UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DISEÑO Y FABRICACIÓN DE COMPUTADORA DE 32 BITS PARA SATÉLITE EXPERIMENTAL

T E S I S

Que para obtener el título de

INGENIERO ELÉCTRICO-ELECTRÓNICO

P R E S E N T A:

DEL MORAL PEREA BERNARDO

DIRECTOR DE TESIS:
Dr. Esaú Vicente Vivas

Ciudad Universitaria, 2014

Jurado asignado:

Presidente: Dr. Salvador Landeros Ayala

Secretario: Dr. Esaú Vicente Vivas

Vocal: Ing. José Salvador Zamora Alarcón

1er. Suplente: Dr. José Ismael Martínez López

2do. Suplente: Dra. Fatima Moumtadi

Esta tesis se realizó en:
Instituto de Ingeniería, U.N.A.M.

Director de tesis:
Dr. Esaú Vicente Vivas

Firma

A MIS PADRES
De cuyos auténticos cabellos grises
yo soy una de las principales causas

Agradecimientos

Gracias Dios por la vida que mediste, por la familia que me permites compartir, por mis amigos que son como mi familia y sobre todo por permitirme terminar una etapa mas en mi vida.

A mis padres Juan y Elisa por su apoyo y sacrificio incondicional, por su comprensión, cariño y confianza depositada en mi todo este tiempo.

A mi hermano Miguel Angel por compartir y disfrutar grandes momentos en mi vida pese a los enojos.

A todos los demás miembros de mi familia: tíos, primos, abuelos, sin excepción. Por los que no están cerca y por quien ya no esta pero que sigo sintiendo su apoyo.

A Mitzi por su comprensión y espera para terminar este proyecto así como el apoyo y cariño que siempre he sentido por parte de su familia.

A todos mis compañeros de carrera y de la prepa por los buenos y malos momentos.

A mis amigos de la infancia Carlos, Ralme por esos felices momentos.

A mis compañeros de laboratorio M.I. Emilio Jiménez por su gran apoyo en este trabajo, a Dr. Mario Alberto Mendoza, Ing. Alberto Muñiz, Ing. Miguel Alvarado, Ing. Pedro, Ing. George.

A mi asesor de tesis Dr. Esaú Vicente Vivas, por darme la oportunidad de pertenecer a su grupo de trabajo y sobre todo por la confianza puesta en mi.

Y por último a los profesores de la facultad que me apoyaron dentro de mi formación académica así como a mi segundo hogar la Universidad Nacional Autónoma de México.

Resumen

La presente propuesta de tesis consiste en el diseño, construcción y validación en hardware y software que integrará el subsistema de computadora de vuelo de un satélite pequeño desarrollado en el Instituto de Ingeniería UNAM, basado en dos MCU's, memorias y sensores de protección a algunos factores que se pueden dar en el espacio. Todo este sistema esta contenido en un circuito impreso de pequeñas dimensiones, bajo costo y bajo consumo de energía.

El diseño está basado en el uso de componentes COTS (Commercial Off-The-Shelf) los cuales son componentes comerciales que pueden ser adquiridos a un precio accesible para el público en general y que poseen cierta tolerancia a los efectos presentes en el espacio sin llegar a ser de calidad espacial.

Dicho subsistema se encargará de tareas tales como: la recepción de comandos desde el software de Estación Terrestre (E.T.), envío de telemetría a E.T., comunicación por medio de comandos con los demás subsistemas, acceso y manejo de memoria Flash y lectura de sensores.

Abstract

This thesis work shows the design, manufacture and validation both in hardware and software to integrate the On Board Computer subsystem for a pico-satellite developed at Instituto de Ingeniería UNAM. This is based in two microcontrollers, memorys and protection sensor to protect it from some factors that can be given into the space orbit. All the system were allocated within a printed circuit board of small dimensions, low cost and low power consumation.

The subsystem use COTS (Commercial off the Shelf) components that can be acquired at a reachable price to general public and have tolerance to effects present into the space without being space quality components.

The subsystem is responsible to perform tasks such as: commands reception from earth station, sending telemetry to earth station and communication with other satellite.

Índice General

Resumen **I**

Abstract **II**

Lista de acrónimos **VI**

Lista de Figuras **VIII**

Lista de Tablas **IX**

Introducción **1**

1 Marco de referencia **3**
1.1 Proyecto Cubesat 3
1.2 Componentes COTS 4
1.3 Misiones exitosas 5
1.3.1 Libertad 1 5
1.3.2 GOLIAT 6
1.3.3 E-ST@R 7
1.3.4 ITUpSAT-1 8
1.4 Conclusión 9

2 Arquitectura del satélite experimental **11**
2.1 Estructural Satelital 12
2.2 Subsistema de Potencia 13
2.3 Subsistema de Comunicaciones 14
2.4 Subsistema de Computadora de Vuelo 14
2.5 Cargas útiles 15
2.6 Conclusión 15

3 Desarrollo de la tarjeta de Computadora de Vuelo **18**
3.1 Arquitectura electrónica de la Computadora de Vuelo y su diseño electrónico asociado . . . 18
3.1.1 Microcontrolador TMS570LS1227 . . . 18
3.1.2 Microcontrolador MSP430F1612 . . . 22
3.1.3 Memorias Flash . . . 23
3.1.4 Sensores de Temperatura . . . 25
3.1.4.1 Sensores 1-Wire . . . 25
3.1.4.2 Sensor Interno de MSP430 . . . 26
3.1.5 Protección contra efecto "Latch-up" . . . 27
3.1.5.1 Técnicas de sensado . . . 27
3.1.5.2 Componente Front-End(BQ24314) . . . 28
3.1.5.3 Amplificador MAX4372 y Comparador LM6511 . . . 30
3.1.6 Magnetómetro . . . 36
3.2 Desarrollo del circuito impreso del subsistema Computadora de Vuelo . . . 38
3.2.1 Plataforma de diseño Altium Design 10 . . . 38
3.2.2 Metodología para generación de un PCB . . . 39
3.2.3 Esquemáticos de la Computadora de Vuelo . . . 39
3.2.4 PCB y Tarjeta Electrónica . . . 45
3.3 Conclusión . . . 47

4 Desarrollo de Software de operación de la Computadora de Vuelo **50**
4.1 Plataformas de desarrollo . . . 50
4.1.1 Code Composer Studio(CCS) . . . 50
4.1.2 HalCoGen . . . 52
4.2 Cargado de programas en TMS570LS1227 y MSP430F1612 . . . 53
4.2.1 Arquitectura de programación en el MCU TMS570LS1227 . . . 53
4.2.1.1 JTAG . . . 53
4.2.1.2 UART Bootloader Hércules . . . 54
4.2.2 Arquitectura de programación en el MCU MSP430F1612 . . . 56
4.2.2.1 LaunchPad basado en Interfaz BSL UART MSP430 . . . 56
4.3 Software de operación de Computadora de Vuelo . . . 58
4.3.1 Esquema general de ejecución de comandos del subsistema Computadora de Vuelo . . . 59
4.3.2 Comandos del subsistema Computadora de Vuelo . . . 63
4.3.2.1 Protocolo de comunicación 1-Wire . . . 63
4.3.2.2 Protocolo de comunicación UART . . . 67
4.3.2.3 Protocolo de comunicación SPI . . . 68
4.3.2.4 Protocolo de comunicación I2C . . . 69
4.4 Conclusión . . . 70

5 Pruebas de Validación **73**
5.1 Prueba de cargado de Software al MSP430F1612 74
5.2 Prueba de programación de Hércules vía Bootloader 75
5.3 Prueba de lectura de sensores de temperatura 1-wire 75
5.4 Prueba de circuito contra evento "latchup" 76
5.5 Conclusión 77

6 Conclusiones y Recomendaciones **78**
6.1 Conclusiones 78
6.2 Recomendaciones 79

Apéndice **80**

Lista de acrónimos

ADC	Analogic Digital Converter.
BSL	Bootstrap Loader.
COTS	Commercial of the Shelf.
CV	Computadora de Vuelo.
DMA	Direct Memory Access.
DSP	Digital Signal Processor.
eCAP	Enhanced Capture.
ECC	Error Correcting Code.
ePWM	Enhanced Pulse Width Modulator.
eQEP	Enhanced Quadrature Encoder Pulse.
GOTS	Government of the Shelf.
GPIO	General-purpose Input-Output.
I2C	(ó IIC) Inter-Integrated Circuit.
IDE	Integrate Development Environment.
LEO	Low Earth Orbit.
LIN	Local Interconection.
MCU	Microcontroller.
N2HET	Next Generation High-End Timers.
OBC	On Board Computer.
PCB	Printed Circuit Board.
P-POD	Poly Picosatellite Orbital Deployer.
RTOS	Real TIme Operation System.
SIMO	Slave input - Master output.
SOMI	Slave output - Master Input.
SPI	Serial Peripheral Interface.
TI	Texas Instruments.
UART	Universal Asynchronous Receiver-Transmitter.

Lista de Figuras

1.1 Estructura estándar 1U para un Cubesat 4
1.2 Satélite GOLIAT 7
1.3 Satélite E-ST@R 8
1.4 Satélite ITUpSAT 8

2.1 Propuesta de Satélite Experimental por parte del IINGEN UNAM 11
2.2 Formas y dimensión de Tarjetas electrónicas 12
2.3 Modelo CAD de estructura satelital 3U de la compañía Pumpkin 13
2.4 Modelo 3D de subsistema de potencia 13
2.5 Modelo 3D de subsistema del comunicaciones 14
2.6 Modelo de cámaras como carga útil para Satélites Cubesat 15

3.1 Diagrama a bloques del MCU TMS570LS1227 21
3.2 Diagrama a bloques del MCU MPS430F1612 23
3.3 Gráfica comparativa entre memorias Flash 24
3.4 Configuración de memorias Flash NOR 25
3.5 Conexión de sensores de Temperatura 1-Wire 26
3.6 Función de transferencia típica del sensor de temperatura del MSP430 . . . 26
3.7 Sensado de corriente en lado bajo 27
3.8 Sensado de corriente en lado alto 28
3.9 Aplicación típica de BQ24314 (Front End) 29
3.10 Resistencia vs corriente del componente BQ24314 30
3.11 Configuración del MAX4372 31
3.12 Configuración de comparador LM6511 32
3.13 Circuito base para divisor de voltaje 32
3.14 Sistema protector de evento Latchup 34
3.15 Transistor NPN 36
3.16 Recomendaciones de conexión para el magnetómetro 38
3.17 Esquemático: Vista general de los principales componentes 40
3.18 Esquemático:Polarización de núcleos y periféricos del TMS570LS1227 . . . 41
3.19 Esquemático:Puertos y periféricos del TMS570LS1227 42

3.20 Esquemático:Polarización de MSP430F1612 y puertos de conexión 43
3.21 Esquemático:Polarización de memorias Flash NOR, protecciones y puertos de conexión 44
3.22 Tarjeta electrónica de Computadora de Vuelo sin rutear 45
3.23 Vista superior e inferior de la Computadora de Vuelo en 2D 46
3.24 Vista superior e inferior de la Computadora de Vuelo en 3D 46
3.25 Computadora de Vuelo dentro de la estructura en 3D 47

4.1 Pantalla de código y depurador de Code Composer Studio 51
4.2 Pantalla de la Plataforma HalCoGen 52
4.3 Pantalla de la Plataforma HalCoGen 53
4.4 Forma de cargado de UART Bootloader 55
4.5 Ejemplo del uso del UART Bootloader con PC 56
4.6 Conexión via BSL UART 57
4.7 Diagrama de flujo de tareas de C.V. 60
4.8 Diagrama de flujo de operación durante error de C.V. 62
4.9 Protocolos de comunicación empleados en la Computadora de Vuelo 63
4.10 Operaciones básicas de protocolo 1-Wire 64
4.11 Registro de ROM code de sensores 1-Wire 65
4.12 Scratchpad de sensores 1-Wire 65
4.13 CRC de sensores 1-Wire 66
4.14 Registro de temperatura de sensores 1-Wire 67
4.15 Líneas ocupadas en bus SPI 68
4.16 Configuración I2C 69
4.17 Condiciones de START y STOP 70
4.18 Trama de datos I2C 70

5.1 Kit de desarrollo Launchpad MSP-EXP430G2553 73
5.2 Kit de desarrollo LAUNCHXL-TMS57004 74
5.3 Tarjeta electrónica con MCU MSP430F1612 para pruebas de validación . . 75
5.4 PCB construida para evento latchup 76
5.5 Señal de error y voltaje de salida de CI BQ24314 76

Lista de Tablas

3.1 Asignación de pines del MCU TMS570LS1227 21
3.2 Asignación de pines del MCU MSP430F1612 23
3.3 Tabla de resistencias de sensado de corriente recomendadas 31
3.4 Tabla de consumo de MSP430Fxxxx 35

4.1 Asignación de GPIOs del MSP430G2553 para implementación de BLS UART 57
4.2 Comandos utilizados en dispositivos con comunicación 1-Wire 67

Introducción

El objetivo general que se presenta en este trabajo de tesis es diseñar, construir y validar el hardware que integrará el subsistema de computadora de vuelo de un satélite pequeño usando componentes COTS y como objetivo particular, este subsistema estará contenido dentro de un satélite experimental con el cual se pretende entrenar y atraer a jóvenes generaciones al mundo de la Ciencia y la Tecnología mediante su uso en aulas de clases, Tecnológicos, Universidades, Posgrados y en Centros de Investigación como es el caso de SATEDU (Satélite Educativo).

La metodología utilizada para alcanzar los objetivos de este trabajo de investigación se describe a continuación. Primero se plantearon las necesidades de la computadora de vuelo y se determinó las tareas que el subsistema debía realizar. Una vez planteadas las bases, se seleccionaron los componentes electrónicos, los cuales fueron elegidos con un rango de operación de temperatura amplio como por ejemplo de -40 °C a 125 °C o en su defecto que fueran de grado automotriz. Se eligieron los protocolos de comunicación dentro del subsitema y los protoclos de comunicación hacia el exterior del subsistema. El siguiente paso fue realizar pruebas con todos los componentes posibles mediante el uso de protoboard, kits de desarrollo de microcontroladores y muestras de circuitos integrados pedidos directamente al fabricante. Una vez teniendo algunas pruebas se procedió al diseño del diagrama esquemático, con el cual se pretende realizar el circuito impreso para finalmente ser construido y ensamblado en su totalidad; después realizar la validación del subsistema mediante pruebas de software y en un futuro pruebas de vibración, radiación y vacío.

El contenido de este documento está estructurado en 6 capítulos. En el primero se hace referencia a lo que es el proyecto Cubesat y algunas de las especificaciones necesarias para ser parte de éste. Por otra parte se hace referencia a lo que son los componentes COTS así como la filosofía tomada por la NASA para su uso y algunos Satélites de dimensiones pequeñas, los cuales han sido misiones exitosas.

En el segundo capítulo se describe un panorama general del Satélite experimental (sin llegar a ser el definitivo) y los subsistemas que actualmente están siendo diseñados por un

grupo de ingenieros del Instituto de Ingeniería UNAM. En mi caso a mi me corresponde el subsistema de computadora de vuelo el cual es descrito en este trabajo de investigación.

En el capítulo tres se describe como fue desarrollado el subsistema de computadora de vuelo. Se habla de los circuitos integrados utilizados, una breve descripción de éstos y su implementación en hardware para un uso correcto. Dentro del mismo capítulo están los pasos que se siguieron para la realización del diagrama esquemático, el diseño de la tarjeta electrónica y se muestran algunas imágenes de como podría ser la tarjeta electrónica final mediante modelos en 3D.

En el capítulo cuarto se aborda el como se programan los microcontroladores, plataformas a utilizar para su uso y una idea general mediante diagramas de flujo de como puede ser el software de operación del subsistema de computadora de vuelo.

En el capítulo cinco se presentan las pruebas realizadas de este trabajo de investigación. Las pruebas fueron realizadas con el uso de protoboard, kits de desarrollo y circuitos integrados, algunos comprados y otros obtenidos como muestras gratis pedidas al fabricante.

Por último, el capítulo seis hace un recuento de los logros alcanzados acerca de este trabajo de investigación, así como recomendaciones para un diseño futuro en hardware y una actualización en software del subsistema de computadora de vuelo.

Capítulo 1

Marco de referencia

En el año 1960, el desarrollo de satélites artificiales solía ser muy costoso, de alto riesgo y muy complicado. Su construcción estaba reservada a grandes empresas aeroespaciales o a gobiernos a través de sus agencias espaciales como es el caso de los satélites Morelos I y II con una inversión de 150 millones de dólares. En ambos casos, se trataba de un satélite modelo HS 376, que era el más comercial de la época, con una forma cilíndrica, una longitud de 6.62 m (desplegado) y un peso de 645.5 kilogramos en órbita geoestacionaria. Afortunadamente esto ya no es así, ya que los cambios tecnológicos están propiciando el desarrollo de satélites que pueden ser diseñados y construidos por universidades y empresas con relativamente pocos recursos.

1.1 Proyecto Cubesat

El proyecto Cubesat [1]fue desarrollado por la "California Polytechnic State University, San Luis Obispo" y por "Standford University's Space Systems Development Lab". El programa Cubesat creó oportunidades de lanzamiento para las universidades que antes no podían acceder al espacio. Con mas de 60 Universidades y Preparatorias participando en el programa, los beneficios educativos son enormes. Los estudiantes desarrollan habilidades y obtienen experiencia necesaria para tener éxito en la industria aeroespacial.

El programa también beneficia a empresas privadas y a gobiernos, constituyendo una vía de acceso para colocar cargas útiles en el espacio de forma económica, así como la creación de importantes oportunidades educativas para los futuros lideres de la industria. El programa Cubesat ofrece prácticas, confiables y rentables oportunidades de lanzamiento de satélites pequeños con cargas útiles. Para ello se ofrece:

- Un diseño físico estándar.

- Sistemas de despliegue de vuelo estándar denominado P-POD(Poly Picosatellite Orbital Deployer) compatible con diversos sistemas de lanzamiento internacionales.
- Coordinación de documentos requeridos y licencias de exportación.
- Proceso ágil de integración.
- Confirmación de un despliegue correcto e información de telemetría.

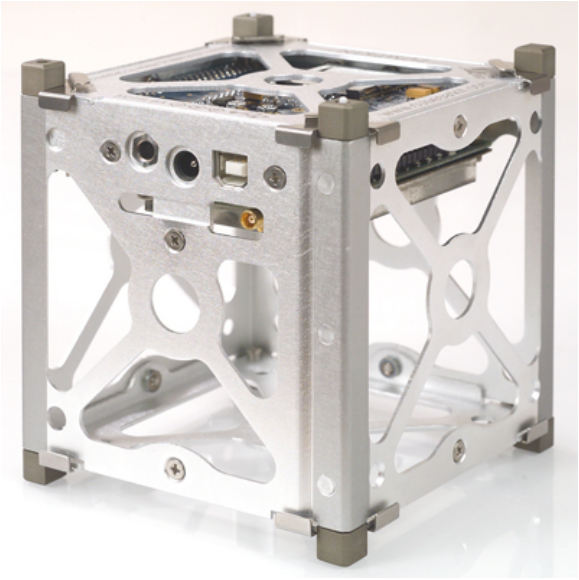


Figura 1.1: Estructura estándar 1U para un Cubesat

1.2 Componentes COTS

Un estudio en el pasado de Satelites Cubesat [2] demostró que se presentan mas fallas de las normales debido a diseños de componentes personalizados. Para optimizar la confiabilidad de sistemas satelitales y asegurar aun mas las misiones, se iniciaron investigaciones para explorar variables de diferentes componentes. Con un desarrollo ilimitado en tiempo y pocos recursos, los componentes COTS son una buena inversión.
Los componentes COTS [3]son productos comerciales que no han sido modificados específicamente para su uso en el espacio.
Un artículo comercial de acuerdo con la parte 2 del Reglamento Federal de Adquisiciones (FAR) se define como:

- Cualquier artículo utilizado para propósitos no gubernamentales, que sea vendido con licencia para todo el público en general.

- Un artículo que evolucionó a partir de uno comercial, esencialmente nuevos modelos o actualizaciones.
- Cualquier combinación de artículos cumpliendo con la definición de artículo comercial, si este es combinado y vendido comercialmente. Un ejemplo sería una computadora o un sistema de video que es una combinación de artículo comercial, a pesar de que el propio sistema puede ser una configuración única.

En resumen este es un listado de algunos de los beneficios de utilizar componentes COTS:

- Reduce costos.
- Buena disponibilidad.
- Reducción de riesgos.
- Actualizaciones del vendedor.
- Factible en base a lo previsto (sin diseño y ensayo).
- Ciclo de desarrollo mas corto.
- Más flexible, escalable y configurable.

NASA estableció una filosofía de diseño a favor de los productos COTS. Adopto la "regla 80-20" que afirma que si un producto COTS o GOTS(Government off the Shelf) reune el 80% de requerimientos funcionales, estos deben ser adoptados a pesar del 20% restante de los requerimientos.

En resumen, NASA ha compartido un sinnúmero de factores que hace exitosos a los componentes COTS.

1.3 Misiones exitosas

A continuación se mencionan algunas misiones de nanosatélites Cubesat exitosas las cuales fueron elegidas por una razón. Estas misiones tienen en común que su subsistema de computadora de vuelo es de la compañía Pumpkin Inc y está basado en una tarjeta electrónica con un microcontrolador de la familia MSP430 Texas Instruments. Su computadora de vuelo es compatible con MCUs de la misma familia ya que el comprador puede elegir entre 3 MCUs distintos los cuales son F1612, F1611 y F169.

1.3.1 Libertad 1

Con una inversión de 800 millones de pesos colombianos (5,333,542 MXN),el Libertad 1 es el primer satélite construido en Colombia con asesoría de EU [4]. Basado en un

Picosatélite Cubesat, el satélite fue puesto en la órbita LEO el 17 de Abril de 2007 desde el cosmodromo de Baikonur, Kazakhastan.
Características de Libertad 1:

- 4 MCU's a bordo
- Dimensiones: 10x10x10 cm
- Peso:995 gramos
- Downlink:437.405 MHz
- Uplink:145.825 MHz
- Potencia de transferencia de bajada:400 mW
- Potencia de transferencia de subida máxima:35 W
- Consumo de corriente:5 mA (Stand by)
- Consumo de corriente en transmisión:135 mA
- Baterías: 2 Ion-Litio de 7.2 V, 950 mAh
- Corriente máxima de baterías: 1.9 A

1.3.2 GOLIAT

Goliat [5, 6]es una demostración de tecnología Cubesat de la "Bucharest University and Bucharest Polytechnic University" bajo la coordinación de la "Romanian Space Agency" (ROSA), figura1.2. Fue lanzado el 13 de Febrero del 2012 desde el Centre Spatial Guyanais a la órbita LEO. El principal objetivo del proyecto fué formar un equipo multidisciplinario de estudiantes los cuales al finalizar el proyecto debían diseñar, operar e integrar una misión completa de un nanosatélite.

Desde un punto de vista científico, el Cubesat Rumano integró 3 experimentos en órbita LEO.

- Dose-N: Determinación de dosis total de radiación en la órbita.
- SAMIS: Detección de micro-meteoritos en órbita.
- CICLOP: Cámara digital equipada con un lente personalizado.

De igual importancia fueron las nuevas soluciones y tecnologías que el GOLIAT probaría:

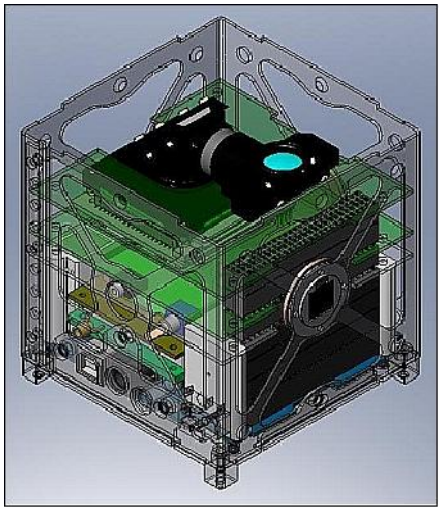


Figura 1.2: Satélite GOLIAT

Fuente de energía de bajo ruido y el algoritmo de determinación de altitud (GPS, Magnetómetro, TLE).

El subsistema de computadora a bordo o OBC por sus siglas en inglés está basado en 2 MCU MSP430.

Adicionalmente, varios subsistemas están controlados por microprocesadores, la transferencia entre subsistemas es SPI e interfaz serial.

A su vez integra un microcontrolador MSP430 para el módulo de transmisión y otro para el subsistema de potencia así como un DSP para la cámara.

1.3.3 E-ST@R

Proyecto Cubesat [7] diseñado y desarrollado por estudiantes del Politécnico di Torino, Turin Italia, figura1.3. Aparte de la experiencia ganada, el principal objetivo fue la demostración del subsistema de Control y Orientación de apuntamiento basado en tecnología de 3 ejes, incluyendo una unidad de medición inercial. El segundo objetivo fue probar componentes comerciales y materiales en el espacio.

La computadora de vuelo se basó en una unidad de procesamiento desarrollada por la compañía Pumpkin que consiste en un microprocesador de Texas Instruments (MSP430, 16 bits)con un sistema operativo en tiempo real.

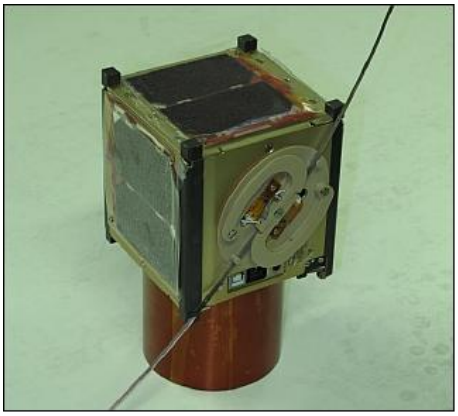


Figura 1.3: Satélite E-ST@R

1.3.4 ITUpSAT-1

Pico satélite [8] diseñado y desarrollado por estudiantes del ITU (Istanbul Technical University) Istanbul, Turquía, figura1.4. Los objetivos de la misión fueron capturar imágenes con la carga útil CMOS y estudiar el comportamiento del sistema de estabilización pasiva del Cubesat.

La computadora a bordo utilizó el módulo de vuelo FM430 de la compañía Pumpkin. Cuenta con una tarjeta de interfaz SD, un puerto USB y un conector externo para cargar. El FM430 esta equipado con un MSP430F1611, un micro-controlador de TI de bajo consumo de 16 bit. Cuenta con varios periféricos como I2C, SPI, UART y también soporta DMA.

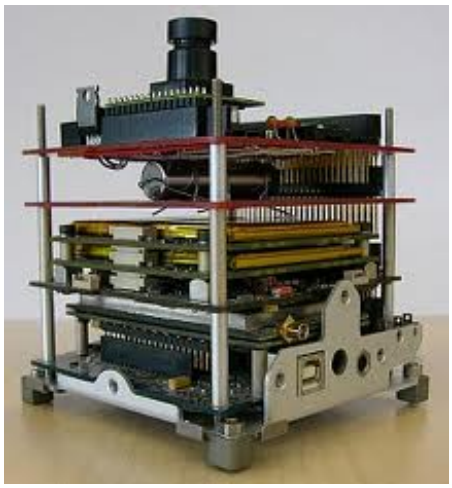


Figura 1.4: Satélite ITUpSAT ensamblado

1.4 Conclusión

El desarrollo de tecnología satelital para dispositivos puestos en la órbita baja de la Tierra es cada vez mas viable. Hoy en día existe una base sólida de misiones exitosas las cuales están documentadas y son ejemplos de satélites diseñados con relativamente bajo presupuesto y hechos por un equipo de trabajo compuesto por alumnos desde licenciatura hasta alumnos de doctorado. En este capítulo se mencionaron solo algunas misiones existentes donde sus diseñadores pertenecen a países desarrollados y no desarrollados tecnológicamente. Con esto se pretende llamar la atención en nuestro país a alumnos a este gran campo de conocimiento.

Referencias

[1] "CubeSat". URL:www.cubesat.org/index.php/about-us. (Consulta: Feb.2014).

[2] "COTS". URL:https://ncurbd.cur.org/ncur/archive/Display_NCUR.aspx?id=20035. (Consulta: Feb.2014).

[3] Academy of Aerospace Quality. "Commercial Off-the-Shelf (COTS) Tutorial". URL:aaq.auburn.edu/node/697. (Consulta: Feb.2014).

[4] Universidad Sergio Arboleda. "Libertad 1". URL:www.usergioarboleda.edu.co/proyecto_espacial/. (Consulta: Ene.2014).

[5] "GOLIAT". URL:www.goliat.ro/index.php. (Consulta: Ene.2014).

[6] eoPortal Directory. "Goliat". URL:https://directory.eoportal.org/web/eoportal/satellite-missions/g/goliat. (Consulta: Ene.2014).

[7] eoPortal Directory. "E-ST@R" (Educational SaTellite @ politecnico di toRino).URL:https://directory.eoportal.org/web/eoportal/satellite-missions/e/e-star. (Consulta: Ene.2014).

[8] eoPortal Directory. "ITUpSat-1(Istanbul Technical University PicoSatellite-1)".URL:https://directory.eoportal.org/web/eoportal/satellite-missions/i/itupsat-1. (Consulta: Ene.2014).

Capítulo 2

Arquitectura del satélite experimental

Este capítulo da un panorama general del Satélite Experimental. Describe una primera propuesta de los subsistemas contenidos dentro del satélite sin tener que ser el definitivo. Algunas imágenes son modelos 3D de los subsistemas diseñados por el equipo de trabajo.

A continuación se muestra un diagrama a bloques general del diseño propuesto del Satélite Experimental por parte del grupo de trabajo del Instituto de Ingeniería UNAM.

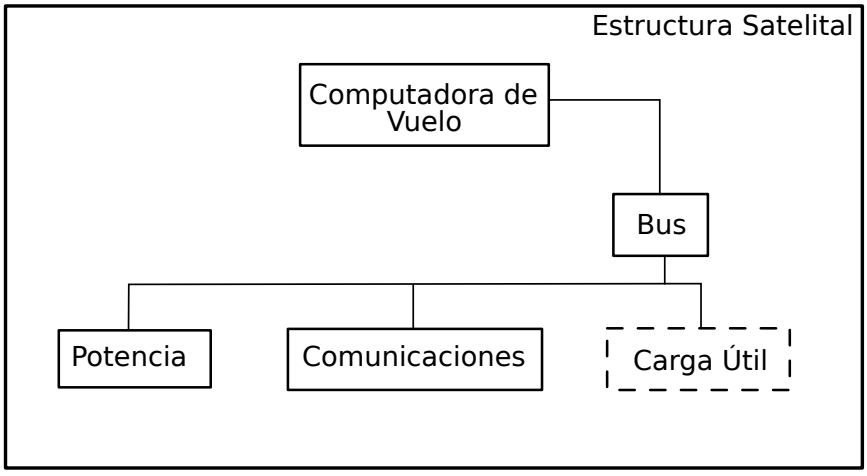


Figura 2.1: Propuesta de Satélite Experimental por parte del IINGEN UNAM

El diagrama muestra los subsistemas que actualmente están en proceso de diseño y cabe mencionar que este diseño podría no ser definitivo ya que en un futuro se podría agregar alguna carga útil como una cámara (linea punteada) u otro subsistema como el de control y orientación de apuntamiento.

La primera propuesta del grupo de trabajo que desarrolla el Satélite Experimental, fue definir las dimensiones que tendrían las placas de circuito impreso donde se colocaría toda

la electrónica que le daría vida al Satélite.

Todos los circuitos impresos serán compatibles con la estructura satelital comercial de la compañía Pumpkin Inc (figura 2.2), por tal motivo fué necesario tomar las medidas estándar de sus PCB(Printed Circuit Board) y se generaron dos diseños, el primero con dimensiones de 92 mm X 95 mm y el segundo de 94.9 mm X 91.8 mm. Los diseños cuentan con recortes en algunos de sus lados para evitar contacto con la estructura o para permitir el paso de cables. Cada tarjeta tiene un conector de costilla de terminales largas para ser apiladas una sobre la otra.

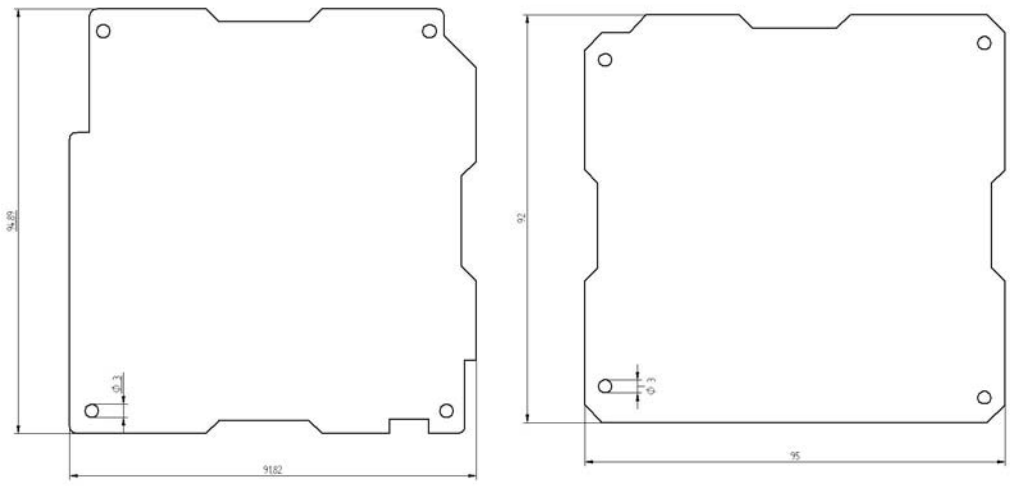


Figura 2.2: Formas y dimensión de Tarjetas electrónicas

De esta forma el Satélite Experimental queda conformado por los siguientes subsistemas: Estructura[1], Computadora de Vuelo, Subsistema de Potencia, Subsistema de Comunicaciones y cargas útiles.
A continuación se dará una breve explicación de las tareas a realizar de cada subsistema así como sus características principales para lograr un desempeño óptimo en conjunto con la Computadora de Vuelo.

2.1 Estructural Satelital

Como se mencionó antes, el diseño de las PCB's está pensado para ser compatible con la estructura de la compañía Pumpkin Inc., figura2.3 sin embargo el Satélite no depende de esta estructura puesto que se puede generar una estructura especial para este Satélite Experimental siendo su única limitante las dimensiones y la posición de los orificios que sujetaran la estructura con las tarjetas apiladas una sobre la otra.

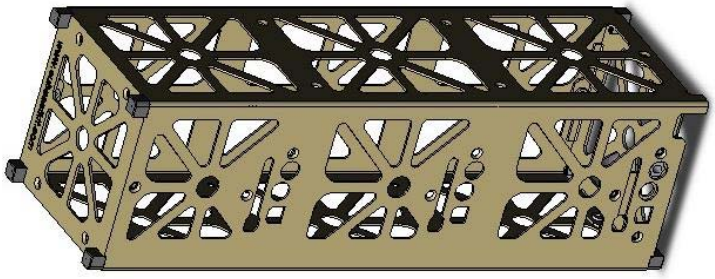


Figura 2.3: Modelo CAD de estructura satelital 3U de la compañía Pumpkin

2.2 Subsistema de Potencia

El subsistema de Potencia está constituido por dos tarjetas, una de ellas contiene los circuitos de carga de baterías para almacenar la energía así como un circuito calefactor para asegurar el rango de operación de las baterías en base a su temperatura. La segunda se encarga del acondicionamiento de los voltajes y circuitos de control que habilitarán los buses de alimentación de cada subsistema, circuitos protectores contra efecto "Latch-up", un circuito cortador de hilo el cual se encargará de liberar las antenas al momento de liberar el Satélite y por último dos interruptores, remover antes de vuelo y el llamado "Kill Switch" que mantienen totalmente apagado al sistema antes de su liberación, figura2.4.

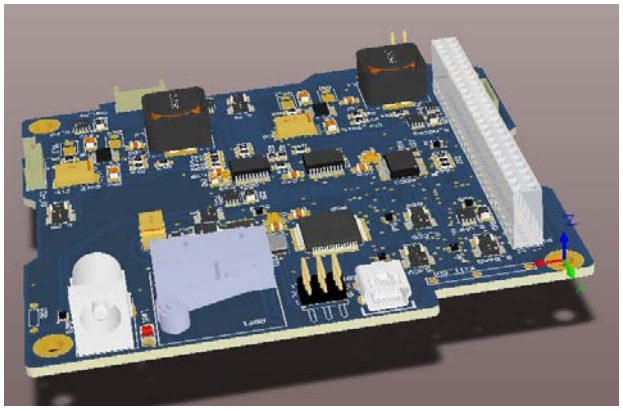


Figura 2.4: Modelo 3D de subsistema de potencia

El subsistema de Potencia está constituido por un microcontrolador de la familia MSP430 de Texas Instrument el cual se comunica con la Computadora de Vuelo para

enviar información como voltajes, corrientes y temperaturas, está diseñado para conectarse con un arreglo de paneles solares y convertidores de DC a DC.

2.3 Subsistema de Comunicaciones

El subsistema de Comunicaciones está constituido por una tarjeta electrónica la cual contiene un radio módem que opera en la banda VHF/UHF, circuitos protectores contra evento "Latch-up", sensores de temperatura y un microcontrolador de la familia MSP430 de Texas Instruments, figura2.5.

Este subsistema cuenta con dos formas de recepción y transmisión de datos, una por medio de un conector USB y la otra por medio del radio módem, lo cual esta pensado para trabajar de forma alámbrica o inalambricamente con el Satélite.

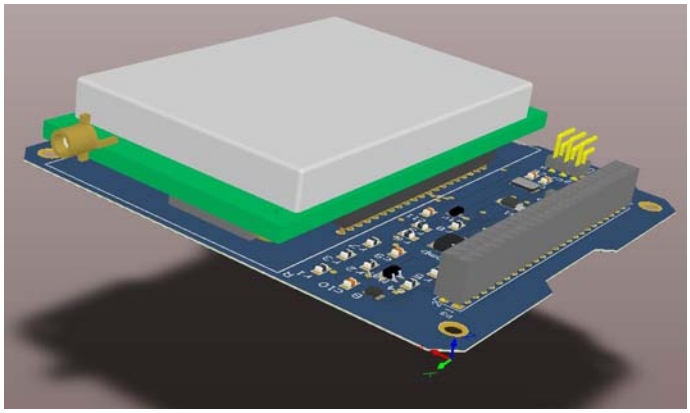


Figura 2.5: Modelo 3D de subsistema del comunicaciones

2.4 Subsistema de Computadora de Vuelo

El subsistema de Computadora de Vuelo está basado en dos microcontroladores de Texas Ins., el primero de la familia HERCULES (TMS570) y el segundo de la familia MSP430 el cual sirve como respaldo del primero permitiendo su reprogramación vía UART.

Cuenta con circuitos de protección contra efecto "Latch-up", memorias Flash para almacenamiento de datos, un magnetómetro como prueba de validación y sensores de temperatura 1-Wire los cuales no solo estarán distribuidos en este subsistema sino en todo el Satélite. La Computadora de Vuelo se encarga de recabar todas las temperaturas

del satélite a través del bus.
En el capítulo siguiente se detalla más a fondo el diseño de la Computadora de Vuelo.

2.5 Cargas útiles

El Satélite Experimental ha sido diseñado para desarrollar expansiones que permitan adaptarse al mismo sistema sin necesidad de hacer cambios al circuito impreso de la Computadora de Vuelo.

Es posible llevar cargas útiles varias debido a que la Computadora de Vuelo cuenta con diferentes periféricos del microcontrolador principal ruteados al conector como son el bus I2C, SPI, 1-Wire, UART y canales de conversión Analógico/Digital. Gran parte del diseño de Satélites, es basado en la carga útil que tendrá el Satélite, asi, podemos encontrar dentro de algunas misiones documentadas cargas útiles como cámaras de luz visible e infrarrojo, validación de algoritmos del subsistema de control de orientación y apuntamiento, detectores de micrometeoritos y medidores de radiación entre otros.

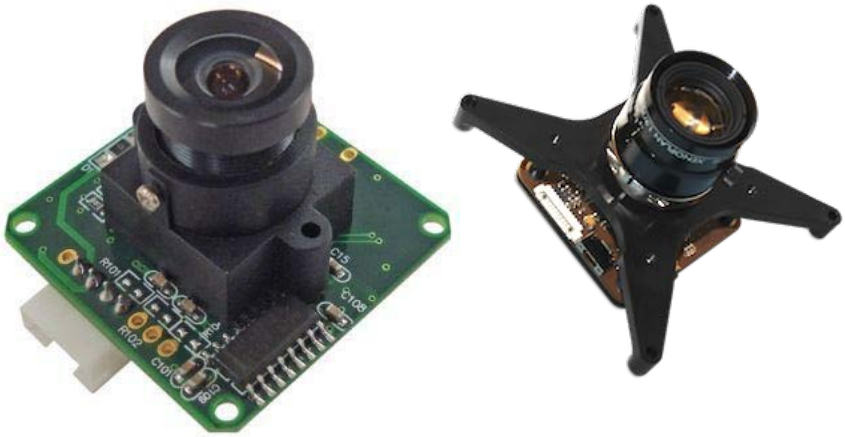


Figura 2.6: Modelo de cámaras como carga útil para Satélites Cubesat

2.6 Conclusión

Por el momento el satélite experimental cuenta con tres subsistemas actualmente en proceso de manufactura sin embargo está diseñado para soportar una carga útil como podría ser una cámara u otro subsistema como el de control y orientación de apuntamiento. Esto se logra seleccionando un conector grande que permita dejar lineas de conexión disponibles

para nuevos subsistemas y tanto la computadora de vuelo como potencia están diseñados para soportar estas expansiones.

Cabe mencionar que los nuevos subsistemas diseñados deben cumplir con las especificaciones de dimensión de tarjetas, voltajes de operación definidos por el subsistema de potencia y contar con el mismo protocolo de comunicación por el cual se intercambiara información entre susistemas dentro del satélite. Finalmente dependiendo del diseño final y el numero de tarjetas electrónicas se seleccionará la estructura que puede ser 1U, 2U o 3U que son las medidas estándar dadas por el proyecto Cubesat.

Referencias

[1] CubeSat kit. URL:www.cubesatkit.com. (Consulta: Feb.2014).

Capítulo 3

Desarrollo de la tarjeta de Computadora de Vuelo

Este capítulo consta del diseño y desarrollo del subsistema satelital de Computadora de Vuelo partiendo de la elección de los componentes y su diagrama de configuración, la plataforma de desarrollo empleada y modelos CAD en 3D de un prototipo final de todo el subsistema.

3.1 Arquitectura electrónica de la Computadora de Vuelo y su diseño electrónico asociado

3.1.1 Microcontrolador TMS570LS1227

El TMS570LS1227 pertenece a la familia de microcontroladores de grado automotriz y alto rendimiento para sistemas seguros. Su arquitectura incluye lo siguiente:

- CPU's duales en "Lockstep"
- CPU y memoria logica Built-In Self-Test (BIST)
- ECC en Flash y SRAM
- Paridad en memorias periféricas
- Capacidad de retroalimentación en periféricos I/O's

El microcontrolador TMS570LS1227 integra un CPU ARM Cortex-R4F punto flotante que ofrece una eficiencia de 1.66 DMIPS/MHz y es configurable a 180MHz ofreciendo hasta 298 DMIPS[1, 2]. El dispositivo soporta el formato de palabra fijo "big-endian".

El TMS570LS1227 tiene integrados 1.25 MB en Flash, 192 kB en RAM y 64kB de Flash para emular EEPROM. La memoria Flash es no-volátil, borrable y programable eléctricamente, implementa una interfaz de bus de datos con ancho de 64 bits. La Flash utiliza 3.3V (así como los I/O)para las operaciones de programado, borrado y lectura.

El MCU cuenta con periféricos característicos en aplicaciones basadas en control de tiempo real e interfaces de comunicación múltiple: (N2HETs, ePWM, eCAP, eQEP, ADCs, MibSPI, SPI, LIN, SCI, DCAN, I2C, Ethernet y Flexray).

Con características de seguridad integradas, una basta elección de comunicaciones y control de periféricos, el TMS570LS1227 es una solución ideal para aplicaciones de control en tiempo real y alto rendimiento con requerimientos de seguridad críticos.

Algunas de sus aplicaciones son:

- Sistemas de frenado (ABS y ESC).
- Dirección asistida eléctricamente (EPS).
- Sistema de manejo de Baterías.
- Sistema de asistencia para conducción.
- Aeroespacial y Aeronáutica.

Diseño electrónico asociado al microcontrolador TMS570LS1227

Para el uso adecuado del microcontrolador se agregaron capacitores de filtrado de $0.1\mu F$ en cada uno de los pines de polarización y se agregó también un oscilador de 16 MHz.

A continuación se muestra la tabla de asignación de pines para el microcontrolador así como su diagrama de bloques en la figura 3.1.

PIN	NOMBRE	DESCRIPCIÓN
3	I2C_SCL	I2C Reloj.
4	I2C_SDA	I2C Datos.
6	SPI3NCS/GIO	Seleccion de dispositivo SPI o GIO.
9	GIOA[2]	Entrada de propósito general.
13	CAN3X	Bus CAN o GIO.
14	GIOA[5]	Entrada de propósito general.
16	GIOA[6]	Entrada de propósito general.

PIN	NOMBRE	DESCRIPCIÓN
18	OSCIN	A cristal externo.
19	KELVIN_GND	Tierra Kelvin para oscilador.
20	OSCOUT	A cristal externo.
22	GIOA[7]	Entrada de propósito general.
24	SPI4NCS[0]	SPI4 Selección de dispositivo.
25	SPI4CLK	SPI4 Reloj.
30	SPI4SIMO[0]	SPI4 SIMO.
31	SPI4SOMI[0]	SPI4 SOMI.
34	TEST	Prueba de habilitación.
38	SCIRX	Receptor SCI.
39	SCITX	Transmisor SCI.
46	nPORRST	Power-on reset.
64	AD1IN[20]	Entrada analógica compartida ADC1/ADC2.
65	AD1IN[21]	Entrada analógica compartida ADC1/ADC2.
66	ADREFHI	Referencia alta para ADC.
67	ADREFLO	Referencia baja para ADC.
68	VSSAD	Tierra para ADCs.
69	VCCAD	Polarización para ADCs.
70	ADIN[09]	Entrada analógica compartida ADC1/ADC2.
71	AD1IN[01]	Entrada analógica para ADC1.
89	CAN1TX	CAN1 transmisor o GPIO.
91	MIBSPI1NCS[5]	MIBSPI1 selección de dispositivo.
92	N2HET1[26]	N2HET1 o GPIO.
93	MIBSPI1SIMO[0]	MIBSPI1, SIMO.
94	MIBSPI1SOMI[0]	MIBSPI1, SOMI.
95	MIBSPI1CLK	MIBSPI1 Reloj.
96	MIBSPI1NENA	MibSPI1 habilitador o GPIO.
97	MIBSPI5SOMI[1]	MIBSPI5SOMI o GPIO.
98	MIBSPI5SOMI[0]	MIBSPI5SOMI o GPIO.
105	MIBSPI1NCS[0]	MIBSPI1 selección de dispositivo.
106	MIBSPI1SIMO[1]	MIBSPI1SIMO o GPIO.
107	N2HET1[28]	N2HET1 o GPIO.
108	TMS	JTAG Prueba de selección.
109	nTRST	JTAG Prueba de reset a Harware.
110	TDI	JTAG Prueba de entrada de datos.
111	TDO	JTAG Prueba de salida de datos.
112	TCK	JTAG Prueba de reloj.
113	RTCK	JTAG Regreso de reloj.

PIN	NOMBRE	DESCRIPCIÓN
116	nRST	Sistema de Reset.
117	nERROR	ESM. Indica error de alta la gravedad.
125	N2HET1[14]	N2HET1 o GPIO.
140	N2HET1[18]	N2HET1 o GPIO.
141	N2HET1[20]	N2HET1 o GPIO.

Tabla 3.1: Asignación de pines del MCU TMS570LS1227

Las terminales de polarización de I/O's (Vccio) son: 10, 26, 42, 104, 120, 134 y 136.
Las terminales de polarización de núcleo (Vcc) son: 17, 29, 45, 48, 49, 57, 87, 101, 114, 123, 137 y 143.
Las terminales de referencia (Vss) son: 11, 21, 27, 28, 43, 44, 47, 50, 56, 88, 102, 103, 115, 121, 122, 135, 138 y 144.

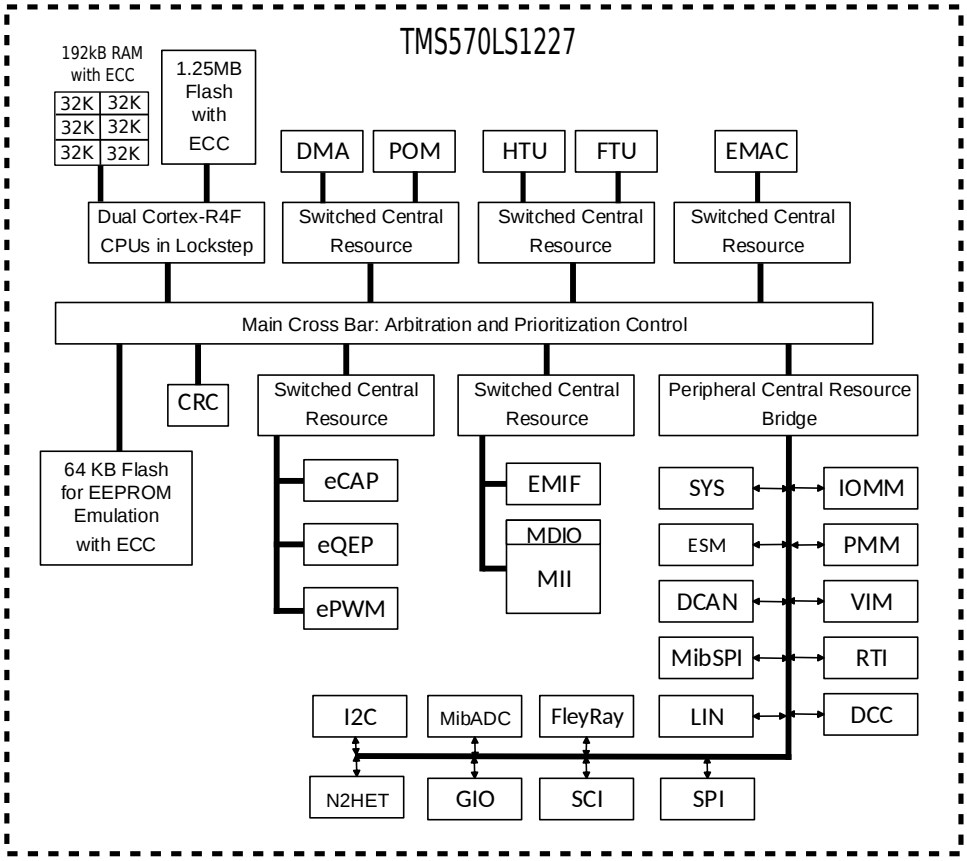


Figura 3.1: Diagrama a bloques del MCU TMS570LS1227

3.1.2 Microcontrolador MSP430F1612

La familia de microcontroladores MSP430 de Texas Instruments[3, 4, 5] de bajo consumo de energía consiste de varios dispositivos con diferentes conjuntos de periféricos especificados para diversas aplicaciones. La arquitectura, combinada con 5 modos de operación en bajo consumo es óptimo para lograr una larga vida de baterías en aplicaciones portables.

El MSP430 incorpora un poderoso CPU RISC de 16 bits, registros de 16 bits, periféricos, un sistema de reloj flexible que se interconecta usando un bus de direccionamiento de memoria común Von-Neumann(MAB), un bus para datos de memoria(MDB), 55kB en Flash y 5kB en SRAM.

La serie de microcontroladores MSP430 15x/16x/151x cuentan con: timers de 16 bits, convertidores AD y DA de 12 bits, comunicación UART, I2C, DMA y 48 lineas I/O.
Para el uso adecuado del microcontrolador se agregaron capacitores de filtrado de $0.1\mu F$ en cada uno de sus pines de polarización y se agregó un oscilador de 8 MHz.
A continuación se muestra la tabla de asignación de pines para el microcontrolador así como su diagrama de bloques.

PIN	NOMBRE	DESCRIPCIÓN
1	DVcc	Fuente de Voltaje Digital, terminal positiva.
8	XIN	Entrada de Oscilador de Cristal.
9	XOUT	Salida de Oscilador de Cristal.
13	P1.1	BSL, Transmitir.
15	P1.3	I/O Digital de Propósito General.
20	P2.0	I/O Digital de Propósito General.
21	P2.1	I/O Digital de Propósito General.
22	P2.2	BSL, Recibir.
23	P2.3	I/O Digital de Propósito General.
24	P2.4	I/O Digital de Propósito General.
25	P2.5	I/O Digital de Propósito General.
29	SDA	Modo I2C, Entrada de Datos.
31	SCL	Modo I2C, Reloj.
32	P3.4/UTXD0	Modo UART, Transmitir.
33	P3.5/URXD0	Modo UART, Recibir.
38	P4.2	I/O Digital de Propósito General.
39	P4.3	I/O Digital de Propósito General.
44	P5.0	Modo SPI, Selección.
45	P5.1	Modo SPI, SIMO.

PIN	NOMBRE	DESCRIPCIÓN
46	P5.2	Modo SPI, SOMI.
47	P5.3	Modo SPI, Reloj.
57	TCK	Reloj para cargado mediante bootstrap loader.
58	RST/NMI	BSL, Inicio.
59	P6.0	I/O Digital de Propósito General.
60	P6.1	I/O Digital de Propósito General.
62	AVss	Fuente de Voltaje Analógica, terminal negativa.
63	DVss	Fuente de Voltaje Digital, terminal negativa.
64	AVcc	Fuente de Voltaje Analógica, terminal positiva.

Tabla 3.2: Asignación de pines del MCU MSP430F1612

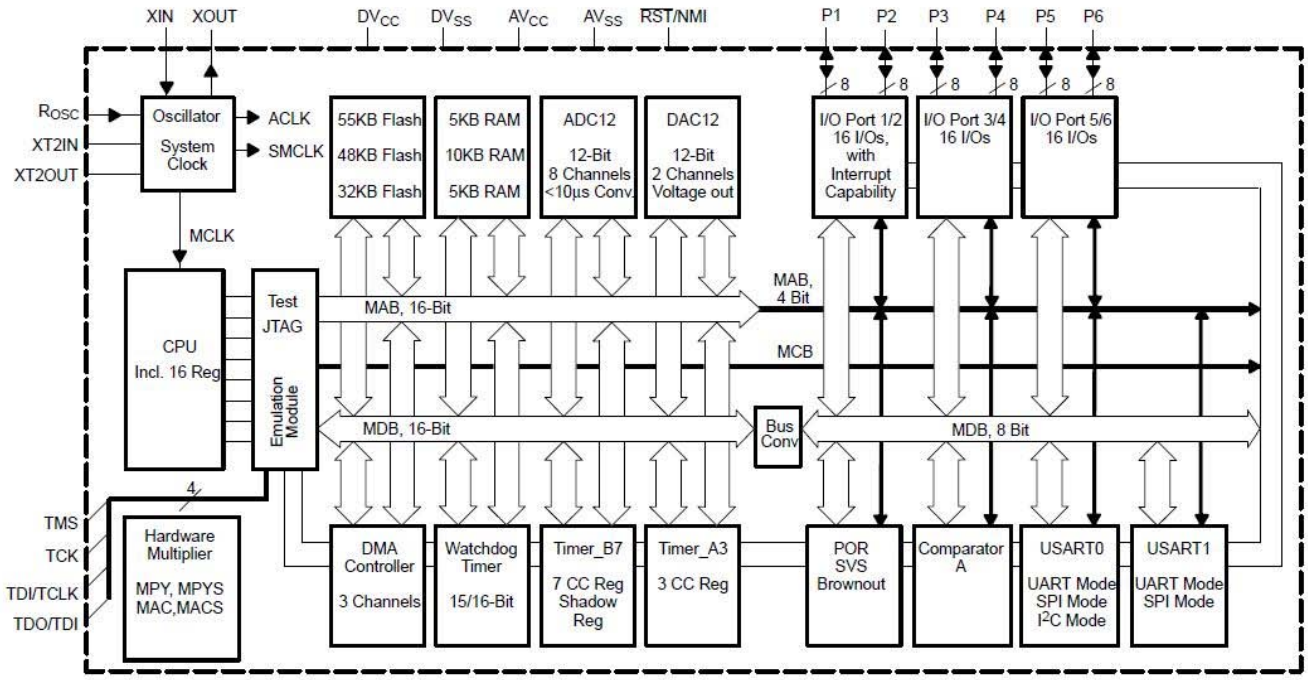


Figura 3.2: Diagrama a bloques del MCU MPS430F1612

3.1.3 Memorias Flash

Las dos principales tecnologías dominantes del mercado hoy en día son las memorias Flash no volátiles: NOR y NAND[6, 7, 8]. Las memorias Flash NOR fueron las primeras y fueron introducidas por Intel en 1988 revolucionando un mercado que era dominado por dispositivos EPROM y EEPROM. Las memorias Flash NAND fueron introducidas por

Toshiba en 1989.

Las memorias Flash se han convertido en una poderosa tecnología de almacenamiento de estado sólido ampliamente utilizadas en dispositivos electrónicos, móviles y otras aplicaciones. Las memorias Flash NAND fueron diseñadas con tamaño de celdas pequeños y sus principales usos se orientan al almacenamiento de datos de alta densidad como cámaras digitales y USB de estado sólido. Por otro lado, las memorias Flash NOR se usan generalmente para almacenar datos de código y ejecución directa como teléfonos celulares y PDA's.

La figura 3.3 muestra una gráfica comparativa entre memorias Flash NOR y Flash NAND donde el centro es una medida baja y el extremo es una medida alta.

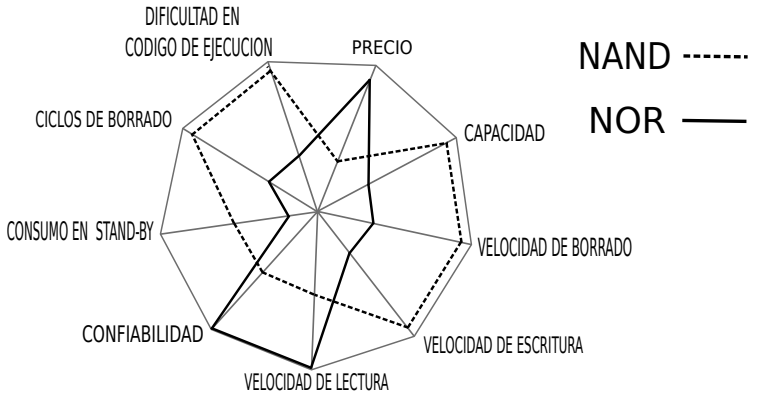


Figura 3.3: Gráfica comparativa entre memorias Flash

Con base en esta gráfica lo ideal para la CV sería ocupar memorias Flash tipo NOR para el cargado de programa y almacenado de nuevo programa y las memorias Flash NAND para el almacenamiento de datos recibidos de cargas útiles como podrían ser imágenes, telemetría entre otros. El problema es que la disposición de memorias NAND es muy poca puesto que son muy comerciales además el protocolo de comunicación deseado en la tarjeta es SPI y estas memorias difícilmente se encuentran diseñadas para este protocolo de comunicación por lo que se optó por el uso de memorias Flash NOR para almacenado de programa de ejecución y tambien para telemetría. Además las memorias NOR son más confiables y el tiempo de desarrollo de software operativo es menor.

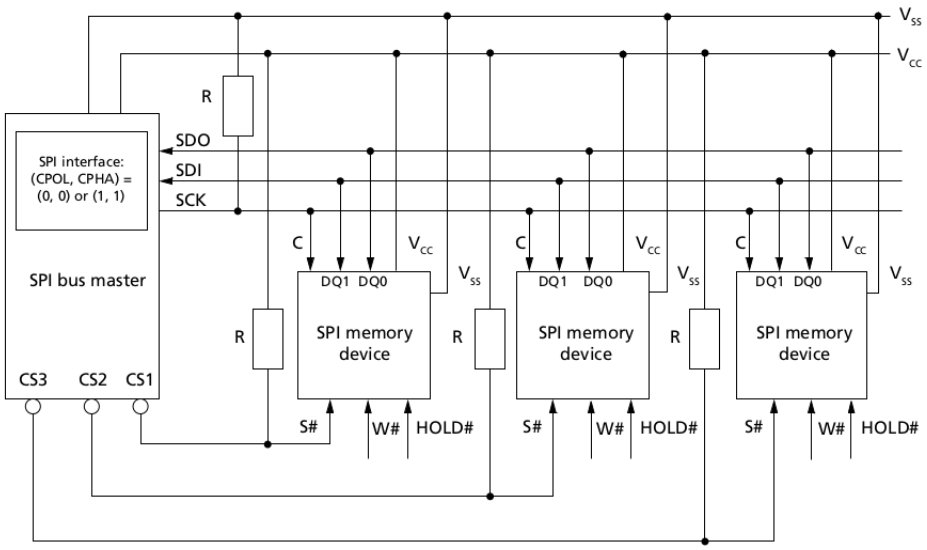


Figura 3.4: Configuración de memorias Flash NOR

3.1.4 Sensores de Temperatura

Una parte importante dentro de las tareas de la Computadora de Vuelo es el monitoreo constantemente de la temperatura de algunos circuitos o algún punto en específico del Satélite. Por otro lado un sensor es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas.

El Satélite consta básicamente con dos tipos de sensores, el sensor de temperatura DS18b20 de Maxim Semiconductors[9]y el segundo, el sensor interno del microcontrolador MSP430.

Los sensores de temperatura estarán distribuidos en todo el Satélite y la CV será la encargada de reunir toda esta información.

3.1.4.1 Sensores 1-Wire

El termómetro digital DS18b20 proporciona mediciones de temperatura en grados Celsius con resolución de 9 a 12 bits. El sensor se comunica a través de un bus 1-Wire, figura3.5 el cual requiere únicamente una línea de datos(y tierra) para comunicación con un microprocesador. Su rango de operación es de -55°C a +125°C con precisión de +/- 0.5 °C sobre rangos de -10°C a +85°C. Además el DS18B20 puede alimentarse de la línea de datos (en modo "parasite power"), eliminando la necesidad de una fuente externa, sin embargo, esta forma no es muy recomendable para temperaturas por encima de +100°C,

ya que el DS18B20 puede fallar en la comunicación debido a las altas corrientes de fuga que pueden existir a estas temperaturas.

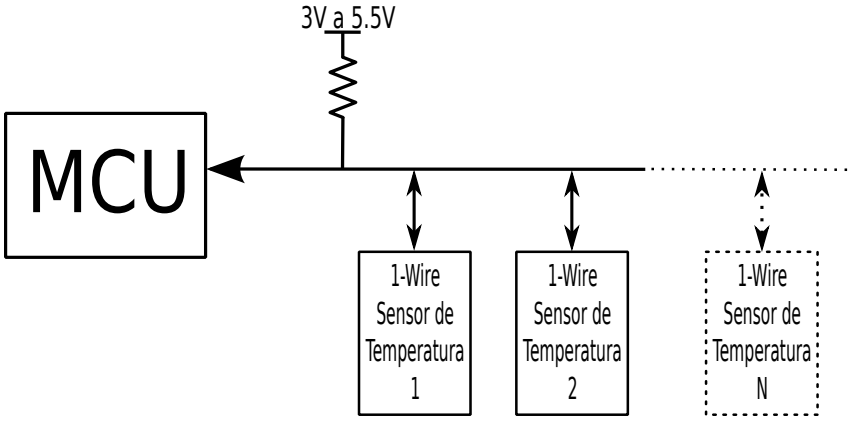


Figura 3.5: Conexión de sensores de Temperatura 1-Wire

3.1.4.2 Sensor Interno de MSP430

Como se mencionó con anterioridad, una de las tareas de la CV es monitorear las temperaturas de algunos dispositivos. El MSP430F1612 tiene un sensor de temperatura interno que se encuentra conectado a un canal del ADC, esto permite medir de manera precisa la temperatura del MCU sin necesidad de sensores adicionales. Este método es replicado en varios subsistemas dentro del Satélite.

La figura 3.6 muestra la función de transferencia del sensor de temperatura.

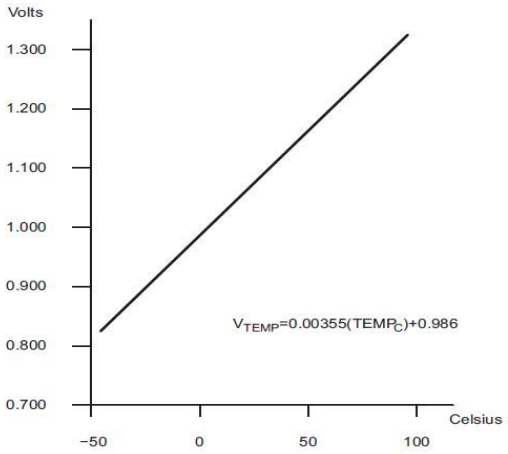


Figura 3.6: Función de transferencia típica del sensor de temperatura del MSP430

3.1.5 Protección contra efecto "Latch-up"

Muchos circuitos CMOS son sensibles al efecto "latchup" que es una de las mayores preocupaciones cuando se evalúan dispositivos CMOS para aplicaciones espaciales[10].

El efecto "latchup" es un fenómeno donde un dispositivo semiconductor es sometido a un estado de alta corriente como resultado de interacción entre transistores bipolares NPN y PNP. Este efecto puede conducir a la destrucción de un circuito semiconductor, empaquetado o sistema . La magnitud de la corriente es tal que generalmente semiconductores de silicio, aluminio y cobre fallan al punto que algunas veces los empaquetados se funden así que lo primero que se hizo fue decidir como solucionar este problema.

3.1.5.1 Técnicas de sensado

Existen diversas técnicas para control y sensado de corriente así como diversas aplicaciones [11, 12]. Cada una tiene ventajas y desventajas, en esta sección se mencionan solo dos.

Sensado de corriente en Lado Bajo

Esta técnica ocupa una resistencia de sensado en serie en dirección de la carga a tierra. En este caso la corriente generalmente fluye en una dirección (unidireccional).

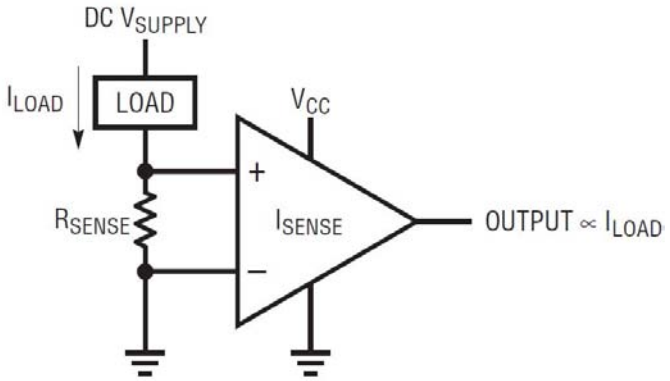


Figura 3.7: Sensado de corriente en lado bajo

Ventajas :
- La entrada de voltaje es baja.
- El voltaje de salida es referenciado a tierra.
- Fácil diseño de suministro.

Desventajas :
- Carga elevada de conexión directa con tierra.
- No se detecta corriente en la parte alta de la carga causada por un corto circuito.

Sensado de corriente en Lado Alto

Esta técnica ocupa una resistencia de sensado en serie en dirección de polarización a la carga. En este caso la corriente generalmente fluye en una dirección (unidireccional).

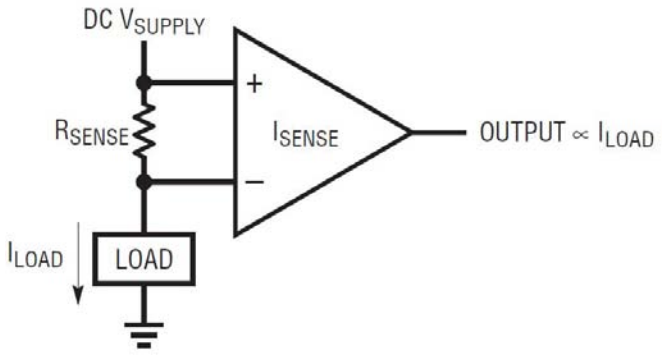


Figura 3.8: Sensado de corriente en lado alto

Ventajas :
- La carga va directamente a tierra.
- La carga no se activa por corto accidental al polarizar.
- Se detecta una sobrecorriente en la parte alta de la carga.

Desventajas :
- Los voltajes en modo común son altos.
- La salida tiene que estar balanceada con los niveles de operación del sistema.

3.1.5.2 Componente Front-End(BQ24314)

El componente Front-End es un circuito integrado que nos proporciona protección contra sobre voltajes, sobre corrientes y cargado de baterías aunque este último no es el caso[13]. El BQ24314 monitorea continuamente los voltajes de entrada y corrientes de entrada(figura 3.9). En caso de una condición de sobre voltaje, el circuito integrado elimina la alimentación del circuito de carga (en nuestro caso apaga la carga) abriendo un interruptor interno. En una condición de sobre corriente, este limita la corriente del sistema a un valor y si la sobrecorriente persiste, apaga al sistema después de un periodo. Ademas el circuito integrado vigila su propia temperatura y este se apaga si su temperatura es muy elevada.

El circuito integrado puede ser controlado por un procesador y proporcionar información de estados como las condiciones de falla de la carga. La CV contiene cinco circuitos integrados de este tipo, uno para cada microcontrolador y los restantes monitorean las memorias Flash. Las señales de falla y habilitación de los microcontroladores son monitoreados por el subsistema de Potencia, puesto que este decide después de varias fallas si

todo el subsistema deberá o no apagarse, mientras que las señales de las memorias Flash son monitoreadas localmente para apagarlas en caso de estar cargando datos y que se presente una condición de sobrecorrientes o voltajes.

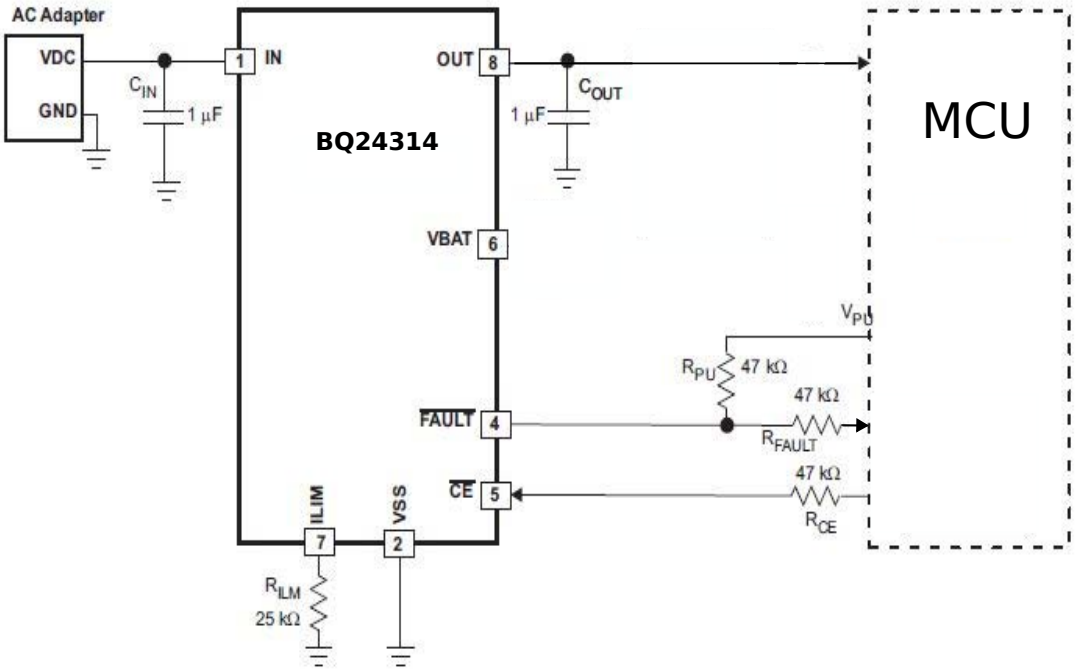


Figura 3.9: Aplicación típica de BQ24314 (Front End)

El valor de sobrecorriente puede ser programado mediante una resistencia y este valor se puede aproximar a la ecuación siguiente.

$$I_{OCP} \approx 25 \div R_{ILIM} (\text{corriente en A, resistencia en kohm}) \tag{3.1}$$

Otra método de aproximación para elegir la resistencia que limita la corriente es mediante la figura 3.10 en la que se observa la relación de resistencia contra corriente dentro del margen especificado por el fabricante.

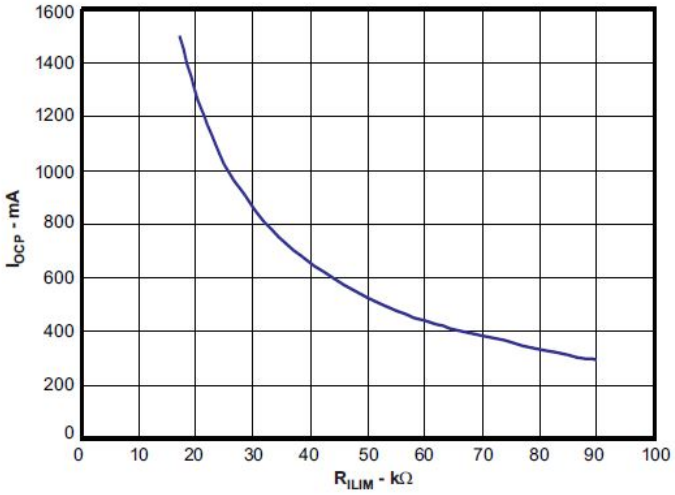


Figura 3.10: Resistencia vs corriente del componente BQ24314

Diseño electrónico asociado al componente BQ24314

Para el caso del microcontrolador Hércules (TMS570) fue necesario calcular su máxima corriente de operación y esta se obtuvo de acuerdo a la frecuencia del reloj con que opera en 1.5 ma/MHz. En nuestro caso el MCU opera a 16 MHz pero su corriente máxima se calculó para operar a 180MHz (frecuencia máxima) obteniendo una corriente de 300mA. Una vez obtenido este valor se procedió a calcular la resistencia para el circuito BQ24314 respecto a la fórmula mencionada anteriormente.

$$R_{ILIM} \approx \frac{25}{I_{OCP}} \approx 83.33k\Omega \tag{3.2}$$

3.1.5.3 Amplificador MAX4372 y Comparador LM6511

El componente Front-End es muy útil por todas las características que tiene, pero su única limitante es que tiene un rango de sensado de corriente entre 300mA y 1.5A. Esto es un gran problema puesto que las memorias Flash NOR y el MSP430 operan en un rango aproximado de 20mA. En esta subsección se describe el circuito diseñado para resolver este problema así como los componentes requeridos.

Amplificador MAX4372H

El MAX4372 es un circuito integrado de bajo costo, precision y amplificador de sensado de corriente en la parte alta de la carga, figura 3.11. La corriente fluye a través de una resistencia de sensado generando un voltaje de sensado amplificado. El MAX4372

cuenta con tres ganancias, 20, 50 y 100 [V/V] y esta se elige con la terminación del empaquetado T,F y H respectivamente. Fue necesario poner un capacitor de filtrado en la terminal de polarización y únicamente la resistencia de sensado.

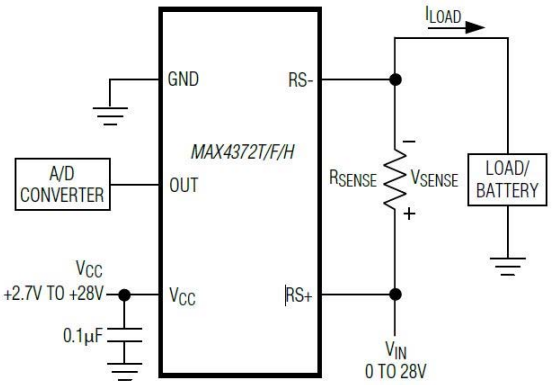


Figura 3.11: Configuración del MAX4372

Para escoger el valor de la ganancia y resistencia a usar en el sistema primeramente se utilizó la siguiente tabla.

Escala completa de corriente de carga, ILOAD(A)	Resistor de sensado de corriente, RSENSE ($m\Omega$)	Ganancia (V/V)	Escala completa de voltaje de salida VOUT(V)
0.1	1000	20 50 100	2.0 5.0 10.0
1	100	20 50 100	2.0 5.0 10.0
5	20	20 50 100	2.0 5.0 10.0
10	10	20 50 100	2.0 5.0 10.0

Tabla 3.3: Tabla de resistencias de sensado de corriente recomendadas

En nuestro caso el valor seleccionado para la resistencia fue de 1Ω y la ganancia seleccionada fue de 100 V/V. Una vez que se obtiene la ganancia y la resistencia de sensado se tiene un voltaje de salida que varía proporcionalmente con la corriente sensada. Para tener la relación entre la corriente sensada y el voltaje de salida, el fabricante nos proporciona la siguiente fórmula:

$$Vo = (G[V/V]) \times (Rsense[\Omega]) \times (i[A]) \tag{3.3}$$

De donde:

G: Ganancia.
Rsense: Resistencia de sensado.
i: Corriente deseada para observar que voltaje de salida se obtendrá.

Comparador LM6511

Con una fuente de alimentación de 2.7 a 3.6 V y tiempo de respuesta de 180 ns, el LM6511 funciona como comparador de un voltaje de referencia, figura 3.12. Para nuestro caso se fija un voltaje de referencia y éste se compara con el voltaje obtenido del MAX4372.

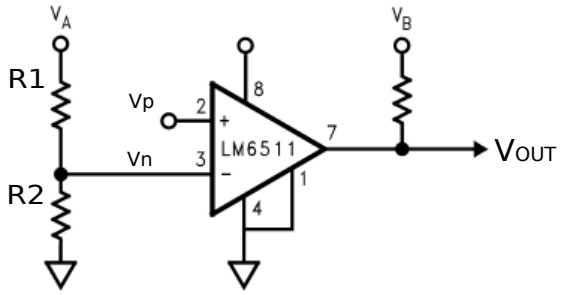


Figura 3.12: Configuración de comparador LM6511

Para fijar el voltaje de referencia se realizó un divisor de voltaje basado en la figura 3.13 y quedando la ecuación 3.4 :

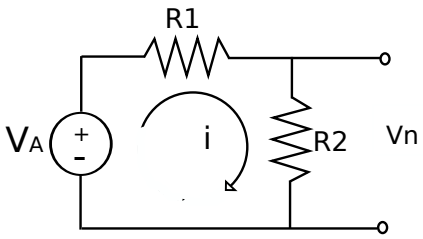


Figura 3.13: Circuito base para divisor de voltaje

$$V_{R2} = V_A \frac{R_2}{R_1 + R_2} \tag{3.4}$$

Teniendo en cuenta que VOUT es el producto de una ganancia Av multiplicada por la diferencia de Vp y Vn, es decir:

$$\text{VOUT} = \text{Av (Vp - Vn)}$$

Se observa que si Vp es menor a Vn se obtiene una ganancia negativa que, debido a la configuración del comparador, da un voltaje de salida (VOUT) igual a 0. Y por el contrario, un voltaje mayor en Vp nos entregaría un voltaje igual al de polarización (3.3V).

Para encontrar los valores de resistencias y obtener el voltaje de referencia basta con despejar la ecuación (3.4) y proponer un valor de resistencia con lo que la ecuación quedará de la siguiente forma:

$$R_1 = \frac{(R_2)(V_A)}{V_{R2}} - R_2 \tag{3.5}$$

Diseño electrónico asociado a la protección contra evento "Latchup"

Una vez descritos los componentes requeridos [14, 15] se procede a explicar el sistema completo para proteger las memorias y el microcontrolador MSP430F1612 contra el efecto "Latchup". En un principio estaba pensado emplear únicamente los dos circuitos antes mencionados con los cuales se sensaba la corriente y el voltaje de sensado era recibido por el comparador para obtener una señal de alarma. Esta sería monitoreada por el MCU del subsistema de Potencia por medio de interrupciones pero esta protección sería por medio de software consumiendo ciclos de reloj y el tiempo sería prolongado. En nuestro caso todo este problema se resolvió mediante hardware como se observa en la figura 3.14.

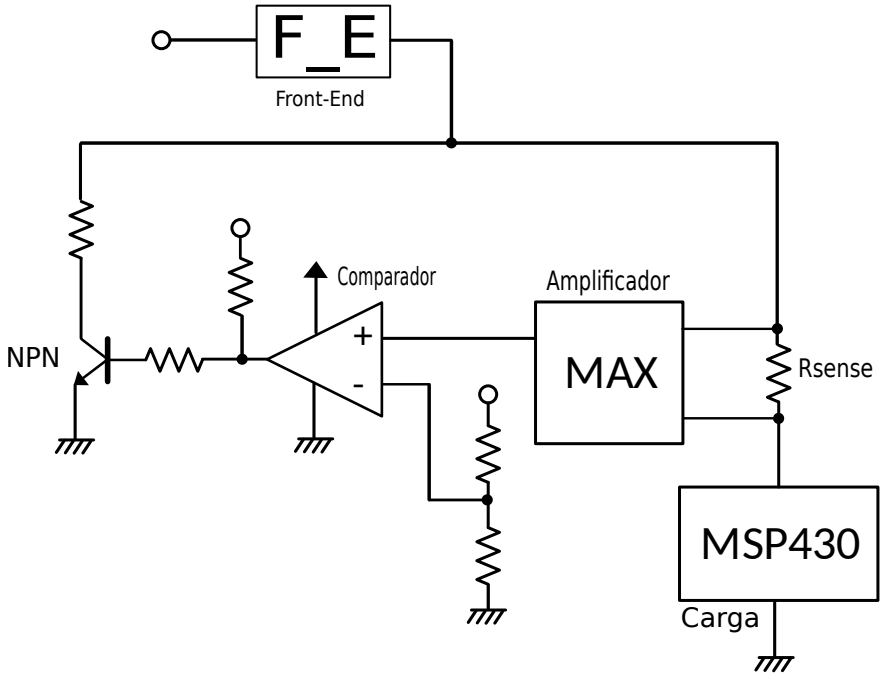


Figura 3.14: Sistema protector de evento Latchup

En este sistema, el CI BQ24314 monitorea constantemente la corriente de polarización de la carga. En caso de presentarse un consumo de sobrecorriente, ésta es detectada por el MAX4372. El voltaje de sensado esta conectado a la terminal positiva del comparador, de esta forma se compara el voltaje de sensado con un voltaje de referencia obteniendo un pulso de alarma que activa a un transistor diseñado para consumir 300mA y de esta manera activar el Front-End BQ24314.

Todo esto se logra vía Hardware y la velocidad de reacción es mayor. El CI BQ24314 se encarga de mandar una señal de aviso al subsistema de Potencia o al MCU de CV (en caso de ser las memorias) y estos se encargan de habilitar la carga o mantenerla apagada algún tiempo.

Para el caso del MSP430F1612 su consumo de corriente se basa en la siguiente tabla.

Familia de Dispositivo	Nombre de Pin	Voltaje(V) Min | Max	CPU IMAX (μA/MHz)	Analógico IMAX(μA)	Comentarios
x11x1A	Vcc	1.8 3.6	350	Comp_A=60	C11x1:300μA/MHz max
F12x	Vcc	1.8 3.6	350	Comp_A+=60	
F11x2,12x2	Vcc	1.8 3.6	350	ADC10=1200, IREF=400	
F13x,14x[1]	Avcc, Dvcc	1.8 3.6	560	Comp_A=60 ADC12=1600 IREF=800	F13x, 14x:Comp_A, ADC12 F14x1:Comp_A
F15x,16x,161x	Avcc, Dvcc	1.8 3.6	600	Comp_A=60 ADC12=1600 IREF=800, DAC12=1500	Salidas de DAC sin carga

Tabla 3.4: Tabla de consumo de MSP430Fxxxx

De donde podemos concluir que:

$$\text{Imax}=600[\mu\text{A/MHz}] \times \text{f(sistema) [MHz]}$$

El MSP430F1612 trabajará a una frecuencia de 8MHz obteniendo así una corriente de 4.8mA pero se decidió fijar un consumo máximo de 17mA, debido a que aquí no se agregan los valores de corriente al momento de cargado de programa que es de 7mA más la corriente requerida por los periféricos. Para el caso de las memorias NOR también se fijó una corriente máxima de 17mA. De esta manera y con base en la ecuación (3.3)

$$Vo = (100[V/V])\ X\ (1[\Omega])\ X\ (.017[A]) = 1.7V \tag{3.6}$$

Con respecto a la ecuación (3.5) y dando un valor de 1 $k\Omega$ a R2

$$R_1 = \frac{(1k)(3.3V)}{1.7V} - 1k = 941.176\Omega \tag{3.7}$$

Por lo tanto se fijaron los valores del comparador de R1 Y R2 en $1k\Omega$.

En cuanto al transistor 2N2222A con una $\beta_{MIN} = 75$, V_{CE}=0.3V y un V_{BE}=0.6V, figura 3.15.

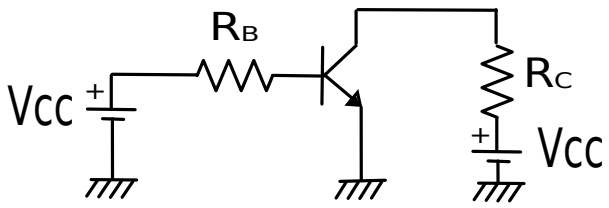


Figura 3.15: Transistor NPN

En la malla del Colector se tiene:

$$Vcc - (R_C)(I_C) - V_{CE} = 0 \tag{3.8}$$

Por lo tanto:

$$R_C = \frac{Vcc - V_{CE}}{I_C} = \frac{(3.3V) - (0.3V)}{300mA} = 10\Omega \tag{3.9}$$

En la malla de la Base se tiene:

$$Vcc - (R_B)(\frac{I_C}{\beta}) - V_{BE} = 0 \tag{3.10}$$

Por lo tanto:

$$R_B = (\frac{V_{CC} - V_{BE}}{I_C})(\beta) = \frac{(3.3V - 0.6V)}{300mA}(75) = 675 \tag{3.11}$$

Por los valores comerciales de resistencias se fijo un valor para $R_B = 650\ \Omega$ y $R_C = 10\ \Omega$

3.1.6 Magnetómetro

El magnetómetro de la marca Honeywell HMC5883L es de montaje superficial diseñado en un módulo multichip para el sensado de campo magnético con una interfaz digital para algunas aplicaciones como brújula. Algunas de sus características son las siguientes:

- Sensor Magnetoresistivo en 3 ejes.
- Voltaje de operacion bajo.
- Interfaz digital I2C.

- Software y algoritmos disponibles.

El HMC5883L utiliza tecnología Magnetoresistiva anisotrópica la cual provee ventajas por encima de otras tecnologías de sensores magnéticos. La construcción de estado sólido de estos sensores con muy baja sensibilidad transversal diseñados para medir la magnitud y dirección de campo magnético con un intervalo de mili gauss y una resolución de 8 gauss. Los sensores magnéticos de Honeywell se encuentran dentro de los sensores mas sensibles y fiables de la industria.

El circuito de sensado magnetoresistivo se compone de 3 sensores y circuitos de soporte para aplicaciones específicas para medir campo magnético. Con la fuente de energía aplicada, el sensor convierte cualquier campo magnético incidente en las direcciones de los ejes sensibles a una salida de voltaje diferencial. El sensor Magnetoresistivo está hecho de un hilo delgado de níquel hierro y se modela como un elemento de banda resistiva. En presencia de un campo magnético un cambio en los elementos resistivos del puente causa un correspondiente cambio en el voltaje a través de las salidas del puente.

Estos elementos resistivos están alineados entre sí para tener un eje sensible en común que proporcionará un cambio positivo de voltaje con campos magnéticos crecientes en la dirección sensible. Debido a que la salida es solo proporcional a la componente del campo magnético a lo largo de su eje, se colocan puentes de sensores adicionales en direcciones ortogonales para permitir la medición exacta del campo magnético en cualquier orientación.

Además de mantener todos los componentes que pueden contener materiales ferrosos (níquel, etc) a cierta distancia del sensor en ambos lados del PCB, se recomienda también que no haya cobre conductor cerca del sensor en cualquiera de las capas de PCB. Por tal motivo se procuró hacer las conexiones como lo muestra la imagen 3.16.

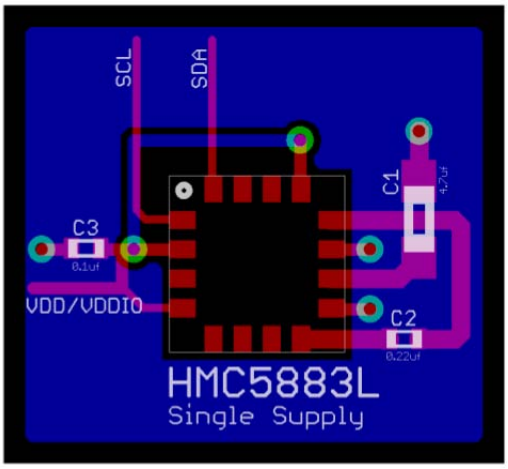


Figura 3.16: Recomendaciones de conexión para el magnetómetro

3.2 Desarrollo del circuito impreso del subsistema Computadora de Vuelo

El Satélite Experimental se desarrolla como un sistema digital de alta calidad tanto en hardware como en software, por tal razón, cada uno de sus subsistemas se debe desarrollar con base en herramientas que generen productos de alta calidad. En el caso del hardware, se emplean herramientas de diseño asistido por computadora para generar circuitos impresos de alta calidad, los cuales se mandan a fabricar puesto que la densidad de las pistas y el tamaño de los componentes complica demasiado la fabricación del impreso.

3.2.1 Plataforma de diseño Altium Design 10

El desarrollo de la tarjeta impresa se realizó con la ayuda del software Protel DXP por la compañía Altium versión 10,[16] y esto se debe a las características favorables que tiene como son:

- Todos los planos de diseño se integran bajo el mismo espacio de trabajo (esquemáticos, PCBs, simulaciones, bibliotecas adicionales de componentes, etc.).
- Amplias bibliotecas de símbolos de componentes.
- Ruteo manual y automático.
- Sincronizan entre diagrama esquemático y PCB.
- Reglas de ruteo.

- Verificación de errores.
- Vista final en 3D.
- Generación de archivos de fabricación de PCBs, etc.

3.2.2 Metodología para generación de un PCB

El proceso de diseño y manufactura de circuitos electrónicos mediante el uso del software Altium Design 10, se puede describir de manera general como sigue:

1.-Creación del proyecto de la tarjeta electrónica.
2.-Generación de componentes que no estén dentro del programa.
3.-Captura del esquemático.
4.-Verificación del diseño eléctrico del esquemático.
5.-Generación de lista de redes.
6.-Generación de la tarjeta impresa con las dimensiones adecuadas.
7.-Posicionamiento de componentes en la tarjeta mediante técnicas de diseño.
8.-Ruteo de redes de forma manual o automática.
9.-Verificación final de las reglas de diseño.
10.-Generación de archivos de salida para la manufactura.

3.2.3 Esquemáticos de la Computadora de Vuelo

A continuación se presentan los esquemáticos del diseño de la Computadora de Vuelo. Se dividió en secciones para tener una mejor visualización del diseño y al final los componentes principales se conectan mediante puertos en un esquemático, figuras 3.17, 3.18, 3.19, 3.20, 3.21.

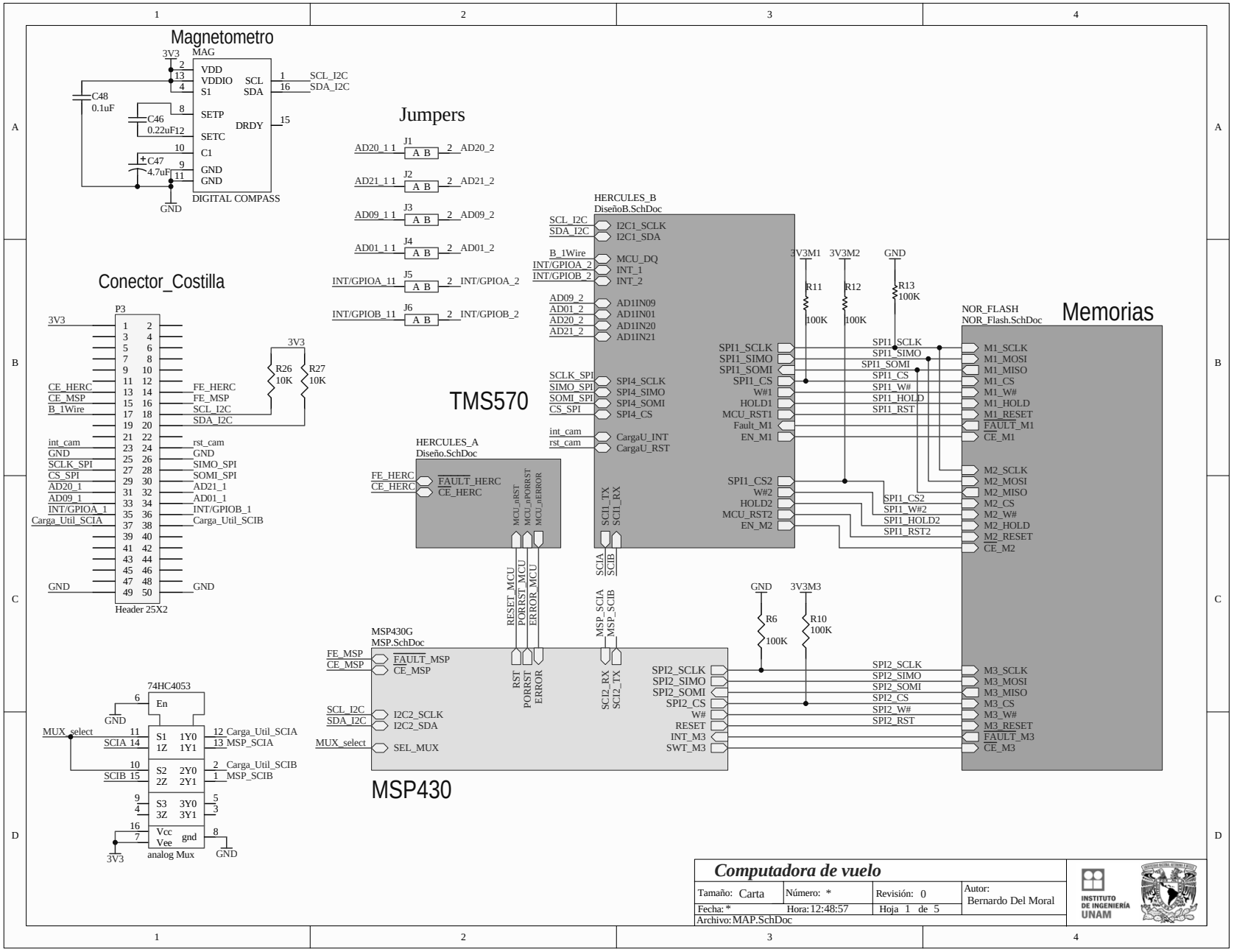


Figura 3.17: Esquemático: Vista general de los principales componentes

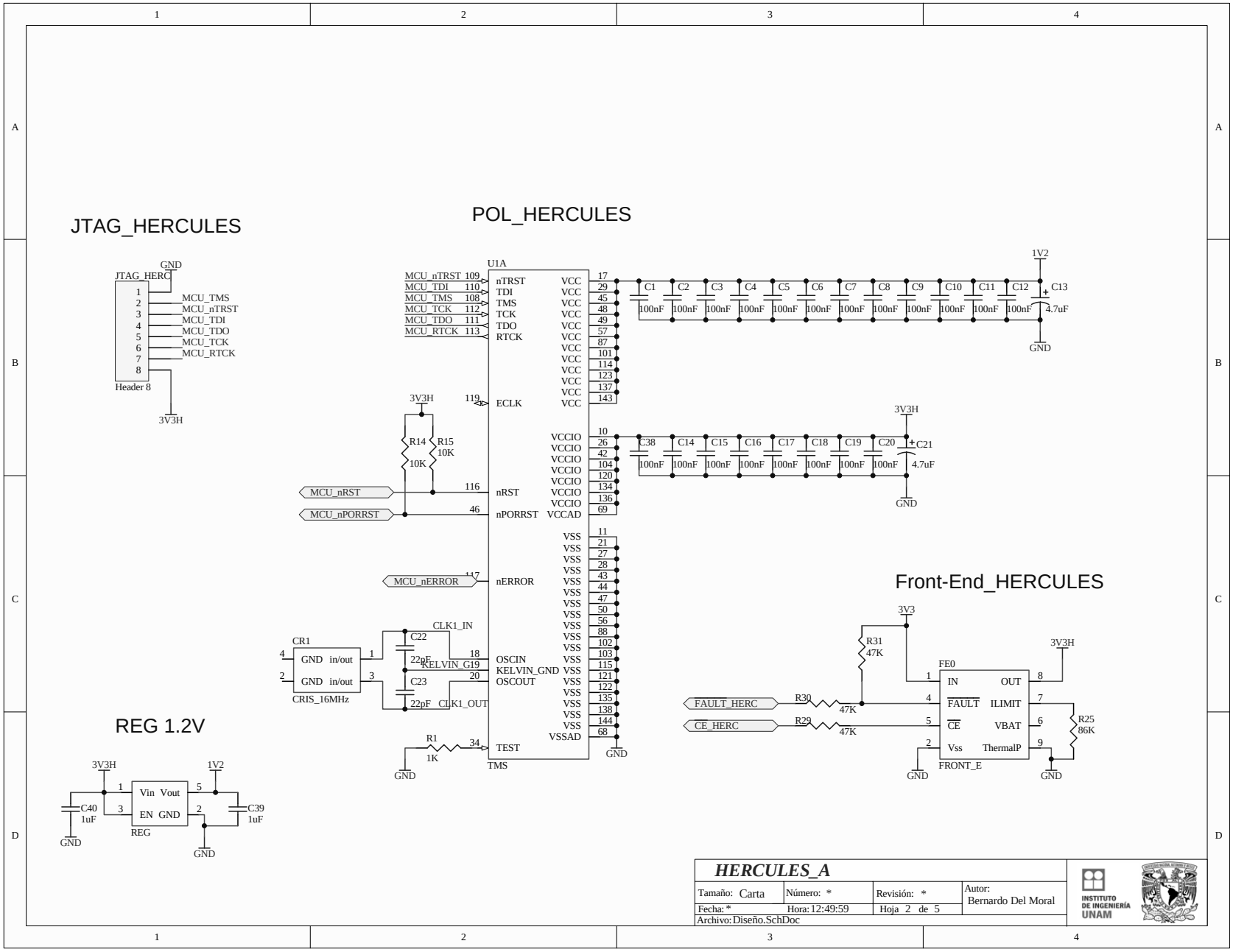


Figura 3.18: Esquemático:Polarización de núcleos y periféricos del TMS570LS1227

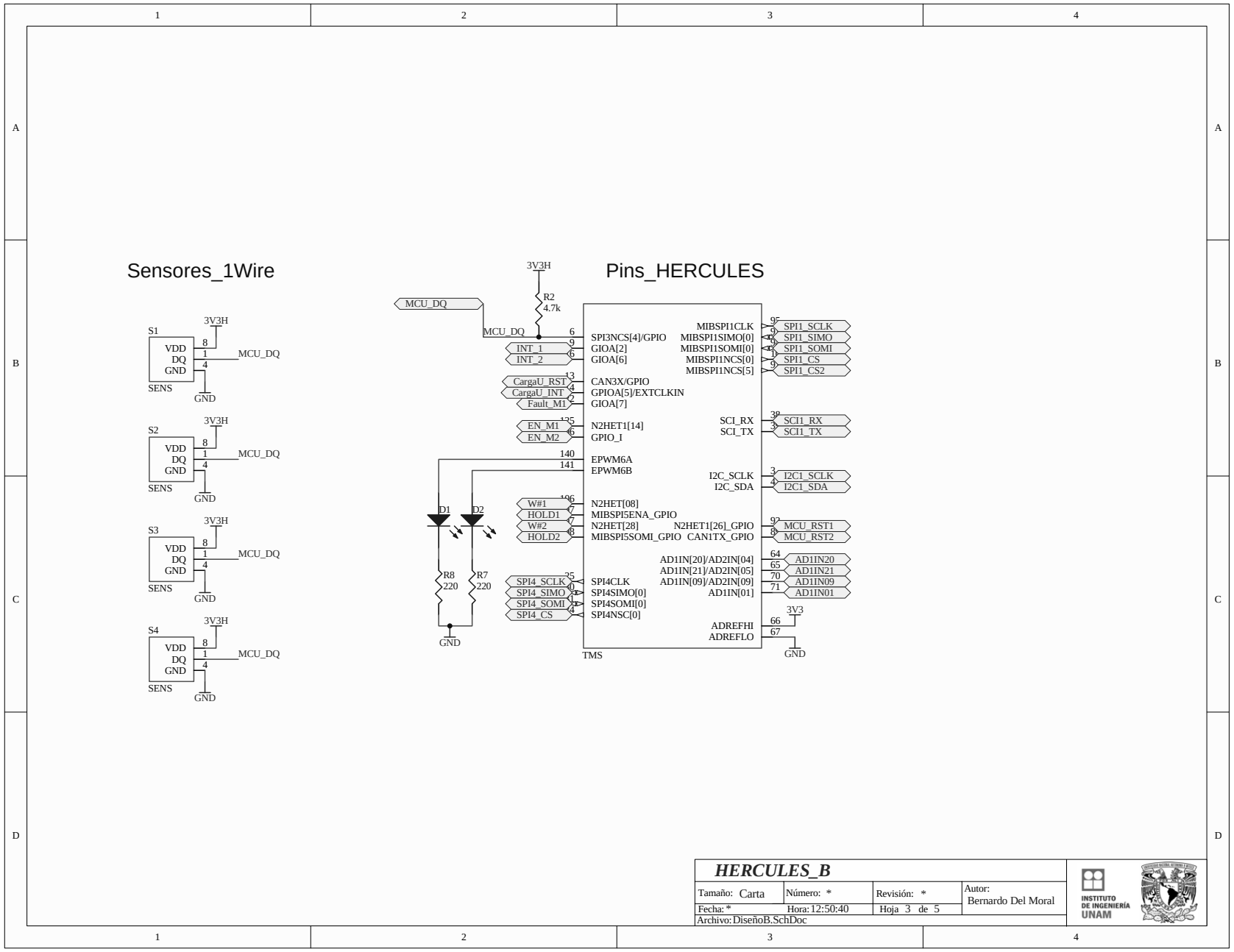


Figura 3.19: Esquemático:Puertos y periféricos del TMS570LS1227

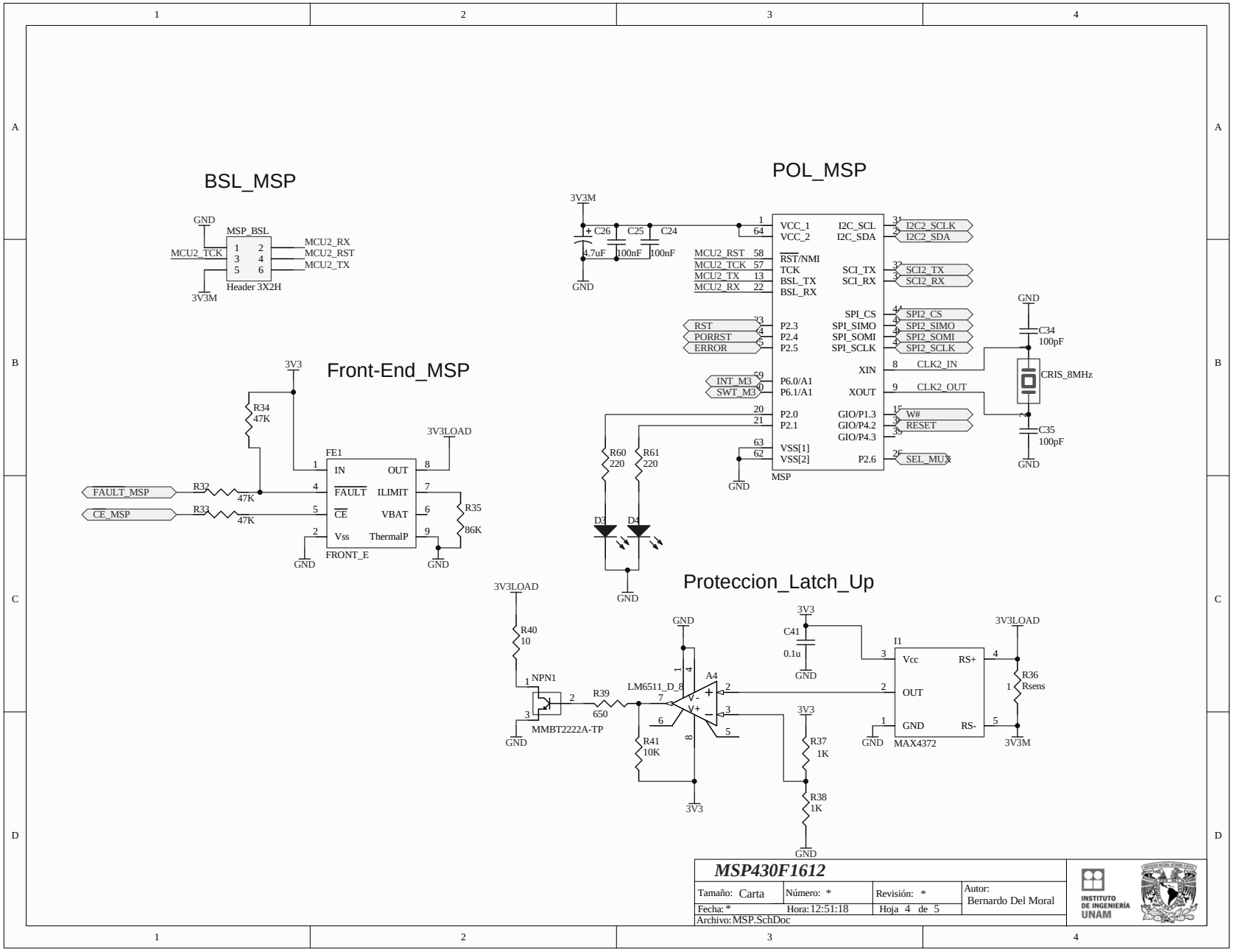


Figura 3.20: Esquemático:Polarización de MSP430F1612 y puertos de conexión

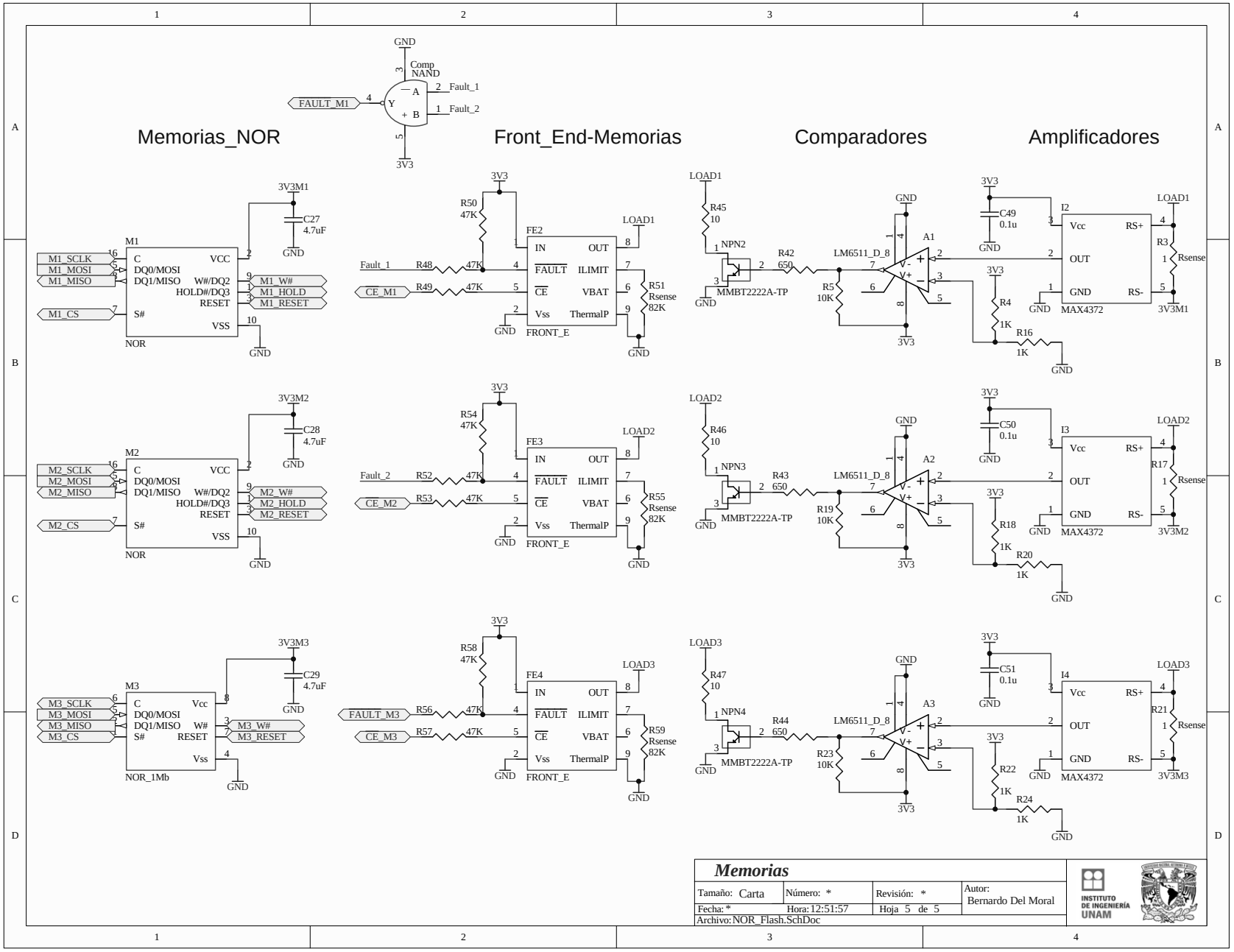


Figura 3.21: Esquemático:Polarización de memorias Flash NOR, protecciones y puertos de conexión

3.2.4 PCB y Tarjeta Electrónica

Otra de las características de Altium Protel es una vista preliminar de nuestra PCB en 2D y 3D figuras 3.22, 3.23, 3.24, 3.25. Esto fue de gran utilidad ya que se observó la tarjeta desde una vista superior e inferior así como una aproximación de su interacción con la estructura que la rodea.

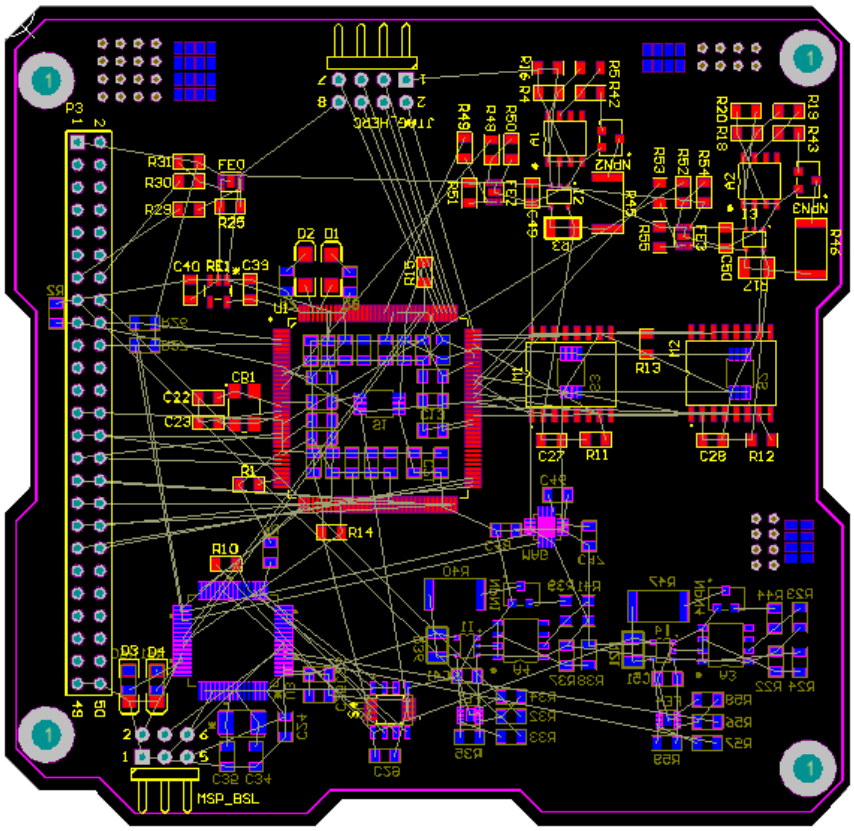


Figura 3.22: Tarjeta electrónica de Computadora de Vuelo sin rutear

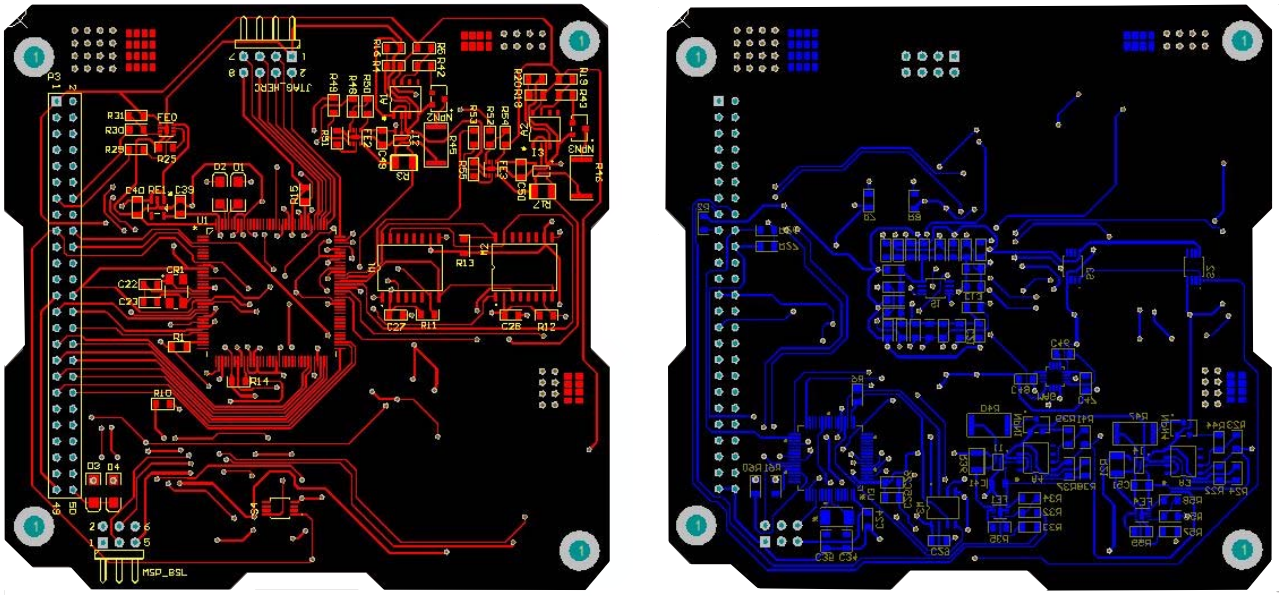


Figura 3.23: Vista superior e inferior de la Computadora de Vuelo en 2D

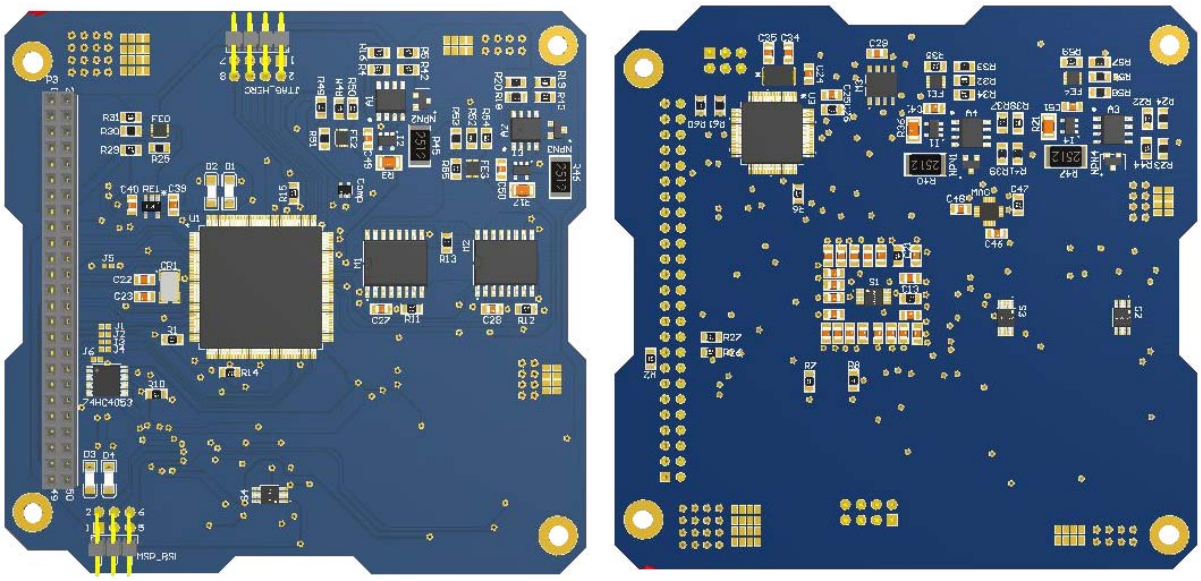


Figura 3.24: Vista superior e inferior de la Computadora de Vuelo en 3D

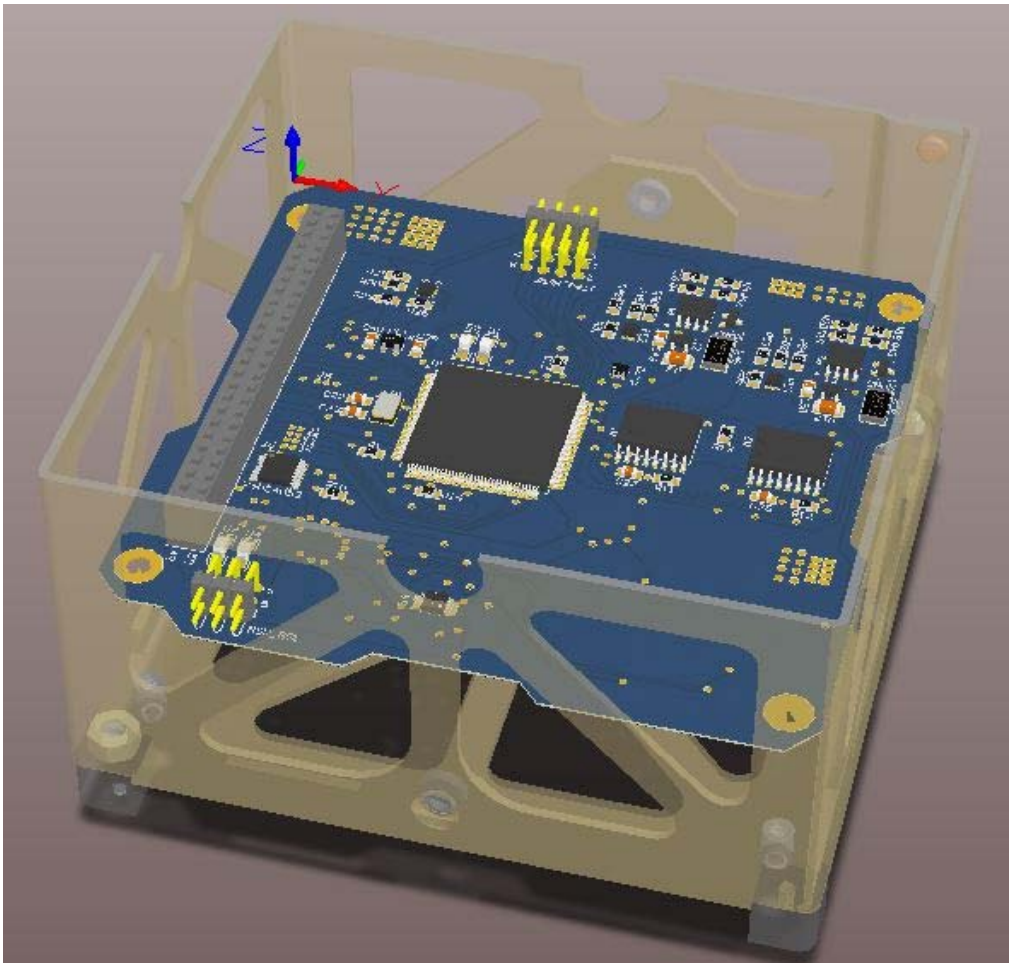


Figura 3.25: Computadora de Vuelo dentro de la estructura en 3D

3.3 Conclusión

Gran parte de los circuitos integrados seleccionados, son comerciales puesto que se encuentran disponibles para el público en general. Existen CI endurecidos contra radiación sin embargo su costo era excesivo y el propósito de este trabajo de investigación era un diseño de bajo costo. tambien ya hay documentación acerca de componentes utilizados en misiones así como registros de pruebas de radiación a componentes como es el caso del microcontrolador MSP430F1612 dando buenos resultados. Uno de los retos al que nos enfrentamos con la computadora de vuelo es lograr un correcto funcionamiento por lo que el diseño y manufactura es el primer paso, restarán pruebas de validación de radiación, vibración y vació.

Referencias

[1] Texas Instruments Hércules, Technical Reference Manual. SPNU515A-September 2013. "TMS570LS12x/11x 16/32-Bit RISC Flash Microcontroller". URL:http://www.ti.com/product/tms570ls1227. (Consulta: Feb.2014).

[2] Texas Instruments Hércules, Datasheet. SPNS192A-October 2012. "TMS570LS1227 16- and32-Bit RISC Flash Microcontroller". URL:http://www.ti.com/product/tms570ls1227. (Consulta: Feb.2014).

[3] Texas Instruments MSP430 Family, Application Report. SLAVA335C-October 2009. "Powering the MSP430 from a High Voltage Input using the TPS62122". URL:http://www.ti.com/product/msp430f1612. (Consulta: Feb.2014).

[4] Texas Instruments MSP430 Family, Datasheet. SLAS368G. "MSP430F15x, MSP430F16x, MSP430F161x Mixed Signal Microcontroller". URL:http://www.ti.com/product/msp430f1612. (Consulta: Feb.2014).

[5] Texas Instruments MSP430 Family, User's Guide. SLAU144J. "MSP430x1xx Family User's Guide". URL:http://www.ti.com/product/msp430f1612. (Consulta: Feb.2014).

[6] Toshiba. "NAND vs. NOR Flash Memory Technology Overview". URL:http://umcs.maine.edu/~cmeadow/courses/cos335/Toshiba%20NAND_vs_NOR_Flash_Memory_Technology_Overviewt.pdf. (Consulta: Feb.2014).

[7] Micron. "NOR | NAND Flash Guide". URL:http://www.micron.com/. (Consulta: Feb.2014).

[8] M-Systems OCTOBER-02. "Two Flash Technologies Compared: NOR vs. NAND". URL:focus.ti.com/pdfs/omap/diskonchipvsnor.pdf. (Consulta: Feb.2014).

[9] Maxim Semiconductor, Datasheet. "DS18B20". URL:http://www.maximintegrated.com/. (Consulta: Nov.2013).

[10] B.L. GREGORY and B.D. Shafer, "Latchup in CMOS Integrated Circuits" IEEE Trans. Nucl. Sci., 20, No 6,293 (1973).

[11] Linear Technology. Aplication Note 105 December 2005. "Current Sense Circuit Collection". URL:http://www.linear.com. (Consulta: Ene.2014).

[12] Maxim Semiconductors, Tutorial 746. "High-side Current-Sense Measurement: Circuits and Principles". URL:http://www.maximintegrated.com/. (Consulta: Ene.2014).

[13] Texas Instruments, Datasheet. "BQ24314 Front-End". URL:http://www.ti.com/product/bq24314a. (Consulta: Dic.2013).

[14] Maxim Semiconductors, Datasheet. "MAX4069-4072 Bidirectional, High-Side, Current-Sense Amplifiers with Reference". URL:http://www.maximintegrated.com/. (Consulta: Dic.2013).

[15] Boylestad Nashelsky. Electrónica: Teoría de circuitos y dispositivos Electrónicos. Decima edición. Pearson.

[16] Tutoriales Altium desde ventana Help en el software o directamente deURL:http://wiki.altium.com

- "Getting Started with Altium Designer".
- "Component, Model and Library Concepts".
- "Creating Library Components Tutorial".
- "Building an Integrated Library".

Capítulo 4

Desarrollo de Software de operación de la Computadora de Vuelo

4.1 Plataformas de desarrollo

Para el desarrollo de software de operación fue necesario el uso de ambientes de desarrollo los cuales se pueden obtener directamente de la página del fabricante de los microcontroladores (Texas Instruments). Este software fue diseñado en lenguaje de programación C y también fue de gran utilidad un software gráfico que facilitó la programación. A continuación se describen las Plataformas de Desarrollo utilizadas y sus características.

4.1.1 Code Composer Studio(CCS)

CCS es un ambiente de desarrollo integrado(IDE) de TI [1] 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 de TI, 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 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 integrados avanzados de TI que resulta en un entorno de desarrollo rico en características de peso para desarrolladores de sistemas integrados, figura 4.1.

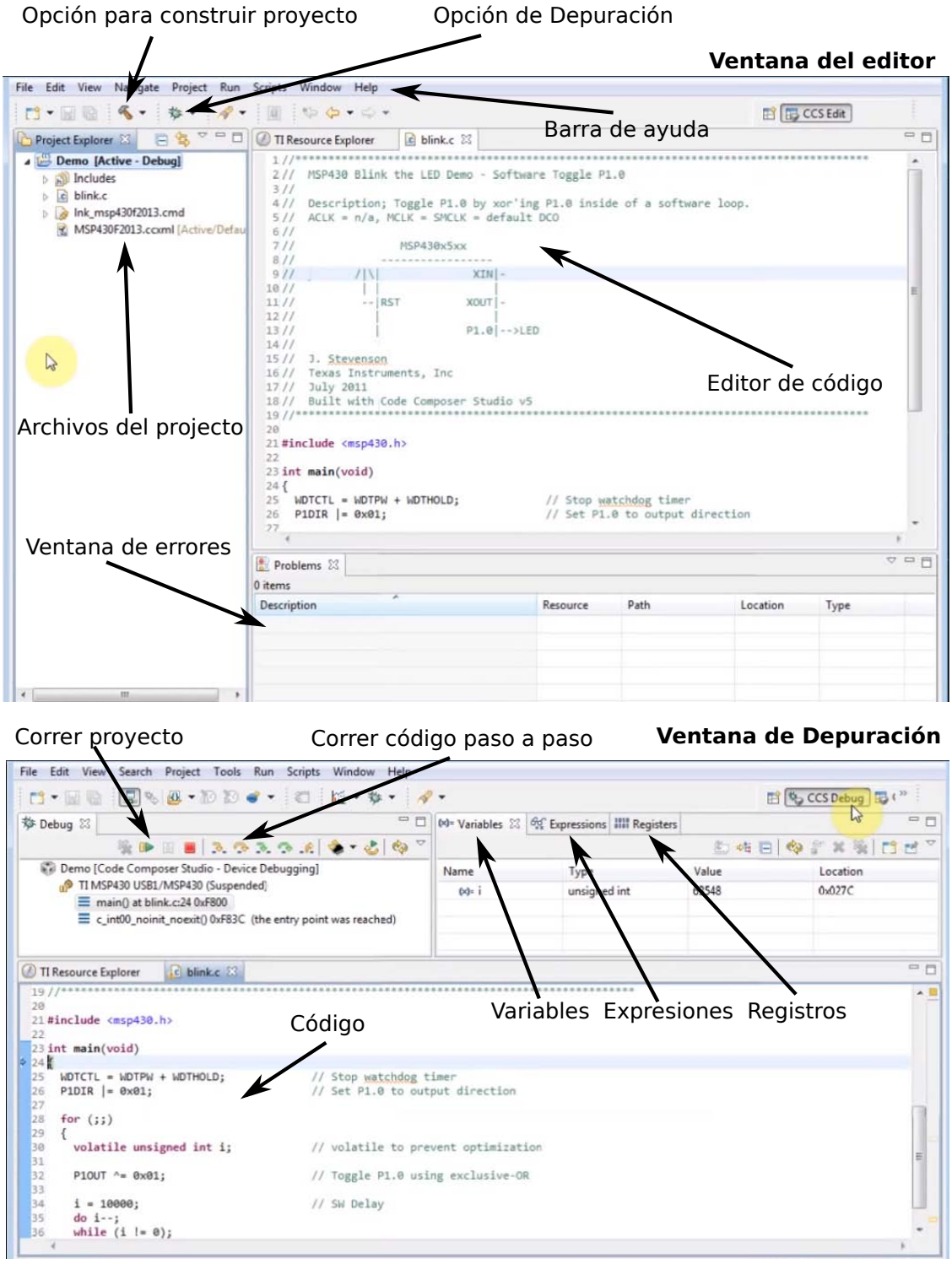


Figura 4.1: Pantalla de código y depurador de Code Composer Studio

4.1.2 HalCoGen

HalCoGen (por sus siglas en ingles Hardware Abstraction Layer Code Generation) contiene módulos de software con acceso directo al MCU y es responsable de la inicialización del sistema,[2]. HalCoGen permite a los usuarios generar "drivers" del dispositivo para microcontroladores de seguridad (Hércules). 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 microcontrolador. Una vez que el dispositivo es configurado, el usuario puede generar código de inicialización y controlador de periféricos, los cuales pueden ser importados dentro de CCS, espacio de trabajo IAR o Keil uVision.

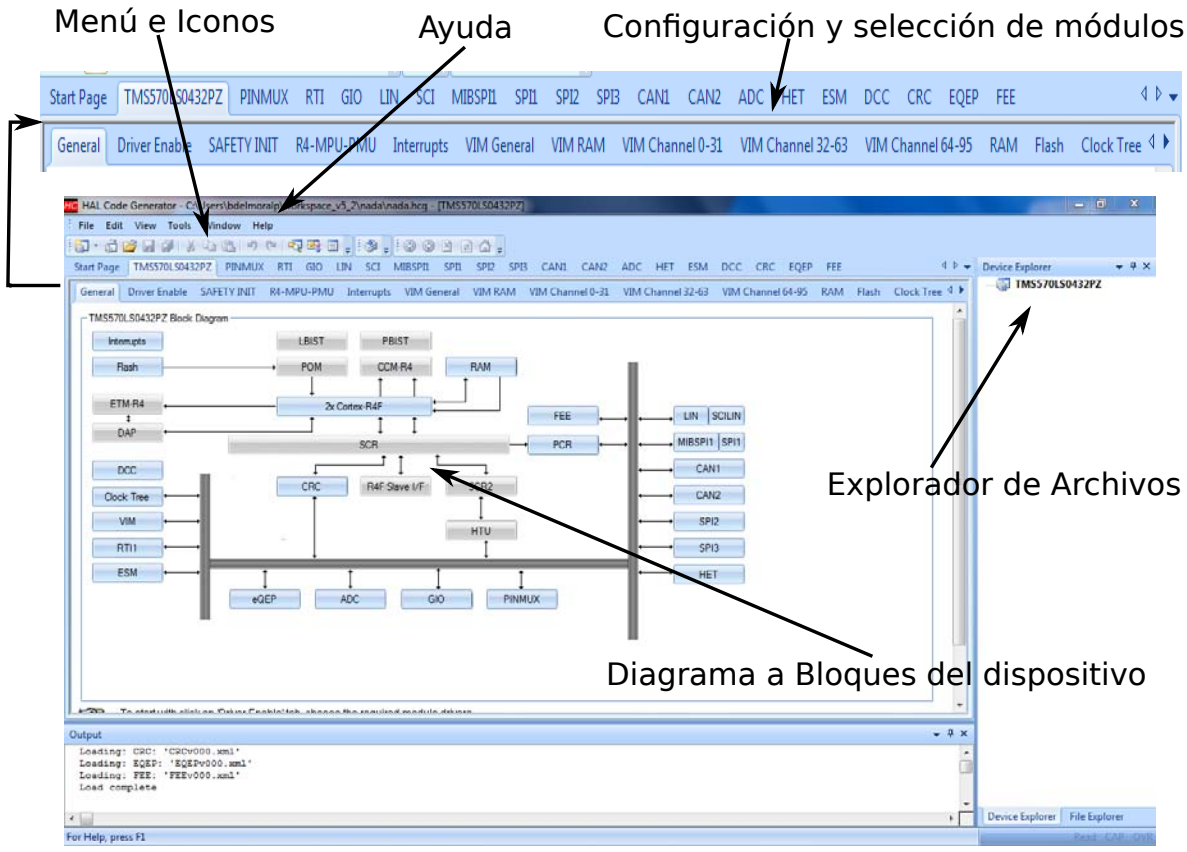


Figura 4.2: Pantalla de la Plataforma HalCoGen

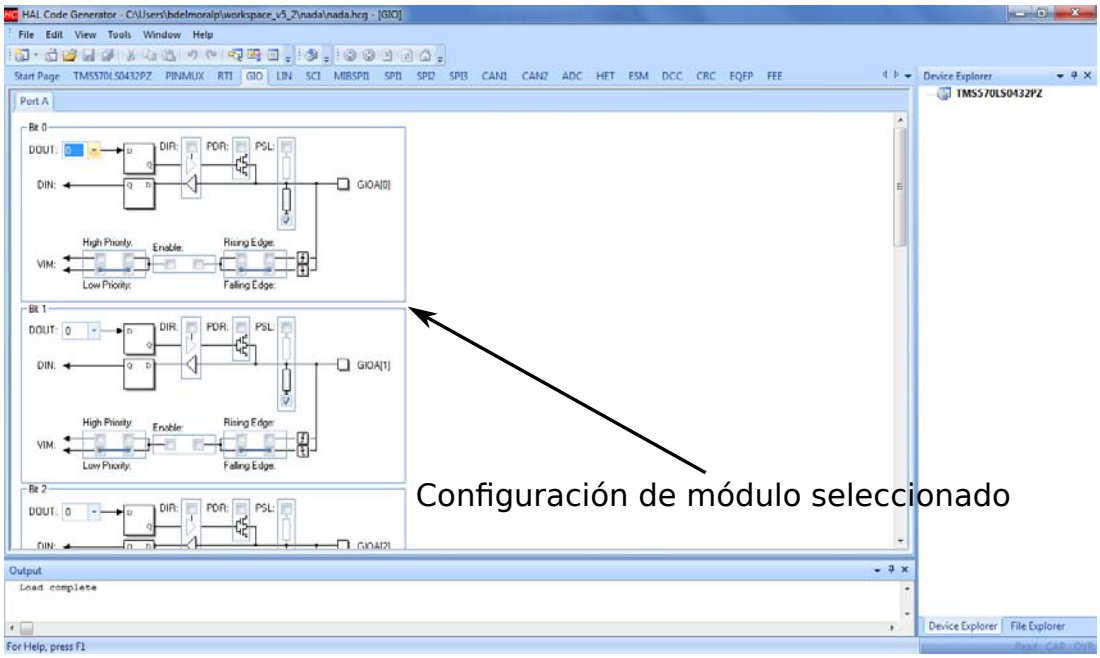


Figura 4.3: Pantalla de la Plataforma HalCoGen

4.2 Cargado de programas en TMS570LS1227 y MSP430F1612

Un requerimiento importante para muchos sistemas basados en memoria Flash es la habilidad de actualizar su firmware una vez que está instalada una primera versión en el producto final. Esta habilidad es referida como una programación dentro de la aplicación. Por ejemplo, el UART Bootloader para el MCU Hércules provee un medio para escritura, lectura y borrado de sección definidas en el programa de la Memoria Flash que generalmente posee el código de aplicación del usuario.

4.2.1 Arquitectura de programación en el MCU TMS570LS1227

4.2.1.1 JTAG

La serie de microcontroladores TMS570LS, por ejemplo el TMS570 USB stick, tienen abordo un emulador XDS100v2 el cual puede ser usado como conectividad JTAG para debug y programación de Flash, [3]. Este puede ser usado para propósitos de evaluación. Por otra parte, existen emuladores que también pueden ser soportados y nos dan una conectividad JTAG más veloz comparada con el XDS100v2.

En este caso se hablará del emulador XDS100v2 el cual viene incluido dentro del Launchpad Hércules TMS570LS04.

Características del XDS100

- El emulador XDS100 es un diseño de hardware JTAG con interfaz USB de muy bajo costo de Texas Instruments.
- El emulador XDS100 provee acceso JTAG a dispositivos de Texas Instrument basados en JTAG.
- Es compatible con el ambiente de desarrollo Code Composer Studio.
- Existen 3 versiones de XDS100. La versión XDS100v1, XDS100v2 y XDS100v3.

Características del emulador XDS100v2.

- Soporta alta velocidad USB (480 Mbits/s)
- Soporta los siguientes núcleos de procesamiento: TMS320C28x, TMS320C54x, TMS320C55x, TMS320C64x+, TMS320C674x, TMS320C66x, ARM 9, ARM Cortex R4, ARM Cortex A8, ARM Cortex A9 and Cortex M3 (Requiere CCSv4.2.2 o mayor).
- Detección de cable desconectado
- Detección de perdida de energía de la tarjeta

Procedimiento de uso del emulador XDS100v2

1.-Instalar CCSv5.1x

2.-Conectar el hardware XDS100.
Conectar el cable USB de la PC al hardware XDS100. Conectar la tarjeta JTAG de acuerdo a la configuración de pines. Aparecen pequeñas ventanas emergentes para informar al usuario de que el hardware USB se reconoce y que se ha instalado correctamente.

3.- Se inicia CCS y se hace una nueva configuración de tarjeta.
Seleccionar XDS100 como tipo de conexión.
Seleccionar el dispositivo correcto.

4.2.1.2 UART Bootloader Hércules

Esta sección describe como trabajar y personalizar la aplicación UART Bootloader. El Bootloader es proporcionado como código fuente que permite la personalización completa

del gestor de arranque. El Bootloader se ha construido y validado usando CCSv5 en el HDK TMS570LS12X y puede encontrarse directamente en la página de Texas Instruments.

El código Bootloader esta implementado en C en cuanto que el código ensamblador ARM Cortex RF4 se usa únicamente cuando es necesario.
El Bootloader es compilado en modo ARM 32-Bit y está basado en el protocolo Ymodem el cual envía datos en bloques de 1024 bytes. Cuenta con detección de errores y es aplicado a la transmisión y recepción de datos.

Operación del UART Bootloader

El Bootloader UART es construido con CCSv5 y cargado a través del puerto JTAG en la parte baja de la memoria de programa (0x0000).

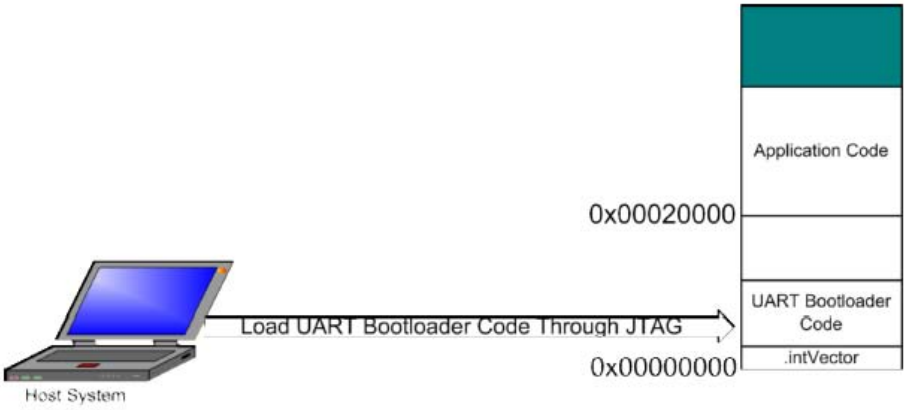


Figura 4.4: Forma de cargado de UART Bootloader

Después de un Reset al dispositivo, el código de inicio copia la Flash API del Bootloader a la SRAM y ejecuta el Bootloader en Flash.
Primero se debe verificar si el pin GPIO_A7 está presionado. Si el GPIO es puesto en bajo, el código de aplicación es actualizado. El GPIO_A7 puede ser habilitado en ENABLE_UPDATE_CHECK en el archivo cabecera bl_config.h. Después se verifica la bandera en 0x0007FF0. Si la bandera es un número válido (0x5A5A5A5A) el Bootloader se dirigirá al código de aplicación ubicado en 0x00020000. Si la bandera no es un número válido, este configurará el UART, después actualizará el código de aplicación del usuario. Después de que todo el código de aplicación se haya programado correctamente, tambien se actualiza la bandera.

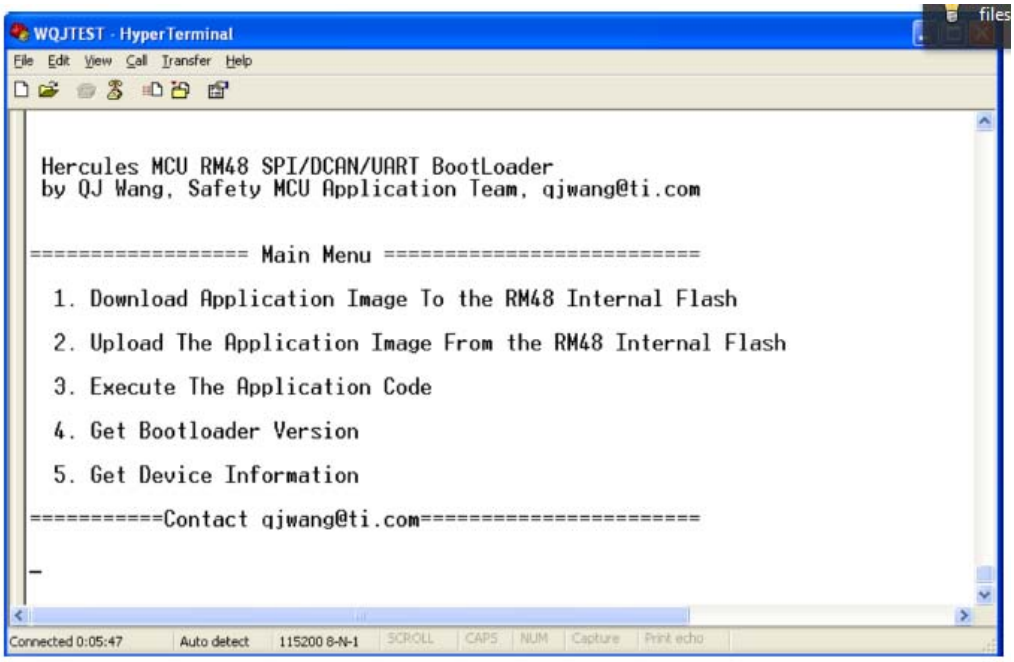


Figura 4.5: Ejemplo del uso del UART Bootloader con PC

En este caso el reto no es comunicarse con el MCU a través de la PC sino lograr la reprogramación de este utilizando el MSP430F1612 montado dentro del subsistema de CV mediante el Bootloader vía UART, figura 4.5.

4.2.2 Arquitectura de programación en el MCU MSP430F1612

A continuación se detalla la forma de implementar una correcta programación del MCU MSP430F1612 desde su conexión en hardware hasta los pasos a seguir en software.

4.2.2.1 LaunchPad basado en Interfaz BSL UART MSP430

El kit de desarrollo MSP-EXP430G2, es un kit experimental para la línea de dispositivos MSP430Gxxx,[4, 5, 6]. Este cuenta con un emulador abordo con interfaz USB la cual también puede ser utilizada como interfaz UART (COM PORT) con un baudaje de 9600.

Implementación

Esta aplicación estaba diseñada para un dispositivo MSP430G2231 aunque para este caso se contaba con un MSP430G2553 dentro del kit de desarrollo EXP430G. Debido a la diferencia de paridad entre el MSP430-EXP430G2 Launchpad UART (9600 baudios,

sin paridad) y la especificación de la BSL MSP430 UART (9600 baudios, paridad par), el MSP430G2553 comprueba el byte entrante desde el PC y añade la paridad par, si es necesario, para el dispositivo de destino MSP430.

Conexión de Hardware

La implementación del LaunchPad basado en Interfaz BSL UART MSP430 requiere todos los pines del puerto 1(P1) del MSP430G2553, figura 4.6. Es necesario utilizar el push button S2 del Launchpad para generar la secuencia de entrada BSL del dispositivo MSP430. La siguiente tabla lista la asignación de GPIOs del MSP430G2553.

GIO PIN	DESCRIPCION
P1.0	Pin de conexion TCK a la tarjeta MSP430 con pines JTAG dedicados
P1.1	Pin de transmision UART a PC (conectado al pin UART RX del PC)
P1.2	Pin de receptor UART de PC (conectado al pin UART TX del PC)
P1.3	Entrada de push button.
P1.4	Pin de reset a tarjeta MSP430.
P1.5	Pin de prueba a tarjeta MSP430.
P1.6	Pin de receptor UART de la tarjeta MSP430 (conectado a pin BSL TX de la tarjeta MSP430)
P1.7	Pin de transmisor UART a la tarjeta MSP430 (conectado a pin BSL RX de la tarjeta MSP430)

Tabla 4.1: Asignación de GPIOs del MSP430G2553 para implementación de BLS UART

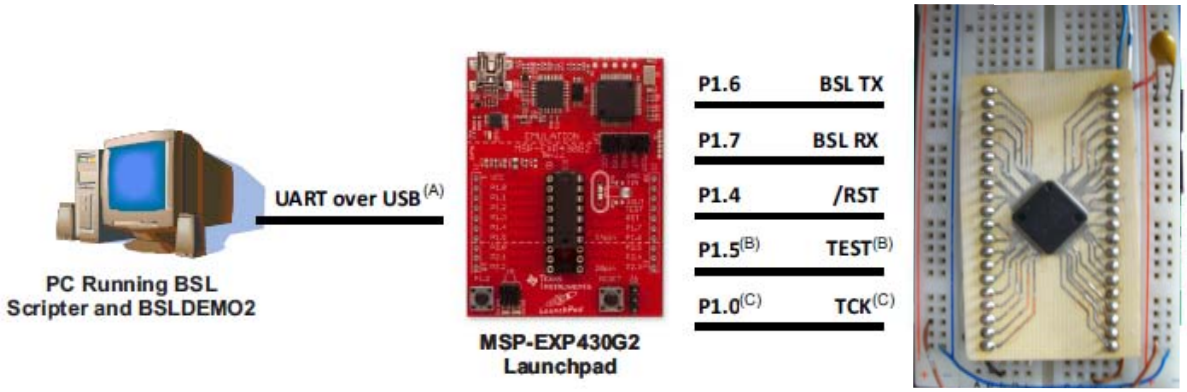


Figura 4.6: Conexión via BSL UART

Usando LaunchPad MSP430 basado en Interfaz BSL

El procedimiento de uso se basa en los siguientes pasos:

1.- Compilar el código de LaunchPad MSP430 basado en Interfaz BSL el cual es obtenido

en archivo .zip en el reporte de la aplicación. Este puede utilizarse usando la versión de prueba del software de desarrollo CCS. Importar el proyecto sin dar click en la opción "Copy projects into workspace". El código fuente está ligado dentro del proyecto Code Composer Studio y no debe intentar copiarse el proyecto CCS dentro de otro directorio de espacio de trabajo. Esto podría generar problemas al compilar.

2.-Actualizar el firmware compilado dentro del MSP430G2553. El archivo asociado .zip dado en este reporte de aplicación también contiene archivo binario Intel-Hex o TI-TXT, los cuales pueden ser usados con el MSP430 Flasher (`http://processors.wiki.ti.com/index.php/MSP430_Flasher_-_Command_Line_Programmer`).

3.- Desconectar el USB del Launchpad y conectar el Launchpad con el dispositivo MSP430 como se describió previamente. Asegurarse que los jumpers 3 y 5 en el Launchpad están conectados. Si el dispositivo MSP430 se energiza independientemente es importante verificar que la señal del nivel liberado por el MSP430G2553 en el Launchpad no debe exceder el "Voltaje aplicado a cualquier pin" especificado en el datasheet del dispositivo MSP430.

4.- Reconectar el USB para reenergizar el Launchpad. Después de un segundo, el MSP430 en el Launchpad automáticamente generará la secuencia de entrada BSL. Si la invocación de la tarjeta BSL es exitosa, ambos LEDs rojo y verde en el Lauchpad deberán prender. El LED verde indica que el pin UART TX del MSP430G2553 está en alto mientras que el rojo indica indica el pin TCK en alto. Si la invocación falla los dos LED prenderán alternándose continuamente. El MSP430G2553 en el Lauchpad solo puede generar la entrada de secuencia BSL al inicio después del reset. Por lo tanto, con el fin de regenerar la secuencia de entrada de BSL, es necesario un reset a el MSP430G2552 del Launchpad pulsando el botón de reinicio S1.

5.- Correr el BSL scripter para la comunicación con el BSL. Este deberá ejecutarse con el puerto serial correcto (COM PORT) el cual es asignado a el Launcpad por la PC. El COM PORT es llamado "MSP430 Application UART".

4.3 Software de operación de Computadora de Vuelo

Una parte importante dentro del desarrollo de la Computadora de Vuelo fue definir la secuencia de como el subsistema realizará las tareas, la comunicación con los demás subsistemas y los protocolos de comunicación para la obtención de datos dentro del mismo subsistema. En esta sección se abordará la manera en como deben realizarse las tareas así como los protocolos utilizados.

4.3.1 Esquema general de ejecución de comandos del subsistema Computadora de Vuelo

El diseño de este esquema fue pensado para lograr gran confianza en la ejecución de los comandos por tal motivo cuenta con "checksums" en gran parte del diseño así como cierto tiempo de espera en la ejecución de comandos para descartar bucles infinitos. Los demás subsistemas estarán diseñados de la misma manera que la CV y de esta manera podemos mejorar la confiabilidad en la operación del Satélite Experimental.

El diagrama de la figura 4.7 muestra una aproximación de cómo serían los pasos a seguir de la Computadora de Vuelo. Está basado manteniendo una conexión con E.T. (Estación Terrestre), es decir, E.T. se comunica con el Satélite por medio del subsistema de Comunicaciones y la Computadora de Vuelo se mantiene en espera de un comando de tarea. En ese momento E.T. determina reprogramación, descarga de telemetría o cálculo y descarga de telemetría en ese preciso momento.

Inicialmente se pensó hacer arbitraje en el bus I2C entre el Microcontrolador principal de la CV y el microcontrolador de Comunicaciones fungiendo los dos como maestros en el bus. De esta manera Comunicaciones avisaría a CV la llegada de una conexión con E.T. y por ende la ejecución de una tarea. Pensando en un evento donde el principal objetivo fuera el comando de reprogramación de CV y que en ese preciso momento el apoderado del bus I2C fuera la misma CV, es decir, CV realizaría una tarea pendiente y dejaría en segundo plano la reprogramación, se pensó en incluir una línea de interrupción generada por Comunicaciones para avisar a CV la llegada de un comando por parte de E.T., de esta manera la CV se mantiene en todo momento como maestro del bus y así eliminamos la probabilidad de dejar la tarea de reprogramación en segundo plano.

La reprogramación nos permite la actualización de software para un mejor desempeño del Satélite. Para esto se tiene pensado construir dos satélites, uno es el que estará en órbita y otro exactamente igual estará en Tierra. El Satélite en órbita nos arrojara datos de problemas presentes en el espacio y con estos datos se puede mejorar y validar mediante pruebas con nuestro Satélite en Tierra esos errores y una vez mejorado el diseño se procede a reprogramar nuestro Satélite en órbita.

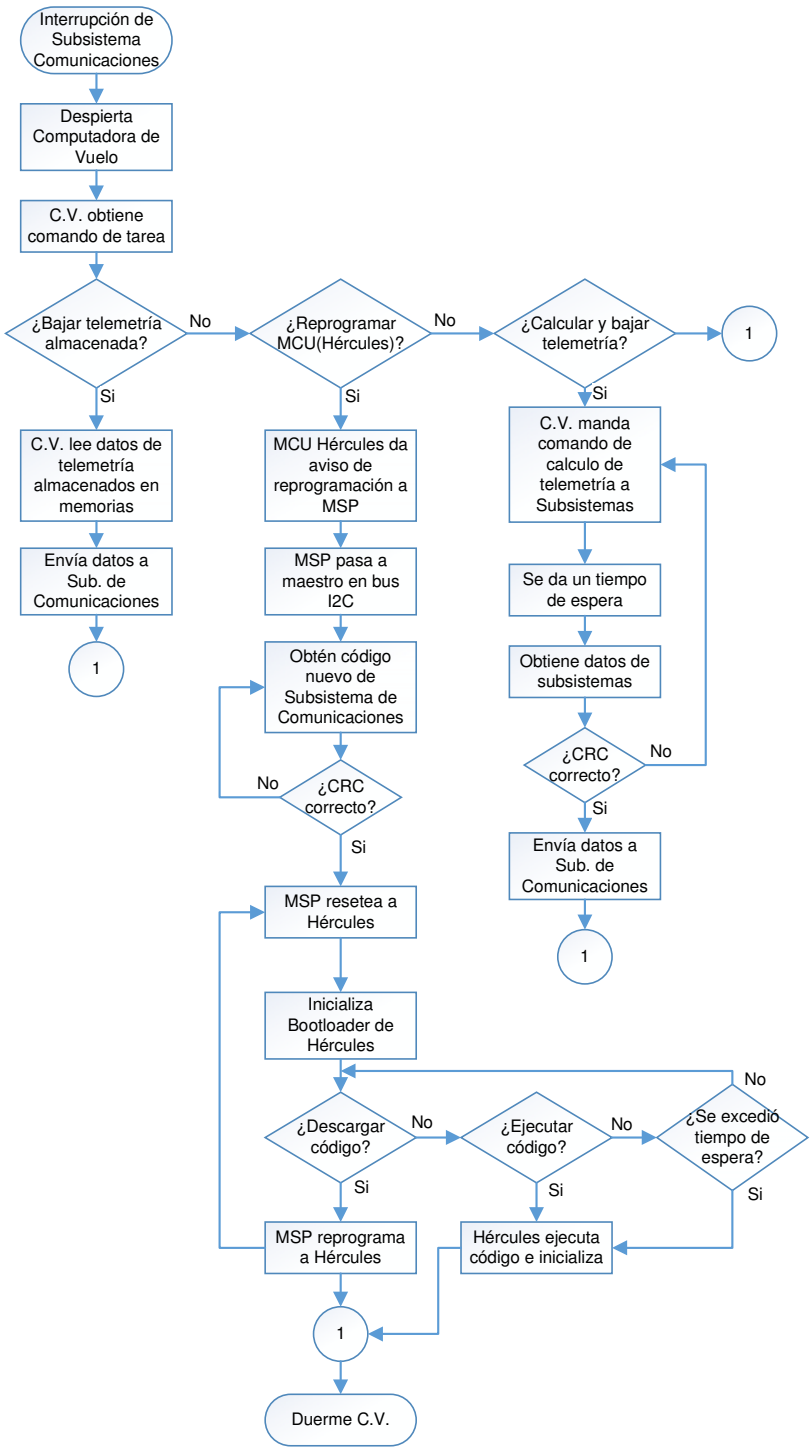


Figura 4.7: Diagrama de flujo de tareas de C.V.

El software y hardware de CV está pensado para tener buena confiabilidad por tal motivo en el diagrama de la figura 4.8 se observa como el MSP430 se encuentra monitoreando en todo momento al MCU Hércules y al momento de ocurrir errores, el MSP430 puede darle un reset al Hércules o reprogramarlo si es necesario mediante un programa base precargado desde Tierra en la memoria Flash NOR que maneja el MSP430.

La memoria Flash conectada al MSP430 estará fraccionada en dos segmentos, una parte para el software base y la otra para un programa nuevo. En muchas ocasiones la línea de vista entre el Satélite y Tierra no nos otorgará mas que unos minutos (aproximadamente 15 min en el mejor de los casos) impidiéndonos la subida de nuevo programa al Satélite. Por tal motivo se pretende enviar y almacenar tramas de datos y ejecutar el código de actualización una vez obtenido el software en su totalidad. Este software de reprogramación estará contenido en la segunda parte de la memoria Flash.

Las memorias Flash son monitoreadas por el CI front-end (BQ24314) para el evento "latchup". Cada MCU monitorea a su respectiva memoria conectada. Las señales de error se conectaron como interrupciones al MCU de tal modo que mediante una operación con las memorias, si se presenta el evento latchup, los microcontroladores saltan a la rutina de interrupcion la cual consiste en desenergizar a las memorias, tomar un tiempo de espera y reintentar el cargado, borrado o lectura de datos. Si el problema persiste "n" número de veces se aborta la tarea y se continua con otra.

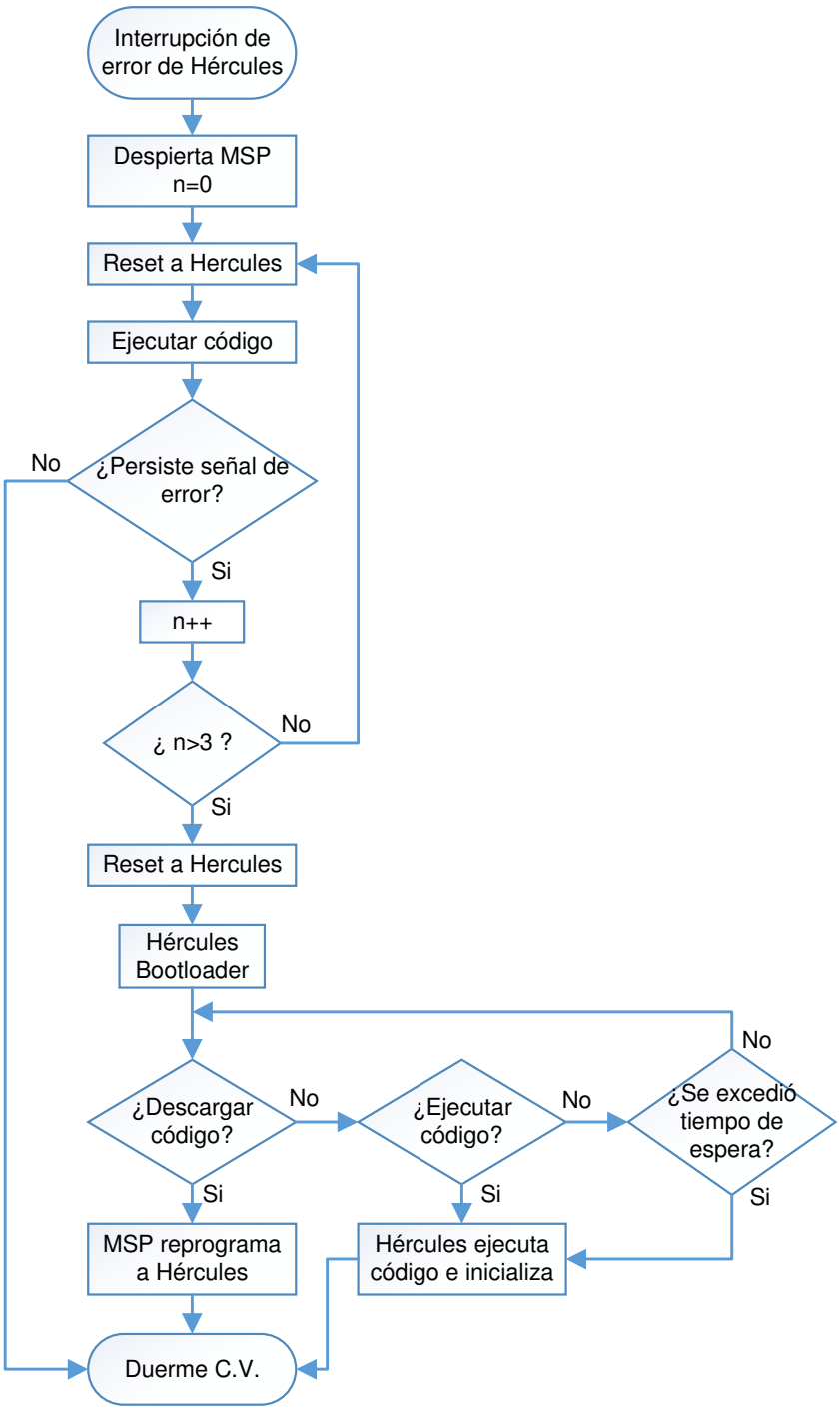


Figura 4.8: Diagrama de flujo de operación durante error de C.V.

4.3.2 Comandos del subsistema Computadora de Vuelo

La Computadora de Vuelo cuenta con 4 canales de comunicación que son los siguientes: UART el cual es utilizado para la reprogramación del microcontrolador Hércules por medio del MSP430, I2C con el cual el Hércules logra la comunicación con los demás subsistemas así como la obtención de datos del magnetómetro y para el caso del MSP430 obtención de nuevo programa que actualizará al Hércules, SPI que es el protocolo empleado por las memorias Flash NOR y 1-Wire para la obtención de temperaturas dadas por los DS18B20 distribuidos por todo el Satélite Experimental. Los protocolos de comunicación interactúan en el Subsistema como lo muestra la figura 4.9.

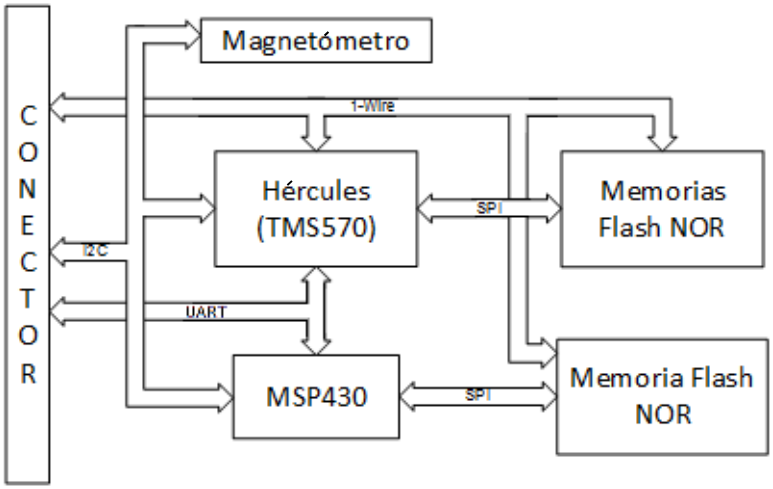


Figura 4.9: Protocolos de comunicación empleados en la Computadora de Vuelo

4.3.2.1 Protocolo de comunicación 1-Wire

1-Wire es un protocolo de comunicaciones en serie diseñado por Dallas Semiconductor (Ahora Maxim Semiconductor) el cual está basado en un bus maestro y "n" número de esclavos, [7, 8, 9, 10].A continuación se detalla el protocolo de comunicación.

Operaciones básicas

Hasta el momento no hay ningún microcontrolador con un canal de comunicación 1-Wire en el mercado sin embargo un microcontrolador es capaz de generar fácilmente señales de tiempo necesarias para este tipo de comunicación. Una vez descritos con anterioridad los requerimientos en Hardware para el uso de sensores 1-Wire, se debe aclarar que el sistema debe ser capaz de generar pulsos de $1\mu s$ de forma repetida y precisa

como velocidad estándar o pulsos de 0.25 microsegundos para trabajar este protocolo a gran velocidad y saber que las operaciones de comunicación no deben ser interrumpidos cuando se generan. Las cuatro operaciones básicas para el manejo del bus 1-Wire son: reset, escribe bit 1, escribe bit 0 y lee bit. El tiempo que se tarda en realizar un bit de comunicación es llamado "slot bit" en las hojas de datos del dispositivos. A su vez la función Byte es una construcción basada en la operación de múltiples bits. La figura 4.10 muestra las formas para cada operación asi como los tiempos recomendados para lograr la comunicación 1-Wire.

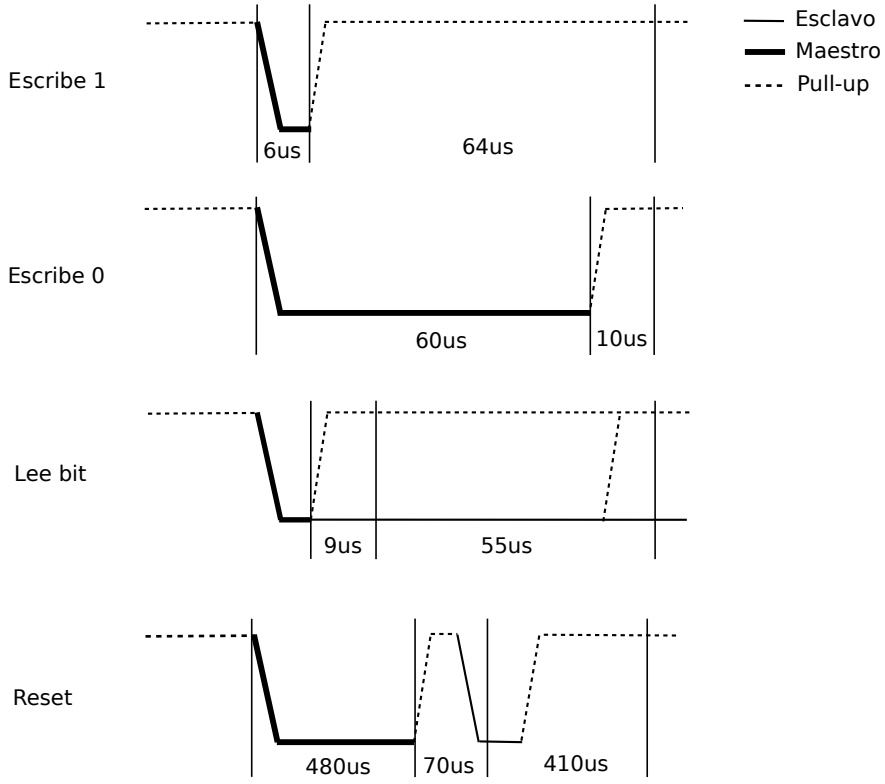


Figura 4.10: Operaciones básicas de protocolo 1-Wire

Con las cuatro operaciones básicas es necesario realizar algunas funciones con el sensor de temperatura DS18B20.

ROM Code (Código en ROM)

Cada DS18B20 contiene un código único de 64 bits guardado en ROM. Los 8 bits menos significativos del "ROM code" son el código de familia que para este caso es: 28h (este valor

nos permite obtener lecturas por el mismo bus 1-Wire pero de otra familia de dispositivos como memorias EPROM y EEPROM de Maxim Semiconductors). Los siguientes 48 bits contienen un número serial único. Los 8 bits más significativos contienen el Checado de Redundancia Cíclica (CRC) el cual se calcula a partir de los primeros 56 bits del ROM code, figura 4.11.

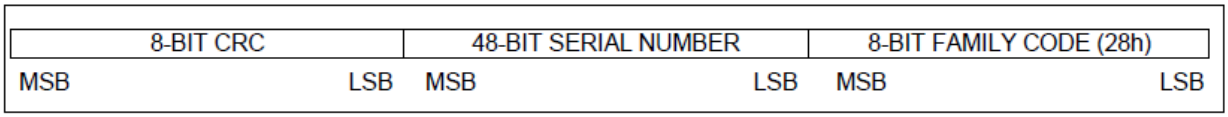


Figura 4.11: Registro de ROM code de sensores 1-Wire

Memoria Interna

La memoria del dispositivo DS18B20 consiste de un "Scratchpad" SRAM con almacenamiento no volátil EEPROM para señales de alarma así como configuración de registros. El byte 0 y 1 del "scratchpad" contiene la lectura de temperatura del bit menos significativo al más significativo respectivamente. Byte 2 y 3 proveen acceso a los registros TH y TL. Byte 4 contiene datos de registro de configuración. Bytes 5, 6 y 7 son reservados para uso interno del dispositivo. El "scratchpad" también contiene CRC y es en su byte 8 (solo lectura) el cual se obtiene a partir del byte 0 al 7 del "scratchpad", figura 4.12.

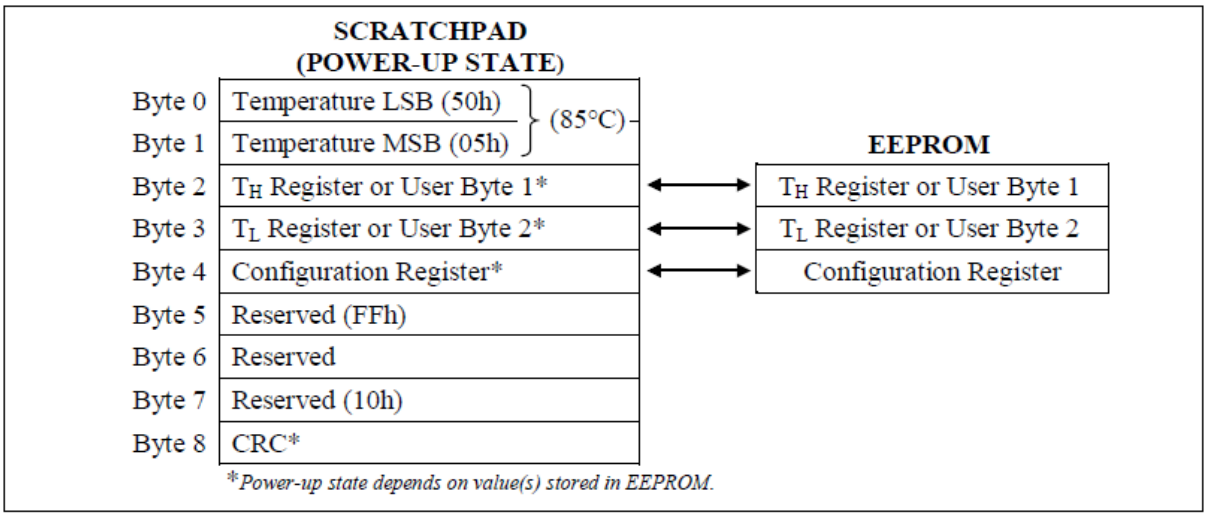


Figura 4.12: Scratchpad de sensores 1-Wire

CRC

El CRC es dado como parte del ROM code y el byte 8 en la memoria "scratchpad" del DS18B20. El CRC se calcula a partir de los bytes anteriores almacenados, este cambia según la información que contenga la memoria scratchpad. El CRC provee al bus maestro un método de validación para los datos obtenidos del sensor. Para verificar si los datos leídos son correctos, el bus maestro recalcula el CRC a partir de los datos recibidos y lo compara con el CRC leído. Los valores comparados del CRC y la decisión para continuar con alguna operación son determinadas completamente por el bus maestro. El DS18B20 no cuenta con algún circuito interno que genere una secuencia para proceder si el CRC del DS18B20 no coincide con el valor generado por el bus maestro.

La función polinomial equivalente es: CRC=X8+X5+X4+1

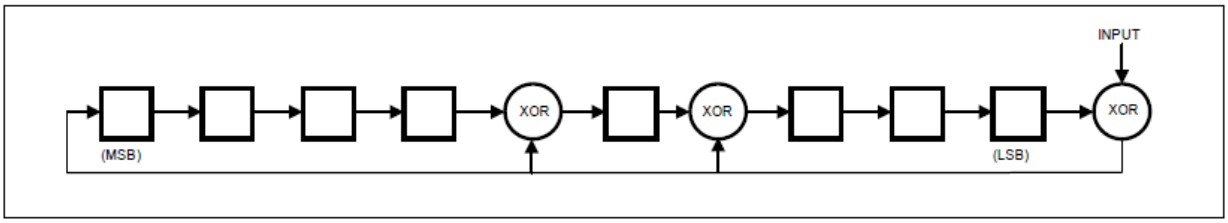


Figura 4.13: CRC de sensores 1-Wire

La ubicación de los caminos de realimentación representados por compuertas XOR, o la presencia de coeficientes en la expresión polinomial, determinan propiedades del CRC y la habilidad del algoritmo para localizar ciertos tipos de errores en el dato, figura 4.13.

Lectura de Temperaturas

La salida de temperatura del sensor se obtiene en grados Celsius. El valor es almacenado en un registro de 16 bits signado con complemento a dos. El bit de signo indica temperaturas positivas o negativas: para números positivos S=0 y para números negativos S=1, figura 4.14.

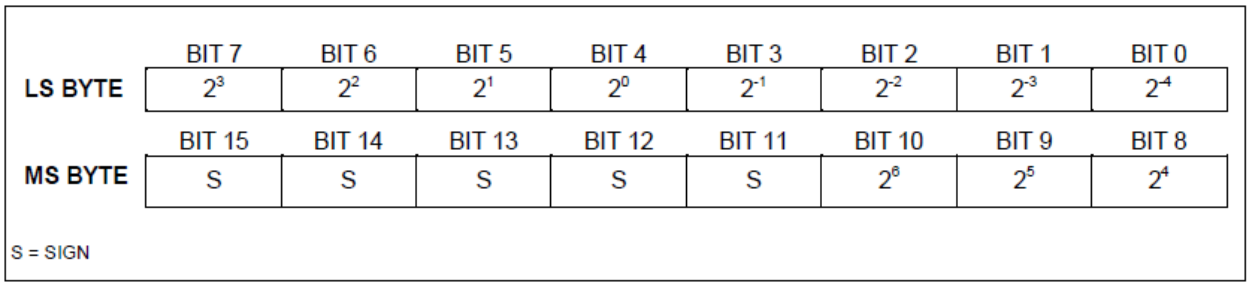


Figura 4.14: Registro de temperatura de sensores 1-Wire

Comandos del DS18B20

Existen comandos para interactuar con los sensores de temperatura. En este caso solo se habla de los más utilizados y se dará una breve descripción.

COMANDO	DESCRIPCION
SEARCH ROM [F0h]	Identifica los ROM codes de todos los dispositivos en el Bus.
READ ROM [33h]	Comando utilizado cuando hay únicamente un dispositivo en el Bus.
MATCH ROM [55h]	Comando que permite al maestro direccionar un dispositivo.
SKIP ROM [CCh]	Comando utilizado para direccionar a todos los dispositivos.
CONVERT T [44h]	Comando para conversión de temperatura.
READ SCRATCHPAD [BEh]	Este comando permite al maestro obtener los datos del scratchpad.

Tabla 4.2: Comandos utilizados en dispositivos con comunicación 1-Wire

4.3.2.2 Protocolo de comunicación UART

El USART, llamado SCI(Serial Comunications Interface), puede funcionar como un sistema de comunicaciones full duplex o bidireccional asíncrono, adaptándose a multitud de periféricos y dispositivos que transfieren información de esta manera, tales como monitor CRT o una PC. También puede trabajar en modo síncrono unidireccional o "half duplex" para soportar periféricos como memorias, conversores, etc. En resumen el USART puede trabajar de tres maneras:

- Asíncrona(Full duplex, bidireccional)
- Síncrona-Maestro(Half duplex, unidireccional)
- Síncrona-Esclavo(Half duplex, unidireccional)

En esta forma de comunicación serie, se usa la norma RS-232 elaborada por EIA(Electronics Industry Association), donde cada palabra de información o dato se envía independientemente de los demás. Suele constar de 8 o 9 bits y van precedidos por un bit de inicio y detrás de ellos un bit de fin.

En el protocolo asíncrono RS-232 la frecuencia en bauds (bits por segundo) a la que realiza la transferencia se debe efectuar a un valor normalizado: 330, 600, 1200, 2400, 4800, 9600, 19200, 38400, etc.

4.3.2.3 Protocolo de comunicación SPI

El bus SPI es un estándar de comunicaciones desarrollado por Motorola[11], utilizado para la transferencia de información entre circuitos integrados. Aplicaciones típicas de SPI son I/Os, memorias y convertidores analógico-digital. Cada circuito conectado al bus puede actuar como transmisor y receptor al mismo tiempo, por lo que este tipo de comunicaciones serial es "fullduplex". Los dispositivos conectados al bus son definidos como maestro y esclavos. Un maestro es aquel que inicia la transferencia de información sobre el bus generando las señales de reloj y control.
Este tipo de bus emplea 3 líneas principales y una línea extra por cada dispositivo conectado como esclavo al bus, figura 4.15.

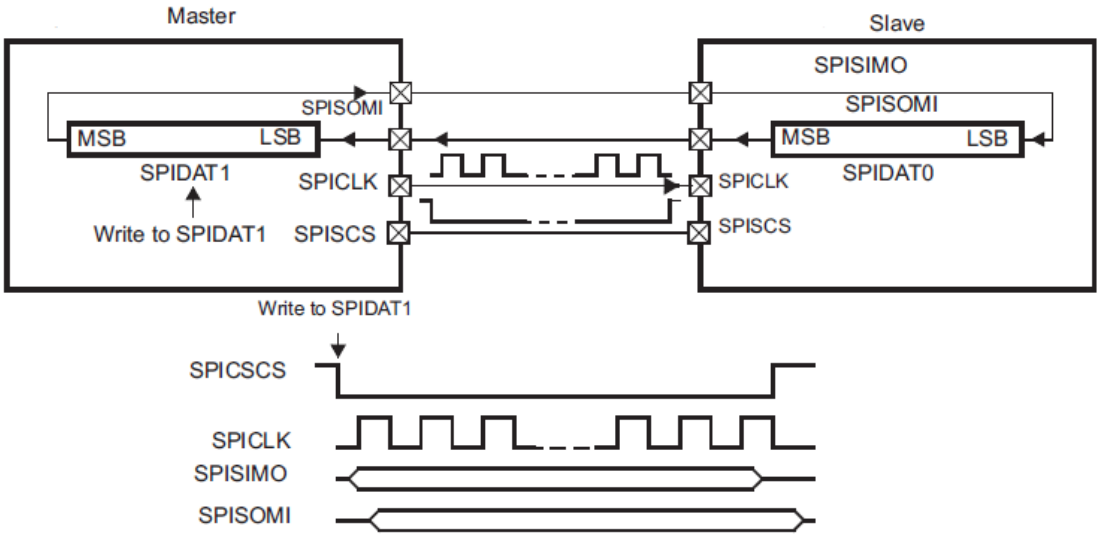


Figura 4.15: Líneas ocupadas en bus SPI

SCLK(Serial Clock): Línea de reloj generada por el maestro que sincroniza la transferencia de datos.
SIMO(Slave Input Master Output): Línea dedicada a enviar datos de Maestro a Esclavo.
SOMI(Slave Output Master Input): Línea dedicada a enviar datos de Esclavo a Maestro.
CS (Chip Select): Los esclavos son seleccionados por el dispositivo maestro a través de esta línea. Solo puede estar habilitado un dispositivo a la vez.

Las memorias Flash NOR utilizan el protocolo de comunicación SPI. Estas memorias cuentan con 4 líneas de datos (antes mencionadas) además de una línea de Reset, una de protección de escritura y una última línea que nos permite detener el proceso o continuar con el dependiendo del nivel lógico que tenga.

El conector de costilla cuenta con otro puerto SPI del Microcontrolador principal (Hércules) dedicado a cualquier carga útil dentro del Satélite Experimental para obtener una mejor versatilidad.

4.3.2.4 Protocolo de comunicación I2C

El protocolo de comunicación serie I2C fue desarrollado por Philips para cubrir sus propias necesidades en la implementación de diversos productos electrónicos que requerían una elevada interconexión de circuitos impresos, [12, 13]. El protocolo I2C (Inter Integrated Ciruit) es un bus en serie multimaestro el cual utiliza únicamente dos líneas para la transferencia de información entre los elementos acoplados al bus, ambas con resistencias de "pull-up". Una línea se dedica a datos (es bidireccional) y es llamada SDA; la otra es la línea de reloj que sirve para la sincronización (unidireccional) llamada SCL, figura 4.16. Los datos en la línea SDA únicamente son válidos cuando la línea SCL mantiene una frecuencia controlada por el maestro.

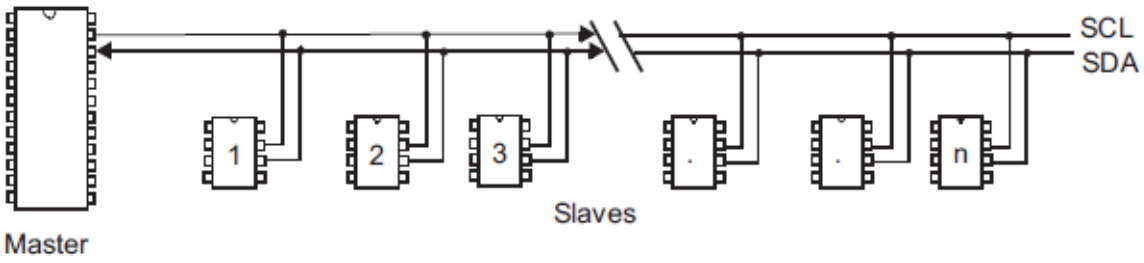


Figura 4.16: Configuración I2C

Es común el uso de velocidades como 10 kbit/s (modo de baja velocidad) o 100 kbit/s (modo estándar), pero revisiones recientes permiten velocidades de 400 kbit/s, 1 Mbit/s y 3.4 Mbit/s. Cada dispositivo es identificado por una única dirección de 7 bits aunque recientes revisiones permiten direcciones de 10 bits con algunas diferencias en la secuencia del protocolo. La secuencia de comunicación básica siempre será iniciada y terminada por el maestro mediante condiciones de START y STOP y también deberá indicar que tipo de transmisión de dato hará mediante un bit R/W (lectura o escritura). La condición de START ocurre cuando la línea SDA está en bajo mientras SCL está en alto y la condición de STOP ocurre cuando SCL está en alto justo antes de SDA, figura 4.18.

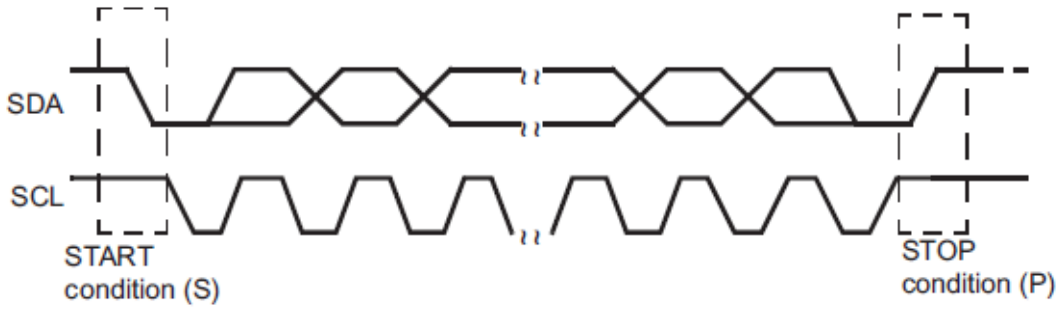


Figura 4.17: Condiciones de START y STOP

El módulo I2C opera con formato de datos en conjunto de bytes. Los datos son transmitidos con el bit más significativo al inicio y cada mensaje es seguido por un bit de reconocimiento en modo receptor. El primer byte después de la condición de START siempre consiste de 8 bits que comprende 7 bits de direccionamiento y un bit para lectura o escritura (R/W) o 8 bits de datos.

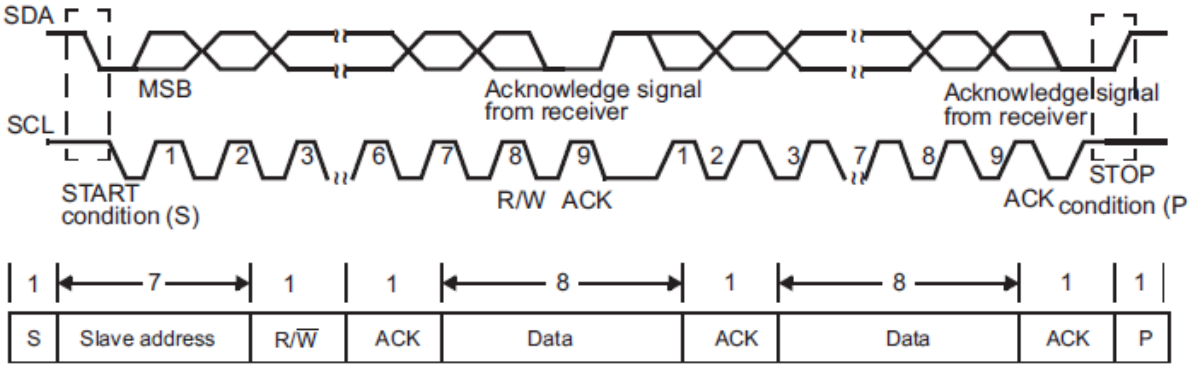


Figura 4.18: Trama de datos I2C

4.4 Conclusión

Aunque en este capítulo se describe la arquitectura de programación para cada microcontrolador así como el diferente software a utilizar, otro de los grandes retos es diseñar, validar y verificar el software de operación del subsistema. Se presentó un diagrama de flujo de como podría ser la operación del subsistema sin embargo éste esta sujeto a modificaciones dependiendo de los subsistemas finales dentro del satélite. Un diseño final en software estará definido por todo el equipo de trabajo y ademas la computadora de vuelo esta diseñada para lograr actualización de software estando en órbita.

Referencias

[1] Texas Instruments, CCS. "Code Composer Studio (CCS) Integrated Development Environment (IDE)". URL:http://www.ti.com/tool/ccstudio. (Consulta: Oct.2013).

[2] Texas Instruments, HalCoGen. "HalCoGen. URL:http://processors.wiki.ti.com/index.php/HALCoGen. (Consulta: Ene.2014).

[3] Texas Instruments Wiki. "XDS100". URL:processors.wiki.ti.com/index.php/XDS100#XDS100v2_Features. (Consulta: Feb.2014).

[4] Texas Instruments Wiki. "BSL (MSP430)". URL:processors.wiki.ti.com/. (Consulta: Feb.2014).

[5] Texas Instruments. "MSP430 Programming Via the Bootstrap Loader User's Guide (SLAU319)". URL:www.Ti.com.(Consulta: Feb.2014).

[6] Texas Instruments. "MSP430x5xx/MSP430x6xx Family User's Guide (SLAU208)". URL:www.Ti.com. (Consulta: Feb.2014).

[7] Maxim Semiconductors, Datasheet. "DS18B20". URL:http://www.maximintegrated.com/. (Consulta: Abr.2014).

[8] Maxim Semiconductors, Application Note 187 Mar 28, 2002. "1-Wire Search Algorithm". URL:http://www.maximintegrated.com/. (Consulta: Abr.2014).

[9] Maxim Semiconductors, Application Note 126 May 30, 2002. "1-Wire Communication Through Software". URL:http://www.maximintegrated.com/. (Consulta: Abr.2014).

[10] Maxim Semiconductors, Application Note 162 Mar 08, 2002. "Interfacing the DS18x20/DS1822 1-Wire Temperature Sensor in a Microcontroller Environment". URL:http://www.maximintegrated.com/. (Consulta: Abr.2014).

[11] Jose M. Angulo Usategui. "Microcontroladores Pic, Diseño Práctico de Aplicaciones" 2da Parte 16F87x, pag 155-158. Ed Mc Graw Hill.

[12] Philips Semiconductors. "The I2C Bus Specification, version 2.1.". URL:http://www.philips.com.January2008..(Consulta: Abr.2014).

[13] NXP, UM10204 Oct-2012. "I2C-bus specification and user manual". URL:http://www.nxp.com/documents/other/UM10204_v5.pdf. (Consulta: Abr.2014).

Capítulo 5

Pruebas de Validación

Actualmente la manufactura de la tarjeta electrónica esta lista y de igual forma el pedido de todos los componentes electrónicos involucrados ya esta colocado por lo que para el tiempo en que se realice la defensa de este tema de tesis se podría tener el primer prototipo del circuito de CV.

Los resultados que se esperan son respaldados en la experiencia acumulada de trabajo en este tipo de desarrollos por parte del Instituto de Ingeniería de la UNAM. Ademas se subraya que durante el desarrollo de esta tesis, fue posible realizar muchas pruebas de validación en circuitos diseñados específicamente para estas y con el uso de "protoboard".

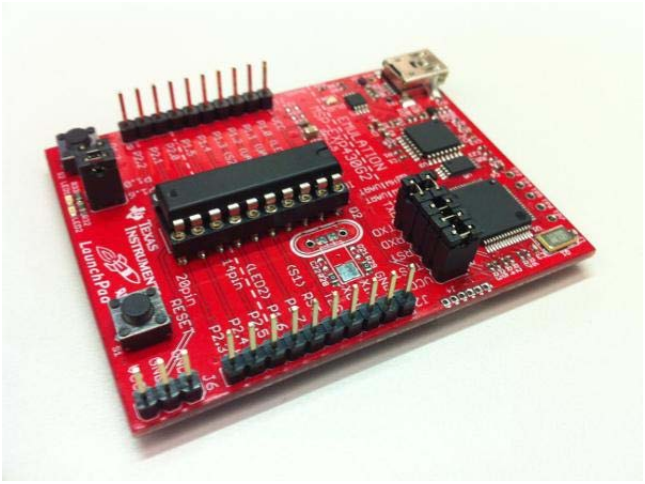


Figura 5.1: Kit de desarrollo Launchpad MSP-EXP430G2553

Gran parte de los circuitos integrados dentro de la Computadora de Vuelo son de la marca Texas Instruments así como los dos microcontroladores incluidos dentro del

subsistema. Para algunas pruebas se contó con dos tarjetas de desarrollo de bajo costo las cuales contienen microcontroladores de la misma familia que la Computadora de Vuelo logrando con esto una gran aproximación a nuestro diseño total. El primero es el kit de desarrollo Launchpad MSP-EXP430G2553, figura 5.1 el cual contiene un microcontrolador de la familia MSP430Gxxxx y el segundo es el LAUNCHXL-TMS57004, figura 5.2 el cual contiene un microcontrolador de la familia TMS570(Hércules).

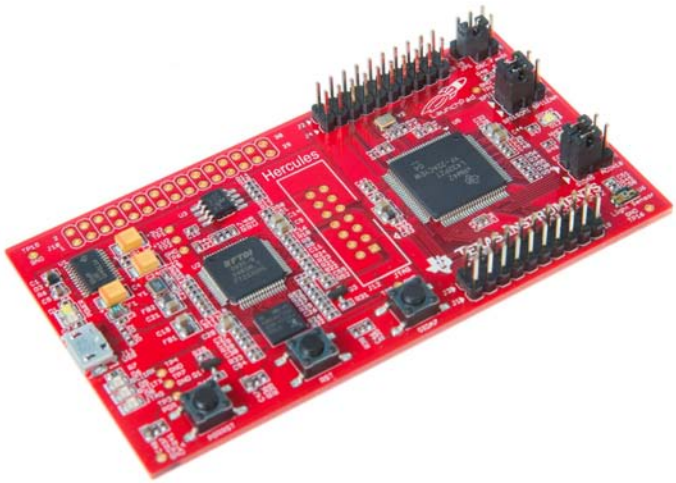


Figura 5.2: Kit de desarrollo LAUNCHXL-TMS57004

Las pruebas de validación que se mencionan a continuación se lograron con circuitos obtenidos con muestras gratuitas pedidas a Texas Instruments, algunos comprados y unos pocos que se tenían en el laboratorio del Instituto de Ingeniería exceptuando las tarjetas de desarrollo que fueron compradas.

5.1 Prueba de cargado de Software al MSP430F1612

Para validar la programación del microcontrolador MSP430F1612 se fabricó un PCB que integra a este microcontrolador y se le cargó un pequeño programa con la ayuda de una protoboard y el "lauchpad" MSP-EXP430G2553 basado en la interfaz BSL. Las conexiones necesarias se muestran en el capítulo 4 y en la figura 5.3 se observa el circuito construido que ya había sido mostrado previamente.

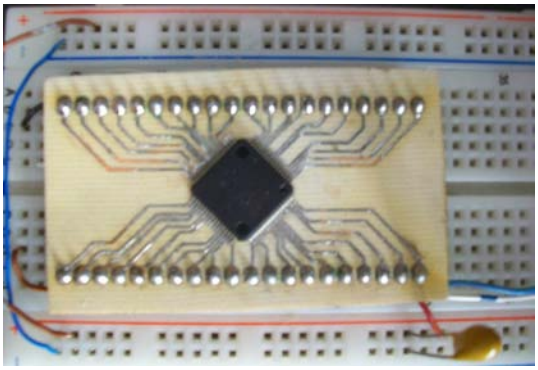


Figura 5.3: Tarjeta electrónica con MCU MSP430F1612 para pruebas de validación

5.2 Prueba de programación de Hércules vía Bootloader

Para la validación de programación del MCU Hércules solo fue necesario el LAUNCHXL-TMS57004 y una PC con hyperterminal el cual es un programa que nos permite la comunicación serie entre PC y otro dispositivo. Se descargó al inicio el software del Bootloader ya que en comparación con el MSP430 este no lo tenía precargado. Después con la ayuda de la hyperterminal y el protocolo Y módem se cargo un software básico dentro de MCU montado en el mismo kit de desarrollo y se validó su funcionamiento. Cabe destacar que en esta prueba la computadora fungía como programador pero para nuestro caso necesitamos que el MSP430 sea quien programe. De esta forma será necesario recortar código innecesario y rediseñar partes del software dedicado a uso con el MSP430.

5.3 Prueba de lectura de sensores de temperatura 1-wire

Estas pruebas se realizaron con dos sensores de temperatura 1-Wire con empaquetado TO-92 los cuales pueden ser fácilmente montados en una "protoboard" así como el kit de desarrollo LAUNCHXL-TMS57004. Aunque gran parte del software ya es dado por el fabricante, los comandos principales tuvieron que ser diseñados para trabajar con el MCU Hércules ya que no hay microcontroladores que cuenten con este periférico. Con la ayuda del software HalCoGen se pudo generar relativamente fácil las funciones necesarias.

Para validar una correcta lectura de temperatura en los sensores, se hizo variar la temperatura de uno de ellos mediante una fuente de calor y de ese modo se obtuvo una medición de la temperatura ambiente y otra de un valor mayor. Esto más el CRC que otorgan los sensores fueron valores suficientes para su validación.

5.4 Prueba de circuito contra evento "latchup"

Para esta prueba se realizaron varias PCB entre ellas la mostrada en la figura 5.4. En un inicio se simuló el circuito en Proteus y posteriormente se validó físicamente. Como carga se puso un potenciómetro de tal modo que al variar su resistencia el consumo de corriente era mayor o menor y una vez que este consumo sobrepasaba el programado mediante una resistencia al CI, este nos daba una señal de falla y el voltaje caía. La figura 3.14 del capítulo 3 muestra el circuito armado.

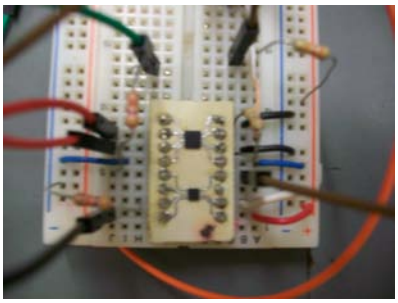


Figura 5.4: PCB construida para evento latchup

En la figura 5.5 se muestra en la parte superior la señal de falla y en la parte inferior el voltaje de salida. Se puede apreciar como al momento de detectar una sobrecorriente, el circuito intenta restablecer el voltaje catorce veces pero como el error persiste, se abre el circuito y por lo tanto el voltaje cae hasta volver a ser restablecido por la señal que habilita al circuito BQ24314.

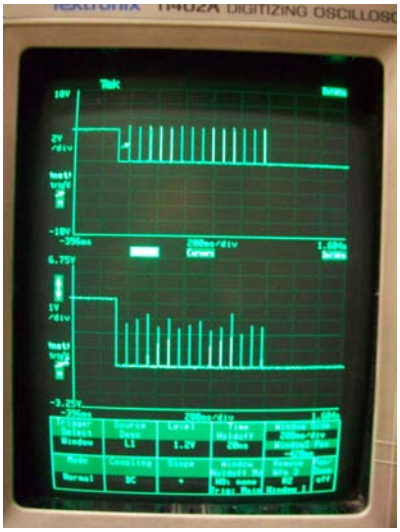


Figura 5.5: Señal de error y voltaje de salida de CI BQ24314

5.5 Conclusión

Las pruebas de validación realizadas son solo algunas sin embargo estas pruebas más la experiencia de desarrollo por parte del equipo de trabajo del IINGEN UNAM garantizan un buen diseño del subsistema. Las pruebas realizadas se lograron con dos kits de desarrollo los cuales llevan a bordo microcontroladores de la misma familia que los utilizados en el subsistema como también componentes electrónicos conseguidos como muestras del fabricante y otros comprados aquí en México, garantizando así que si existen componentes comerciales.

Capítulo 6

Conclusiones y Recomendaciones

6.1 Conclusiones

De la presente tesis desarrollada se concluyen los siguientes aspectos:

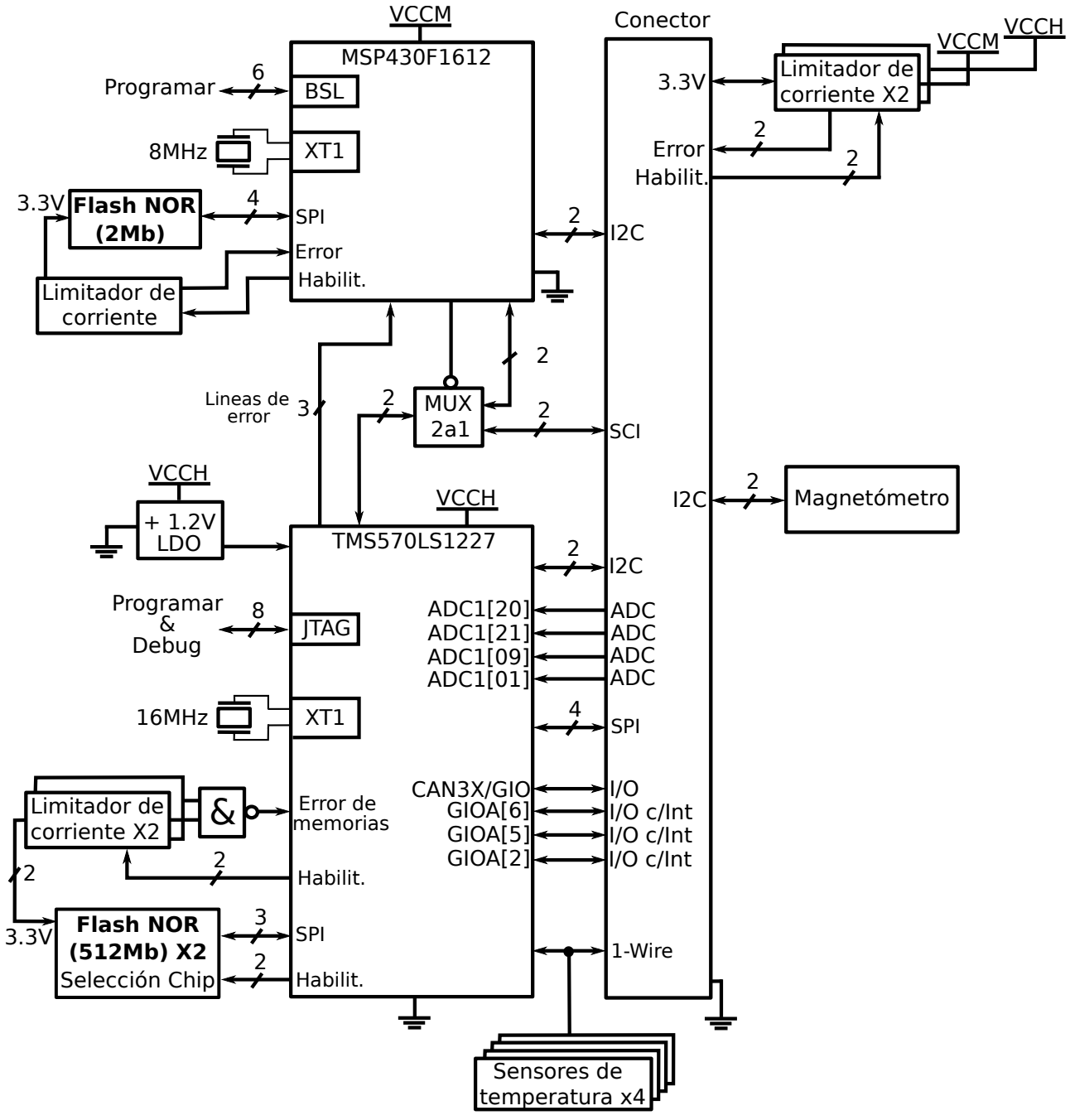
- Se efectuó el diseño de una Computadora de Vuelo cumpliendo con el objetivo de utilizar componentes COTS de bajo costo y pequeñas dimensiones, el uso de un microcontrolador de 16 bits como apoyo y validado ya en órbita así como el uso de un MCU principal de 32 bits de gran confiabilidad.
- El diseño de la CV fue pensado para evitar ciertos problemas que pudieran surgir en el espacio como el evento "latchup" y la tolerancia a temperaturas extremosas.
- La tarjeta electrónica se diseñó para tener compatibilidad con el estándar Cubesat, esto es, dimensiones, ubicación de orificios sujetadores con demás subsistemas así como recortes necesarios para el paso de cables y espacio necesario entre tarjetas y estructura.
- Se probó y desarrolló software y hardware de circuitos integrados como los que componen el sistema contra evento "latchup" y los sensores de temperatura.
- Se logró la programación de los dos microcontroladores abordo mediante el Bootloader y con la fabricación de tarjetas electrónicas para su validación.
- Puesto que un Satélite es diseñado en base a su misión y cargas útiles, se rutearon periféricos al conector de costilla como ADC, SPI, 12C, UART, 1-Wire y varios GPIO con interrupciones para de esta manera permitir versatilidad al elegir algún tipo de carga útil con alguna especificación en cuanto a su protocolo de comunicación.
- Se colocaron memorias de almacenamiento tipo Flash NOR, dos con capacidad de 512Mbit para uso en tareas de almacenamiento de datos de telemetría básica (voltajes, corrientes, temperaturas, etc) y telemetría científica(imágenes) y una de 1 Mbit para almacenamiento de programa.

6.2 Recomendaciones

- Usar un reloj en tiempo real abordo que permita a la CV ubicarse en el tiempo y mediante una interrupción avise de ciertas tareas programadas a determinada hora decidida por ordenes de la estación terrestre.
- El uso de un sistema operativo en tiempo real(RTOS). Existen RTOS específicamente para dispositivos Texas Instrument. Algunos sistemas operativos soportados por el MCU Hércules son ETAS, FreeRTOS, Sciopta.
- Gran parte de la CV se validó para su funcionamiento en hardware y software. Pruebas adicionales de vacío radiación y vibración son necesarias para una validación total del subsistema.
- Desarrollo de una interfaz de ET para realizar pruebas simulando una comunicación Satélite-Tierra.

Apéndice

Apéndice A: Diagrama a bloques de C.V.



Apéndice B: Conector