacto se caracteriza por ser en un breve periodo de tiempo, con gran amplitud y gran frecuencia de hasta 1 MHz. Para que un satélite pueda alcanzar el espacio, es necesario que sea transportado en un vehículo lanzador y sin importar el tamaño del satélite, éste será expuesto a grandes aceleraciones, vibraciones y un gran número de impactos. Los impactos son debido al encendido de los cohetes dentro del vehículo lanzador. La prueba de impacto pretende simular la fuerza que puede llegar a presentar el satélite al momento de su vuelo al espacio. Para tal fin, la prueba puede consistir en fijar el cubesat en una mesa. Después de esto, se coloca en un péndulo un contrapeso y enseguida se libera para poder ejercer una fuerza o impacto en el cubesat. Finalmente resta realizar pruebas de funcionalidad y verificar que el satélite opera con éxito.

4.3.1. Pruebas de vibración

Como se mencionó con anterioridad, la prueba de vibración es de suma importancia ya que es considerada una prueba necesaria y requerida por la compañía lanzadora. Debido a que México no cuenta hasta ahora con toda la infraestructura necesaria para poder realizar todas las pruebas, el satélite sólo ha sido sometido a la prueba de vibración. La prueba de vibración permitirá al equipo de trabajo verificar los siguientes puntos:

- Validar que todos los circuitos eléctricos en cada subsistema fueron soldados correctamente.
- Verificar que objetos móviles como cables y conectores no interfieran con otros componentes internos pudiendo llegar a dañarse.
- Asegurar que la estructura y el satélite serán capaz de sobrevivir a las vibraciones generadas por el vehículo lanzador.

Las pruebas de vibración estuvieron a cargo del Instituto Politécnico Nacional y dichas pruebas se basaron en el estándar GSFC-STD-7000 [34] desarrollado por NASA. La prueba consistió en someter al satélite a una vibración aleatoria en cada uno de los tres ejes perpendiculares entre sí durante un minuto. Para realizar la prueba vibratoria, el satélite fue colocado en un test-POD que cumple con las mismas dimensiones del P-POD o dispositivo de expulsión. La imagen 4.22 presenta, del lado izquierdo la máquina vibratoria instalada en ESIME Ticoman y del lado derecho, la interacción entre test-POD y satélite.

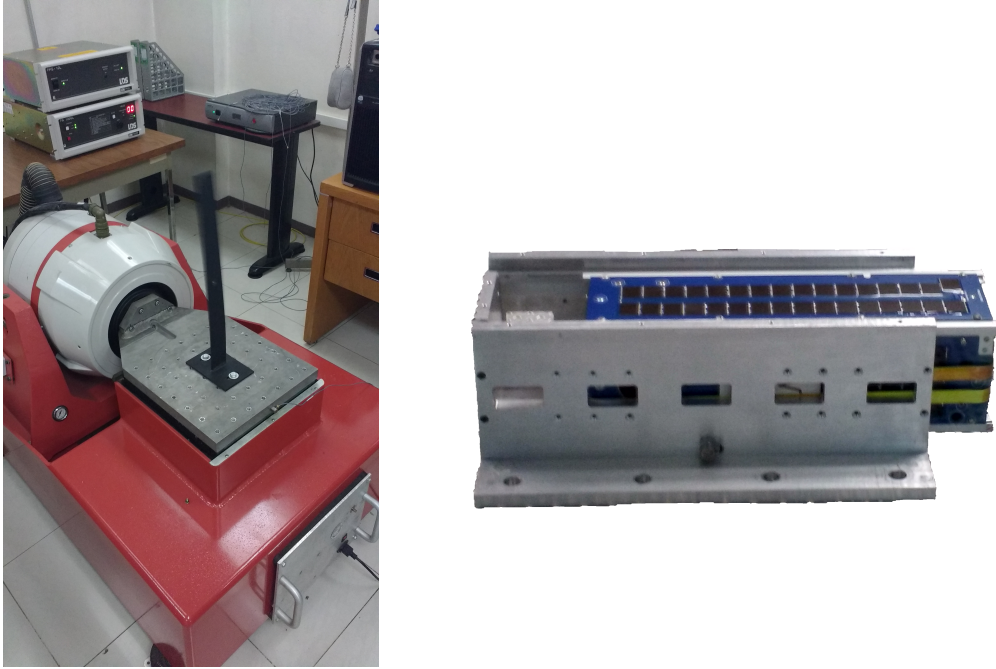


Figura 4.22: Máquina vibratoria y test-POD

A medida que el satélite era vibrado en cada eje perpendicular, el equipo de trabajo de la UNAM realizó mediante el uso del software de estación terrena, pruebas para validar que el satélite continuara funcionando correctamente. Estas pruebas de validación nos permitieron generar una serie de documentos presentados en el apéndice C de este trabajo de tesis, en el cual se registraba cualquier anomalía observada y los valores o datos leídos desde el satélite. El criterio seguido para verificar que el satélite continuara operando correctamente fue el siguiente:

- Ausencia de daños mediante una inspección visual.
- Discrepancias entre los perfiles generados a partir de la prueba sinusoidal, antes y después de la prueba de vibración aleatoria.
- Errores o lecturas de telemetría diferentes a los valores de referencia.

Finalmente, las pruebas de vibración fueron exitosas debido a que el satélite sobrevivió a las vibraciones aleatorias en cada eje propuestas por el estándar GSFC-STD-7000. Cabe mencionar que las pruebas descritas en el estándar corresponden a niveles de vibración por encima de los valores reales. Con esto, queda confirmado que el cubesat y todas las tarjetas electrónicas dentro de éste, tienen la capacidad de sobrevivir a cualquier lanzador que sea contratado. La imagen 4.23 presenta el cubesat y test-POD montados sobre la máquina de vibración en sus ejes X y Z.



Figura 4.23: Vibración de cubesat en eje X y Z

Capítulo 5

Conclusiones y Trabajo futuro

El objetivo principal del presente trabajo de tesis fue diseñar software operativo y hacer uso de diversos protocolos de comunicación para poder obtener y transmitir datos de telemetría y descarga de imágenes a una estación terrena. Todo el software fue implantado dentro de la computadora de vuelo de un satélite cuyas dimensiones, peso y tamaño están especificados dentro del estándar cubesat. La computadora de vuelo está basada en un microcontrolador de 32 bits de Texas Instruments y cuenta con un magnetómetro triaxial, un reloj en tiempo real, diversos sensores de temperatura, protecciones de sobrecorriente contra efecto latch up y dos memorias flash externas.

Por lo tanto, del presente trabajo de tesis se concluyen los siguientes aspectos:

1. Se ensamblaron todos los circuitos electrónicos dentro del subsistema de computadora de vuelo.
2. Mediante el desarrollo de software básico se verificó que todos los componentes y elementos dentro del subsistema funcionaran correctamente. Esto se logró realizando la lectura de registros que contienen identificadores o valores fijos los cuales permiten validar una correcta comunicación entre microcontrolador y dispositivos.
3. Se implementó el bus I2C como principal protocolo de comunicación entre los subsistemas satelitales y, debido a que la computadora de vuelo es el dispositivo maestro dentro del bus, se desarrolló software que impide que éste quede en un modo no operativo ante cualquier problema de comunicación presente.
4. Se desarrolló e implementó exitosamente el protocolo de comunicación UART para ser usado con el módulo de comunicación en banda S. Mediante la descarga de imágenes satelitales en tamaño grande y pequeño se validó su correcto funcionamiento.
5. Se desarrolló e implementó exitosamente el protocolo 1-wire con el fin de realizar la lectura de 20 sensores de temperatura dispersos en todo el cubesat. Mediante la lectura de los ID's en cada sensor y su respectiva lectura de temperatura se validó el correcto funcionamiento del bus.
6. Se logró un correcto control y operación de la cámara satelital permitiendo la captura de imágenes en tamaño grande y pequeño, así como la modificación de sus registros de control.
7. Se desarrolló e implementó exitosamente el protocolo de comunicación serie SPI, tanto en el reloj en tiempo real como en las dos memorias flash NOR. El uso del reloj en tiempo real se validó con la configuración de sus registros de control y con la lectura de la fecha y hora. La comunicación entre microcontrolador y memorias flash se validó con la lectura de identificadores y, posterior a esto, se logró verificar que el microcontrolador almacena de forma correcta telemetría e imágenes satelitales.
8. Se generó estructura de comandos internos y externos para dar confiabilidad, tanto a la comunicación entre subsistemas como a la comunicación con estación terrena.
9. Se definieron, desarrollaron e implementaron comandos operativos dentro de la computadora de vuelo que permiten la captura, manejo, almacenamiento y descarga de imágenes y telemetría satelital. Para verificar que los comandos operativos implementados en la computadora de vuelo funcionaran exitosamente, se hizo uso de un software de estación terrena mediante el cual se puede observar un correcto despliegue de imágenes y telemetría.

10. Se desarrollaron comandos autónomos que debe realizar la computadora de vuelo, de tal forma que el satélite tenga la capacidad de obtener telemetría durante su vuelo orbital o ser desenergizado en caso de porcentaje de baterías bajo y/o presencia de temperaturas extremas.
11. Mediante una prueba de vibración aleatoria realizada, se validó que la computadora de vuelo es capaz de sobrevivir las vibraciones generadas por cualquier vehículo lanzador.

5.1. Trabajo futuro

A continuación se presentan algunas consideraciones que pueden ser incluidas para mejorar la confiabilidad del software y hardware de la computadora de vuelo.

1. En el presente trabajo de tesis se describieron dos de los principales efectos causados por la radiación espacial. El efecto latchup que se manifiesta como un consumo excesivo de corriente, dañando principalmente circuitos con gran escala de integración. El segundo es el efecto upset que se manifiesta cambiando el valor de bits dentro de registros de memoria. Hasta el momento, la computadora de vuelo únicamente está protegida contra el efecto latchup por medio de hardware. En cuanto al efecto upset, éste puede ser mitigado por software haciendo uso de códigos detectores y correctores de errores. Para el caso del microcontrolador principal, éste cuenta con código corrector de único error y detección de doble error en su memoria flash para almacenado de programa y datos. Para el caso de las memorias flash NOR externas, éstas cuentan con protección de datos por software y hardware por lo que también se recomienda implementar esta protección.
2. Todo el software presentado se desarrolló bajo el lenguaje de programación C. Como trabajo futuro se pretende implementar un sistema operativo en tiempo real, el cual puede tener ciertas ventajas sobre el lenguaje de programación usado.
3. Aunque la mayoría de los circuitos que componen a la computadora de vuelo son de grado automotriz, soportando temperaturas de -10°C a $+80^{\circ}\text{C}$, se recomienda realizar la prueba térmica para poder validar un sistema satelital completo.

Apéndice A

Diagrama a bloques de computadora de vuelo

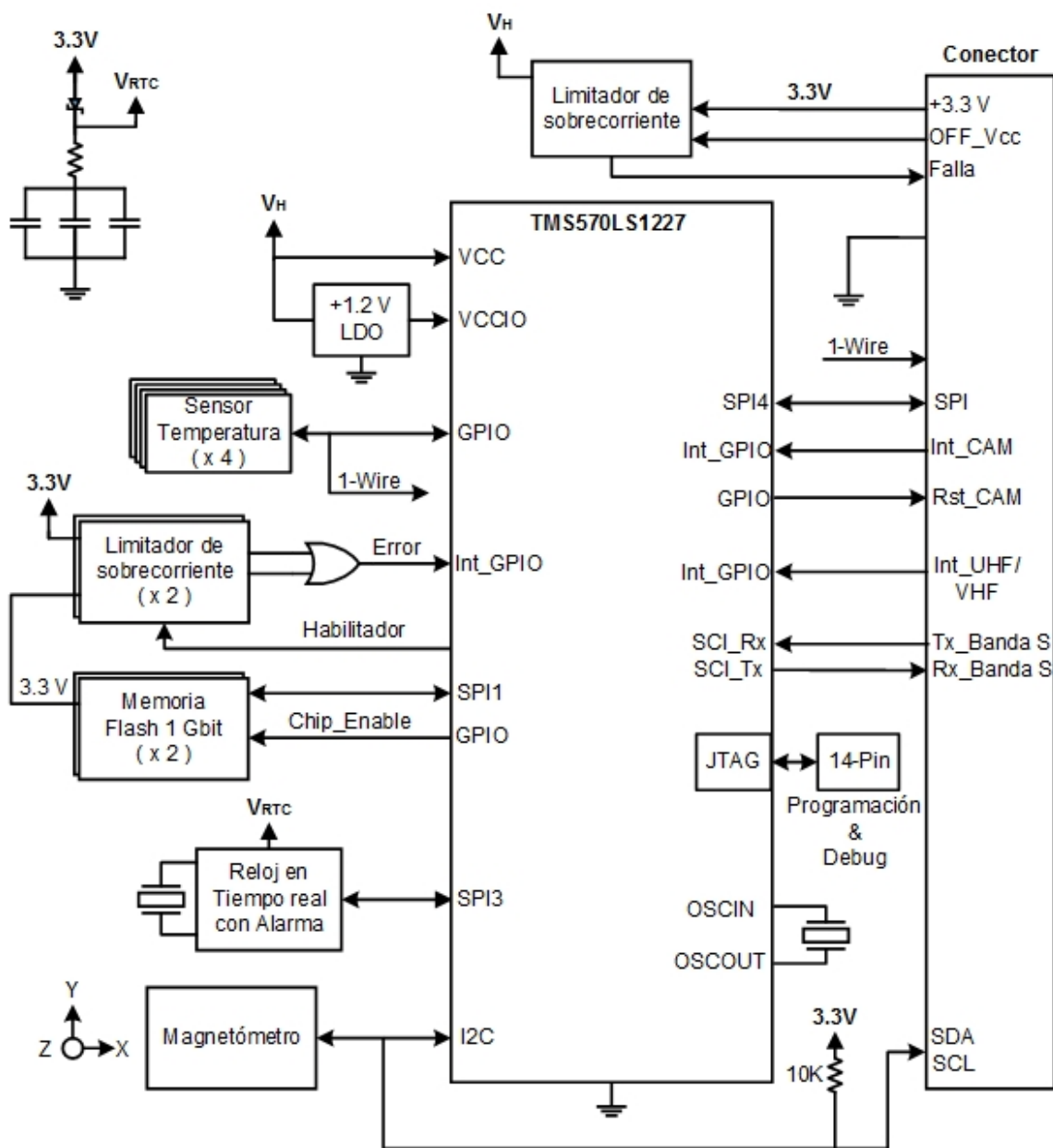


Figura A.1: Diagrama de computadora de vuelo

Apéndice B

Diagrama a bloques de reloj en tiempo real

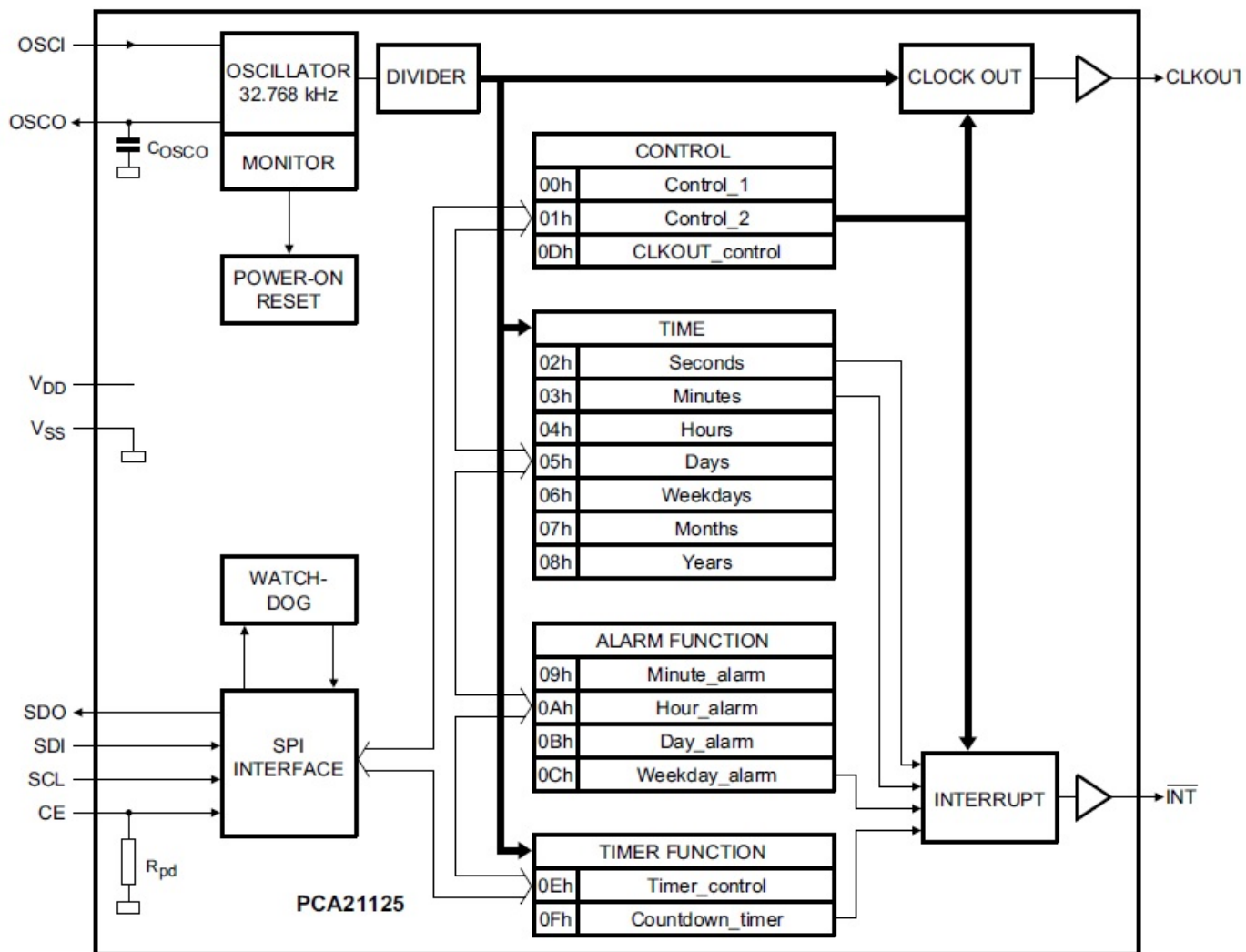



Figura B.1: Diagrama a bloques de reloj en tiempo real

Apéndice C

Documentos generados a partir de pruebas de vibración

Primera Prueba

| | | | |
|--|---|---|--------|
| |  | Formas para validación satelital mediante uso de software de prueba. PNN-UNAM-PPF3.DOCX | |
| | | Fecha: 07/06/2016 Hora Ed: 23:00 | |
| | | Ed.: 0.1 | Pag: 1 |

Prueba o pruebas realizadas: Vibración eje X senoidal y aleatoria

Nombre de personas que realizan prueba: García González Alma

Llenado de forma: Del Moral Perca Bernardo

Fecha y hora en formato HH / MI - DD / MM / AAAA: 3:41 pm 16 / 06 / 16

Lugar de prueba: ESIME Ticoman, Laboratorio IPAE

Forma YY1

Sección A.- Inspección visual de panel

| | Observaciones | Status |
|------------------|------------------------|----------|
| Panel X+ | No se observaron daños | Correcto |
| Panel X- | No se observaron daños | Correcto |
| Panel Y+ | No se observaron daños | Correcto |
| Panel Y- | No se observaron daños | Correcto |
| Resultado Global | | |

Sección B.- Voltajes y corrientes de paneles sin incidencia de luz

| | Voltaje [V] | Corriente [A] |
|----------|-------------|---------------|
| Panel X+ | 1.596 | 0 |
| Panel X- | 1.590 | 0 |
| Panel Y+ | 1.050 | 0 |
| Panel Y- | 1.050 | 0 |

Sección C.- Voltajes y corrientes de paneles con incidencia de luz

| | Voltaje [V] | Corriente [A] |
|----------|-------------|---------------|
| Panel X+ | 6.582 | |
| Panel X- | 3.7 | 0.136 |
| Panel Y+ | 6.97 | 0.257 |
| Panel Y- | 6.936 | 0.292 |

Observaciones

Se hizo uso de foco de 40W, no se observaron daños

Figura C.1: Prueba de vibración en eje X. Hoja 1.


| | | | |
|--|---|---|--------|
| |  | Formas para validación satelital mediante uso de software de prueba. PNN-UNAM-FPF3.DOCX | |
| | | Fecha: 07/06/2016 Hora Ed: 23:00 | |
| | | Ed.: 0.1 | Pag: 2 |

| Forma YY2 | | | | | |
|------------------|--------------|---------------|------------------|---------------|--------|
| | Valor medido | | Valor esperado | | Status |
| | Voltaje [V] | Corriente [A] | Voltaje [V] | Corriente [A] | |
| C. Vuelo | 3.272 | 0.088 | 3.05 - 3.4 | 0.06 - 0.08 | |
| EPS 3.3 | 3.298 | 0.004 | 3.05 - 3.4 | 0.005 - 0.01 | |
| EPS 5.0 | 5.136 | 0.009 | 4.9 - 5.2 | 0.1 - 0.4 | |
| ADS 3.3 | 3.244 | 0.291 | 3.05 - 3.4 | 0.25 - 0.3 | |
| ADS 7.4 | 6.106 | N/A | Voltaje de batt. | N/A | |
| COM 3.3 | 3.269 | 0.014 | 3.05 - 3.4 | 0.005 - 0.02 | |
| COM 5.0 | 5.064 | 1.193 | 4.9 - 5.2 | 0.1 - 0.4 | |
| GPS 3.3 | N/A | N/A | N/A | N/A | N/A |
| GPS 5.0 | 6.468 | 0.006 | Voltaje de batt. | 0.1 - 0.14 | |
| CAM 5.0 | 5.184 | 0.775 | 4.9 - 5.2 | 0.2 - 0.6 | |
| B.S. 3.3 | 3.243 | 0.274 | 3.05 - 3.4 | 0.15 - 0.18 | |
| BATT | 6.718 | 2.764 | 2.9 - 8.4 | 0.3 - 0.8 | |

| Forma YY3 | | | | | |
|------------------|--------------|---------------|------------------|---------------|--------|
| | Valor medido | | Valor esperado | | Status |
| | Voltaje [V] | Corriente [A] | Voltaje [V] | Corriente [A] | |
| Panel X+ | | | N/A | N/A | |
| Panel X- | | | N/A | N/A | |
| Panel Y+ | | | N/A | N/A | |
| Panel Y- | | | N/A | N/A | |
| C. Vuelo | 3.278 | 0.117 | 3.05 - 3.4 | 0.06 - 0.08 | |
| EPS 3.3 | 3.302 | 0.014 | 3.05 - 3.4 | 0.005 - 0.01 | |
| EPS 5.0 | 5.122 | 0.140 | 4.9 - 5.2 | 0 - 0.01 | |
| ADS 3.3 | 0 | 0 | 0 - 0.05 | 0 - 0.01 | |
| ADS 7.4 | 0.014 | N/A | Voltaje de batt. | N/A | |
| COM 3.3 | 3.333 | 0.009 | 0 - 0.05 | 0 - 0.01 | |
| COM 5.0 | 5.264 | 1.201 | 0 - 0.05 | 0 - 0.01 | |
| GPS 3.3 | N/A | N/A | N/A | N/A | N/A |
| GPS 5.0 | 0.512 | 0 | 0 - 0.05 | 0 - 0.01 | |
| CAM 5.0 | 0.002 | 0 | 0 - 0.05 | 0 - 0.01 | |
| B.S. 3.3 | 0.621 | 0 | 0 - 0.01 | 0 - 0.01 | |
| BATT | 7.062 | 1.806 | 2.9 - 8.4 | 0.3 - 0.8 | |

N/A: No aplica

Figura C.2: Prueba de vibración en eje X. Hoja 2.

| | | | |
|--|---|---|--------|
| |  | Formas para validación satelital mediante uso de software de prueba. PNN-UNAM-FPF3.DOCX | |
| | | Fecha: 07/06/2016 Hora Ed: 23:00 | |
| | | Ed.: 0.1 | Pag: 3 |

| Forma YY4 | | | | | | | |
|------------------|--------------|---------------|--------------------|----------------|---------------|-----------|---|
| | Valor medido | | | Valor esperado | | | Status |
| | Voltaje [V] | Corriente [A] | Temp [°C] | Voltaje [V] | Corriente [A] | Temp [°C] | |
| Comp. Vuelo | 3.266 | 0.087 | 31 | 3.05 - 3.4 | 0.06 - 0.08 | 20 - 30 | Correcto |
| EPS 3V3 | 3.298 | 0.008 | 36 | 3.05 - 3.4 | 0.005 - 0.01 | 20 - 30 | Correcto |
| EPS 5V0 | 5.256 | 0.137 | 29 | 4.9 - 5.2 | 0.1 - 0.4 | 20 - 30 | Correcto |
| COM 3V3 | 3.270 | 0.010 | 31 -107 | 3.05 - 3.4 | 0.005 - 0.02 | 20 - 30 | Se observó falla en sensor de temperatura |
| COM 5V0 | 5.188 | 1.168 | 27 | 4.9 - 5.2 | 0.1 - 0.4 | 20 - 30 | Correcto |
| BATERIA | 6.565 | 2.2 | 31 | 2.9 - 8.4 | 0.3 - 0.8 | 20 - 30 | Correcto |

| | | |
|---------|----------------|-----------|
| % Carga | 0.098 (0x00EA) | 0 - 100 % |
|---------|----------------|-----------|

Checar factor de conversión

| Forma YY5 | | | |
|------------------|---|-----------------------------|----------|
| | Nota | Observaciones | Status |
| Giroscopio | Mientras el satélite este inmóvil, la lectura en eje X, Y y Z debe ser aproximado a cero. | Aproximado a cero en 3 ejes | Correcto |
| Acelerómetro | Usted deberá leer un aproximado a 9.8 [m/s ²] en el eje Z mientras el satélite este en posición vertical. | Sin errores | Correcto |
| Magnetómetro | Realice 3 lecturas moviendo el satélite por cada una. Las lecturas deberán diferir por cada movimiento. | Sin errores | Correcto |

| | Valor medido | Valor esperado | Status |
|-------------|------------------|-------------------|----------|
| Presión | 0.78668 [bar] | 0.77 - 0.79 [bar] | |
| Temperatura | [°C] | 20 - 30 [°C] | |
| Diagnóstico | 0080 hexadecimal | 0080 hexadecimal | Correcto |

Figura C.3: Prueba de vibración en eje X. Hoja 3.

| | | |
|--|---|---|
| |  | Formas para validación satelital mediante uso de software de prueba. PNN-UNAM- FPF3.DOCX |
| | | Fecha: 07/06/2016 Hora Ed: 23:00 |
| | | Ed.: 0.1 Pag: 4 |

| Forma YY6 | | | |
|------------------|-------------------------|---------------------------|----------|
| Observaciones | | | |
| | Sentido de giro derecha | Sentido de giro izquierda | Status |
| Motor X | No se observó daño | No se observó daño | Correcto |
| Motor Y | No se observó daño | No se observó daño | Correcto |
| Motor Z | No se observó daño | No se observó daño | Correcto |

| Forma YY7 | | | | | | | | | | | | | | | | | | | | |
|------------------|-------|---|---|---|---|---|---|---|-----|---|---|---|---|---|---|---|---|---|---|---|
| Valor esperado | D:SYB | 1 | 2 | 3 | M | 1 | 2 | 3 | syb | 1 | 2 | 3 | m | 1 | 2 | 3 | - | © | & | o |
| Valor capturado | D:SYB | | | | M | | | | syb | | | | m | | | | - | © | & | o |


| Observaciones |
|---------------|
| |

| Forma YY8 | | | |
|---------------------|-----------------|----------------|--------|
| Ambiente de prueba | Interior | Exterior | |
| Marque la adecuada: | | | |
| | Valor capturado | Valor esperado | Status |
| Estado | | Y | |
| Latitud | | N/A | |
| Longitud | | N/A | |
| Altitud | | N/A | |
| Fecha | | N/A | |
| Hora | | N/A | |

| Observaciones |
|-------------------------------------|
| Checar GPS, responde 00 hexadecimal |

Figura C.4: Prueba de vibración en eje X. Hoja 4.

Segunda prueba.

| | | | |
|--|---|---|--------|
| |  | Formas para validación satelital mediante uso de software de prueba. PNN-UNAM-FPF3.DOCX | |
| | | Fecha: 07/06/2016 Hora Ed: 23:00 | |
| | | Ed.: 0.1 | Pag: 1 |

Prueba o pruebas realizadas: Vibración eje Y senoidal y aleatoria

Nombre de personas que realizan prueba: García González Alma

Elencado de formas: Ramírez Santiago Pedro

Fecha y hora en formato HH / MI - DD / MM / AAAA: 7:15 pm 16/06/16

Lugar de prueba: Laboratorio IPAE ESIME Ticomán.

Forma YY1

| Sección A.- Inspección visual de panel | | |
|--|--|----------|
| | Observaciones | Status |
| Panel X+ | Sin daños en la celda con rayadura en el epóxico | |
| Panel X- | No se observaron daños | Correcto |
| Panel Y+ | No se observaron daños | Correcto |
| Panel Y- | No se observaron daños | Correcto |
| Resultado Global | | |


| Sección B.- Voltajes y corrientes de paneles sin incidencia de luz | | |
|--|-------------|---------------|
| | Voltaje [V] | Corriente [A] |
| Panel X+ | 1.566 | 0 |
| Panel X- | 1.428 | 0 |
| Panel Y+ | 1.170 | 0 |
| Panel Y- | 1.164 | 0 |

| Sección C.- Voltajes y corrientes de paneles con incidencia de luz | | |
|--|-------------|---------------|
| | Voltaje [V] | Corriente [A] |
| Panel X+ | 6.234 | 0.209 |
| Panel X- | 6.198 | 0.010 |
| Panel Y+ | 6.570 | 0.183 |
| Panel Y- | 6.558 | 0.264 |

| Observaciones |
|--|
| Se hizo uso de foco de 40W y lampara led de 8W |

Figura C.5: Prueba de vibración en eje Y. Hoja 1.

Segunda prueba

| | | | |
|--|---|---|--------|
| |  | Formas para validación satelital mediante uso de software de prueba. PNN-UNAM-FPF3.DOCX | |
| | | Fecha: 07/06/2016 Hora Ed: 23:00 | |
| | | Ed.: 0.1 | Pag: 2 |


| Forma YY2 | Valor medido | | Valor esperado | | Status |
|-----------|--------------|---------------|------------------|---------------|--------|
| | Voltaje [V] | Corriente [A] | Voltaje [V] | Corriente [A] | |
| C. Vuelo | 3.268 | 0.068 | 3.05 - 3.4 | 0.06 - 0.08 | |
| EPS 3.3 | 3.299 | 0.066 | 3.05 - 3.4 | 0.005 - 0.01 | |
| EPS 5.0 | 5.132 | 0.012 | 4.9 - 5.2 | 0.1 - 0.4 | |
| ADS 3.3 | 3.244 | 0.288 | 3.05 - 3.4 | 0.25 - 0.3 | |
| ADS 7.4 | 5.643 | N/A | Voltaje de batt. | N/A | |
| COM 3.3 | 3.269 | 0.011 | 3.05 - 3.4 | 0.005 - 0.02 | |
| COM 5.0 | 5.056 | 1.167 | 4.9 - 5.2 | 0.1 - 0.4 | |
| GPS 3.3 | N/A | N/A | N/A | N/A | N/A |
| GPS 5.0 | 5.636 | 0.005 | Voltaje de batt. | 0.1 - 0.14 | |
| CAM 5.0 | 5.062 | 0.779 | 4.9 - 5.2 | 0.2 - 0.6 | |
| B.S. 3.3 | 3.244 | 0.292 | 3.05 - 3.4 | 0.15 - 0.18 | |
| BATT | 6.055 | 2.454 | 2.9 - 8.4 | 0.3 - 0.8 | |

| Forma YY3 | Valor medido | | Valor esperado | | Status |
|-----------|--------------|---------------|------------------|---------------|--------|
| | Voltaje [V] | Corriente [A] | Voltaje [V] | Corriente [A] | |
| Panel X+ | | | N/A | N/A | |
| Panel X- | | | N/A | N/A | |
| Panel Y+ | | | N/A | N/A | |
| Panel Y- | | | N/A | N/A | |
| C. Vuelo | 3.301 | 0.151 | 3.05 - 3.4 | 0.06 - 0.08 | |
| EPS 3.3 | 3.340 | 0.004 | 3.05 - 3.4 | 0.005 - 0.01 | |
| EPS 5.0 | 5.202 | 0.048 | 4.9 - 5.2 | 0 - 0.01 | |
| ADS 3.3 | 0.002 | 0 | 0 - 0.05 | 0 - 0.01 | |
| ADS 7.4 | 0.008 | N/A | Voltaje de batt. | N/A | |
| COM 3.3 | 3.309 | 0.011 | 0 - 0.05 | 0 - 0.01 | |
| COM 5.0 | 5.192 | 1.198 | 0 - 0.05 | 0 - 0.01 | |
| GPS 3.3 | N/A | N/A | N/A | N/A | N/A |
| GPS 5.0 | 0.408 | 0 | 0 - 0.05 | 0 - 0.01 | |
| CAM 5.0 | 0 | 0 | 0 - 0.05 | 0 - 0.01 | |
| B.S. 3.3 | 0.620 | 0 | 0 - 0.01 | 0 - 0.01 | |
| BATT | 6.055 | 0.984 | 2.9 - 8.4 | 0.3 - 0.8 | |

N/A: No aplica

Figura C.6: Prueba de vibración en eje Y. Hoja 2.

Segunda prueba

| | | | |
|--|---|---|--------|
| |  | Formas para validación satelital mediante uso de software de prueba. PNN-UNAM-FPF3.DOCX | |
| | | Fecha: 07/06/2016 Hora Ed: 23:00 | |
| | | Ed.: 0.1 | Pag: 3 |

| Forma YY4 | Valor medido | | | Valor esperado | | | Status |
|-------------|--------------|---------------|-----------|----------------|---------------|-----------|--------|
| | Voltaje [V] | Corriente [A] | Temp [°C] | Voltaje [V] | Corriente [A] | Temp [°C] | |
| Comp. Vuelo | 3.299 | 0.124 | 30 | 3.05 - 3.4 | 0.06 - 0.08 | 20 - 30 | |
| EPS 3V3 | 3.332 | 0.004 | 29 | 3.05 - 3.4 | 0.005 - 0.01 | 20 - 30 | |
| EPS 5V0 | 5.22 | 0.075 | 30 | 4.9 - 5.2 | 0.1 - 0.4 | 20 - 30 | |
| COM 3V3 | 3.329 | 0.15 | 29 | 3.05 - 3.4 | 0.005 - 0.02 | 20 - 30 | |
| COM 5V0 | 5.062 | 1.191 | 28 | 4.9 - 5.2 | 0.1 - 0.4 | 20 - 30 | |
| BATERIA | 6.317 | 0.648 | 30 | 2.9 - 8.4 | 0.3 - 0.8 | 20 - 30 | |

*


| | |
|---------|-----------|
| % Carga | 0 - 100 % |
|---------|-----------|

| Forma YY5 | Nota | Observaciones | Status |
|--------------|---|--------------------------|--------|
| Giroscopio | Mientras el satélite este inmóvil, la lectura en eje X, Y y Z debe ser aproximado a cero. | Aproximado a 0 en 3 ejes | |
| Acelerómetro | Usted deberá leer un aproximado a 9.8 [m/s ²] en el eje Z mientras el satélite este en posición vertical. | Sin errores | |
| Magnetómetro | Realice 3 lecturas moviendo el satélite por cada una. Las lecturas deberán diferir por cada movimiento. | Sin errores | |

| | Valor medido | Valor esperado | Status |
|-------------|------------------|-------------------|--------|
| Presión | 0.78642 [bar] | 0.77 - 0.79 [bar] | |
| Temperatura | [°C] | 20 - 30 [°C] | |
| Diagnóstico | 0080 hexadecimal | 0080 hexadecimal | |

Figura C.7: Prueba de vibración en eje Y. Hoja 3.

Segunda prueba.

| | | | |
|--|---|---|--------|
| |  | Formas para validación satelital mediante uso de software de prueba. PNN-UNAM-FPF3.DOCX | |
| | | Fecha: 07/06/2016 Hora Ed: 23:00 | |
| | | Ed.: 0.1 | Pag: 4 |

| Forma YY6 | | | |
|---------------|-------------------------|---------------------------|--------|
| Observaciones | | | |
| | Sentido de giro derecha | Sentido de giro izquierda | Status |
| Motor X | Sin errores | No se observó daño | |
| Motor Y | Sin errores | No se observó daño | |
| Motor Z | Sin errores | No se observó daño | |

| Forma YY7 | | | | | | | | | | | | | | | | | | | | |
|-----------------|-------|---|---|---|---|---|---|---|-----|---|---|---|---|---|---|---|---|---|---|---|
| Valor esperado | D:SYB | 1 | 2 | 3 | M | 1 | 2 | 3 | syb | 1 | 2 | 3 | m | 1 | 2 | 3 | - | © | & | o |
| Valor capturado | D:SYB | | | | M | | | | syb | | | | m | | | | - | © | & | o |

| Observaciones |
|---------------|
| |


| Forma YY8 | | | |
|---------------------|-----------------|----------------|--------|
| Ambiente de prueba | Interior | Exterior | |
| Marque la adecuada: | | | |
| | Valor capturado | Valor esperado | Status |
| Estado | | Y | |
| Latitud | | N/A | |
| Longitud | | N/A | |
| Altitud | | N/A | |
| Fecha | | N/A | |
| Hora | | N/A | |

| Observaciones |
|--------------------------------|
| - Checar GPS valor arrojado 00 |

> se logró enlace con banda 5

Figura C.8: Prueba de vibración en eje Y. Hoja 4.

Tercera prueba.

| | | | |
|--|---|---|--------|
| |  | Formas para validación satelital mediante uso de software de prueba. PNN-UNAM-PPF3.DOCX | |
| | | Fecha: 07/06/2016 Hora Ed: 23:00 | |
| | | Ed.: 0.1 | Pag: 1 |

Prueba o pruebas realizadas: Vibración eje Z senoidal y aleatoria

Nombre de personas que realizan prueba: García González Alma

Llenado de formato: Ramírez Santiago Pedro

Fecha y hora en formato HH / MI - DD / MM / AAAA: 12:55 pm 17/06/16

Lugar de prueba: Laboratorio IPAE ESIME Ticomán

Forma YY1

Sección A.- Inspección visual de panel

| | Observaciones | Status |
|------------------|--|--------|
| Panel X+ | Sin daños en la celda con rayadura en el epóxico | |
| Panel X- | No se observaron daños. | |
| Panel Y+ | No se observaron daños. | |
| Panel Y- | No se observaron daños | |
| Resultado Global | | |

Sección B.- Voltajes y corrientes de paneles sin incidencia de luz

| | Voltaje [V] | Corriente [A] |
|----------|-------------|---------------|
| Panel X+ | 1.344 | 0 |
| Panel X- | 1.350 | 0 |
| Panel Y+ | 1.044 | 0 |
| Panel Y- | 1.044 | 0 |

Sección C.- Voltajes y corrientes de paneles con incidencia de luz


| | Voltaje [V] | Corriente [A] |
|----------|-------------|---------------|
| Panel X+ | 6.612 | 0.115 |
| Panel X- | 9.264 | 0.012 |
| Panel Y+ | 9.054 | 0.297 |
| Panel Y- | 12.3 | 0.263 |

Observaciones

Se utilizó foco de 40W y lámpara led de 8W

Figura C.9: Prueba de vibración en eje Z. Hoja 1.

Tercera prueba

| | | | |
|--|---|---|--------|
| |  | Formas para validación satelital mediante uso de software de prueba. PNN-UNAM- FPF3.DOCX | |
| | | Fecha: 07/06/2016 Hora Ed: 23:00 | |
| | | Ed.: 0.1 | Pag: 2 |


| Forma YY2 | Valor medido | | Valor esperado | | Status |
|-----------|--------------|---------------|------------------|---------------|--------|
| | Voltaje [V] | Corriente [A] | Voltaje [V] | Corriente [A] | |
| C. Vuelo | 3.271 | 0.079 | 3.05 - 3.4 | 0.06 - 0.08 | |
| EPS 3.3 | 3.298 | 0.005 | 3.05 - 3.4 | 0.005 - 0.01 | |
| EPS 5.0 | 5.12 | 0.004 | 4.9 - 5.2 | 0.1 - 0.4 | |
| ADS 3.3 | 3.243 | 0.288 | 3.05 - 3.4 | 0.25 - 0.3 | |
| ADS 7.4 | 6.342 | N/A | Voltaje de batt. | N/A | |
| COM 3.3 | 3.268 | 0.010 | 3.05 - 3.4 | 0.005 - 0.02 | |
| COM 5.0 | 5.132 | 0.162 | 4.9 - 5.2 | 0.1 - 0.4 | |
| GPS 3.3 | N/A | N/A | N/A | N/A | N/A |
| GPS 5.0 | 6.280 | 0 | Voltaje de batt. | 0.1 - 0.14 | |
| CAM 5.0 | 5.138 | 0.772 | 4.9 - 5.2 | 0.2 - 0.6 | |
| B.S. 3.3 | 3.245 | 0.264 | 3.05 - 3.4 | 0.15 - 0.18 | |
| BATT | 6.570 | 1.802 | 2.9 - 8.4 | 0.3 - 0.8 | |

| Forma YY3 | Valor medido | | Valor esperado | | Status |
|-----------|--------------|---------------|------------------|---------------|--------|
| | Voltaje [V] | Corriente [A] | Voltaje [V] | Corriente [A] | |
| Panel X+ | | | N/A | N/A | |
| Panel X- | | | N/A | N/A | |
| Panel Y+ | | | N/A | N/A | |
| Panel Y- | | | N/A | N/A | |
| C. Vuelo | 3.279 | 0.121 | 3.05 - 3.4 | 0.06 - 0.08 | |
| EPS 3.3 | 3.298 | 0.009 | 3.05 - 3.4 | 0.005 - 0.01 | |
| EPS 5.0 | 5.15 | 0.132 | 4.9 - 5.2 | 0 - 0.01 | |
| ADS 3.3 | 0 | 0 | 0 - 0.05 | 0 - 0.01 | |
| ADS 7.4 | 0.014 | N/A | Voltaje de batt. | N/A | |
| COM 3.3 | 3.3 | 0.010 | 0 - 0.05 | 0 - 0.01 | |
| COM 5.0 | 5.066 | 1.198 | 0 - 0.05 | 0 - 0.01 | |
| GPS 3.3 | N/A | N/A | N/A | N/A | N/A |
| GPS 5.0 | 0.266 | 0 | 0 - 0.05 | 0 - 0.01 | |
| CAM 5.0 | 0 | 0 | 0 - 0.05 | 0 - 0.01 | |
| B.S. 3.3 | 0.620 | 0 | 0 - 0.01 | 0 - 0.01 | |
| BATT | 7.038 | 0.712 | 2.9 - 8.4 | 0.3 - 0.8 | |

N/A: No aplica

Figura C.10: Prueba de vibración en eje Z. Hoja 2.

Tercera prueba

| | | | |
|--|---|---|--------|
| |  | Formas para validación satelital mediante uso de software de prueba. PNN-UNAM-FPF3.DOCX | |
| | | Fecha: 07/06/2016 Hora Ed: 23:00 | |
| | | Ed.: 0.1 | Pag: 3 |

| Forma YY4 | | | | | | | |
|-------------|--------------|---------------|-----------|----------------|---------------|-----------|--------|
| | Valor medido | | | Valor esperado | | | Status |
| | Voltaje [V] | Corriente [A] | Temp [°C] | Voltaje [V] | Corriente [A] | Temp [°C] | |
| Comp. Vuelo | 3.304 | 0.117 | 28° | 3.05 - 3.4 | 0.06 - 0.08 | 20 - 30 | |
| EPS 3V3 | 3.248 | 0.004 | 29 | 3.05 - 3.4 | 0.005 - 0.01 | 20 - 30 | |
| EPS 5V0 | 5.198 | 0.134 | 27 | 4.9 - 5.2 | 0.1 - 0.4 | 20 - 30 | |
| COM 3V3 | 3.335 | 0.014 | 27 | 3.05 - 3.4 | 0.005 - 0.02 | 20 - 30 | |
| COM 5V0 | 5.204 | 1.22 | 27 | 4.9 - 5.2 | 0.1 - 0.4 | 20 - 30 | |
| BATERIA | 7.050 | 0.8. | 29 | 2.9 - 8.4 | 0.3 - 0.8 | 20 - 30 | |

*


| | | | |
|---------|--|-----------|--|
| % Carga | | 0 - 100 % | |
|---------|--|-----------|--|

| Forma YY5 | | | |
|--------------|---|--------------------------|--------|
| | Nota | Observaciones | Status |
| Giroscopio | Mientras el satélite este inmóvil, la lectura en eje X, Y y Z debe ser aproximado a cero. | Aproximado a 0 en 3 ejes | |
| Acelerómetro | Usted deberá leer un aproximado a 9.8 [m/s ²] en el eje Z mientras el satélite este en posición vertical. | Sin errores | |
| Magnetómetro | Realice 3 lecturas moviendo el satélite por cada una. Las lecturas deberán diferir por cada movimiento. | Sin errores | |

| | Valor medido | Valor esperado | Status |
|-------------|------------------|-------------------|--------|
| Presión | 0.76788 [bar] | 0.77 - 0.79 [bar] | |
| Temperatura | [°C] | 20 - 30 [°C] | |
| Diagnóstico | 0080 hexadecimal | 0080 hexadecimal | |

Figura C.11: Prueba de vibración en eje Z. Hoja 3.

Tercera prueba

| | | | |
|--|---|---|--------|
| |  | Formas para validación satelital mediante uso de software de prueba. PNN-UNAM-FPF3.DOCX | |
| | | Fecha: 07/06/2016 Hora Ed: 23:00 | |
| | | Ed.: 0.1 | Pag: 4 |

| Forma YY6 | | | |
|---------------|-------------------------|---------------------------|--------|
| Observaciones | | | |
| | Sentido de giro derecha | Sentido de giro izquierda | Status |
| Motor X | Sin errores | No se observo daño | |
| Motor Y | Sin errores | No se observo daño | |
| Motor Z | Sin errores | No se observo daño | |

| Forma YY7 | | | | | | | | | | | | | | | | | | | | |
|-----------------|-------|---|---|---|---|---|---|---|-----|---|---|---|---|---|---|---|---|---|---|---|
| Valor esperado | D:SYB | 1 | 2 | 3 | M | 1 | 2 | 3 | syb | 1 | 2 | 3 | m | 1 | 2 | 3 | - | © | & | o |
| Valor capturado | D:SYB | | | | M | | | | syb | | | | m | | | | - | © | & | o |

| Observaciones |
|---------------|
| |

| Forma YY8 | | | |
|---------------------|-----------------|----------------|--------|
| Ambiente de prueba | Interior | Exterior | |
| Marque la adecuada: | | | |
| | Valor capturado | Valor esperado | Status |
| Estado | | Y | |
| Latitud | | N/A | |
| Longitud | | N/A | |
| Altitud | | N/A | |
| Fecha | | N/A | |
| Hora | | N/A | |

| Observaciones |
|---|
| → Se realizó la solicitud de latitud, longitud y altitud a GPS. |

→ Se logro enlace con banda S

Figura C.12: Prueba de vibración en eje Z. Hoja 4.

Bibliografía

- [1] Peter Fortescue, John Stark, Graham Swinerd (2003). *Spacecraft systems engineering*. Editorial Wiley tercera edición.
- [2] Mission Design Division Staff Ames Research Center, NASA. (Septiembre 2015). *Small Spacecraft Technology State of the Art*. Reporte.
- [3] Cal Poly, San Luis Obispo (2014). *CubeSat Design Especification REV 13*. Reporte.
- [4] Satellite Missions Database, de <https://directory.eoportal.org/web/eoportal/satellite-missions>. (Consulta: Enero 2016).
- [5] Bruce A. Campbell, Samuel Walter McCandless Jr. (1996). *Introduction to Space Science and Spacecraft Applications*. Editorial Gulf Publishing Company.
- [6] Philips Semiconductors (2000). *The I2C bus specification*. Manual técnico.
- [7] Kenneth A. LaBel, Michele M. Gates, Amy K. Moran. *Commercial Microelectronics Technologies for Applications in the Satellite Radiation Environment*. Artículo.
- [8] Astronautical Development, LLC (2009). *AstroDev Helium Radios*. Manual de usuario. Rev 0.9.
- [9] Microhard Systems Inc. (2015). *Nano Series n2420*. Documentación de producto.
- [10] Texas Instruments. (Abril 2015). *TMS570LS12x/11x 16/32-Bit RISC Flash Microcontroller*. Manual de referencia técnico.
- [11] NXP Semiconductors (Noviembre 2009). *PCA21125 SPI-bus Real-Time Clock and calendar*. Datasheet.
- [12] Texas Instruments. (Marzo 2012). *KeyStone Architecture Serial Peripheral Interface (SPI)*. Guía de usuario.
- [13] Intersil. (Agosto 2007). *SPI Protocol and Bus Configuration of Multiple DCPs*. Guía de referencia.
- [14] Texas Instruments. (Julio 2009). *TMS320x281x Serial Communications Interfaces (SCI)*. Guía de referencia.
- [15] Microchip. (2008). *1-Wire Communication with PIC Microcontroller*. Nota de aplicación.
- [16] IDS Imaging. *UI-1220SE-M*. Hoja de datos.
- [17] Wertz, J.R. y Larson, W.J. (1999). *Space Mission Analysis and Design*. Space Technology Library 3ª Edición.
- [18] André Emile Heunis. (2014). *Design and Implementation of Generic Flight Software for a CubeSat*. Tesis de maestría.
- [19] Jim Greene. (2003). *Implementing an I2C reset*. Analog Devices. Nota de aplicación.
- [20] Silicon Labs. *Serial Communications*. Documento.
- [21] Maxim Semiconductor *DS18B20-PAR 1-Wire Parasite Power Digital Thermometer*. Nota de aplicación.
- [22] Maxim Integrated. (2002). *1-Wire Search Algorithm*. Nota de aplicación 187.
- [23] NXP Semiconductors. (2014). *PCA21125 SPI bus Real Time Clock and Calendar*. Nota de datos.
- [24] Honeywell. (2010). *3-Axis Digital Compass IC HMC5883L*. Nota de datos.

- [25] Micron *Micron Serial NOR Flash Memory N25Q00AA*. Manual de referencia técnico.
- [26] Texas Instruments. (2013) *Inizialitation of hercules ARM Cortex R4F Microcontrollers*. Reporte de aplicación.
- [27] Texas Instruments. (2015) *Safety Manual for TMS570LS12x and 11x Hercules ARM-Based Safety Critical Microcontrollers*. Guía de usuario.
- [28] Texas Instruments. (2013). *F021 Flash API*. Guía de referencia.
- [29] André Emile Heunis. (2014). *Design and Implementation of Generic Flight Software for a CubeSat*. Tesis de maestría.
- [30] Texas Instruments. (2013). *Code Composer Studio Integrated Development Environment (IDE)*. URL:<http://www.ti.com/tool/ccstudio>
- [31] Texas Instruments. *XDS100v2 JTAG Debug Probe (14-pin TI version)*. URL:<http://www.ti.com/tool/TMDSEMU100V2U-14T>
- [32] Texas Instruments. (2014). *HalCoGen*. URL:<http://processors.wiki.ti.com/index.php/Halcogen>
- [33] Joshua Debes, Nathan Howard, Ryan Harrington, Richard Cobb, Jonathan Black. *Rapid Build and Space Qualification of CubeSats* Air Force Institute of Technology. Artículo.
- [34] NASA GODDARD Space Flight Center. *General Environmental Verification Standard*. Estándar.
- [35] Martin Johnsson. (2012). *Development of a Shock Test Facility for Qualification of Space Equipment*. Tesis de maestría.