



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**ACTUALIZACIÓN DE UNA COMPUTADORA
DE VUELO PARA UN SATÉLITE
EDUCATIVO**

T E S I S

QUE PARA OBTENER EL TÍTULO DE

INGENIERO ELÉCTRICO ELECTRÓNICO

P R E S E N T A :

GARCÍA ILLESCAS MIGUEL ÁNGEL

DIRECTOR:

DR. ESAÚ VICENTE VIVAS



México D.F.

2009

AGRADECIMIENTOS

A mis padres por el apoyo incondicional que me han brindado

A la UNAM, mi segundo hogar, por haberme dado una formación integral, por haberme ayudado a ser una mejor persona

A los profesores de la Facultad de Ingeniería de la UNAM por compartir sus conocimientos y experiencias

A mis amigos: Sofía, Alejandra, Giles y compañeros de carrera, por su compañía y apoyo

Al Dr. Esaú Vicente Vivas por dirigir esta tesis

Actualización de una Computadora de Vuelo para un Satélite Educativo

Índice	Página
Capítulo 1. Proyecto CubeSat y satélites educativos	1
1.1 Introducción	1
1.2 Proyecto CubeSat	1
1.2.1 Características principales de los picosatélites CubeSat	2
1.2.1.1 Arquitectura y dimensiones	2
1.2.1.2 Sistema de lanzamiento	3
1.2.2 Proyectos CubeSat a nivel mundial	3
1.2.2.1 Universidad Aalborg (Dinamarca)	4
1.2.2.2 Universidad Wurzburg (Alemania)	4
1.3 Satélites educativos a nivel internacional y en México	5
1.3.1 Proyecto EyasSat	5
1.3.1.1 Recursos de hardware y software de EyasSat	5
1.3.1.2 Recursos de aprendizaje de EyasSat	6
1.3.2 Proyecto SATEDU (satélite educativo)	6
 Capítulo 2. Arquitectura del satélite educativo SATEDU	 9
2.1 Introducción	9
2.2 Subsistema de potencia	9
2.3 Subsistema de comunicaciones	10
2.4 Subsistema de estabilización	11
2.5 Subsistema de sensores	12
2.6 Segmento terrestre para supervisión y mando	13
2.7 Subsistema de computadora de vuelo	13
 Capítulo 3. Selección de la unidad de procesamiento de la nueva computadora de vuelo	 15
3.1 Introducción	15
3.2 Necesidades de la anterior CV	15
3.2.1 Energía	15
3.2.2 Memoria	15
3.2.3 Limitantes de diseño	15
3.2.4 Rezago tecnológico	16
3.3 Expectativas de la nueva unidad de procesamiento de la CV	16
3.3.1 Arquitectura	16

3.3.2 Menor consumo de energía	16
3.3.3 Memoria interna	16
3.3.4 Memoria Externa	17
3.3.5 Recursos de comunicaciones	17
3.4 Análisis y comparación de las ofertas disponibles de microcontroladores en el mercado	17
3.5 Justificación en la selección de la nueva unidad de procesamiento	20
Capítulo 4. Arquitectura de la nueva computadora de vuelo	22
4.1 Introducción	22
4.2 Unidad de procesamiento de la CV	22
4.3 Memoria Interna del AT91SAM7SE512	22
4.3.1 Memoria Flash	22
4.3.2 Memoria SRAM	23
4.3.3 Memoria ROM	23
4.4 Memoria Externa de la CV	26
4.4.1 Memoria SRAM	26
4.4.2 Memoria Flash SPI	29
4.5 Recursos de comunicaciones	30
4.5.1 Unidad USART	31
4.5.2 Unidad TWI	31
4.5.3 Unidad DBGU	31
4.5.4 Unidad SPI y decodificador SN74HC42	31
4.6 Bus CAN	33
4.6.1 Controlador CAN MCP2515	33
4.6.2 Transmisor-Receptor CAN MCP2551	35
4.7 Sensores de temperatura	36
4.7.1 Sensor DS18S20	36
4.7.2 ROM Code	37
4.7.3 Memoria interna	37
4.7.4 Lectura de temperaturas	38
4.8 Sistemas de protección efecto latch-up	39
4.8.1 Sensor de corriente MAX4071	39
4.8.2 Comparador LM6511	40
4.8.3 Protección de efecto Latch-up para el AT91SAM7SE512	42
4.8.4 Protección de efecto Latch-up para la memoria RAM	42
4.8.5 Convertidor A/D	43

Capítulo 5. Modo de carga de software	45
5.1 Introducción	45
5.2 AT91SAM Boot Program	45
5.2.1 Limitaciones de hardware y software del Boot Program	45
5.2.2 Comandos de AT91SAM Boot Program	46
5.3 Protocolo Xmodem	47
5.4 SAM-BA (Boot Assistant)	49
Capítulo 6. Validación del diseño de la CV	53
6.1 Introducción	53
6.2 Comunicación SPI	53
6.3 Sensores de temperatura	55
6.3.1 Búsqueda de ROM Code	56
6.3.2 Lectura de temperaturas	58
6.4 Protecciones efecto latch-up	59
6.5 Controlador de Bus CAN	62
6.5.1 Recepción de mensajes	62
6.5.2 Transmisión de mensajes	65
Capítulo 7. Desarrollo de la tarjeta electrónica	67
7.1 Introducción	67
7.2 Diagrama electrónico de la computadora de vuelo	67
7.3 Diseño del PCB de la computadora de vuelo	67
7.4 Análisis de resultados del diseño de la CV	71
Capítulo 8. Conclusiones y recomendaciones	74
8.1 Conclusiones	74
8.2 Recomendaciones	75
Apéndice A	76
A.1 Señales del AT91SAM7SE512	76
Apéndice B	79
B.1 Programa asociado a la comunicación SPI	79
B.2 Programa asociado a los sensores de temperatura 1-Wire	81
B.3 Programa asociado a los sistemas de protección efecto latch-up	85
Referencias	87

Objetivos

La presente tesis tiene como objetivo principal realizar el nuevo modelo de diseño del subsistema “computadora de vuelo” del satélite educativo (SATEDU) desarrollado en el Instituto de Ingeniería de la UNAM.

Las principales razones para modificar el diseño de la computadora de vuelo (CV) son:

- Ahorro de energía. El microcontrolador principal de la actual CV, SAB80C166, funciona con 100 mA de corriente y una fuente de 5 V mientras, cada uno de los dos módulos de memoria SRAM funciona con una corriente de 80 mA y una fuente de 5 V. Por lo tanto la actual CV requiere de 1 W de potencia para trabajar correctamente.
- Recursos. La CV limita su uso a la programación de 256 kBytes de memoria SRAM externa a través de dos módulos de 128kBytes. Además de la memoria externa SRAM, los componentes que integra la actual CV son: 1 microcontrolador SAB80C166, 1 microcontrolador PIC16F876, 2 módulos de memoria SRAM de 128 kBytes, 4 módulos de memoria flash SPI de 8 MBytes, 1 Convertidor A/D 1-Wire, 3 Sensores de temperatura 1-Wire, 3 sistemas de protección efecto latch-up (microcontroladores y memoria SRAM) y 8 canales de comunicación serial a través de 2 multiplexores.
- Limitantes de espacio. El circuito impreso de la CV cuenta con un área cuadrada de 8.9 cm por lado y dos capas, la superior y la inferior, que permiten implantar alrededor de 70 componentes distribuidos e interconectados en ambas capas a través de 160 líneas o redes.

Por lo anterior, a través de esta tesis se plantea un nuevo diseño de la CV donde los objetivos particulares son:

- Sustitución de la unidad principal de procesamiento. Considerando como parámetros principales: la disminución en el consumo de energía y la integración de los recursos suficientes de memoria para la ejecución del programa de la CV, se busca una nueva unidad de procesamiento para la CV
- Ampliación de recursos. Se plantea que la nueva CV aumente los recursos de memoria externa, tanto flash como SRAM, integre 5 sensores 1-Wire de temperatura, incluya las protecciones de efecto latch-up necesarias y cuente con un sistema de comunicación, entre la CV y los subsistemas de SATEDU, más eficiente y compacto. También se espera que los componentes trabajen con un sistema de alimentación de 3.3V para reducir el consumo de energía por parte de la CV. El subsistema de potencia de SATEDU proporciona fuentes de alimentación tanto de 5 como 3.3V a los demás subsistemas, incluida la CV.

- Reducción de espacio. A pesar del aumento de memoria SRAM y el incremento de sensores de temperatura, se pretende que la nueva CV emplee una cantidad menor de componentes electrónicos y por lo tanto de conexiones. Esto reduce la posibilidad de fallos a nivel de hardware dejando a cargo de la programación, en su mayor parte, el correcto funcionamiento de la CV.
- Conservar compatibilidad. Se espera que la nueva CV sea compatible con los subsistemas de SATEDU. Esto simplificará la integración de la nueva CV con SATEDU.

Capítulo 1. Proyecto CubeSat y satélites educativos

1.1 Introducción

Los satélites son parte fundamental de la vida moderna y contribuyen al desarrollo tecnológico global. Sirven para propósitos de toda índole siendo los más comunes: monitoreo de cambio climático, agricultura, telemedicina, sistemas de posicionamiento global (GPS) y comunicaciones. Esto hace importante el desarrollo de esta tecnología en nuestro país

Por otro lado, los satélites CubeSat por su reducido tamaño, costo y tiempo de desarrollo facilitan el diseño, construcción, validación y ejecución de pruebas o experimentos típicos en satélites pequeños, como sensores especializados (vacío, radiación, térmicos, de referencia por estrellas, entre otros), cámaras de video y manejo de actuadores diversos. A estos experimentos o pruebas que realizan satélites pequeños en órbita espacial, también se les conoce como cargas útiles.

Adicionalmente, los satélites educativos contribuyen a la formación y capacitación de recursos humanos, en universidades, centros de investigación o institutos, que pueden en un futuro especializarse en este tipo de sistemas o en sistemas relacionados a la tecnología espacial.

Las diferencias entre un satélite educativo y un satélite real, radican principalmente en diseños de los equipos y en la calidad de los componentes. En el caso de electrónica para equipo espacial, debe ser de calificación militar o espacial, aunque en el campo de satélites pequeños es frecuente el empleo de electrónica comercial empleada en la industria automotriz y de telecomunicaciones, siempre y cuando cuenten con protecciones para radiación espacial. De no tener estas protecciones la dosis de radiación acumulada puede llegar a generar el llamado efecto latch-up, el cual al aparecer y en cuestión de microsegundos lleva a la falla catastrófica de los componentes electrónicos por la presencia de consumos excesivos de corriente. En el caso de la electrónica de la computadora de vuelo que se presenta en esta tesis, se incluyen protecciones de este tipo. Se enfatiza que estos módulos no tendrán ningún uso cuando el satélite educativo (SATEDU) se emplee en salones de clase, sin embargo serán de gran utilidad cuando se pretenda utilizar esta tarjeta en un satélite pequeño que se lance al espacio.

El objetivo de este capítulo es conocer las características de los satélites CubeSat y de los satélites educativos ya que guardan una estrecha relación en sus arquitecturas.

1.2 Proyecto CubeSat

El proyecto CubeSat [27] fue desarrollado por el Dr. Jordi Puig-Suari de la Universidad Politécnica Estatal de California (Cal Poly) en San Luis Obispo y el Prof. Bob Twiggs de la Universidad de Stanford en Palo Alto, en el laboratorio de desarrollo de sistemas espaciales de la Universidad de Stanford.

El propósito del proyecto fue establecer un estándar para el diseño de satélites pequeños, con un costo reducido y mínimo tiempo de desarrollo, capaces de ser puestos en órbita y utilizarlos en la realización de experimentos a través de una carga útil como: una cámara, un detector de radiación, un medidor de campo magnético o sistemas de detección de posición.

1.2.1 Características principales de los picosatélites CubeSat

1.2.1.1 Arquitectura y dimensiones

Los picosatélites, de acuerdo a las especificaciones CubeSat, son sistemas en forma cubo con una longitud nominal de 10 cm por lado, tienen un peso no mayor a 1 kg y su centro de masa se encuentra dentro de 2 cm de su centro geométrico.

El acoplamiento de picosatélites permite diseñar sistemas de 2 o 3 Kg que conservan dos de sus dimensiones y una varía entre 227 mm y 340 mm, respectivamente. La figura 1.1 muestra la forma geométrica de un picosatélite CubeSat.

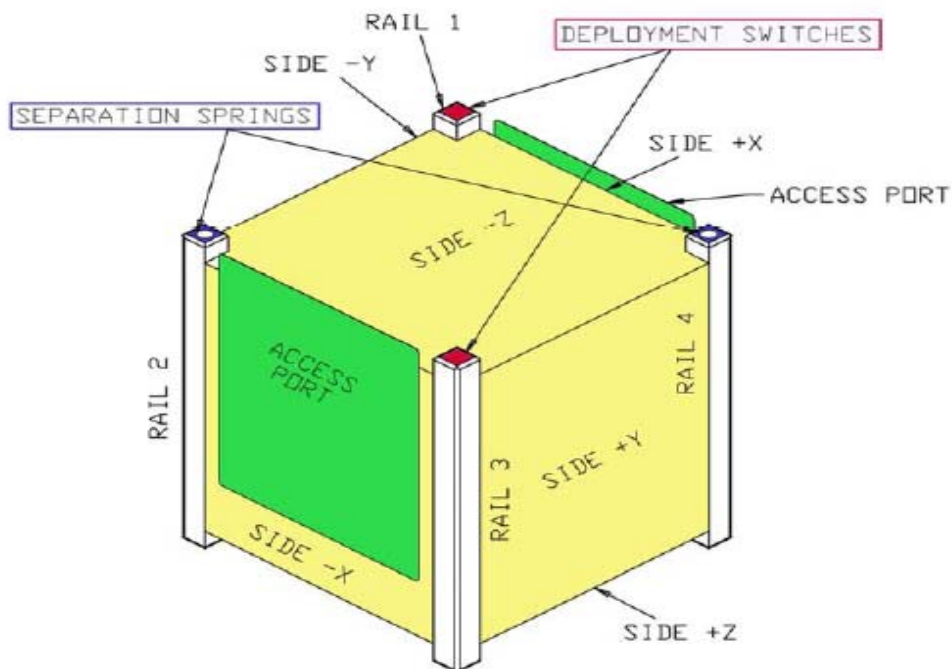


Figura 1.1 Picosatélite CubeSat

Su arquitectura consiste de un conjunto de subsistemas de suministro de energía, determinación y control de posición, sistema de comunicaciones, sistema de sensores, sistema de manejo de comandos (computadora de vuelo) y un sistema de carga útil, principalmente. Tales subsistemas, en un CubeSat, se implantan por medio de tarjetas electrónicas de 9cm, aproximadamente, de lado interconectadas entre sí.

1.2.1.2 Sistema de lanzamiento

Los satélites CubeSat son lanzados al espacio por medio de un sistema especial de lanzamiento, Poly Picosatellite Orbital Deployer (P-POD), que funciona como interfaz entre el vehículo lanzador y los picosatélites en su interior. La figura 1.2 y la 1.3 muestran el sistema P-POD y como sería un picosatélite en órbita, respectivamente.

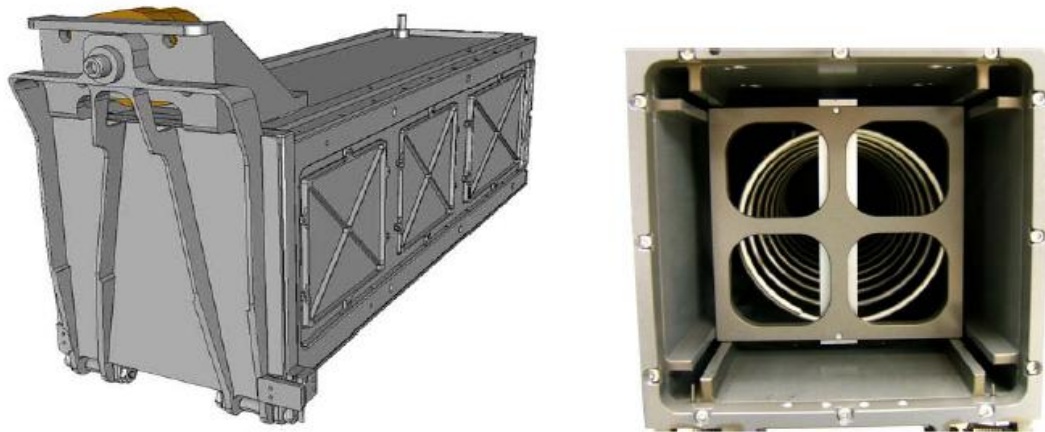


Figura 1.2 Sistema P-POD Vista lateral y sección transversal

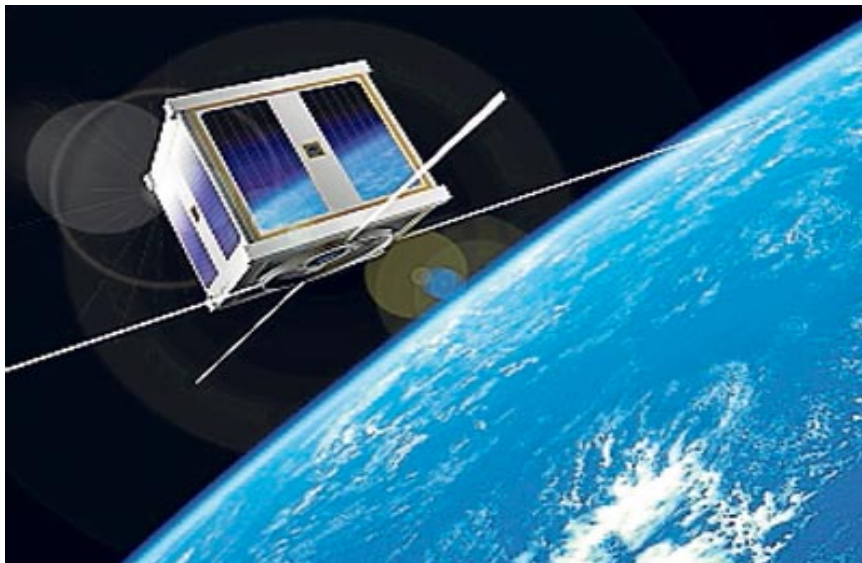


Figura 1.3 Picosatélite CubeSat en órbita

1.2.2 Proyectos CubeSat a nivel mundial

Con el auge de los picosatélites tipo CubeSat, su rápido y su sencillo desarrollo y su bajo costo económico, varias universidades alrededor del mundo han incursionado en el desarrollo de satélites experimentales pequeños. Algunos de los proyectos de picosatélites CubeSat desarrollados se presentan a continuación.

1.2.2.1 Universidad Aalborg (Dinamarca)

Desde el año de 2001 hasta la actualidad, cuenta con dos satélites CubeSat desarrollados y uno más en proceso de desarrollo.

El AUU-CubeSat [1] fue lanzado el 30 de junio del 2003 y se mantuvo operativo en órbita por 2 meses y medio. La disminución en la energía de baterías provocó su incapacidad para continuar trabajando además de que las señales de transmisión fueron muy débiles y por lo tanto los datos recuperados fueron muy escasos. No obstante fue el primer paso de la Universidad Aalborg en el desarrollo de satélites CubeSat.

El AAUSAT-II [2] comenzó su desarrollo desde el año 2003 y fue lanzado en abril del 2008. Con un detector de rayos gamma como carga y un sistema de determinación y control de posición (Attitude Determination and Control System, ADC), el satélite logró la comunicación con la estación terrena de Aalborg y los resultados del experimento fueron exitosos. Actualmente sigue en órbita.

El AAUSAT3 [3] está en fase de desarrollo y se planea usar para establecer un sistema de localización de buques que circulan en las áreas de Groenlandia y cuya responsabilidad de su seguridad está a cargo de la Administración Danesa de Seguridad Marítima (Danish Maritime Safety Administration, DaMSA).

1.2.2.2 Universidad Wurzburg (Alemania)

Lanzado el 27 de octubre del 2005, el satélite UWE-1 [30] (University of Wurzburg's Experimental-Satellite 1) fue desarrollado con el objetivo de analizar la utilidad de técnicas TCP-IP para la telemetría y telecomandos teniendo en cuenta los típicos problemas aeroespaciales de retraso y perturbaciones. Además se desarrolló para poner a prueba microsistemas de determinación de posición (giro y ángulo de declinación/inclinación).

Iniciando su desarrollo en el 2008, UWE-2 [31] es el segundo satélite CubeSat que la Universidad Wurzburg utilizará para realizar pruebas con protocolos de Internet en el espacio. También lo empleará en la determinación de posición y orientación del satélite a través de nuevos sensores de posicionamiento.

Algunos otros proyectos CubeSat se muestran en la tabla siguiente.

Universidad	País	Picosatélite
Universidad Tecnológica de Delft en los Países Bajos.	Países Bajos	Delfi-C3 (nanosatélite)
Instituto Tecnológico de Tokio	Japón	Cute 1.7

Tabla 1.1 Proyectos satelitales tipo CubeSat.

Universidad	País	Picosatélite
Universidad de Tokio	Japón	CubeSat XI-IV, XI-V
Universidad Sergio Arboleda	Colombia	Libertad 1
Universidad de Vigo	España	Xatcobeo
Universidad de Arizona	E.U.A	Rincón 1
Universidad de Ciencias Aplicadas de Aachen	Alemania	COMPASS-1

Tabla 1.1 Proyectos satelitales tipo CubeSat (continuación).

1.3 Satélites educativos a nivel internacional y en México

1.3.1 Proyecto EyasSat

Originalmente desarrollado para el uso en el departamento de Astronáutica en la Academia de la Fuerza Aérea de los Estados Unidos, el proyecto EyasSat [10] nació en enero del año 2003 con el objetivo de preparar estudiantes en el diseño, construcción y validación de pruebas en sistemas satelitales, a través de cursos y prácticas de laboratorio.

Actualmente EyasSat es el primer proyecto satelital desarrollado con fines educativos y comercializado, en universidades y laboratorios, a través de la empresa CSS (Colorado Satellite Services).

1.3.1.1 Recursos de hardware y software de EyasSat

EyasSat no es directamente compatible con la arquitectura CubeSat pues su objetivo no se centró en satélites reales sino en aspectos de entrenamiento en el campo de satélites pequeños. Tiene como subsistemas satelitales: una estructura, subsistema de energía, manejo de datos, comunicaciones, determinación y control de posición y sistema térmico, figura 1.4. Tales subsistemas son capaces de recibir comandos y generar telemetría vía una interfaz grafica de usuario.

EyasSat permite la integración de cargas útiles, desarrolladas por estudiantes, con sus subsistemas.

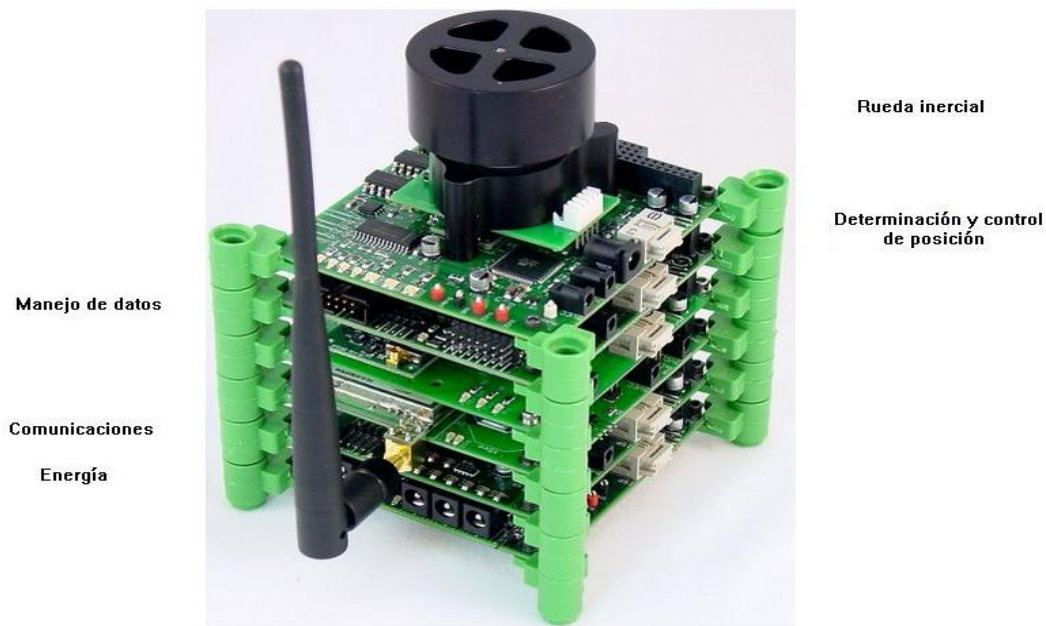


Figura 1.4 EyasSat ensamblado.

1.3.1.2 Recursos de aprendizaje de EyasSat

EyasSat cuenta con manual de usuario para estudiantes, manuales de subsistemas, software para profesores, libros de ejercicios, prácticas de laboratorio, cursos y guías para el entrenamiento de estudiantes o profesionales en el diseño, manejo, validación y verificación de pruebas de satélites pequeños.

1.3.2 Proyecto SATEDU (satélite educativo)

SATEDU [14] es un satélite educativo, diseñado, fabricado y validado completamente en el Instituto de Ingeniería de la UNAM, para ser empleado en laboratorios escolares, aulas de clases, centros tecnológicos, Universidades, Posgrados y Centros de investigación.

SATEDU permitirá entrenar jóvenes generaciones en el mundo de la ciencia y tecnología. Podrá emplearse para desarrollar y validar nuevos subsistemas satelitales, así como nuevas tecnologías para pequeños satélites reales.

SATEDU, figura 1.5, integra los subsistemas básicos de un satélite comercial: estructura, potencia, computadora de vuelo, comunicaciones inalámbricas, sensores de plataforma satelital, estabilización por rueda inercial y sensores de navegación inercial. Los subsistemas de SATEDU son pequeños, portátiles y autónomos, pues integran cada uno de ellos un microcontrolador. SATEDU fue desarrollado con el objetivo de hacerlo compatible con la norma Cubesat.

Otra característica de SATEDU es su bajo costo, que varía alrededor de \$20,000 pesos en partes mientras la adquisición del EyasSat cuesta alrededor de \$120,000 pesos en México.

Tanto el sistema EyasSat como SATEDU incluyen los sistemas que son típicos en satélites pequeños, como los subsistemas de: estructura, potencia, computadora de vuelo, comunicaciones inalámbricas de bajo alcance, sensores básicos de plataforma (temperatura, corriente y voltaje), así como algunos sensores que permiten conocer la orientación del satélite. En cuanto a sus principales diferencias, se tienen las siguientes:

- Los circuitos impresos de EyasSat son multicapa, en tanto que los de SATEDU son de dos capas y fabricados en nuestro país con una excelente calidad. Este factor redujo considerablemente el costo del sistema SATEDU.
- EyasSat emplea sistemas de comunicación inalámbrica comerciales que integran en su satélite. En el caso de SATEDU se realizó un diseño completamente nacional a partir de un circuito integrado de radiofrecuencia programable que opera en la banda de 2.4 GHz. Este factor también redujo el costo del sistema SATEDU.
- Tanto los subsistemas de EyasSat como de SATEDU son subsistemas inteligentes que basan su operación en microcontroladores. En el caso de EyasSat se emplean microcontroladores costosos de amplios recursos de Texas Instruments. En tanto que el sistema SATEDU emplea microcontroladores PIC económicos que se eligieron de acuerdo a la complejidad de las tareas que realiza cada subsistema. Este factor redujo la complejidad de diseño de cada circuito impreso de SATEDU y de igual forma contribuye a reducir los costos.
- El experimento más complicado que realiza el sistema EyasSat consiste en realizar un seguimiento de una fuente luminosa externa por medio de un arreglo pequeño de celdas solares que se usan como sensores. Para este efecto emplean tanto su rueda inercial como sus bobinas de torque magnético para hacer el seguimiento de la fuente luminosa externa. Cabe resaltar que el software que se ejecuta en PCs para interactuar con el sistema EyasSat es bastante austero. En contraparte SATEDU cuenta con un sistema de seguimiento virtual que permite al usuario hacer un seguimiento completamente visual 3D (tercera dimensión) con el apoyo del software de visualización que hace las veces de software de estación terrestre satelital. Este software es altamente visual, único y sin precedentes en sistemas comerciales.
- El sistema SATEDU cuenta además con un modo adicional de validación de funcionamiento que emplea su rueda inercial. Bajo este modo el software de estación terrestre permite enviar tareas a SATEDU, con las que las revoluciones por minuto de la rueda inercial se modulan en diversas formas de onda (senoidal y diente de sierra). Bajo este modo cuando SATEDU se suspende del techo por medio de un hilo, y se le envía un comando como el citado anteriormente, el usuario percibe de manera muy sencilla lo que es la operación automática de un satélite, que en este caso experimenta movimientos físicos periódicos que puede controlar el usuario desde el software de una PC.

En el capítulo siguiente se describen los componentes de SATEDU.

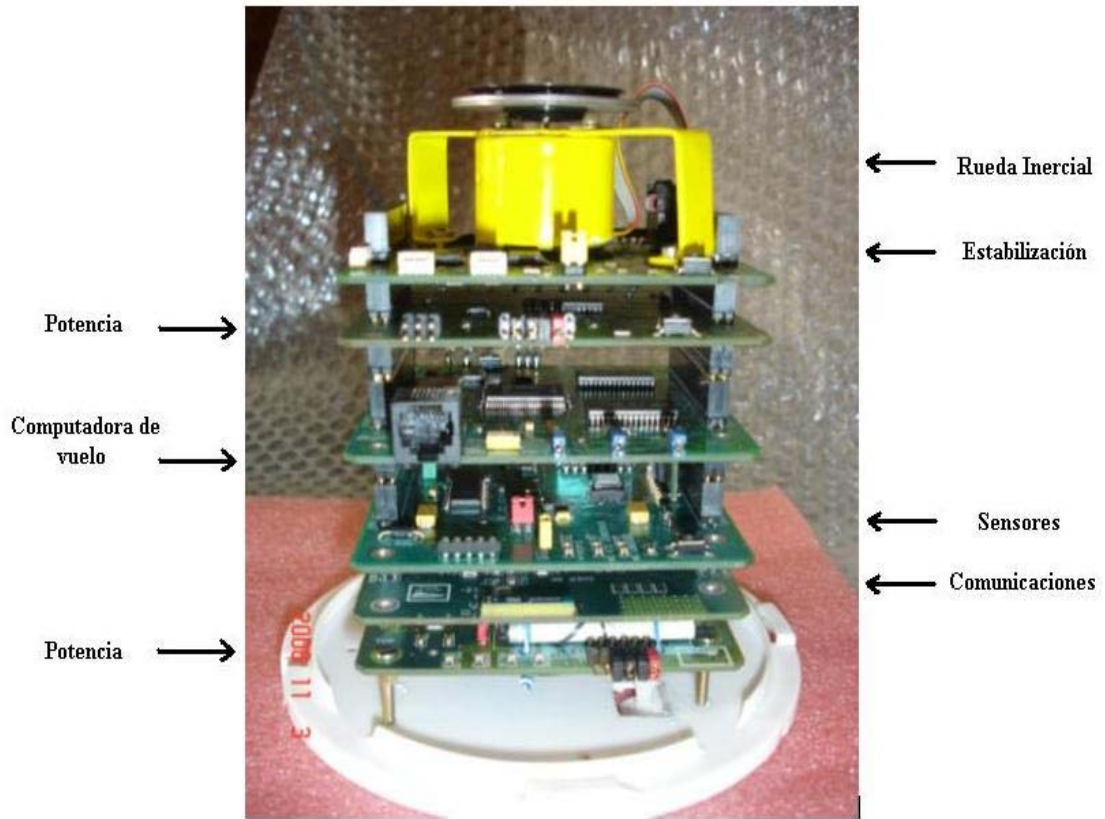


Figura 1.5 SATEDU ensamblado.

Capítulo 2. Arquitectura del satélite educativo SATEDU

2.1 Introducción

En este capítulo se explica la arquitectura empleada en el desarrollo de SATEDU del Instituto de Ingeniería de la UNAM; se describen los subsistemas y sus funciones principales.

Se describen las tareas de la computadora de vuelo (CV) y su relación con los demás subsistemas.

2.2 Subsistema de potencia

El subsistema de potencia se encarga de controlar el encendido y apagado de los subsistemas del satélite, por medio de comandos transmitidos por la CV. Está compuesto de dos tarjetas electrónicas. La primera integra un microcontrolador (PIC 18F2321) y la electrónica que regula y administra las fuentes de energía de los demás subsistemas. La segunda integra 4 baterías de litio y un módulo de carga que acepta dos fuentes de alimentación, una a través de paneles solares y otra a partir de un adaptador AC. Las figuras 2.1 y 2.2 muestran el subsistema de potencia.



Figura 2.1 Subsistema de potencia (Módulo de batería)

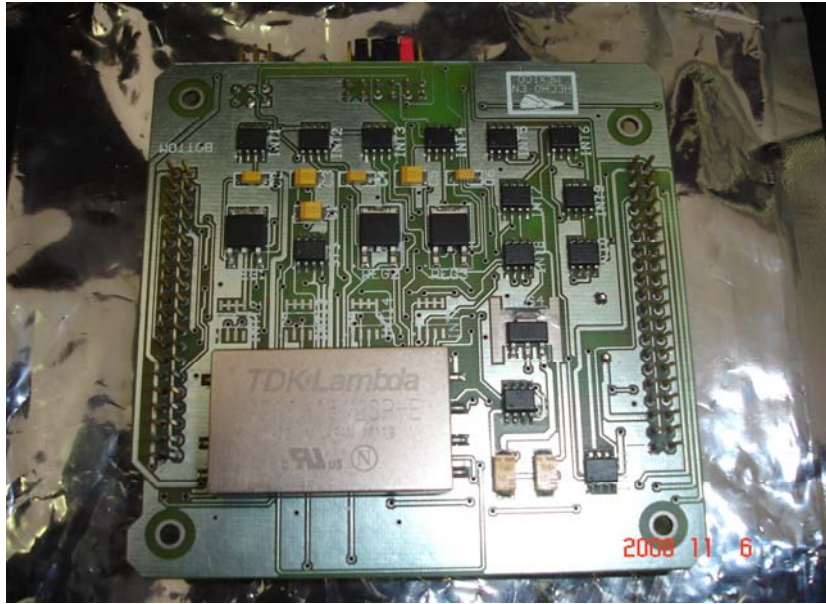


Figura 2.2 Subsistema de potencia (Módulo de administración de energía)

2.3 Subsistema de comunicaciones

Consiste en dos tarjetas: un módulo conectado a la computadora de estación terrestre (ET), donde se encuentra el software que controla el satélite; y una tarjeta de comunicaciones para SATEDU, de igual diseño que los demás subsistemas, cuya función es la recepción y transmisión de datos y mensajes entre el satélite y el software de ET.

La tarjeta de comunicaciones tiene un microcontrolador PIC como medio de comunicación con la CV, de donde también puede recibir y transmitir comandos. Las figuras 2.3 y 2.4 muestran el subsistema de comunicaciones.



Figuras 2.3 y 2.4 Subsistema de Comunicaciones ET y SATEDU.

2.4 Subsistema de estabilización

Cuenta con un microcontrolador (PIC18F4431) que recibe comandos de la CV para controlar la velocidad de una rueda inercial (motor de corriente directa) y de 6 bobinas de torque magnético, para este propósito emplea un puente H y modulación PWM, con las cuales se realiza el manejo de la rueda inercial en lazo abierto hasta este momento.

La figura 2.5 muestra el subsistema de estabilización con una rueda inercial montada.



Figura 2.5 Subsistema de estabilización.

Las pruebas de estabilización por el momento se realizan de forma virtual con Simulink de Matlab y forman parte de una tesis de Maestría en Instrumentación. Adicionalmente, se tiene pensado emplear una mesa suspendida en aire para realizar las primeras pruebas de validación de estabilización en tres ejes, este trabajo también forma parte de la tesis de maestría referida.

La figura 2.6 muestra la mesa suspendida en aire para pruebas de demostración preliminares de estabilización.



Figura 2.6 Mesa suspendida en aire para pruebas de estabilización.

2.5 Subsistema de sensores

Es la tarjeta donde se alojan los sensores de detección de movimiento del satélite, es decir, los sensores que permiten visualizar su posición en tiempo real. Tales sensores son: una brújula electrónica, un acelerómetro de 3 ejes y 3 giróscopos electrónicos que se colocan de manera ortogonal entre si para monitorear el movimiento del satélite en sus respectivos ejes. La figura 2.7 muestra la tarjeta de sensores del satélite.



Figura 2.7 Subsistema de sensores.

2.6 Segmento terrestre para supervisión y mando

Es el software desarrollado para la estación terrestre (ET) bajo la plataforma Visual Basic, que realiza funciones de envío de comandos al satélite, recepción y despliegue de datos y telemetría en la computadora de ET. Puede ser instalado en cualquier PC con Windows XP.



Figura 2.8 Software de estación terrestre.

2.7 Subsistema de computadora de vuelo

La CV de SATEDU se desarrolló para controlar las tareas del satélite tales como: la recepción de comandos desde el software de ET, el envío de telemetría a la ET, la comunicación por medio de comandos con los demás subsistemas, la coordinación de algoritmos de estabilización, el encendido o apagado de cualquiera de los subsistemas, el acceso y manejo de 32 MBytes de memoria masiva tipo Flash y la lectura de los sensores de tecnología 1-Wire (corriente, voltaje y temperatura).

La CV cuenta con un microcontrolador de tecnología RISC de Siemens SAB80C166 [24] de 16 bits, 256 kBytes de SRAM externa para ejecución de programas, un microcontrolador PIC para subir nuevos programas por telemetría, 9 puertos seriales de comunicaciones, plataforma de desarrollo de software en Lenguaje C, todo esto en un circuito impreso de dos capas de manufactura nacional de 8.9 x 8.9 cm.



Figura 2.9 Computadora de vuelo de SATEDU, versión 1, desarrollada completamente en el Instituto de Ingeniería UNAM.

Capítulo 3. Selección de la unidad de procesamiento de la nueva computadora de vuelo

3.1 Introducción

El propósito de este capítulo es exponer las principales razones y necesidades que llevaron a sustituir la unidad de procesamiento de la anterior CV por una actual que, además de ejecutar las tareas de la CV de SATEDU, favoreciera el crecimiento y operatividad de la CV en relación con la ejecución de tareas complejas que lleven a un mejor funcionamiento de un satélite pequeño.

3.2 Necesidades de la anterior CV

La CV anterior [12], del proyecto SATEDU, cuenta con un microcontrolador SAB80C166 [24] como unidad de procesamiento principal. El SAB80C166 realiza las funciones de enviar y recibir comandos a los demás subsistemas, monitorea el consumo de energía de la CV, monitorea la temperatura de la tarjeta, controla el flujo de datos en la memoria flash, coordina el correcto funcionamiento del satélite en conjunto con los subsistemas restantes y toma decisiones ante eventualidades que puedan presentarse y afectar el funcionamiento del satélite. También cuenta con un microcontrolador PIC encargado de enviar una señal reset al SAB80C166 y de ponerlo en modo de carga con ayuda de buffers de tres estados y compuertas lógicas.

Los principales problemas y necesidades que se presentan en la CV de SATEDU y que llevaron a rediseñarla, buscando una nueva unidad de procesamiento, fueron los siguientes.

3.2.1 Energía

Con una fuente de alimentación de 5 [V], el SAB80C166 utiliza hasta 100 [mA] de corriente, en condiciones adversas. La potencia de operación de 0.5 [W] del SAB80C166 se manifiesta con incrementos elevados en la temperatura del microcontrolador y por consecuencia de la CV además de consumir la mayor parte de la energía de las baterías.

3.2.2 Memoria

El SAB80C166 solo cuenta con 1 kByte de memoria RAM y 8 kBytes de memoria ROM. La memoria RAM interna se utiliza para almacenar un programa cargador de aplicaciones en la memoria RAM externa, que es de 256 kBytes, máximo que puede manejar el SAB80C166, distribuidos en dos chips de 128 kBytes.

3.2.3 Limitantes de diseño

La CV anterior tiene 70 componentes electrónicos, aproximadamente, implantados en una tarjeta de 8.9cm por lado. Por lo anterior, se buscó que la nueva unidad de procesamiento contribuyera a minimizar la cantidad de componentes empleados en el diseño de la nueva

computadora de vuelo sin comprometer su desempeño y funcionalidad.

3.2.4 Rezago tecnológico

El SAB80C166 de la CV de SATEDU, al igual que algunos de los componentes que lo conforman, es de tecnología obsoleta. Además de las altas temperaturas que alcanza por el elevado consumo de energía y la poca memoria, la adquisición del SAB80C166 ya no es tan accesible y existen en el mercado una gran cantidad de microcontroladores de diversas marcas que ofrecen un desempeño igual o mucho mejor en procesamiento, consumo de energía, memoria, comunicaciones, periféricos internos y compatibilidad con una amplia gama de periféricos externos.

3.3 Expectativas de la nueva unidad de procesamiento de la CV

Las principales expectativas que influyeron en el análisis y selección de la nueva unidad de procesamiento de la CV fueron las siguientes.

3.3.1 Arquitectura

Debido a las limitantes de diseño, se optó por un microcontrolador como unidad de procesamiento. Un microcontrolador integra, además de una unidad de control y unidad aritmética lógica, componentes de comunicación, convertidores A/D, temporizadores, comparadores y memorias, principalmente, de tal manera que contribuye a disminuir la cantidad de componentes requeridos en el diseño de la CV.

Por otro lado se planteó analizar microcontroladores de arquitectura de 16 y 32 bits. Un microcontrolador de 16 bits tiene la ventaja de trabajar con un consumo menor de energía, sin embargo su memoria, periféricos internos, posibilidades de expansión, precisión y exactitud en procesamiento son limitados, en comparación a uno de 32 bits

Por lo anterior, se decidió buscar una unidad que proporcionara un buen equilibrio entre sus recursos internos, velocidad de procesamiento, posibilidades de expansión y consumo de energía.

3.3.2 Menor consumo de energía

Para minimizar el consumo de energía, se consideró que la unidad de procesamiento trabajara con un sistema de alimentación de 3.3 [V], compatible con los componentes externos. También se consideró que debía disponer de un sistema eficiente en la administración de energía y que proporcionara métodos para optimizar y reducir su consumo.

3.3.3 Memoria interna

Se planteó que el microcontrolador contara con la memoria necesaria para alojar el programa principal, 256 kBytes, y poder ejecutarlo de forma adecuada con ayuda de sus

recursos de memoria ROM o Flash y SRAM.

3.3.4 Memoria Externa

Se consideró que la nueva unidad de procesamiento dispusiera de una interfaz compatible con dispositivos de memoria RAM externa. Dicha memoria permitirá la incorporación de programas más complejos tales como sistemas operativos (SOS) que facilitan el empleo de protocolos de red, sistemas de archivos o aplicaciones más avanzadas que requieren una mejor administración tanto de recursos como de procesamiento.

Se determinó que la interfaz de bus externo manejara como mínimo 2 MBytes de memoria RAM.

3.3.5 Recursos de comunicaciones

Como medio de comunicación entre los subsistemas y la CV, se planteó buscar un protocolo que redujera la cantidad de conexiones y componentes requeridos para su implantación, sin dejar de lado la rapidez de transmisión de datos.

El protocolo Bus CAN [7] se consideró el más adecuado por su sencilla estructura de dos líneas para la transmisión y recepción de mensajes, detección de errores, sistema de prioridades y transmisión de hasta 1 Mbit/s, en comparación a los canales de comunicación serial, bajo el estándar RS232, que se emplearon en la anterior computadora de vuelo.

El Bus CAN permite añadir nuevos nodos al bus o quitar aquellos que resulten defectuosos, sin necesitar componentes extras. De esta manera, mientras la CV de SATEDU utiliza 2 multiplexores para manejar 8 canales de comunicación serial, con los demás subsistemas, el protocolo Bus CAN permite disminuir la cantidad de componentes requeridos para establecer la comunicación entre los subsistemas del satélite.

Al igual que la nueva CV, los subsistemas requieren la sustitución de su actual microcontrolador por uno que integre el protocolo Bus CAN. Esto resulta factible actualizando el microcontrolador PIC por uno que integre el protocolo.

Además del protocolo Bus CAN, se consideró que la nueva unidad de procesamiento debía disponer de los sistemas de comunicaciones SPI, I2C, USART, que sirven como medio de comunicación con los periféricos externos de la CV.

3.4 Análisis y comparación de las ofertas disponibles de microcontroladores en el mercado

Los microcontroladores que cumplieron con las expectativas planteadas en la búsqueda de la nueva unidad de procesamiento y resultaron mejores, en comparación con los de la competencia, fueron el AT91SAM7SE512 de Atmel [6], el SAF-XE164F-96F66LAC de Infineon [13], el H8SX/1544 de Renesas [23] y el PIC24HJ256GP610 de Microchip [21]. La tabla 3.1 muestra sus principales características.

Durante el análisis y comparación de microcontroladores, el H8SX/1544 y el PIC24HJ256GP610 se descartaron como posibles unidades de procesamiento para la nueva CV por las razones siguientes.

El H8SX/1544 siendo de 32 bits, ofrece recursos suficientes de memoria, comunicaciones, interfaz de bus externo, herramientas de carga de software, etc, no obstante funciona con una fuente de alimentación de 5 [V] y opera con 63 [mA] de corriente. El consumo de energía del H8SX/1544 es superior a los restantes microcontroladores analizados.

El PIC24HJ256GP610 siendo de los más completos microcontroladores de 16 bits, no ofrece una interfaz de bus externo por lo cual su utilidad se limita al uso de su memoria interna. Existen microcontroladores microchip de 32 bits de características similares al PIC24HJ256GP610 y que incluyen una interfaz de bus externo, sin embargo solo pueden manejar 128 kBytes de memoria a través de un bus de direcciones y uno de datos, ambos, de 16 bits.

En cuanto al microcontrolador AT91SAM7SE512 (AT91) y el SAF-XE164F-96F66LAC (SAF), sus características son las siguientes.

Energía. Siendo de 32 bits, el AT91 trabaja con solo 31 [mA] de corriente y una fuente de alimentación de 3.3 [V]. El SAF trabaja a 27 [mA] y 3.3 [V]

Memoria interna. El SAF cuenta con 768 kBytes de memoria flash y 82 kBytes de memoria RAM, mayores a las del AT91.





Interfaz externa. Ambos microcontroladores, AT91 y SAF, cuentan con una interfaz de bus externo compatible con dispositivos de memoria RAM externa.

Comunicaciones. Además de los sistemas de comunicaciones estándar, SPI, I2C, unidades USART, el SAF incluye un controlador CAN a diferencia del AT91.

Modo de carga de programas. Tanto el AT91 como el SAF, incluyen herramientas y métodos de cargado de software en su memoria interna o externa. Integran un bootloader en su memoria ROM.

En cuanto a sistemas operativos (S.O) y aplicaciones, el microcontrolador AT91 (núcleo ARM7TDMI) cuenta con una mayor cantidad disponible ya sean de paga o de distribución libre. El SAF limita su aplicación a S.O comerciales además de disponer de una cantidad reducida de aplicaciones desarrolladas bajo su plataforma.

La tabla 3.2 muestra algunos de los S.O disponibles en el mercado para ambos microcontroladores.

Modelo y marca	AT91SAM7SE512 ARM7TDMI 	SAF-XE164F- 96F66L AC 	H8SX/1544 Renesas 	PIC24HJ256GP610 Microchip 
Arquitectura	32 bits RISC	16 bits RISC	32 bits CISC	16 bits RISC
MIPS/Frecuencia Max.	55 [MHz]	66 [MHz]	40 MHz	40 MIPS
Memoria	512 kB Flash 32 kB SRAM	768 kB Flash, 82 kB SRAM	512 kB Flash 24 kB SRAM	256 kB Flash 16 kB SRAM
Interfaz Externa	EBI 8 líneas chip select	EBI 12 MB	EBI 16 MB	NO
Comunicaciones	2 USART 1 SPI 1 I2C (TWI) 1 USB	6 USIC (I2C, SPI, USART) 4 CAN	4 USART 2 SPI 2 I2C 2 CAN	2 USART 2 SPI 2 I2C 2CAN
Timers 16 bits, Unidades CAPCOM, PWM	3Timers/CAPCOM 4 PWM	11 Timers 3 CAPCOM- PWM	6 Timers 6 CAPCOM	9 Timers 8 CAPCOM-PWM
CA/D	CA/D 8 Chs 10 bits	2 CA/D 16 Chs 10bits	1 CA/D 16 Chs 10 bits	2 CA/D 32 Chs 10 o 12 bits
Fuente de alimentación [V]	3.3	3.0 -5.0	5.5	3.3
Temperatura de operación [°C]	-40 a 85	-40 a 85	-40 a 105	-40 a 85
Consumo de energía. ¹⁾	P=102 [mW] I=31 [mA]	P=89 [mW] I=27 [mA]	P=346 [mW] I = 63 [mA]	P=168 [mW] I=51 [mA]
Programación en sistema/Bootloader	Si/Si	Si/Si	Si/Si	Si/Si
Encapsulado y disponibilidad en el mercado	LQFP-128 Pines Disponible	LQFP-100 Pines Disponible	FP-144L Nuevo	TQFP-100 Pines Disponible

.Tabla 3.1 Características principales de los microcontroladores analizados

1) Estos datos fueron obtenidos y calculados bajo situaciones similares para todos los microcontroladores: trabajando a 20 MHz con todos los relojes de periféricos activos y con la fuente mínima de alimentación.



Modelo y marca	AT91SAM7SE512 ARM7TDMI 	SAF-XE164F 96F66L AC 
S.O de tiempo real (RTOS) y S.O de distribución libre	eCos , UC/OS-II MICRIUM CMX-RTX, pC/OS Kernel FreeRTOS, IAR PowerPac uClinux ²⁾	CMX-RTX, TAXKING-C166 KEIL ARTX-166 UC/OS-II MICRIUM

Tabla 3.2 Sistemas operativos disponibles para la unidad de procesamiento.

2) uClinux [9] tiene una aplicación disponible para el núcleo ARM7TDMI por lo que, con el equipo y recursos suficientes, es factible adaptarlo al microcontrolador Atmel.

3.5 Justificación en la selección de la nueva unidad de procesamiento

El microcontrolador seleccionado como la nueva unidad de procesamiento de la CV fue el AT91SAM7SE512 (AT91) de Atmel, basado en el núcleo ARM7TDMI. Las principales razones fueron las siguientes.

Arquitectura. De arquitectura von Neumann, el AT91 permite la ejecución de código de 32 bits o de 16 bits al integrar dos sets de instrucciones de tipo RISC. Esto significa que puede seleccionarse entre exactitud y rendimiento con código de 32 bits o en el desarrollo de programas que requieran una mayor cantidad de instrucciones. El código puede ejecutarse de la memoria interna o externa del microcontrolador con una tasa de 0.9MIPS/MHz.

Otra característica que optimiza el rendimiento del AT91 es su estructura dividida en módulos de control que ejecutan tareas sin necesitar la intervención del procesador o núcleo ARM7TDMI del microcontrolador. Tales tareas son la administración de interrupciones, interfaz externa, memoria, comunicaciones, energía, puertos I/O, principalmente.

Eficiencia energética. Siendo un microcontrolador de 32 bits trabaja a 3.3 [V] y con una corriente de 31 [mA] que puede ser reducida a 26 [µA] cuando el microcontrolador se encuentra en modo idle y sus periféricos no son utilizados.

El microcontrolador AT91 integra dos osciladores, uno interno RC de 22 KHz y uno externo de 20 MHz. El oscilador de 22 KHz proporciona una señal constante de reloj al procesador o núcleo del AT91 y permite ponerlo en modo idle o modo de espera. El

oscilador de 20 MHz puede ser seleccionado como fuente de la señal MCK (Master Clock) la cual es asignada al controlador de memoria y al controlador de interrupciones que se ejecutan permanentemente. La señal MCK también es asignada a los periféricos internos del microcontrolador, conforme se requieren utilizar, es decir a la unidad USART, SPI, I2C y EBI. Cabe mencionar que el AT91 integra un PLL (Phase-Locked Loop) el cual puede utilizarse como multiplicador de frecuencia permitiendo al AT91 trabajar hasta con frecuencias de 55 MHz.

Al trabajar con una corriente de 31 [mA] y disponer de un sistema de administración y control de energía, el AT91 resulta eficiente para aplicaciones pequeñas como SATEDU donde las fuentes de energía son limitadas.

Memoria. El AT91 cuenta con 512 kBytes de memoria flash y 32 kBytes de SRAM suficientes para almacenar y ejecutar el programa de la CV de SATEDU. Además de la memoria interna, el AT91 cuenta con una interfaz externa que maneja hasta 16 MBytes de memoria a través de un bus de direcciones de 23 bits y uno de datos de 16 bits. Esto permite ampliar su uso a aplicaciones más complejas como sistemas operativos, sistemas de archivos o protocolos de red, que requieren una mayor cantidad de espacio.

Modo de carga de software. El AT91 integra un bootloader en su memoria ROM el cual permite, mediante comandos, realizar la carga de software en la memoria interna o externa. Mientras el SAB80C166 emplea un microcontrolador PIC y circuitos de lógica digital para ponerlo en modo de carga, el AT91 utiliza solo una línea. Como medio de transmisión y recepción de datos, el AT91 utiliza dos líneas de la unidad serial DBGU (Debug Unit) durante la carga de software. Esto significa que el AT91 permite disminuir la cantidad de componentes requeridos en el diseño de la nueva CV.

Software. El AT91 dispone de una mayor cantidad de aplicaciones, compiladores, sistemas operativos y recursos de software, comerciales o de distribución libre, pues su procesador es de arquitectura ARM, ARM7TDMI. Los microcontroladores o microprocesadores ARM son empleados en la industria y en el desarrollo de aplicaciones pequeñas y portátiles donde el consumo de energía es un punto vital por ejemplo: celulares, videojuegos, reproductores o bien en SATEDU.

Finalmente y a pesar de no contar con un controlador CAN, el AT91 fue seleccionado como unidad de procesamiento de la nueva CV por sus características de rendimiento energético, flexibilidad, versatilidad, posibilidades de expansión y su relativa sencillez de implantación.

Capítulo 4. Arquitectura de la nueva computadora de vuelo

4.1 Introducción

En este capítulo se describen los componentes que integra la nueva computadora de vuelo del satélite, las características principales de la unidad de procesamiento, los módulos de memoria interna y externa, los sistemas de comunicaciones, los sistemas de protección de efecto latch-up, los sensores de temperatura y así mismo se describe la forma de implantarlos en hardware y establecer la comunicación y control por medio de la unidad principal de procesamiento.

4.2 Unidad de procesamiento de la CV

La nueva CV dispone de un microcontrolador Atmel AT91SAM7SE512 [6] de 32 bits de arquitectura Von Neumann, núcleo ARM7TDMI y set de instrucciones RISC como unidad de procesamiento. El AT91SAM7SE512 cuenta con 3 niveles pipeline (obtener instrucción, decodificarla y ejecutarla), una señal de reloj de 20 MHz con una eficiencia de 0.9MIPS/MHz, memoria interna de 512 kBytes flash y 32 kBytes SRAM, sistema de interrupciones con 8 niveles de prioridad, 8 canales de convertidor A/D con un tiempo por conversión de 2 μ s a 10 bits de resolución y 1.25 μ s a 8 bits, una interfaz de bus externa (EBI) compatible con memorias externas SRAM, SDRAM, PROM, EPROM, EEPROM, NAND Flash y Compaq Flash. Además cuenta con 3 canales Timer/Counter de 16 bits, 4 canales PWM, 1 PLL (phase-locked loop), 2 unidades seriales USART, 1 unidad de comunicación serial SPI, 1 interfaz tipo Two-Wire y un puerto USB, utilizando 88 pines de propósito general y un sistema de multiplexado para el manejo de sus periféricos embebidos. Se alimenta, en su totalidad, con una fuente de 3.3 V y tiene un encapsulado LQFP de 128 pines.

La conexión del microcontrolador con los componentes externos de la nueva CV se muestra en la figura 4.1.

En el apéndice A se muestran las señales del AT91SAM7SE512 utilizadas en el diseño de la CV.

4.3 Memoria Interna del AT91SAM7SE512

4.3.1 Memoria Flash

El AT91SAM7SE512 integra 512 kBytes de memoria flash interna organizada en 2 bancos de 1024 páginas de 256 Bytes con un tiempo de acceso de 33 ns. Cuenta con 32 bits para la protección de 32 regiones de tamaño variable, un detector de apagones BOD (Brownout detector) y es accesible, cualquier momento, en la dirección 0x0010 0000. Por medio de un bit de propósito general NVM (non-volatile memory), modificable a través de los registros del AT91SAM7SE512, puede accederse a la memoria flash en la dirección 0x0 para ejecutar el código de programa dentro de ella al inicio o reseteo del AT91SAM7SE512.

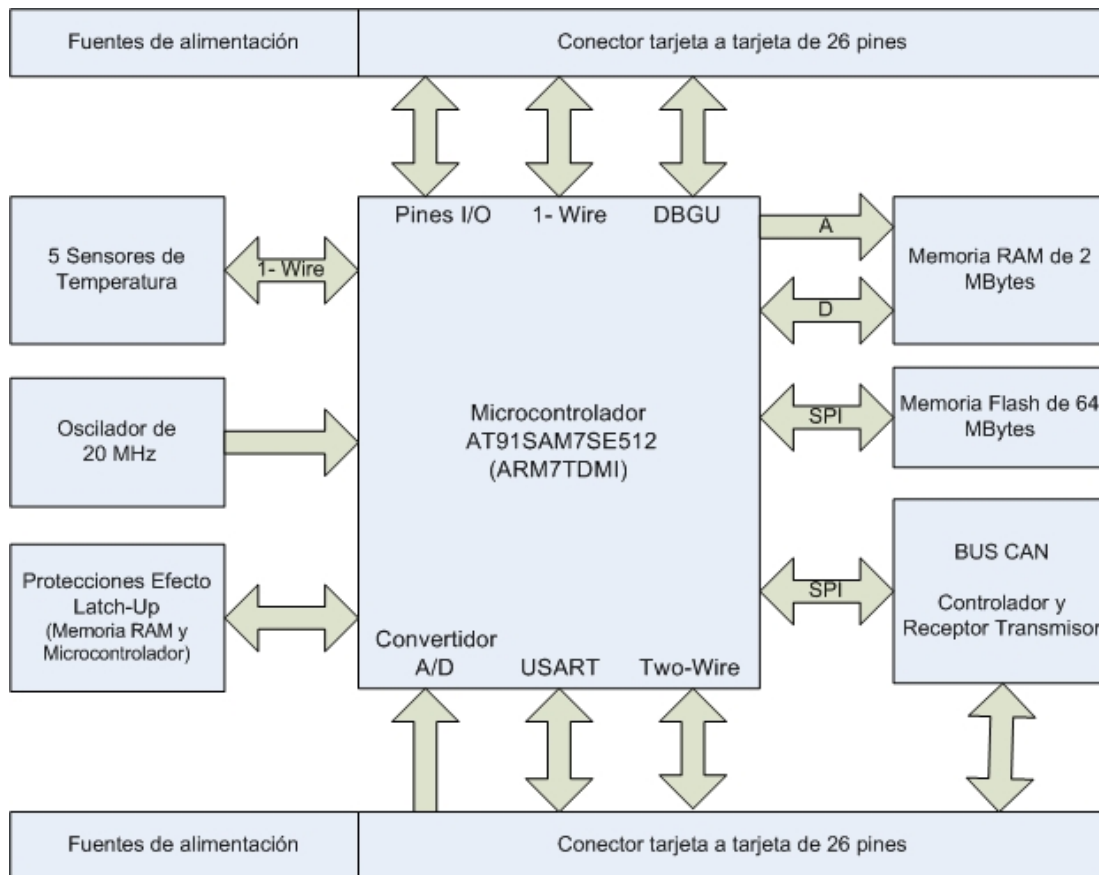


Figura 4.1 Arquitectura de la CV.

4.3.2 Memoria SRAM

El AT91SAM7SE512 integra 32 kBytes de memoria SRAM accesible en la dirección 0x0020 0000 antes del comando Remap. Después del comando Remap es accesible tanto en la dirección 0x0020 0000 como en la dirección 0x0.

El comando Remap cambia la posición de acceso a las memorias, se puede utilizar para iniciar, después de resetear el sistema, la ejecución de código desde la memoria SRAM o la memoria Flash del AT91SAM7SE512.

4.3.3 Memoria ROM

El AT91SAM7SE512 incluye una memoria ROM ubicada siempre en la dirección 0x0030 0000. Por medio de un bit de propósito general NVM puede ser accesible tanto en la dirección 0x0030 0000 como en la 0x00. Almacena el programa SAM-BA boot que permite programar la CV mediante la interfaz gráfica del software libre In-System programmer de la empresa Atmel y a través de una comunicación serial en cualquier computadora con Windows® XP como sistema operativo.

Las figuras 4.2 y 4.3 muestran el mapa de memoria del AT91SAM7SE512 de acuerdo al valor del bit 2 de propósito general NVM.

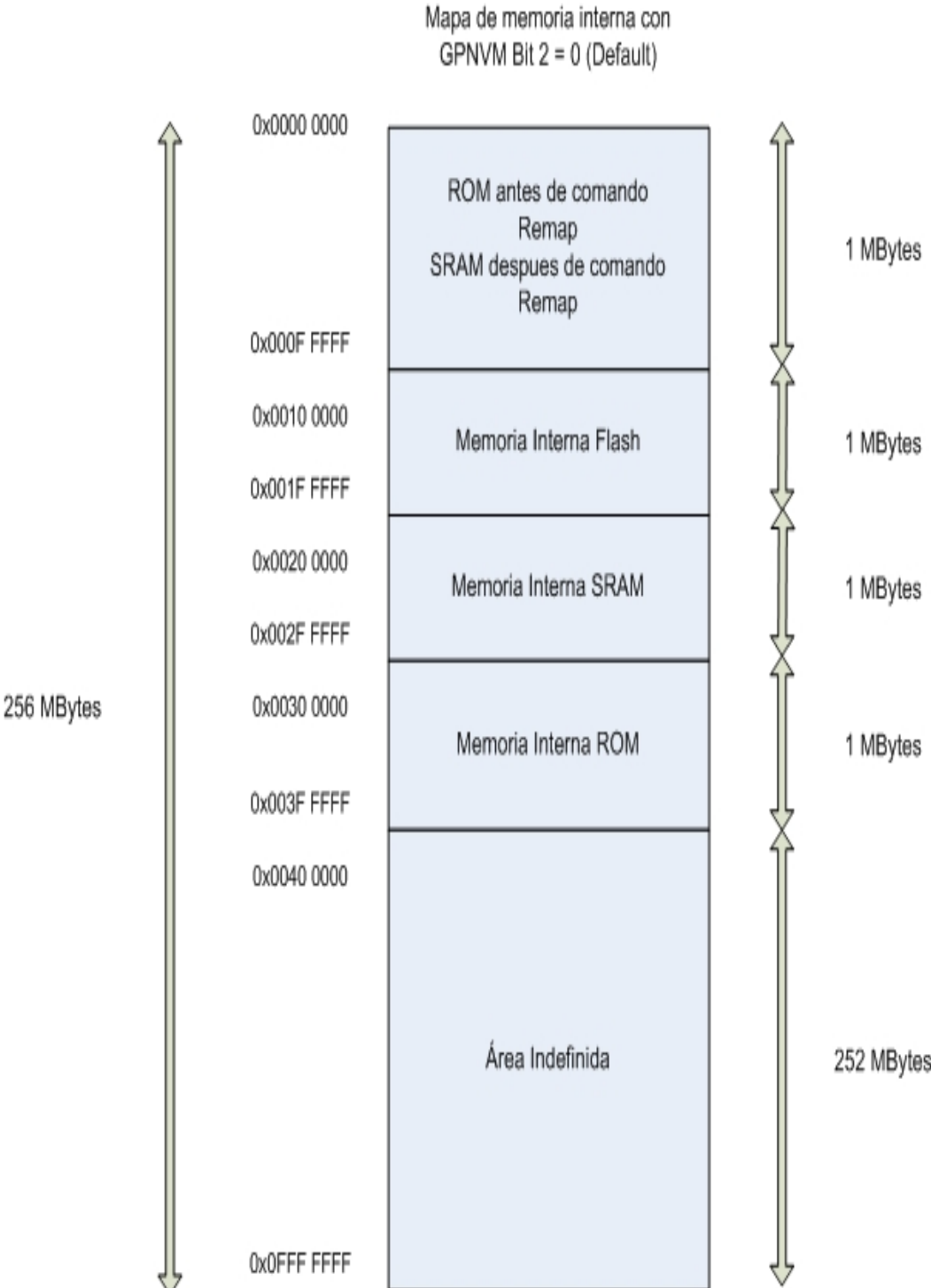


Figura 4.2 Mapa de memoria interna con GPNVM Bit 2 =0

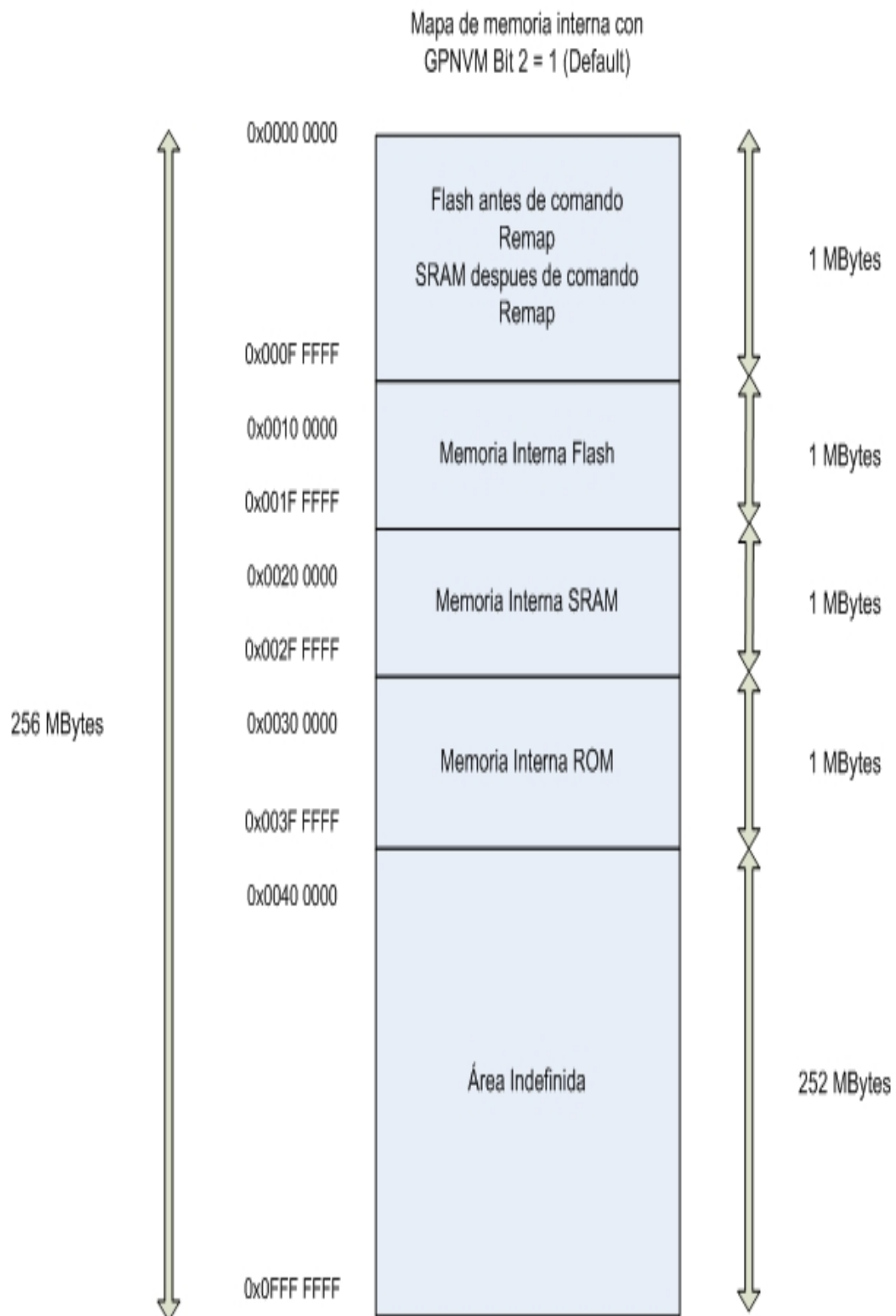


Figura 4.3 Mapa de memoria interna con GPNVM Bit 2 =1

4.4 Memoria Externa de la CV

4.4.1 Memoria SRAM

La CV cuenta con un módulo CY62167DV30 Cypress [8] de 2 MBytes de memoria SRAM externa de encapsulado TSOP-1 (Thin small-outline package) de 48 pines, alimentación de 3.3V, corriente operativa máxima de 37 mA y tiempo de lectura y escritura de 45 ns. Funciona con un bus de direcciones de 20 bits (A0-A19) y un bus de datos de 16 bits (D0-D15).

El AT91SAM7SE512 integra en su mapa de memoria 16 áreas de 256 MBytes, la primera (0) corresponde a las memorias internas, las siguientes 8 están asociadas con la interfaz de bus externa (también conocida como External Bus Interface, EBI), las siguientes 6 son indefinidas y la última (14) corresponde a los periféricos internos del microcontrolador. Por medio de la EBI y en el área 1, la memoria SRAM es accesible a partir de la dirección 0x1000 0000.

Las figuras 4.4 y 4.5 muestran el mapa de memoria del AT91SAM7SE512 donde se aprecian las diversas áreas reservadas para los periféricos internos del microcontrolador y los externos que puede manejar.

Considerando que se utiliza un bus de datos de 16 bits para la lectura y escritura de la memoria y de acuerdo a su tabla de verdad que se muestra en la tabla 4.1. Donde L es nivel lógico bajo (0V), H es nivel lógico alto (3.3V) y X significa que es indistinto su valor. Se observa que modificando solo los pines CE2 y WE pueden ejecutarse las tareas de lectura y escritura en 16 bits dejando BHE, BLE y CE1 en nivel lógico bajo (0V). Además se puede entrar en modo power down reduciendo el consumo de corriente a solo 22 μ A.

La figura 4.6 muestra el esquemático de la memoria SRAM dentro del diseño de la CV.

CE1	CE2	WE	OE	BHE	BLE	Modo
L	H	H	L	L	L	Lectura de bits 0-15
L	H	H	L	H	L	Lectura de bits 0-7
L	H	H	L	L	H	Lectura de bits 8-15
L	H	L	X	L	L	Escritura de bits 0-15
L	H	L	X	H	L	Escritura de bits 0-7
L	H	L	X	L	H	Escritura de bits 8-15
X	L	X	X	X	X	Power Down

Tabla 4.1 Tabla de verdad de la memoria SRAM

AT91SAM7SE512 Mapa de memoria

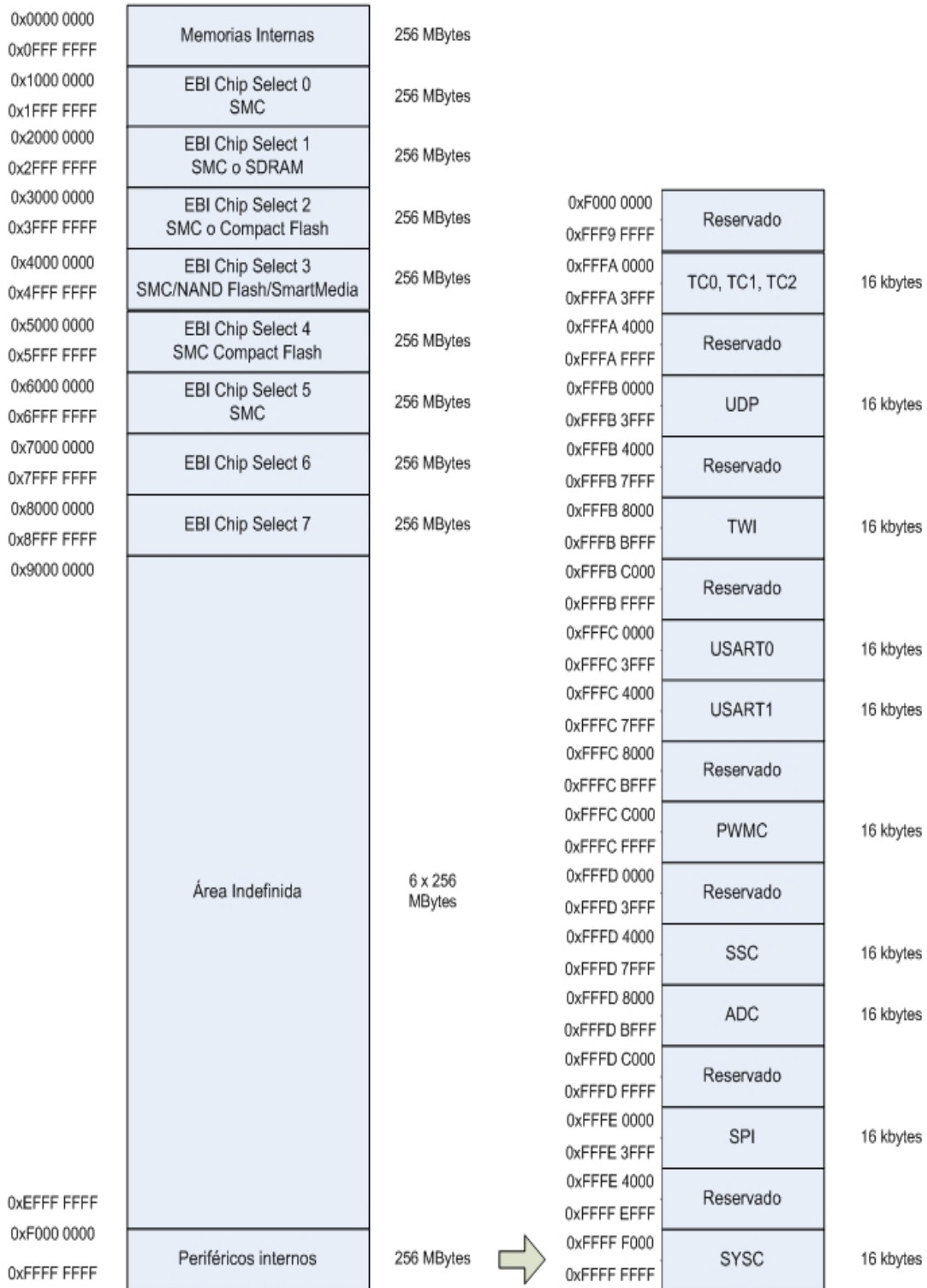


Figura 4.4 Mapa de memoria del microcontrolador AT91SAM7SE512

AT91SAM7SE512 Mapa de memoria

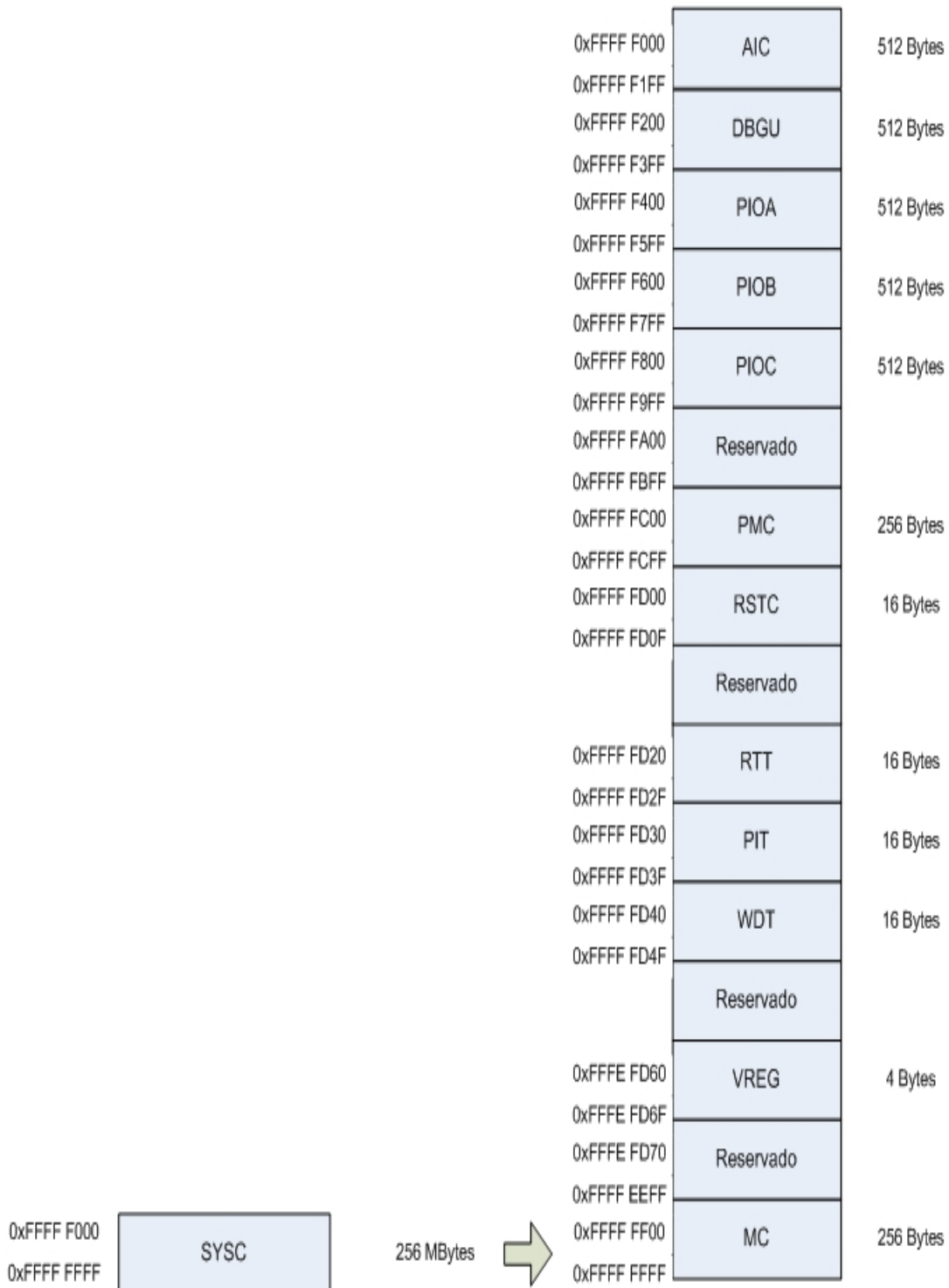


Figura 4.5 Mapa de memoria del microcontrolador AT91SAM7SE512

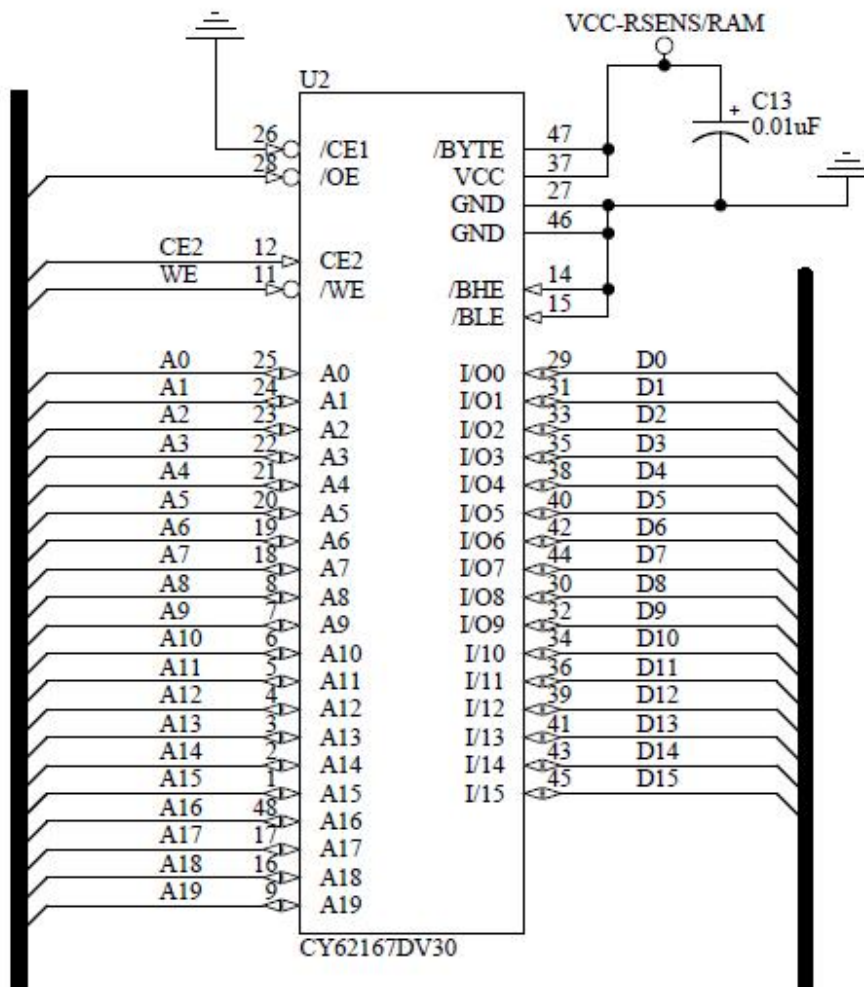


Figura 4.6 Esquemático de memoria SRAM

4.4.2 Memoria Flash SPI

La CV integra 64 MBytes de memoria flash externa, utilizada principalmente para propósitos de almacenamiento de datos, a través de 4 módulos S25FL128P Spansion [25] de 16 MBytes con fuente de alimentación de 2.7 a 3.6 V, corriente operativa de 26 mA y protocolo SPI [15] (Serial Peripheral Interface) como medio de programación.

La figura 4.7 y la tabla 4.2 muestran un esquemático de la conexión de los 4 módulos de memoria SPI de la CV y la organización de la memoria dentro de cada módulo S25FL128P, respectivamente.

Cada módulo tiene	Cada sector tiene	Cada página tiene	
16,777,216	65,536	256	Bytes
65,536	256		Páginas
256			Sectores

Tabla 4.2 Organización del arreglo de memoria.

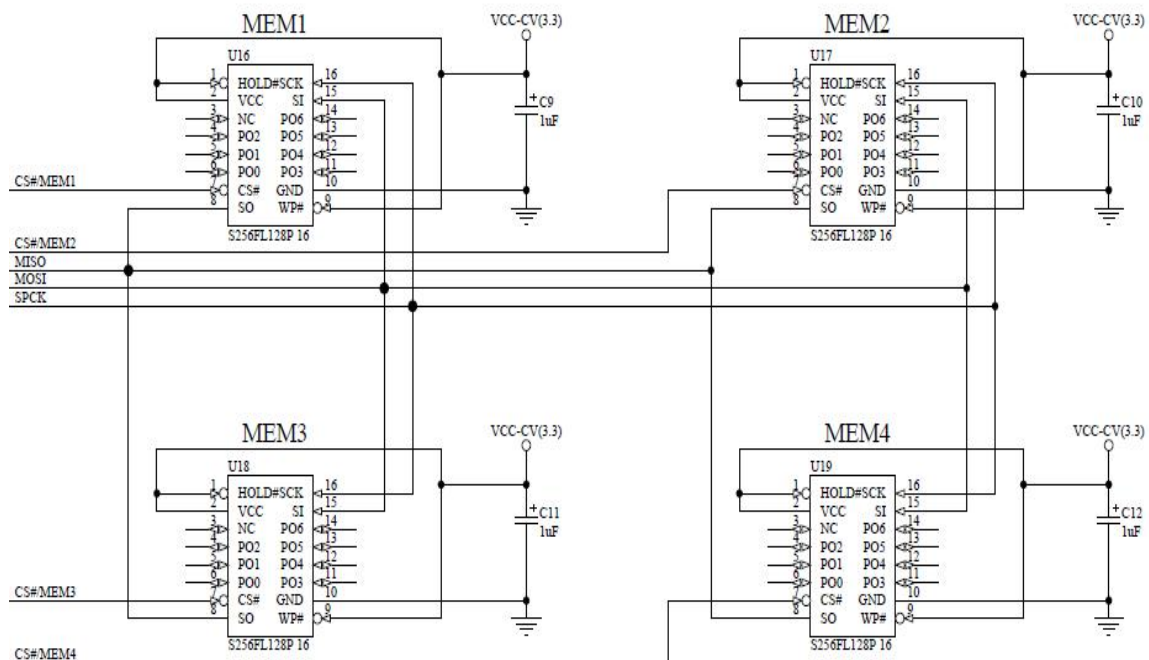


Figura 4.7 Esquemático de memoria Flash SPI

Los comandos utilizados para establecer la comunicación entre la CV y la memoria flash están descritos en la tabla 4.3, funcionan por medio del protocolo SPI con modo de operación 0,0 (polaridad y fase de reloj 0).

Operación	Comando	Descripción	Bytes de direcciones	Bytes de datos
Lectura	READ (03h)	Lee bytes de datos	3	1 a ∞
	RDID (9Fh)	Lee identificación	0	1 a 3
Control de escritura	WREN (06h)	Habilita escritura	0	0
	WRDI (04h)	Deshabilita escritura	0	0
Borrado	SE (20h)	Borra sector	3	0
	BE (C7h)	Borra volumen	0	0
Programación	PP (02h)	Programa página	3	1 a 256
Registro de estatus (SR)	RDSR (05h)	Lee del SR	0	1 a ∞
	WRSR (01h)	Escribe el SR	0	1

Tabla 4.3 Comandos de memoria SPI.

4.5 Recursos de comunicaciones

La CV incluye los siguientes sistemas de comunicación serial, a partir del microcontrolador AT91SAM7SE512.

4.5.1 Unidad USART

La CV proporciona una unidad de comunicación serial USART compatible, por medio de la programación de la longitud de datos, paridad y bits de paro, con una amplia gama de comunicaciones de tipo serial síncronas y asíncronas.

Los modos de operación soportados son:

- 5 a 9 bits full-duplex en comunicación serial asíncrona. Con el manejo del MSB (bit mas significativo) o LSB (Bit menos significativo) como primero en el envío de tramas; 1, 1.5 o 2 bits de paro; paridad par, impar o ninguna.
- 5 a 9 bits full-duplex en comunicación serial síncrona. Con el manejo del MSB o LSB como primero en el envío de tramas; 1 o 2 bits de paro; paridad par, impar o ninguna.
- RS485 sistema de comunicación bidireccional y serial que trasmite a altas velocidades. Trasmite hasta 35 Mbits/s.
- ISO7816 protocolo de interfaz con tarjetas inteligentes (smart cards).
- Modulación y desmodulación de infrarrojos IrDA que permite la conexión con transmisores-receptores IrDA.

4.5.2 Unidad TWI

La unidad TWI (Two Wire Interface), a través de un bus serial de dos líneas (reloj y datos) y con velocidades de hasta 400 kbits/s, puede ser utilizada para conectar dispositivos como: sensores de temperatura, pantalla LCD, Reloj de tiempo real, convertidores A/D y D/A, memorias EEPROM, etc. Compatible con el estándar I2C, la unidad TWI puede actuar como dispositivo maestro o esclavo.

4.5.3 Unidad DBGU

Es una unidad serial UART de dos pines que puede ser utilizada para propósitos de depuración y principalmente como medio de comunicación entre la estación terrestre y el satélite durante el cargado de software.

4.5.4 Unidad SPI y decodificador SN74HC42

La unidad SPI proporciona una comunicación serial síncrona y puede actuar como dispositivo maestro o esclavo a través de sus líneas:

- MISO (Master In Slave Out) Línea de entrada de datos.
- MOSI (Master Out Slave In) Línea de salida de datos

- SPCK (Serial Clock) Línea de reloj.
- NPCS0-NPCS3 (Peripheral Chip Select) Línea de selección de periféricos.

Se puede trabajar con los 4 modos de operación del protocolo SPI. EL modo de operación, tabla 4.4, utilizado para comunicarse con la memoria SPI S25FL128P así como el controlador CAN MCP2515 es el 1 (polaridad y fase de reloj 0).

SPI modo	Polaridad de reloj	Fase de reloj
0	0	1
1	0	0
2	1	1
3	1	0

Tabla 4.4 Modos de operación de la unidad SPI del AT91SAM7SE512

La figura 4.8 muestra un diagrama del comportamiento de las señales para el modo de operación 1 (polaridad de reloj y fase 0).

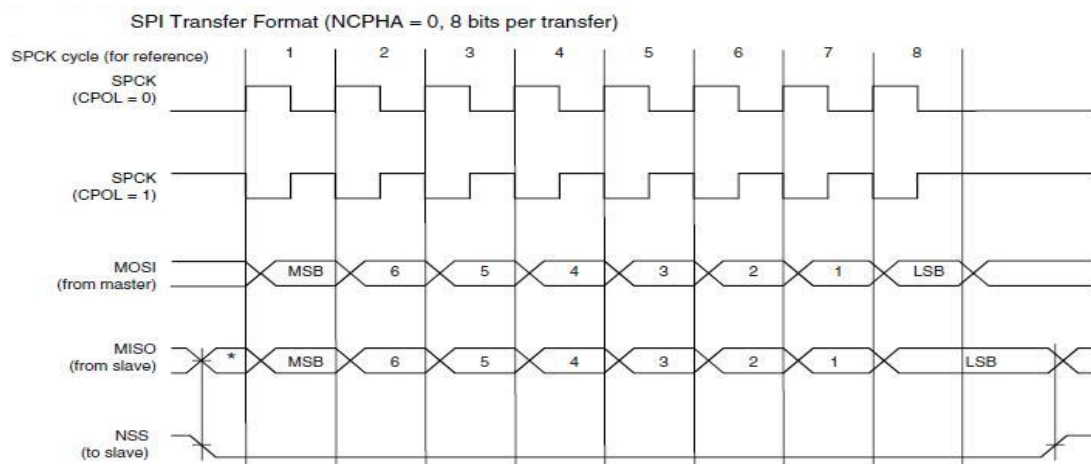


Figura 4.8 Modo de operación SPI con polaridad y fase de reloj 0.

La unidad SPI del AT91SAM7SE512 permite la conexión directa de 1 a 4 dispositivos esclavos por medio de sus líneas NPCS0-NPCS3 (chip select). Sin embargo, se requieren 5 líneas para la comunicación con los 4 módulos de memoria SPI S25FL128P y el controlador CAN MCP2515.

No obstante, el AT91SAM7SE512 permite, a través de sus registros, configurar las líneas NPCS0-NPCS3 para comunicarse hasta con 15 dispositivos. Para esto se utiliza un decodificador SN74HC42 [26] de 4 a 10 líneas por su pequeño tamaño y su compatibilidad con dispositivos de 3.3 V.

De esta manera las líneas NPCS0-NPCS3 de la unidad SPI del AT91SAM7SE512, se programan como entradas del decodificador, cuyas salidas representan las líneas CS (Chip

Select) correspondientes a los módulos de memoria y el controlador CAN. Mientras, las líneas MISO, MOSI y SPCK son las mismas tanto para los módulos de memoria como el controlador CAN.

El diagrama de conexiones del decodificador SN74HC42 y su tabla de verdad son los siguientes:

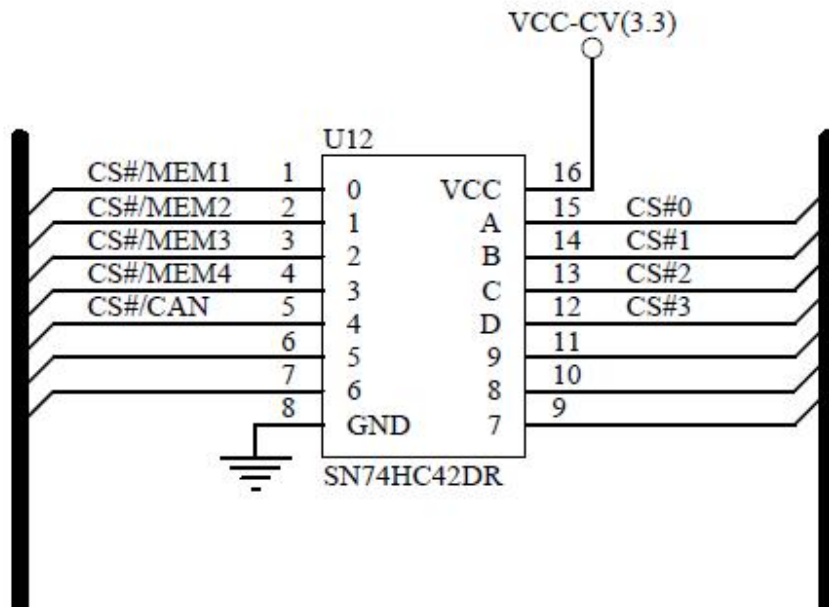


Figura 4.9 Esquemático de decodificador SN74HC42

No	Entradas D,C,B,A				Salidas (Low)
0	L	L	L	L	CS#MEM1
1	L	L	L	H	CS#MEM1
2	L	L	H	L	CS#MEM1
3	L	L	H	H	CS#MEM1
4	L	H	L	L	CS#CAN

Tabla 4.5 Tabla de verdad de decodificador SN74HC42

Donde L es estado lógico bajo (0V) y H estado alto (3.3V).

Por lo tanto, el direccionamiento de los dispositivos conectados al bus SPI se realiza por medio de la programación del AT91SAM7SE512.

4.6 Bus CAN

4.6.1 Controlador CAN MCP2515

Los subsistemas del satélite serán acondicionados para implantar como sistema de comunicaciones entre ellos el protocolo de Bus CAN por su sencilla implantación, a nivel de hardware. El bus CAN, con una tasa de transferencia de hasta 1 Mbit/seg, emplea

únicamente dos líneas para la transmisión y recepción de comandos entre los nodos conectados a dicho bus.

Para hacer posible la implantación del Bus CAN en la CV, se utiliza un controlador CAN, MCP2515 Microchip [19], que se comunica vía SPI (modo 1, polaridad y fase de reloj 0) con el AT91SAM7SE512. Dicho controlador soporta la versión V2.0B del protocolo CAN con una tasa de transferencia de 1Mbit/s y maneja mensajes tipo: estándar, extendido y remotos. Tiene 2 buffers (RXBn) para recibir mensajes y 3 buffers (TXBn) de transmisión, 2 mascarar y 6 filtros, ambos, para la validación de mensajes. Cuenta con un sistema de prioridades, sistema de interrupciones, detección de errores. Tiene 5 modos de operación, siendo los importantes: configuración, normal y reposo (sleep mode).

Además de las líneas de comunicación SPI (SO, SI, SCK, CS), las líneas del MCP2515 utilizadas para su implantación son:

- INT. Genera una interrupción cuando: un buffer TXBn se encuentra disponible para recibir un mensaje del AT91SAM7SE512, un mensaje ha sido recibido en un buffer (RXBn), un error de transmisión o recepción ha ocurrido, etc.
- TXnRTS. Se solicita la transmisión de un mensaje contenido en uno de los 3 buffers de transmisión (TXBn).

El envío y recepción de mensajes por medio del bus CAN se realiza conectando el controlador CAN con un emisor-receptor de la familia MCP2551 Microchip [20]. Esto garantiza la compatibilidad entre el controlador y el emisor-receptor del bus CAN y permite a la CV actuar como un nodo de este sistema de comunicaciones.

La tabla 4.6 muestra los comandos utilizados para establecer la comunicación con el controlador CAN MCP2515.

Instrucción	Formato de instrucción	Descripción
RESET	1100 0000	Reinicia los registros internos a su estado inicial y activa el modo configuración del dispositivo.
Leer	0000 0011	Lee un dato de registro de acuerdo a la dirección solicitada.
Leer RX Buffer	1001 0nm0	De acuerdo a los valores de “n,m” se puede leer un buffer de recepción del MCP2515
Escribir	0000 0010	Escribe un dato en el registro que apunta la dirección.
Escribir TX Buffer	0100 0abc	De acuerdo a los valores de “a,b,c” se puede escribir un buffer de transmisión del MCP2515
Leer Estatus	1010 0000	Lee los estatus de varios bits de transmisión y recepción.

Tabla 4.6 Comandos SPI del controlador de bus CAN MCP2515

Instrucción	Formato de instrucción	Descripción
RX Estatus	1011 0000	Indica que filtro aceptó el mensaje así como que tipo de mensaje fue.
Modificar Bit	0000 0101	Permite modificar bits individuales de un registro. NOTA No todos los registros permiten este comando.

Tabla 4.6 Comandos SPI del controlador de bus CAN MCP2515 (continuación)

4.6.2 Transmisor-Receptor CAN MCP2551

El transmisor y receptor MCP2551 proporciona la interfaz entre las líneas del Bus CAN y el controlador CAN MCP2515. Soporta operación del Bus a 1 Mbits/s e incluye protecciones térmicas, contra apagones, cortos circuito y transiciones de voltajes altos. También permite disminuir efectos de interferencias electromagnéticas y trabaja ante condiciones adversas de ruido.

El MCP2551 funciona con una fuente de 5V, sin embargo sus líneas RXD y TXD son compatibles con el controlador CAN MCP2515 que opera a 3.3 V.

La figura 4.10 muestra el diagrama de conexiones del controlador y el transmisor-receptor de bus CAN.

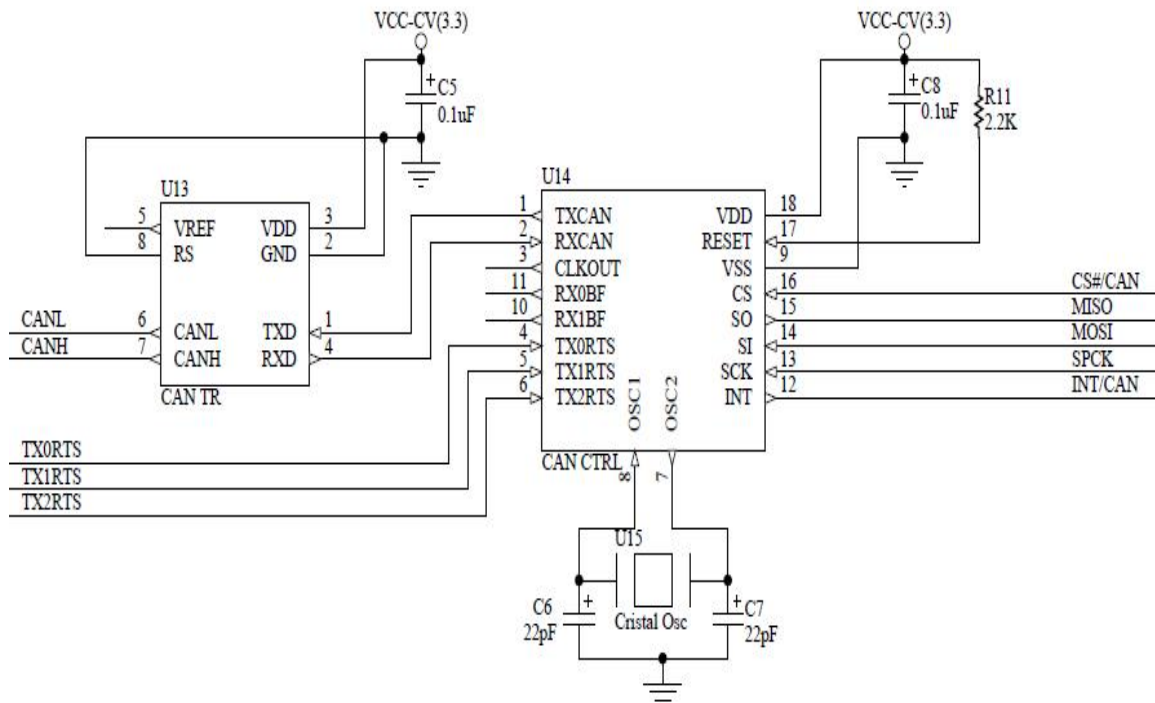


Figura 4.10 Esquemático de nodo de Bus CAN.

4.7 Sensores de temperatura

La tarjeta de la CV cuenta con 5 sensores de temperatura de tipo 1-Wire que permiten conocer su temperatura alrededor de su superficie. Los sensores 1-Wire utilizan una sola línea para la recepción de comandos y el envío de datos a la CV. La figura 4.11 muestra la conexión de los sensores.

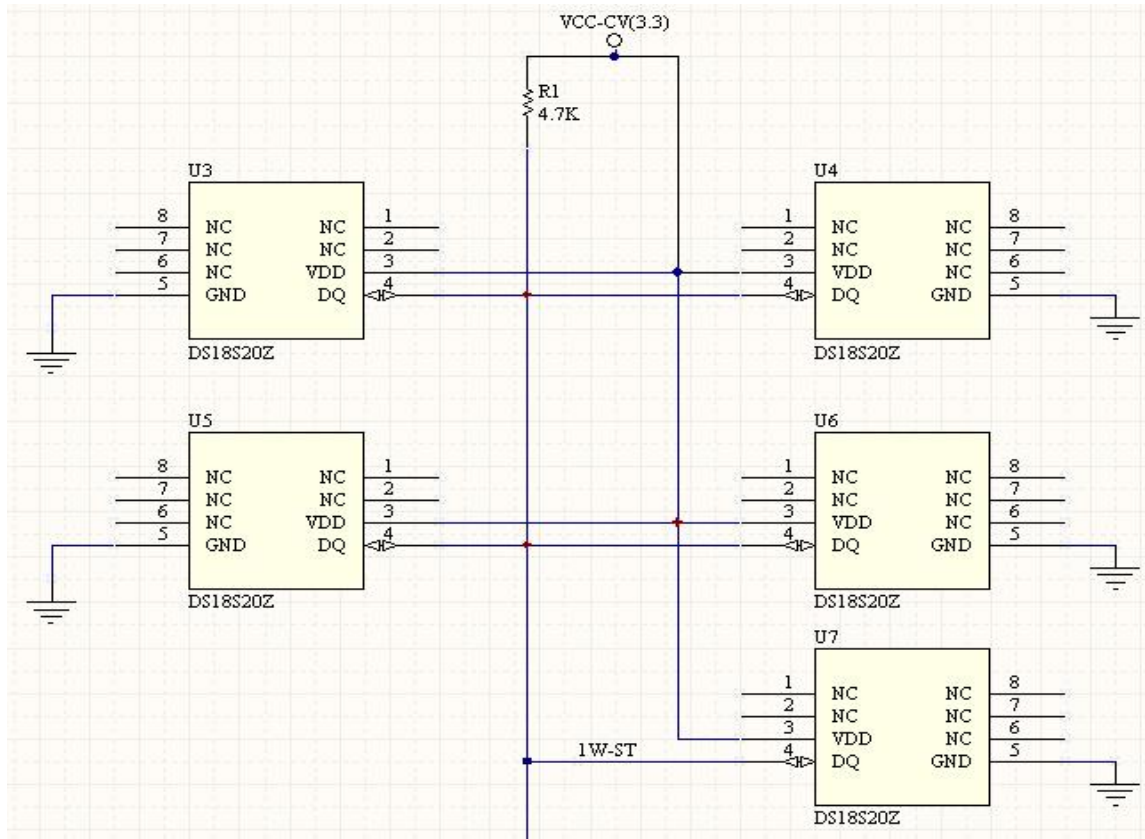


Figura 4.11 Esquemático de sensores de temperatura 1-Wire

4.7.1 Sensor DS18S20

De marca Maxim, el sensor DS18S20 [16] a través del protocolo 1-Wire permite tomar lecturas de temperatura en el rango de -55°C a 125°C con una precisión de 0.5°C y resolución de 9 bits. Tal rango es necesario para equipos que pretendan emplearse en órbita terrestre, si bien esta computadora de vuelo se utilizará en salones de clase como parte de SATEDU, se persigue que el mismo diseño se pueda emplear con modificaciones mínimas en satélites pequeños que operen en el espacio, en donde existe un ambiente agresivo en términos de temperatura (arriba de 100°C y por debajo de -40°C), vacío y radiación.

Con el protocolo 1-Wire un dispositivo maestro puede manejar varios dispositivos esclavos a través de una sola línea que actúa como bus de entrada y salida de datos tanto para el dispositivo maestro como para el esclavo.

En el caso de la CV el dispositivo maestro es el microcontrolador AT91SAM7SE512 y los dispositivos esclavos son los sensores de temperatura DS18S20.

4.7.2 ROM Code

La manera de identificar un dispositivo 1-Wire conectado es por medio de un código de identificación que consiste en un número serial único de 64 bits y de nombre ROM code. El ROM code, figura 4.12, se encuentra almacenado en la memoria ROM interna de los sensores y tiene la estructura siguiente:

Los 8 bits menos significativos representan la familia del dispositivo, 10h en el caso del sensor DS18S20.

Los siguientes 48 bits son un número serial único para cada sensor.

Los 8 bits más significativos representan el CRC (cyclic redundancy check) calculado por medio de los 56 bits menos significativos. El CRC se utiliza para verificar si el ROM code ha sido leído correctamente.

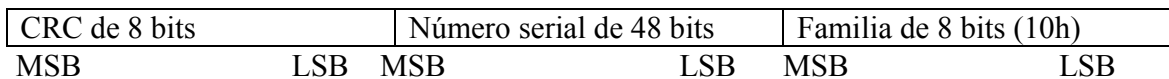


Figura 4.12 ROM code de 64 bits.

Para verificar el ROM code se utiliza el siguiente registro de corrimiento:

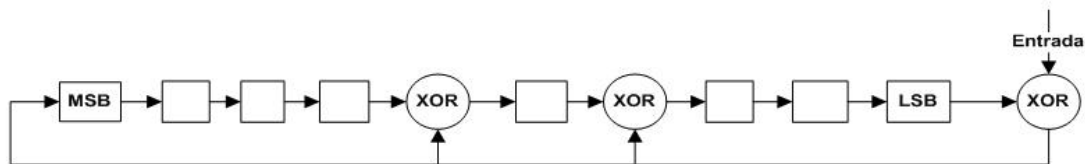


Figura 4.13 Registro de corrimiento para verificar el ROM code.

Comenzando con un valor de cero en los bits del registro de corrimiento, se introduce el ROM code empezando por el bit menos significativo. Si después de haber introducido el ROM code completo, los bits del registro de corrimiento siguen con un valor de cero entonces el ROM code es correcto.

4.7.3 Memoria interna

El sensor de temperatura DS18S20 cuenta con memoria SRAM interna de 9 bytes ordenada de la siguiente manera.

Byte	Nombre
0	T LSB
1	T MSB
2	TH
3	TL
4	Reservado
5	Reservado
6	COUNT REMAIN
7	COUNT PER °C
8	CRC

Tabla 4.7 Memoria SRAM interna del DS18S20.

El byte 0 y el 1, solo de lectura, almacenan el byte menos significativo y el byte más significativo, respectivamente, de la temperatura medida. El byte 2 y el 3 almacenan los límites superior e inferior, respectivamente, de la alarma del sensor DS18S20. El byte 4 y el 5 no se utilizan y en caso de leerlos tiene un valor de FFh. El byte 6 y el 7 permiten una mayor resolución en la lectura de temperaturas, mayor a 9 bits. El byte 8 es el CRC de los anteriores y permite al dispositivo maestro verificar si la lectura de la memoria SRAM fue correcta.

El DS18S20 cuenta con 2 bytes de memoria EEPROM que sirven para almacenar el límite superior TH e inferior TL de la alarma, evitando la programación de los límites cada vez que se enciende el sensor. Los límites determinan el rango de temperaturas aceptables, una temperatura mayor a la almacenada en TH o una menor que la de TL provoca una condición de alarma que activa una bandera dentro del DS18S20, la cual se puede comprobar su estado por medio de un comando.

4.7.4 Lectura de temperaturas

La lectura se obtiene en unidades °C y se almacena en los dos primeros bytes de la memoria SRAM del DS18S20. El byte más significativo es el byte de signo que vale 0 para temperaturas positivas y 1 para temperaturas negativas. El byte menos significativo almacena la temperatura obtenida del sensor. La figura 4.14 muestra los registros T LSB y T MSB que almacenan el valor de la temperatura y el valor del signo de temperatura, respectivamente.

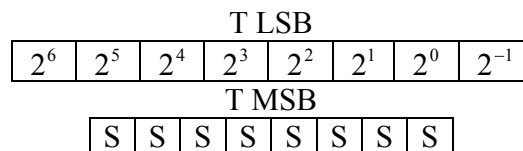


Figura 4.14 Registros T LSB y T MSB de la memoria interna.

Si la variable S vale 0 la temperatura medida es positiva y se obtiene de la fórmula:

$$T = \frac{TLSB}{2} [^{\circ}C]$$

Si la variable S vale 1 la temperatura medida es negativa y se obtiene de la fórmula:

$$T = -\frac{256 - TLSB}{2} [^{\circ}C]$$

En la tabla 4.8 se muestran equivalencias de la temperatura en formato decimal, binario y hexadecimal.

Temperatura	Binario	Hexadecimal
85 °C	0000 0000 1010 1010	AA
25 °C	0000 0000 0011 0010	32
0.5 °C	0000 0000 0000 0001	01
0 °C	0000 0000 0000 0000	00
- 0.5 °C	1111 1111 1111 1111	FFFF
- 25 °C	1111 1111 1100 1110	FFCE
- 55 °C	1111 1111 1001 0010	FF92

Tabla 4.8 Equivalencias de temperatura en decimal, binario y hexadecimal.

4.8 Sistemas de protección efecto latch-up

Cuando un dispositivo semiconductor, como un circuito integrado, se expone a la radiación del espacio, en función de su dosis de carga eléctrica que acumule, podrá experimentar un incremento en su consumo de corriente eléctrica. Al presentarse tal efecto, llamado Latch-Up, y de no detenerlo a tiempo, provocará que el dispositivo deje de funcionar permanentemente.

Una manera de detener el efecto latch-up es desenergizando el dispositivo. En la CV la electrónica necesaria para detectar el efecto latch-up consiste de un sensor de corriente MAX4071 con salida de voltaje y un comparador de voltaje LM6511 que genera una señal eléctrica para que la tarjeta de potencia desenergice el dispositivo que presenta el efecto latch-up.

En la CV se cuenta con protección para la memoria RAM y el microcontrolador AT91SAM7SE512 pues son los componentes de mayor escala de integración y por tanto los más susceptibles de presentar el efecto latch-up ya que pueden estar trabajando por largos periodos de tiempo. La lectura de la corriente consumida se realiza por medio del convertidor analógico digital A/D del AT91SAM7SE512.

4.8.1 Sensor de corriente MAX4071

El sensor MAX4071 [17], figura 4.15, permite medir la cantidad de corriente que circula por una carga. Para esto, la carga se conecta en serie con una resistencia de sensado (Rsense) haciendo que la corriente que circula por dicha resistencia sea la misma que circula por la carga. La fuente que alimenta Rsense y la carga es la misma.

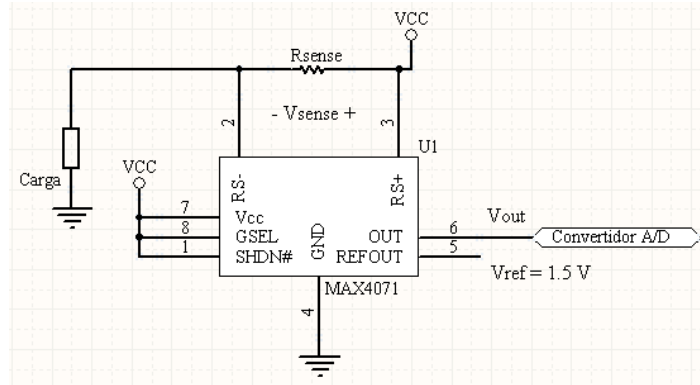


Figura 4.15 Sensor de corriente MAX4071

Como salidas se tienen: un voltaje de referencia en REFOUT de 1.5 V y un voltaje medido con respecto a tierra, V_{out} , que es directamente proporcional a la corriente que fluye por la resistencia R_{sense} y esta dado por:

$$V_{out} = G_{sel} R_{sense} I_{Rsense} + 1.5 \text{ [V]} \text{-----(1)}$$

Donde G_{sel} representa una ganancia que es igual a 100 V/V, cuando el pin GSEL es conectado a la fuente de 3.3 V.

En la tabla 4.9 se muestran los valores recomendados por el fabricante para la selección de R_{sense} de acuerdo al valor de corriente que se desea monitorear y para el caso de una ganancia de 100V/V. Al seleccionar el valor de R_{sense} se debe elegir el mínimo valor posible para que la caída de voltaje en R_{sense} sea mínima. La resistencia R_{sense} debe tener una inductancia pequeña si la corriente a monitorear varía mucho.

Corriente máxima [A]	R_{sense} [mΩ]	V_{sense} [mV]	$V_{out}-V_{ref}$ [V]
10	5	50	5
5	10	50	5
2.5	20	50	5
0.5	100	50	5
0.05	1000	50	5

Tabla 4.9 Valores de R_{sense} del sensor MAX4071

4.8.2 Comparador LM6511

Con una fuente de alimentación de 2.7 a 3.6 V y tiempo de respuesta de 180 ns, el LM6511 [22] funciona como comparador de un voltaje de referencia y el voltaje de salida del sensor de corriente MAX4071. La figura 4.16 muestra el diagrama empleado para su implantación en la CV.

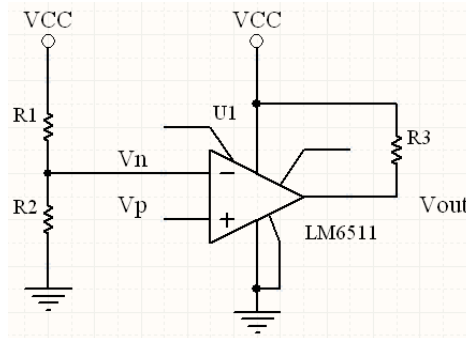


Figura 4.16 Comparador LM6511

El voltaje V_n esta dado por la siguiente ecuación:

$$V_n = R_2 \frac{V_{cc}}{R_1 + R_2} [V]$$

Teniendo en cuenta que V_{out} es el producto de una ganancia A_v multiplicada por la diferencia de V_p y V_n , es decir:

$$V_{out} = A_v (V_p - V_n) \text{-----(A)}$$

Se puede observar que si V_p es menor a V_n se obtiene una ganancia negativa que, debido a la configuración del comparador, da un V_{out} igual a 0.

Para tener como salida V_{cc} se necesita que V_p sea mayor a V_n . Ahora si aplicamos un V_{ref} , voltaje de referencia, a V_p y calculamos el valor de R_1 y R_2 para que V_{ref} sea igual a V_n , entonces tenemos:

$$V_{ref} = V_n$$

$$V_{ref} = R_2 \frac{V_{cc}}{R_1 + R_2}$$

$$V_{ref} (R_1 + R_2) = R_2 V_{cc}$$

$$R_2 (V_{ref} - V_{cc}) = -V_{ref} R_1$$

$$R_1 = R_2 \left(\frac{V_{cc}}{V_{ref}} - 1 \right) [\Omega] \text{-----(2)}$$

De esta manera dado un valor de R_2 se obtiene el de R_1 . Si V_p es menor al voltaje de referencia V_{ref} entonces tenemos a la salida un voltaje igual a 0 y si V_p es mayor al voltaje V_{ref} tenemos una salida igual a V_{cc} .

4.8.3 Protección de efecto Latch-up para el AT91SAM7SE512

Siendo que la corriente máxima de trabajo del AT91SAM7SE512 es de 80 mA. De la tabla 4.9 se observa que el valor de R_{sense} más cercano es el de 100 m Ω con la capacidad de sensar hasta 500 mA. De la ecuación (1) el valor V_{out} para una corriente de 70 mA es:

$$V_{out} = (100)(0.1)(0.07) + 1.5$$

$$V_{out} = 2.2 \text{ [V]}$$

Sustituyendo en la ecuación (2) $V_{ref} = 2.2 \text{ V}$ y $R_2 = 2.7 \text{ k}\Omega$ tenemos que R_1 vale:

$$R_1 = 2.7k (3.3/2.2 - 1)$$

$$R_1 = 1.33 \text{ [k}\Omega\text{]}$$

La protección contra efecto latch-up del AT91SAM7SE512 queda implantada de la siguiente forma:

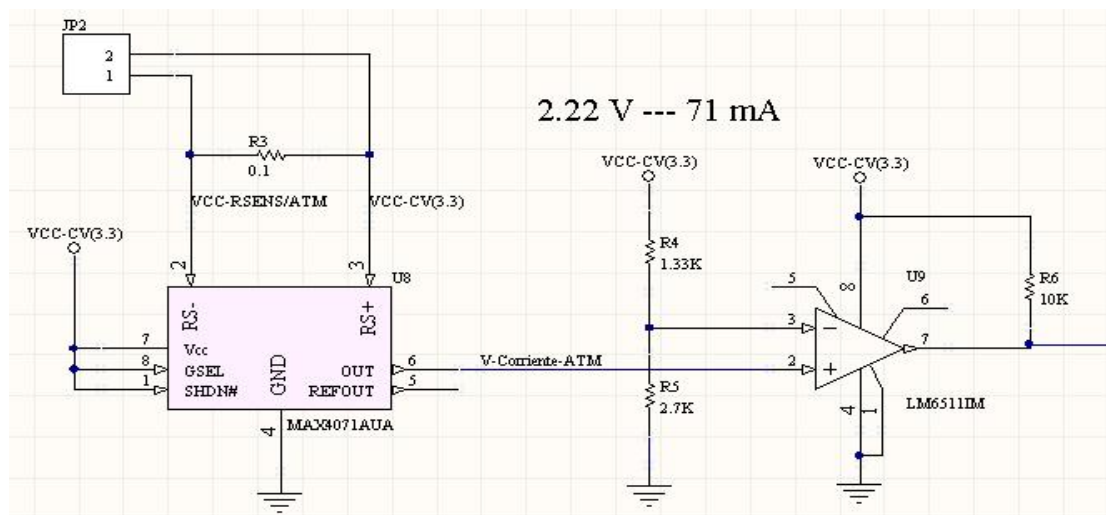


Figura 4.17 Protección efecto latch-up AT91SAM7SE512

Donde $V_{cc-Rsens/ATM}$ se conecta a la carga, microcontrolador AT91SAM7SE512 y $V_{-Corriente-ATM}$ se envía al convertidor A/D para poder tomar lectura de la corriente consumida.

4.8.4 Protección de efecto Latch-up para la memoria RAM

La memoria RAM CY62167DV30 trabaja con una corriente máxima de 37 mA. De la tabla 4.9 se observa que la resistencia de sensado más conveniente es la de 100 m Ω teniendo la capacidad de sensar hasta 500 mA. La de 1 Ω implica un V_{out} mayor de 3.3 V, voltaje de saturación, que impide el uso correcto del comparador.

De la ecuación (1) el valor V_{out} para una corriente de 30 mA es:

$$V_{out} = (100)(0.1)(0.03) + 1.5$$

$$V_{out} = 1.8 \text{ [V]}$$

Sustituyendo en la ecuación (2) $V_{ref} = 2.2 \text{ V}$ y $R_2 = 2.7 \text{ k}\Omega$ tenemos que R_1 vale:

$$R_1 = 2.7k (3.3/1.8 - 1)$$

$$R_1 = 2.26 \text{ [k}\Omega\text{]}$$

La protección contra efecto latch-up de la memoria CY62167DV30 queda implantada de la siguiente forma:

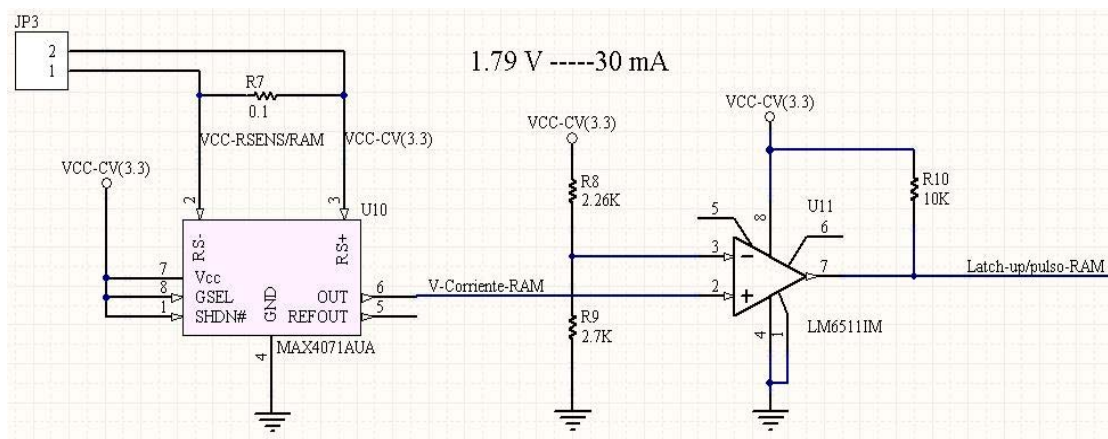


Figura 4.18 Protección efecto latch-up SRAM

Donde $V_{cc-Rsens/RAM}$ se conecta a la carga, la memoria CY62167DV30 y $V-Corriente-RAM$ se envía al convertidor A/D para tomar la lectura de la corriente consumida.

4.8.5 Convertidor A/D

Basado en aproximaciones sucesivas, el convertidor analógico-digital (CA/D) de 10 bits cuenta con 8 líneas analógicas de entrada. Soporta resoluciones de 8 y 10 bits y los resultados de conversión son reportados en un registro común para todos los canales así como en un registro dedicado a cada canal.

La CV utiliza dos líneas del CA/D conectadas a los nodos $V-Corriente-ATM$ y $V-Corriente-RAM$ para medir el voltaje de salida del sensor de corriente y conocer en cualquier momento la corriente que esta consumiendo el microcontrolador y la memoria SRAM.

Las líneas AD4-AD7 son líneas dedicadas del convertidor mientras las líneas AD0-AD3 son configurables como líneas de entrada analógica para uso del CA/D o pueden ser

configuradas como líneas digitales I/O de uso general por medio del controlador de entradas y salidas I/O paralelas (PIO).

Los rangos de voltajes a la entrada de las líneas del convertidor son de 0 [V] al voltaje de referencia ADVREF (3.3V).

Capítulo 5. Modo de carga de software

5.1 Introducción

La CV de SATEDU cuenta con un microcontrolador (PIC16F876) que establece la comunicación entre el subsistema de comunicaciones y su unidad de procesamiento SAB80C166. Tal microcontrolador recibe comandos de la tarjeta de comunicaciones, provenientes de la estación terrena, principalmente para efectuar un reset y poner en modo de carga el SAB80C166.

El microcontrolador SAB80C166 en modo de carga recibe solo 32 bytes de datos por lo cual para cargar el software de aplicación final de SATEDU fue necesario crear un software de cargado de segundo nivel que habilita y a su vez permite el cargado de software en la memoria SRAM externa.

Por lo anterior, el modo de carga de software influyó en la elección de la nueva unidad de procesamiento. Se buscó un microcontrolador que proporcionara herramientas para cargar software en su memoria interna o externa sin necesitar de varios elementos externos.

5.2 AT91SAM Boot Program

El Boot Program [5] integra aplicaciones que permiten la carga de software en las distintas memorias del microcontrolador AT91SAM7SE512 ya sea por medio de una interfaz USB o por medio de la unidad serial DBGU conectada a una PC. El modo de funcionamiento de Boot Program, utilizando la unidad serial DBGU, es el siguiente.

El Boot Program comienza inicializando la unidad DBGU a 115200 Baudios, 8bits, 1 bit de paro y sin paridad y comprueba el auto-baudaje de acuerdo a los pasos descritos en el diagrama de flujo de la figura 5.1.

Una vez que realiza la comprobación del baudaje y resulta satisfactoria, espera de forma continua comandos provenientes del software SAM-BA (Boot Asistant) o de un software particular a través del protocolo Xmodem.

5.2.1 Limitaciones de hardware y software del Boot Program

El Boot Program se copia de la memoria ROM a la SRAM del AT91SAM7SE512 por lo que ocupa una cantidad de memoria para variables y stacks (pilas) haciendo disponible para el usuario un espacio de 24576 bytes. El área del usuario se extiende de la dirección 0x202000 a 0x208000. Los pines utilizados, para el caso de la unidad serial DBGU, son: DRXD y DTXD.

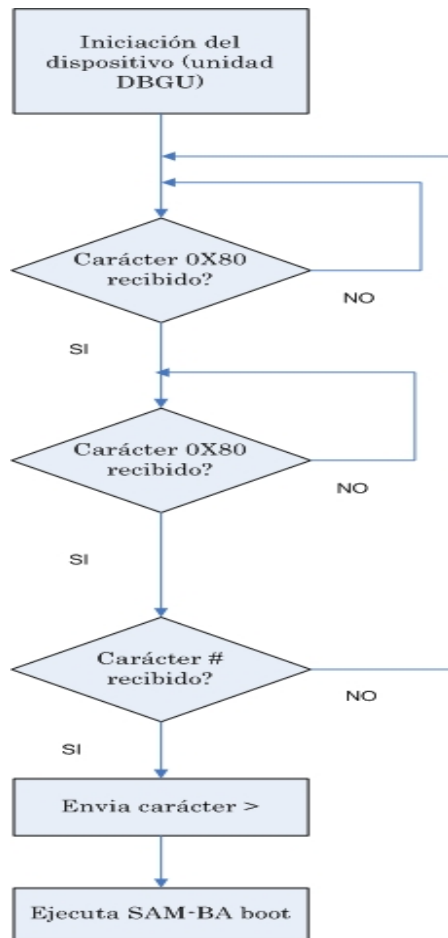


Figura 5.1 Diagrama de flujo de detección de Baudaje.

5.2.2 Comandos de AT91SAM Boot Program

Los comandos que soporta el Boot Program provenientes del software de la estación terrena, tabla 5.1, son los siguientes.

Comando	Acción	Argumentos	Ejemplo
O	Escribe un byte	Dirección, valor#	O 200001,CA#
o	Lee un byte	Dirección#	o 200001,#
H	Escribe media palabra	Dirección, valor#	H 200002,CAFE#
h	Lee media palabra	Dirección#	h 200002,#
W	Escribe una palabra	Dirección, valor#	W 200000,CAFEDECA#
w	Lee una palabra	Dirección#	w 200000,#
S	Envía un archivo	Dirección#	S 200000,#
R	Lee un archivo	Dirección, numero de bytes#	R 200000,1234#
G	Go	Dirección#	G 200200#
V	Despliega versión	Sin argumento	V#

Tabla 5.1 Comando del Boot Program

Comandos de escritura (O, H, W)

Se pueden escribir en el microcontrolador ya sean 8 bits (1 byte), 16 bits (media palabra) o 32 bits (una palabra). Los argumentos son la dirección y el valor en hexadecimal que se desea escribir. La salida es el carácter '>'.

Comandos de lectura (o, h, w)

Al igual que los de escritura, pueden leerse 8bits, 16 bits o 32 bits del microcontrolador. El único argumento es la dirección a leer en hexadecimal. La salida son los 8 bits, 16 bits o 32 bits en formato hexadecimal seguidos del carácter '>'.

Comando de envío de archivo (S)

Se envía un archivo a una dirección del microcontrolador que se especifica en hexadecimal como argumento del comando. La salida es el carácter '>'.

Comando de lectura de archivo (R)

Se recibe un archivo correspondiente a la lectura a partir de una dirección de memoria y en base al número de bytes que se desean leer. Tanto la dirección como el número de bytes se especifican en hexadecimal. La salida es el carácter '>'.

Comando GO (G)

Permite pasar a la dirección de su argumento, en hexadecimal, para ejecutar el código que contiene el microcontrolador a partir de tal dirección. La salida es el carácter '>'.

Los comandos de envío y recepción de archivos utilizan el protocolo Xmodem y cualquier terminal que cuente con dicho protocolo, que es fácil de implantar, puede enviar el archivo de aplicación al microcontrolador de la CV. El tamaño del archivo binario a enviar depende del tamaño de la memoria SRAM interna con que cuenta el microcontrolador. En todos los casos el archivo binario debe ser de menor tamaño que la memoria SRAM pues el protocolo Xmodem consume parte de la memoria para su ejecución.

5.3 Protocolo Xmodem

El protocolo Xmodem [33] empleado soporta paquetes de 128 Bytes de longitud. Utiliza dos caracteres CRC-16 para garantizar la detección de un error de bit.

El protocolo Xmodem con CRC es preciso pues proporciona al transmisor y receptor reportes de transmisión exitosa. Cada paquete o bloque de transferencia tiene la siguiente estructura:

<SOH><blk #><255-blk #><--128 data bytes--><checksum>

Donde.

<SOH> es 1 hex. <blk#> es un número binario que comienza en 1, incrementa en 1 y cambia de FFh a 00h. <255-blk#> es el complemento a 1 de blk#. <128 data bytes> son los bytes de datos a transmitir y en caso de que los datos sean menos de 128 bytes, se utilizan caracteres de relleno para completar el paquete a 128 bytes. <checksum> son 2 bytes CRC.

Además en el protocolo Xmodem se utilizan los caracteres: <eot> (04h), <ack> (05h), <nak> (15h) y <can> (18H).

La figura 5.2 muestra un ejemplo de transmisión por medio del protocolo Xmodem. Donde el Host es la PC y el dispositivo receptor es la tarjeta de la CV. La transmisión comienza cuando el receptor envía un carácter (NAK) al transmisor indicando que está listo para recibir paquetes de datos. El transmisor envía al receptor los paquetes de datos comenzando por SOH y finalizando con los bytes CRC. El receptor comprueba que el paquete se recibió correctamente, calcula el checksum a partir de los datos recibidos y lo compara con los CRC recibidos. En caso de que la recepción del paquete sea correcta, el receptor envía un carácter ACK al transmisor y en caso contrario envía un carácter NAK que puede solicitar la retransmisión del paquete corrupto. Así mismo se puede enviar el carácter CAN para indicar que se ha producido un error irreparable y se aborte la transmisión. En el caso de una correcta transmisión de un paquete, el transmisor incrementa el contador <blk#> y envía el siguiente paquete de datos. La transmisión se termina con el envío de un carácter EOT.

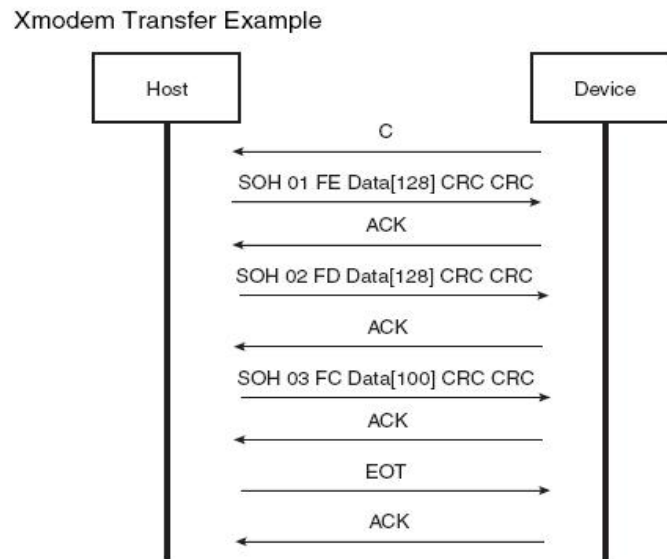


Figura 5.2 Transmisión a través del protocolo Xmodem.

5.4 SAM-BA (Boot Assistant)

La tarjeta que establece la comunicación entre la estación terrestre y la tarjeta de comunicaciones del satélite así como el software para cargar de forma inalámbrica aplicaciones en la CV, serán desarrollados en conjunto con el encargado del subsistema de comunicaciones.

No obstante una vez desarrollado el circuito impreso, en modo paralelo se pueden efectuar pruebas de software por medio de la herramienta SAM-BA [4] de la empresa Atmel.

SAM-BA es un software basado en una biblioteca de enlace dinámico DLL (common dynamic linked library) AT91Boot_DLL.dll de dominio público que permite elaborar aplicaciones particulares para la comunicación entre el Boot Program del microcontrolador y el software de estación terrena. Establece la comunicación entre cualquier PC que cuente con un puerto serial RS232 y tenga como sistema operativo Windows 2000 o Windows XP donde pueda ser instalado.

Cabe mencionar que hay herramientas disponibles, además de SAM-BA, desarrolladas tanto para ambientes Windows o Linux y son de distribución y desarrollo libre (Open Source) lo cual no restringe el uso de sistemas operativos en computadoras.

SAM-BA puede ejecutarse desde una terminal de comandos o a través de una interface gráfica de usuario (GUI) que funciona de la manera siguiente:

- Se conectan las líneas DRXD y DTXD de la CV a una protoboard por medio de un cable conectado en el conector izquierdo o el conector de ET (mini-USB).
- En la protoboard se monta un emisor-transmisor serial ADM3202ARN que opera con voltajes de alimentación de 3.3 V a una tasa de 460 kbps suficiente para la unidad DBGU. El emisor-transmisor ADM3202ARN se comunica a la PC por medio de un puerto serial DB-9 de acuerdo al esquemático de la figura 5.3.
- SAM-BA utiliza la función Scan de AT91Boot_DLL.dll para determinar los dispositivos que se encuentran conectados a la PC.
- SAM-BA despliega una ventana, figura 5.4, donde se selecciona el dispositivo conectado a la PC y su interfaz de comunicación, “COMX” para el caso de la unidad serial DBGU.

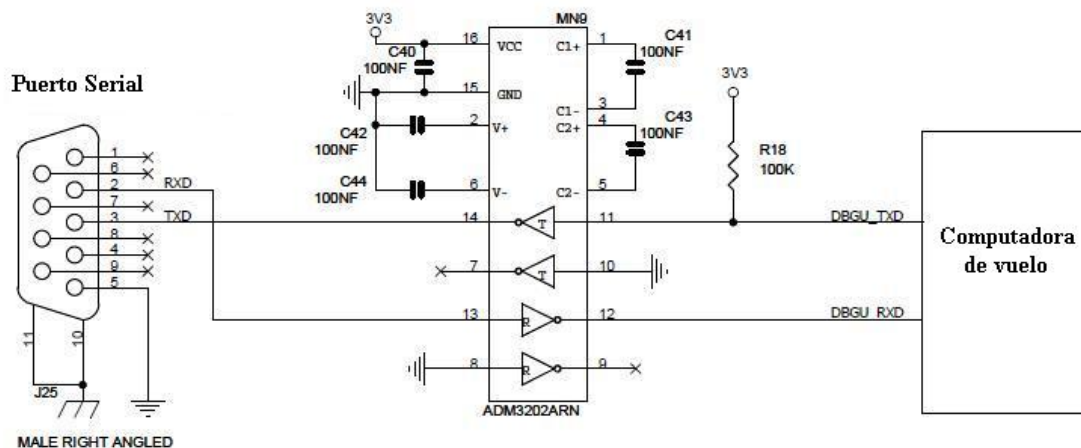


Figura 5.3 Esquemático de comunicación entre la CV y la PC.



Figura 5.4 SAM-BA selección de unidad y medio de comunicación.

- SAM-BA despliega, figura 5.5, un área de memoria máxima de 1024 Bytes en los formatos ASCII, 8 bits (Byte), 16 bits (media palabra) o 32 bits (palabra). La memoria, SRAM o Flash, puede ser leída o editada.
- Para la lectura o despliegue de memoria es necesario especificar la dirección de inicio, el tamaño en bytes (menor a 1024) y el formato para finalmente presionar el botón Refresh y actualizar el área de desplegado de memoria.
- La edición del contenido de memoria, disponible solo para SRAM, se realiza seleccionando la dirección de memoria donde se desea modificar su contenido y anotando el nuevo valor en la ventana emergente, figura 5.6, presionado por ultimo el botón OK.
- SAM-BA ofrece un área, figura 5.7, donde puede seleccionarse la memoria SRAM o la memoria Flash interna del microcontrolador y subir un programa o descargar un programa a través de un archivo binario.

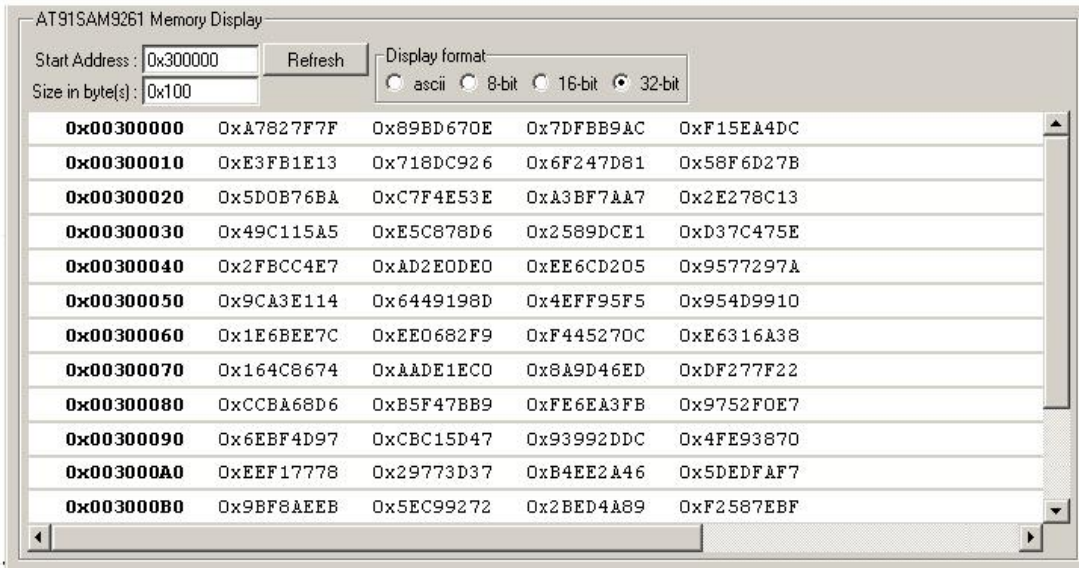


Figura 5.5 SAM-BA Despliegue de memoria.

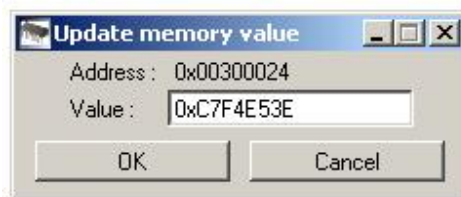


Figura 5.6 SAM-BA Edición de memoria.

- Una vez seleccionado el archivo binario a subir al microcontrolador, se habilita la memoria SRAM o Flash por medio de scripts, se selecciona la dirección de memoria donde se comenzará a subir el programa y se presiona el botón de envío (Send File)

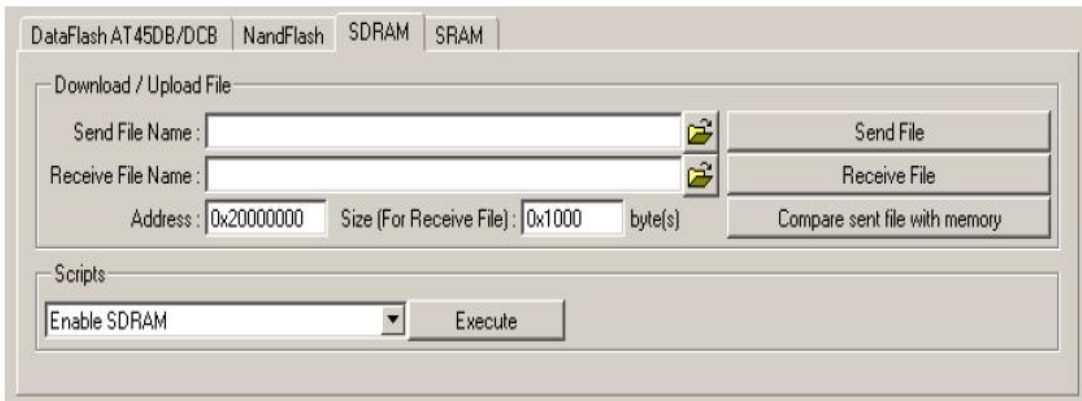


Figura 5.7 SAM-BA Área de carga o descarga de memoria.

- De manera similar se puede recibir un archivo con la variante de que debe especificarse el tamaño de bytes que se desean leer de la memoria del microcontrolador. Además puede compararse un área de memoria a partir de una dirección especificada y en base al tamaño de un archivo especificado en el campo Send File. Una ventana aparece indicando si el archivo y el contenido de la memoria son idénticos o no. La comparación permite verificar si un archivo se subió correctamente al microcontrolador.

Los comandos y sus respuestas aparecerán en un área TLC (Tool Command Language), figura 5.8, donde pueden ejecutarse en modo terminal comandos específicos que proporcionan otro método para ejecutar las tareas previamente mencionadas. Para más información puede consultarse “SAM-BA Boot Asistant” en el apartado de la Bibliografía.

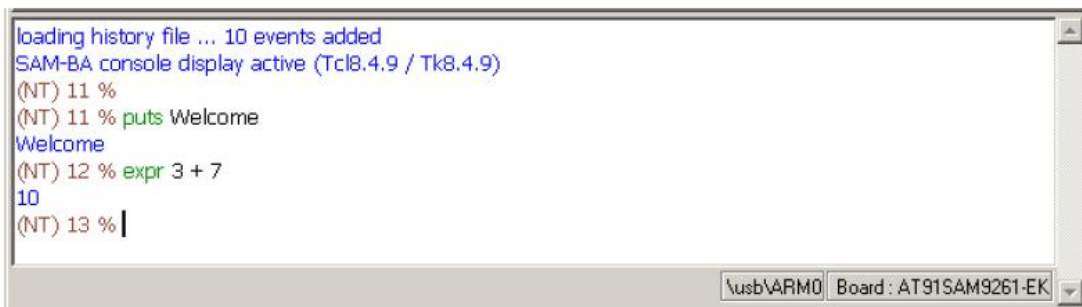


Figura 5.8 SAM-BA. Área TCL

Capítulo 6. Validación del diseño de la CV

6.1 Introducción

El objetivo de este capítulo es validar el diseño de la CV para ponerla a punto y poder utilizarla con SATEDU.

La validación de los circuitos electrónicos, se hizo con ayuda de la herramienta Proteus 7 de Labcenter Electronics, simulador de circuitos electrónicos, al integrar la capacidad de simular microcontroladores, incluyendo algunos basados en el núcleo ARM7TDMI.

El software requerido para validar dichos circuito, fue escrito en lenguaje C para un microcontrolador PIC16F876A con la ayuda de Mplab de Microchip.

El software al ser escrito en lenguaje C, puede ser extrapolado para el microcontrolador AT91SAM7SE512 requiriendo cambios mínimos de acuerdo al compilador que se utilice para su programación.

6.2 Comunicación SPI

Al no disponer de un modelo de simulación para el módulo de memoria Spansion S25FL128P o el controlador CAN MCP2515, se optó por utilizar una memoria 25AA080B SPI, figura 6.1, de 8 kbits de Microchip [18].

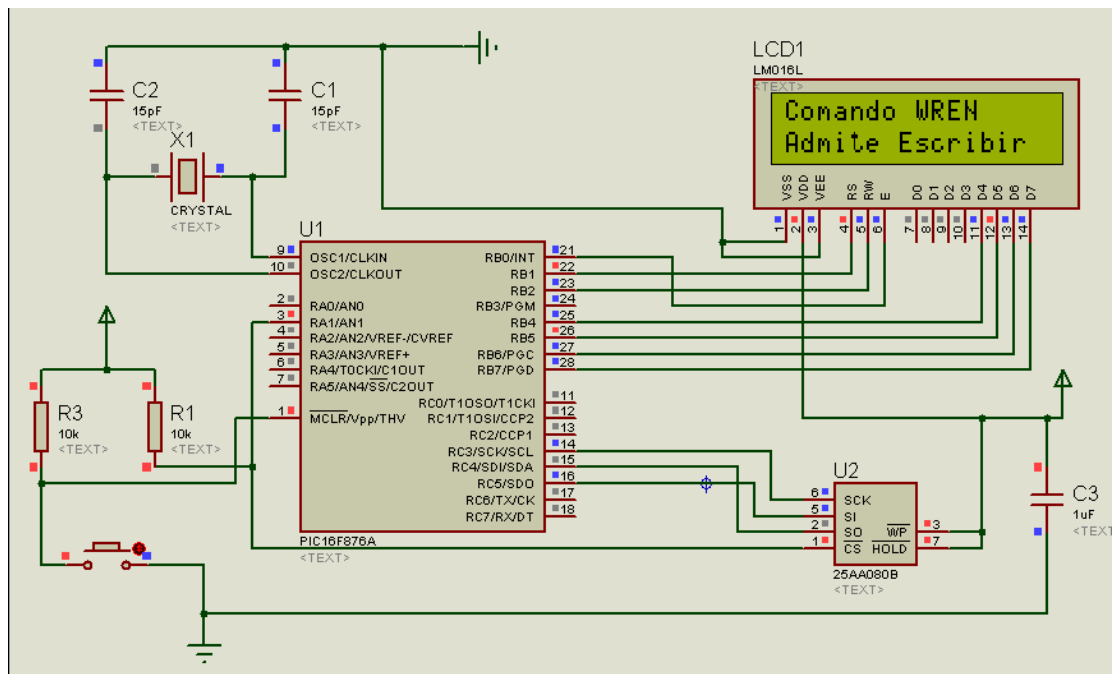


Figura 6.1 Esquemático de simulación de memoria SPI

Siendo la única diferencia entre la memoria Spansion, el controlador CAN y la memoria Microchip, la cantidad de espacio, direcciones y el número asignado a los comandos, el funcionamiento es el mismo. Tanto las memorias como el controlador CAN, emplean polaridad y fase de reloj 0 en el protocolo SPI.

La memoria 25AA080B se encuentra organizada en páginas de 32 Bytes a partir de la dirección 0x0000 a la 0x03FF, sus comandos correspondientes son:

Operación	Comando	Descripción	Bytes de direcciones	Bytes de datos
Lectura	RDSR (05h)	Lee registro de estatus (RE)	0	1
	READ (03h)	Lee bytes o página	2	1 a 32
Escritura	WREN (06h)	Habilita escritura	0	0
	WRSR (01h)	Escribe ER	0	1
	WRITE (02h)	Escribe bytes o página	2	1 a 32

Tabla 6.1 Comandos de memoria Microchip 25AA080B

El software desarrollado, genera un arreglo de 32 Bytes equivalente a una página, habilita la escritura de la memoria y escribe el arreglo en la dirección 0x0000 para posteriormente leerlo. Seguidamente se escribe el valor de 0x08 en el RE y se realiza su lectura.

Para poder visualizar la ejecución del código escrito en el PIC16F876A, el cual se muestra en el apéndice B, se utilizó una pantalla lcd de 16 caracteres por 2 líneas.

Las figuras 6.2 y 6.3 muestran la simulación, en la 6.2 se observa la lectura de la memoria y en la 6.3 la lectura del RE.

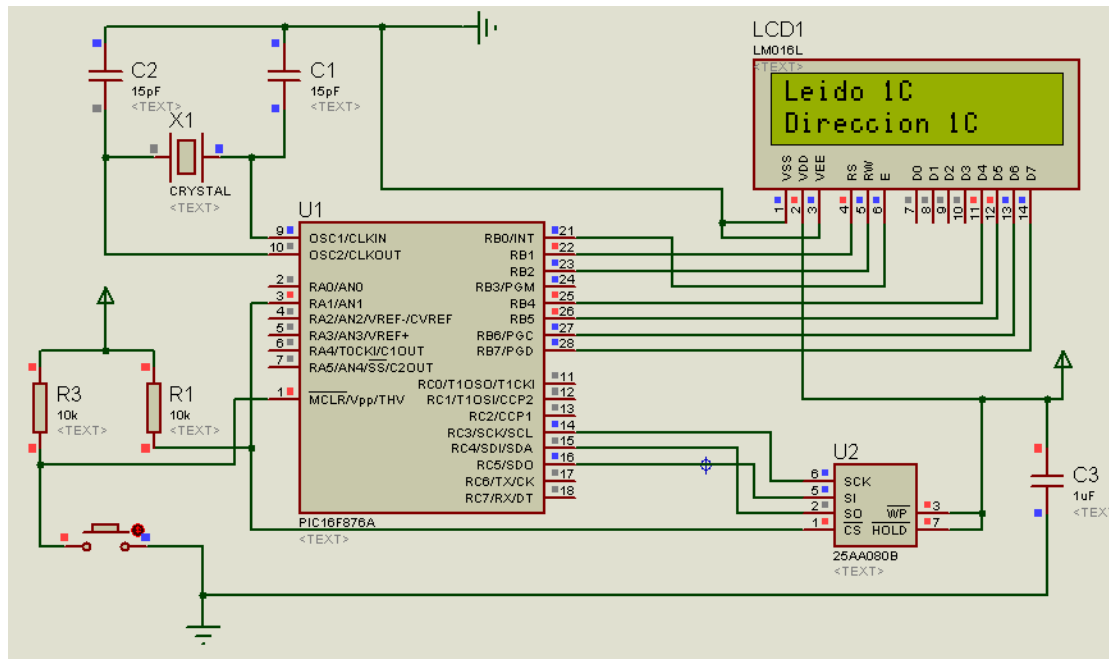


Figura 6.2 Simulación de comunicación SPI.

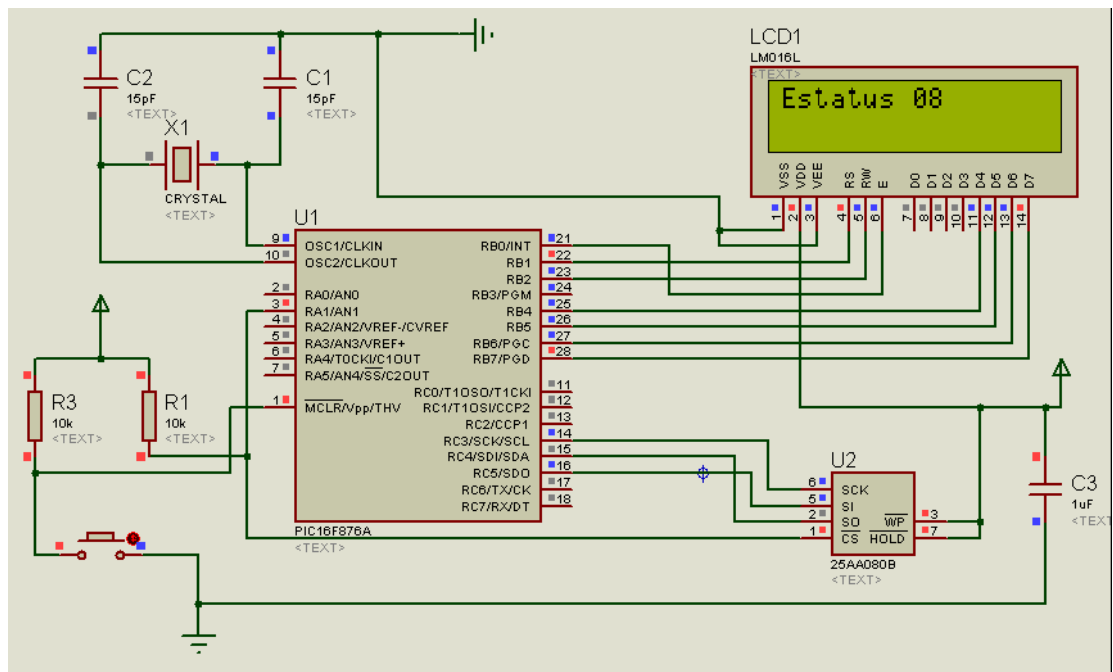


Figura 6.3 Simulación de comunicación SPI.

6.3 Sensores de temperatura

Para validar la comunicación 1-Wire y los sensores de temperatura, se realizó el siguiente circuito:

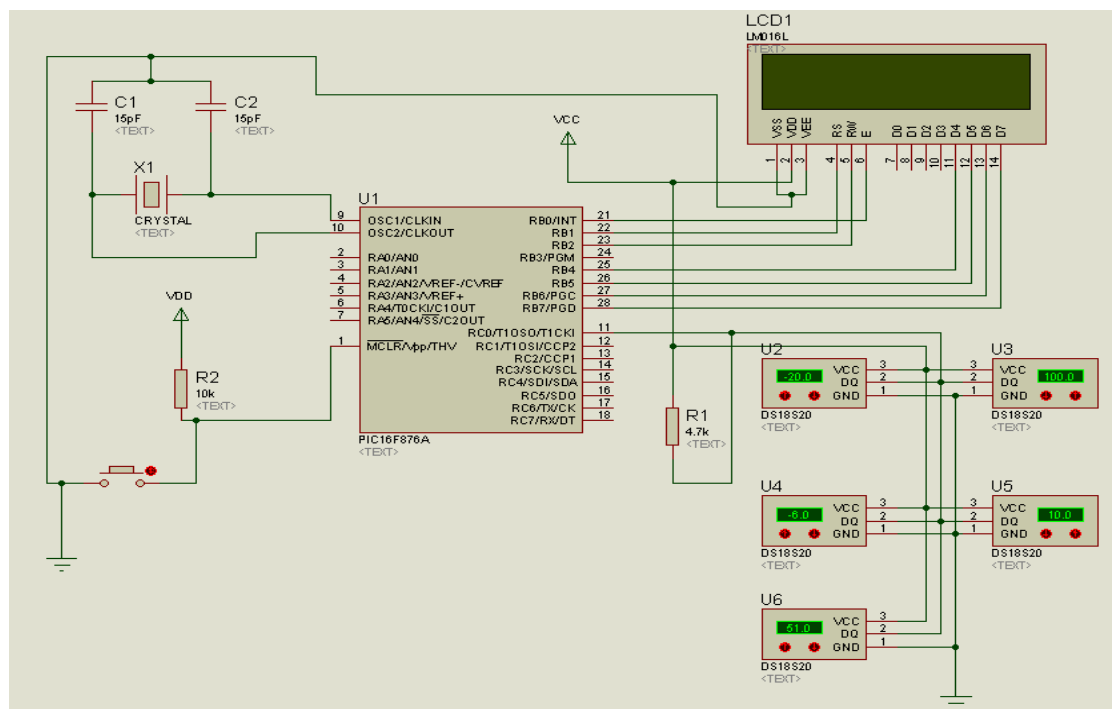


Figura 6.4 Sensores de temperatura 1-Wire

Donde U2 a U6 representan los sensores 1-Wire DS18S20 utilizados en el diseño de la CV.

La simulación realiza la búsqueda del ROM Code asociado a cada sensor de temperatura, solicita a dichos sensores la toma de temperaturas y su envío al microcontrolador (PIC16F876A) para posteriormente mostrarlas en la pantalla lcd.

6.3.1 Búsqueda de ROM Code

El algoritmo que se utiliza para encontrar el ROM Code de los sensores, consiste en un árbol binario de búsqueda y los pasos para su ejecución son los siguientes:

1. Se envía un pulso reset vía 1-Wire a los dispositivos, sensores, conectados al bus.
2. Se envía el comando Search (0xF0) vía 1-Wire para comenzar la búsqueda del ROM Code asociado a los sensores conectado al bus.
3. Los sensores responden enviando el primer bit de su ROM Code. Debido a la presencia de varios sensores, el bit resultante es el de aplicar una compuerta lógica AND a todos los bits son enviados al mismo tiempo por los dichos sensores.
4. Después de que el dispositivo maestro (microcontrolador) recibe el primer bit, los sensores envían el complemento de su primer bit. De igual forma el complemento resulta de aplicar una compuerta AND a todos los bits enviados por parte de los sensores.
5. De acuerdo al diagrama de flujo de la figura 6.5, el dispositivo maestro envía un bit a los sensores. De acuerdo con el valor del bit enviado, los sensores cuyo bit en su ROM Code tenga el mismo valor siguen participando en la búsqueda. Los que tiene un valor diferente entran en un estado de espera hasta detectar el próximo reset 1-Wire.
6. Los pasos 3 a 5 se repiten con los bits 2 a 64 del ROM Code de los sensores. El ROM Code resultante es el perteneciente a uno de los sensores conectados al bus.
7. Para sucesivas búsquedas se repiten los pasos 1 a 6.

Cabe mencionar que cuando los sensores envían el valor del bit asociado a su ROM Code así como su complemento, se puede presentar que ambos tengan el valor de 0. Esto indica la presencia de dispositivos cuyo bit dentro de su ROM Code vale 1 y dispositivos para los cuales vale 0. Este caso de discrepancia se maneja con ayuda de una variable de acuerdo al diagrama de flujo de la figura 6.5.

En el caso de que el valor del bit y el complemento tengan como valor 1, se dice que no hay dispositivos conectados al bus. Esto puede resultar de una mala conexión de los sensores.

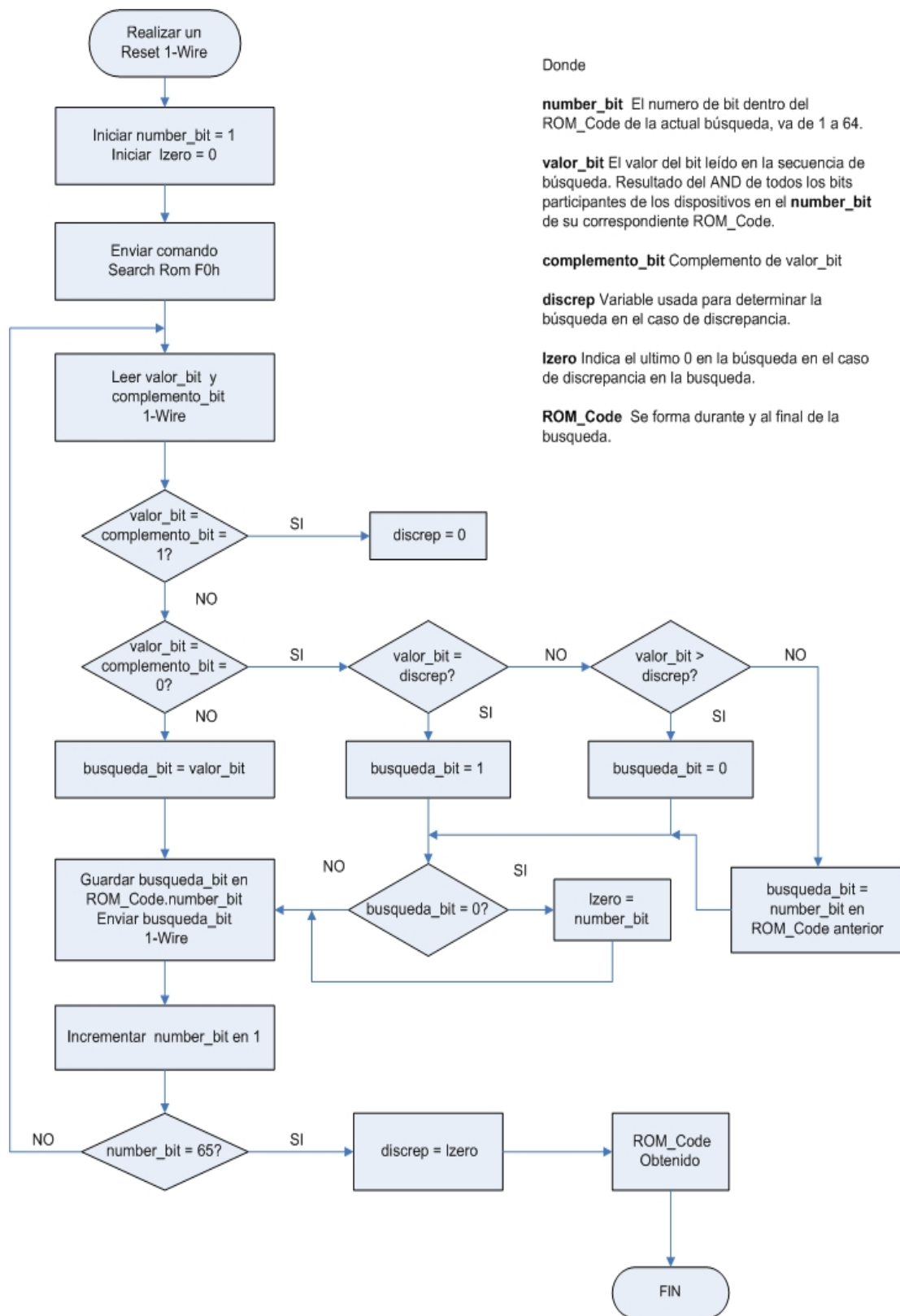


Figura 6.5 Diagrama de flujo de algoritmo de búsqueda de ROM_Code

6.3.2 Lectura de temperaturas

Para obtener las temperaturas asociadas a cada uno de los sensores, el programa desarrollado, mostrado en el apéndice B, envía el comando SKIP ROM (0xCC) seguido de CONVERT T (0x44) para indicar a los sensores que obtengan la lectura de la temperatura.

Después se manda el comando MATCH ROM (0x55) seguido del ROM Code de un sensor y del comando READ SCRATCHPAD (BEh). Cuando el microcontrolador envía el comando READ SCRATCHPAD, el sensor DS18S20 transmite el contenido de la memoria SRAM comenzando por el menos significativo de la localidad 0 hasta el bit mas significativo de la localidad 9 a través del bus. El microcontrolador recibe los primeros 2 Bytes para determinar la temperatura del sensor y a continuación envía un pulso de reset para no seguir recibiendo el contenido de la memoria del sensor.

La lectura se hace con los 5 sensores del circuito de la figura 6.4. El resultado de la temperatura se muestra en la pantalla lcd. Las figuras 6.6 y 6.7 muestran la simulación de los sensores de temperatura, la 6.6 muestra la obtención de un ROM Code y la 6.7 la temperatura asociada a dicho sensor.

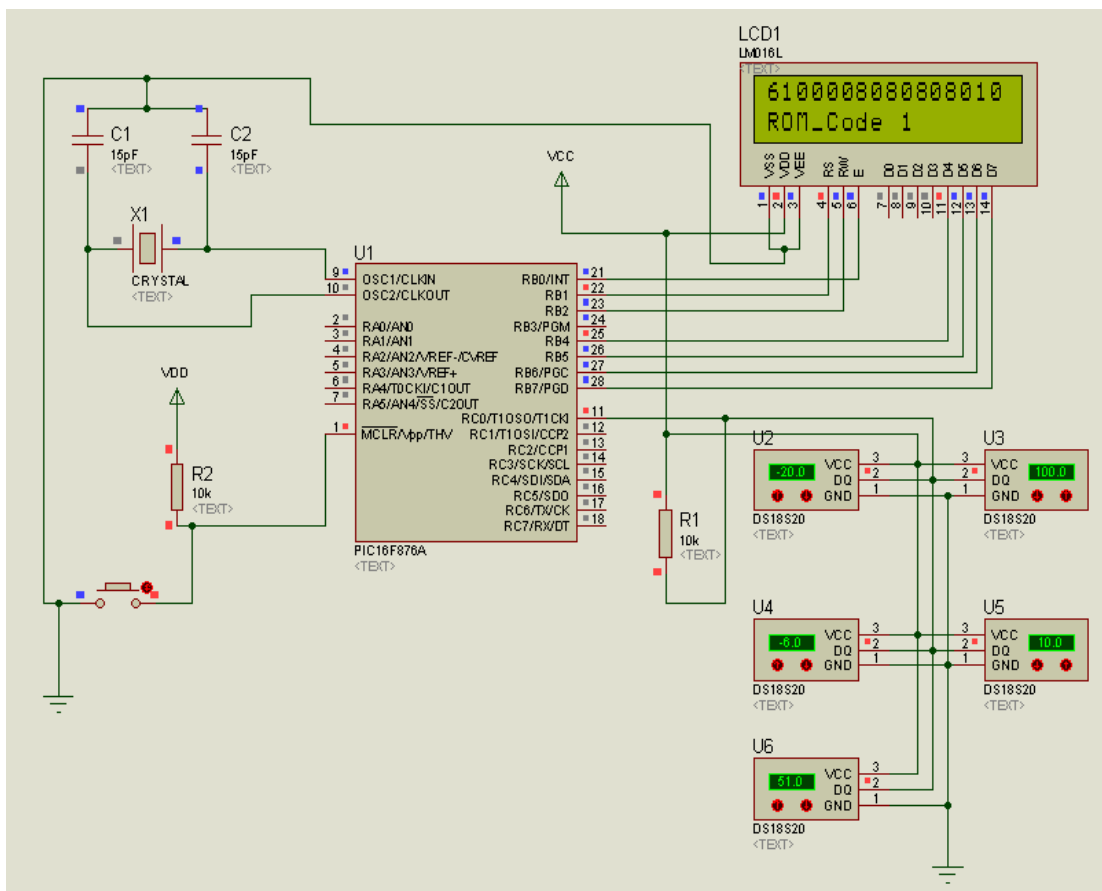


Figura 6.6 Simulación de sensores 1-Wire, ROM Code obtenido

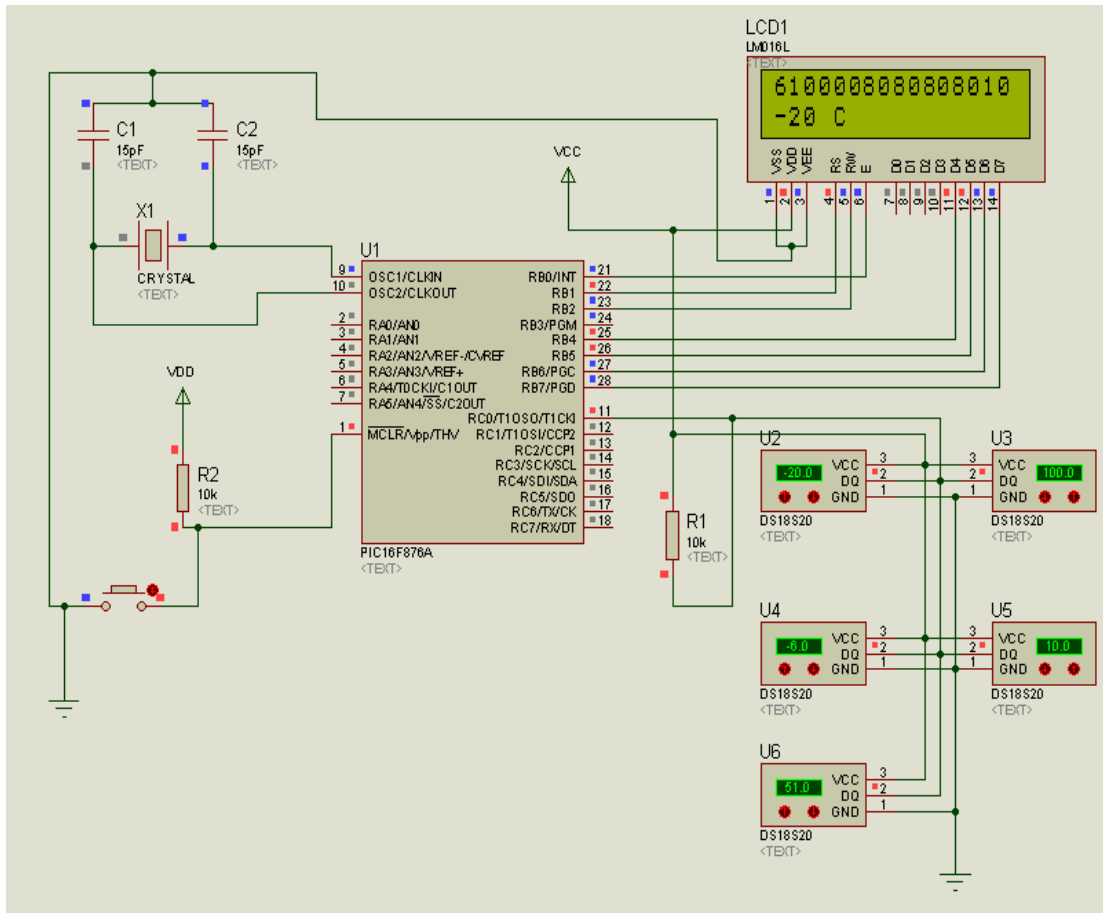


Figura 6.7 Simulación de sensores 1-Wire, temperatura obtenida

6.4 Protecciones efecto latch-up

Para validar los sistemas de protección de efecto latch-up se utilizó el circuito de la figura 6.8. Tal circuito tiene el sensor MAX4071 así como el comparador LM6511. La carga se sustituye por un par de resistencias las cuales permiten simular ya sea el microcontrolador AT91SAM7SE512 o la memoria SRAM CY62167DV30 cuando funcionan con una corriente mayor o menor a la establecida en la protección de efecto latch-up.

Para el AT91SAM7SE512 se utilizan, figura 6.8, las resistencias de 49 y 45 Ω . Por medio de un switch (conmutador) se selecciona una de las resistencias y a la salida del comparador se obtienen 0 V cuando la corriente que circula por la carga es menor a 70 mA y 3.3 V cuando es mayor.

Para la memoria SRAM las resistencias tiene el valor de 105 y 115 Ω . La corriente permitida en la memoria es de 30 mA y de igual forma a la salida del comparador se tienen 0 V cuando la corriente que circula por la memoria es menor a los 30 mA y cuando es mayor se obtienen 3.3 V.

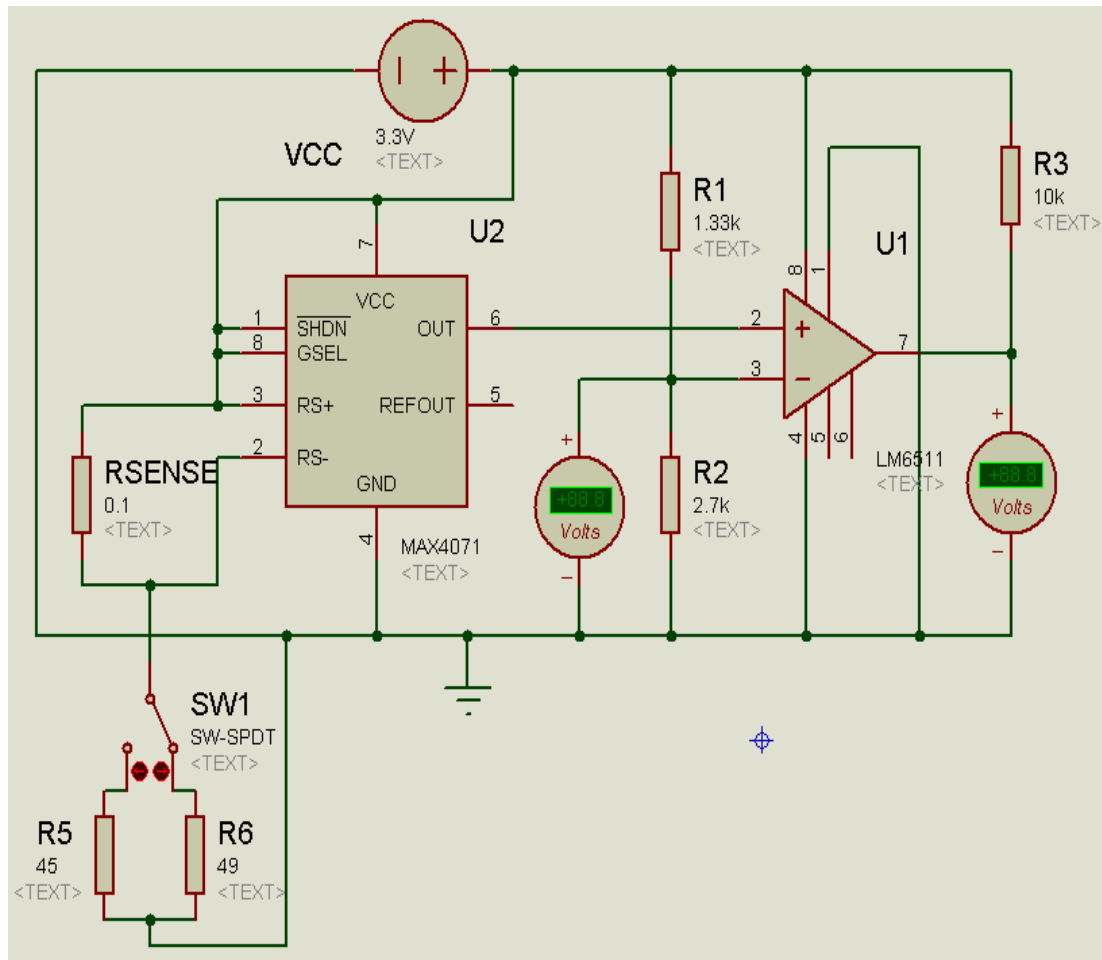


Figura 6.8 Circuito de protección efecto latch-up

Para visualizar el funcionamiento del circuito de protección efecto latch-up, se agregó un microcontrolador (PIC16F876A) que realiza una conversión A/D de la salida del comparador y cuando ésta tiene un valor de 3.3 ± 0.3 V, a través de una pantalla lcd, se muestra una advertencia.

La figura 6.9 muestra el circuito de protección del AT91SAM7SE512 cuando la corriente que utiliza es menor a los 70 mA mientras la figura 6.10 muestra cuando dicha corriente es mayor.

En el apéndice B se muestra el programa utilizado para la simulación de los sistemas de protección de efecto latch-up.

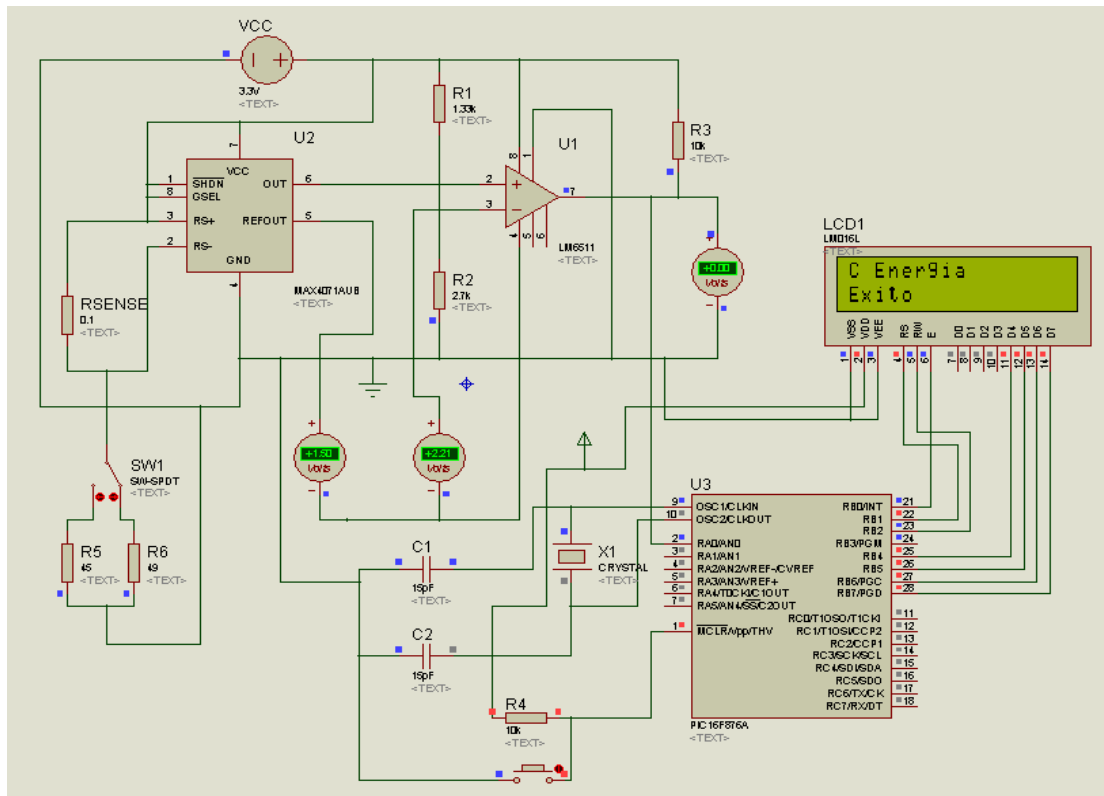


Figura 6.9 Simulación de protección efecto latch-up para el AT91SAM7SE512

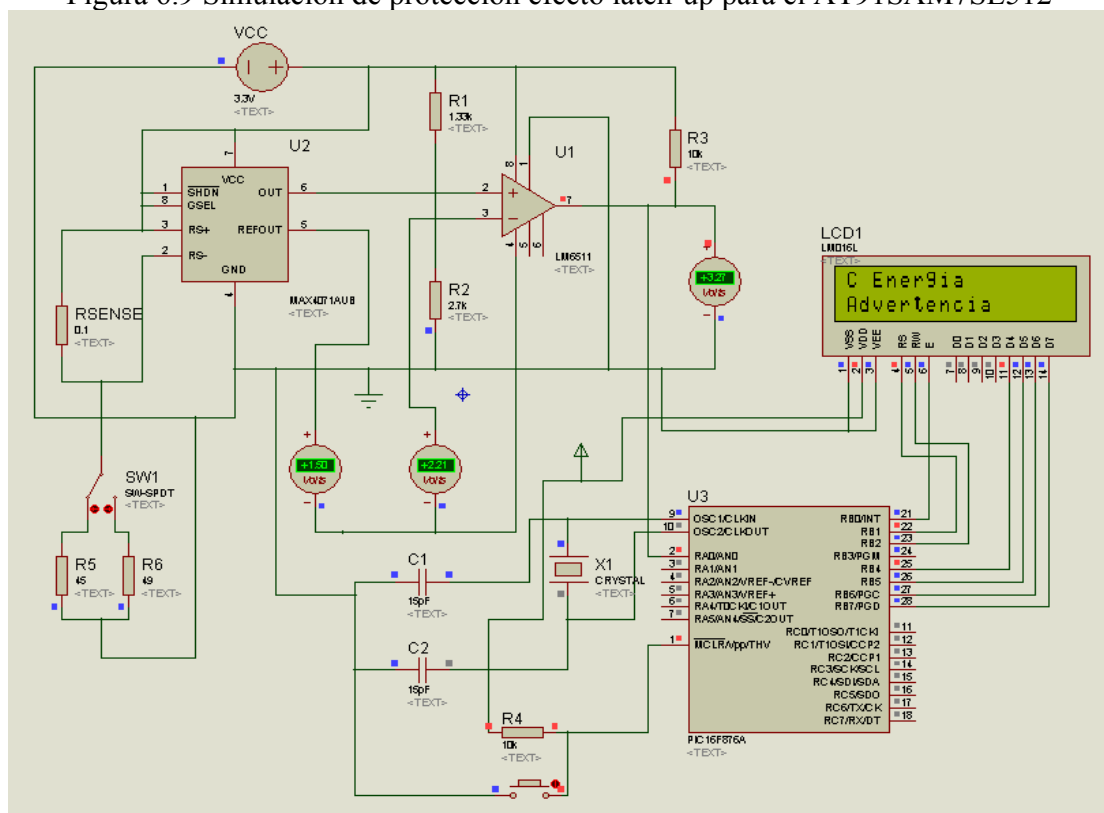


Figura 6.10 Simulación de protección efecto latch-up para el AT91SAM7SE512

6.5 Controlador de Bus CAN

No se dispone de un modelo de simulación para el controlador y el receptor-transmisor del Bus CAN sin embargo, la comunicación entre el microcontrolador AT91SAM7SE512 y el controlador MCP2515 es a través del protocolo SPI, de la misma manera que con la memoria utilizada para simular dicha comunicación.

No obstante se realizó un diagrama de flujo para la transmisión y recepción de mensajes del Bus CAN a través del controlador MCP2515. Dichos diagramas se explican a continuación.

6.5.1 Recepción de mensajes

El algoritmo a seguir, desde la detección de mensajes entrantes, su recepción y almacenamiento en los buffers del controlador CAN así como las indicaciones de si fue recibido exitosamente o se presentó un error, es el siguiente:

- Se monitorea continuamente la detección del inicio de un mensaje entrante.
- Se carga el nuevo mensaje en el buffer ensamblador de mensajes MAB (Message Assembly Buffer).
- Se verifica si un mensaje valido ha sido recibido. En caso de que el mensaje no sea valido, se genera una trama de error y se retorna al inicio.
- Se comprueba si el mensaje cumple los criterios de los filtros para el buffer de recepción RXB0 y si no los cumple, se comprueba que los cumpla para RXB1. Si el mensaje no cumple con los criterios de los buffer de RXB0 y RXB1, se retorna al inicio y el mensaje no es procesado.
- Si cumple los criterios de aprobación de los filtros de RXB0, se comprueba si el bit CANINTF.RX0IF esta desactivado, vale 0, indicando que el buffer de recepción esta disponible para recibir el nuevo mensaje.
- Si el buffer RXB0 no esta disponible para recibir un nuevo mensaje, se verifica si el bit RXB0CTRL.BUKT esta activado. De esta manera el mensaje se envía al buffer RXB1, sin pasarlo por sus filtros y suponiendo se encuentra disponible para recibirlo, para evitar que el buffer RXB0 se sobrescriba.
- Si no hay buffer disponible para recibir el mensaje, se genera un error de sobrecarga y se activa el bit EFLG.RXnOVR. Si esta habilitado el bit CANINTE.ERRIE, se genera una interrupción en el pin INT. Por último se regresa al inicio.
- El mensaje se mueve al buffer disponible y se activa el bit CANINTF.RXnIF correspondiente. Tal bit debe ser borrado por el CPU de la CV para indicar que el buffer esta listo para recibir el siguiente mensaje entrante.

- Se activan los bits RXBnCTRL.FILHIT<2:0> de acuerdo al número de filtro, del respectivo buffer, que aceptó el mensaje.
- Se comprueba si está activo el bit CANINTE.RXnIE y de ser así, se genera una interrupción en el pin INT. Se modifican los bits CANSTAT<3:0> indicando el buffer donde se cargó el mensaje.
- Se comprueba que los bits BFPCTRL.BnBFM y BF1CTRL.BnBFE estén activos, valgan 1, para modificar los pines RXnBF a un nivel bajo e indicar que un mensaje ha sido recibido.
- Se regresa al inicio del algoritmo de recepción de mensajes.

Los pines RXnBF no son necesarios en el diseño de la CV, cuando un mensaje es recibido se produce una interrupción en el pin INT.

El diagrama de flujo de la figura 6.11 muestra el algoritmo de recepción de mensajes del controlador CAN MCP2515.

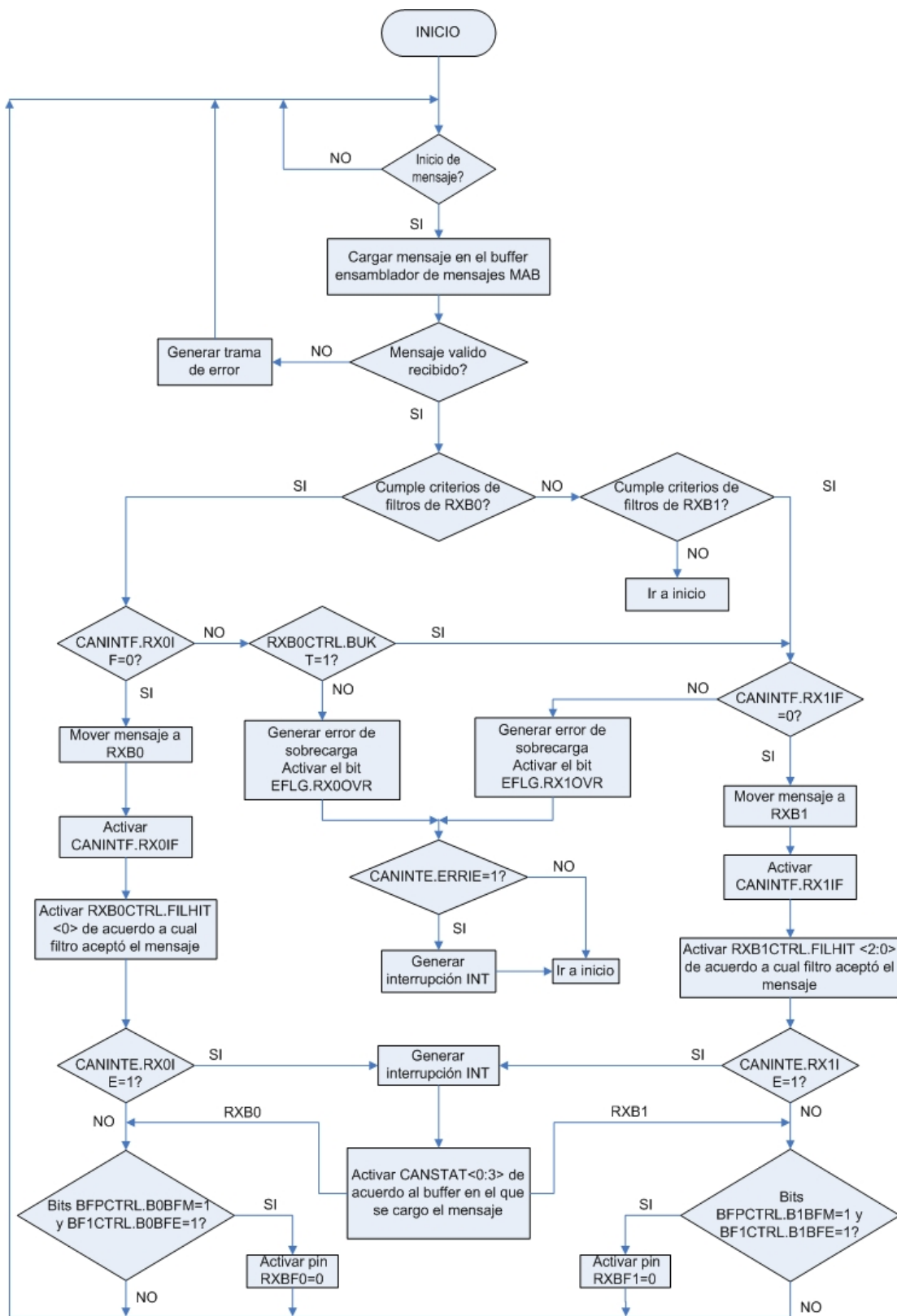


Figura 6.11 Diagrama de flujo de recepción de mensajes del bus CAN

6.5.2 Transmisión de mensajes

Teniendo almacenado un mensaje en un buffer de transmisión y solicitando su envío por medio de los pines TXnRTS durante un flanco de bajada, su envío se hace de la manera siguiente:

- El bit TXBnCTRL.TXREQ se activa, vale 1, y se borran los bits TXBnCTRL.ABTF, TXBnCTRL.MLOA y TXBnCTRL.TXERR que, respectivamente, se activan cuando el mensaje es abortado, hay pérdida de arbitraje durante su envío o ha ocurrido un error durante su transmisión.
- Su comprueba que el Bus CAN este disponible para iniciar la transmisión del mensaje. En caso de no estarlo y que el mensaje no haya sido abortado, se comprueba su disponibilidad continuamente. Si se aborta la transmisión, el bit CANCTRL.ABAT debe ser borrado por el CPU de la CV para poder volver a transmitir un mensaje.
- Se revisa la prioridad del buffer por medio de los bits TXBnCTRL.TXP<1:0>, siendo (1,1) la combinación que da la mayor prioridad al buffer correspondiente, y el buffer que tenga una mayor prioridad es transmitido. Se asegura que cuando dos buffers solicitan el envío de un mensaje, lo haga el buffer de mayor prioridad y el otro permanezca como pendiente hasta poder transmitirlo.
- En caso de que el mensaje no se enviara exitosamente, se comprueba si se presentó un error de mensaje o si se perdió la prueba de arbitraje. Para un error de mensaje, se activa el bit TXBnCTRL.TXERR y se comprueba si el bit CANINTE.MERRE se encuentra activado. Si el bit CANINTE.MERRE está activado, vale 1, se genera una interrupción en el pin INT. Finalmente se activa el bit CANINTF.MERRF que debe ser borrado, por el CPU de la CV, cuando la condición de interrupción sea atendida. Para indicar que un mensaje perdió la prueba de arbitraje, se activa el bit TXBnCTRL.MLOA. El mensaje se vuelve a enviar.
- Si el mensaje se envió exitosamente: Se borra el bit TXREQ para indicar que el buffer no esta pendiente de transmisión. Se verifica si esta activo el bit CANINTE.TXnIE y, en caso de estarlo, se genera una interrupción. Se activa el bit CANINTF.TXnIF para indicar que una interrupción se encuentra pendiente y debe ser borrado por el CPU de la CV.
- Se regresa al inicio para comprobar continuamente si se encuentra en espera la transmisión de un mensaje por parte de un buffer y, en caso de haber mensaje, se inicia su transmisión.

El diagrama de flujo de la figura 6.12 muestra el algoritmo de transmisión de mensajes del controlador CAN MCP2515.

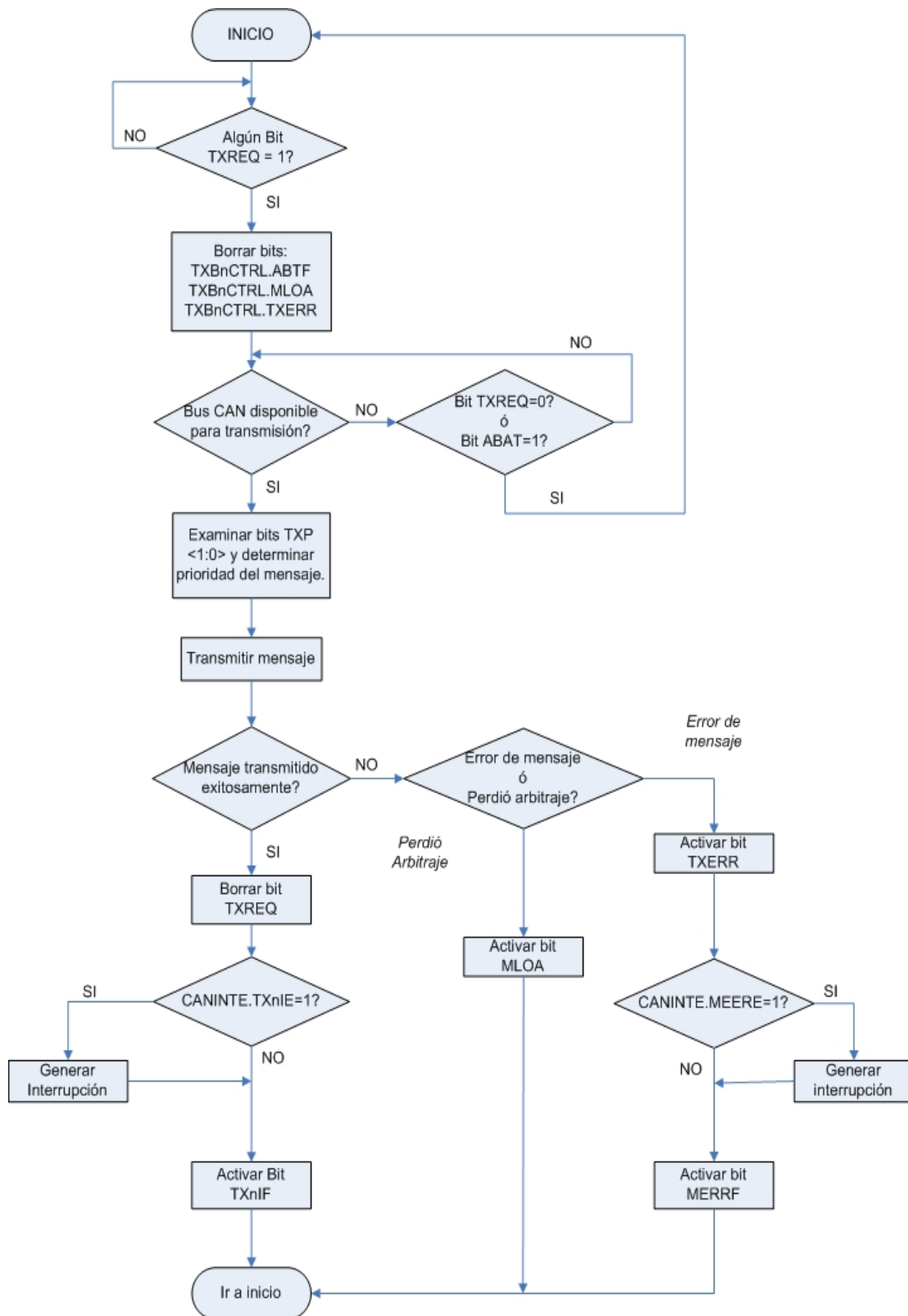


Figura 6.12 Diagrama de flujo de transmisión de mensajes a través del bus CAN

Capítulo 7 Desarrollo de la tarjeta electrónica

7.1 Introducción

El PCB de la computadora de vuelo tiene forma cuadrada de 8.9 cm por lado con componentes en ambas capas, inferior y superior, la mayoría de montaje superficial. El diseño de la CV se hizo utilizando la herramienta Protel DXP [29] buscando la completa compatibilidad con los subsistemas de SATEDU.

En este capítulo se explica el procedimiento de diseño de la CV, se hace un análisis de los resultados obtenidos y se muestra, de manera grafica, la conclusión de la CV.

7.2 Diagrama electrónico de la computadora de vuelo

En el diagrama de la CV, figura 7.1, se muestra el AT91SAM7SE512 conectado a 2 MBytes de memoria SRAM externa, 5 sensores de temperatura 1-Wire, 64 MBytes de memoria flash SPI, protecciones contra efecto latch-up en el AT91SAM7SE512 y en la memoria S25FL128P, controlador de bus CAN MCP2515, transmisor-receptor de Bus CAN MCP2551 y el decodificador SN74HC42.

Las señales de reloj del AT91SAM7SE512 y del controlador CAN MCP2515 se obtienen de un cristal de 20 MHz asociado a cada dispositivo.

Los conectores, derecho e izquierdo, que se utilizarán como interfaz de comunicación entre los subsistemas, se cambiaron de headers de 40 pines por headers de 26, de dos líneas (13 pines por cada una). El conector que establece la comunicación entre la CV y la ET se cambió de un RJ11 a uno de menor tamaño, un mini-USB. El conector mini-USB maneja las señales de la unidad serial DBGU, la fuente de 3.3 V y tierra (GND), sin embargo pueden modificarse las líneas asociadas al conector en caso de utilizar el protocolo USB, disponible en el AT91SAM7SE512, como medio de comunicación entre la CV y la ET.

7.3 Diseño del PCB de la computadora de vuelo

El PCB [11] de la CV tiene una forma cuadrada de 8.9cm por lado, compatible con los subsistemas de SATEDU, con componentes en ambos lados. Las pistas tienen un grosor y separación de 8 mil (1 mil = 0.0254 mm) y las vías son de 38 mil x 20 mil. Para rutear el PCB se colocaron los componentes electrónicos de tal forma que las conexiones entre ellos sean lo más directo posible. Después se rutearon manualmente las señales de los cristales de 20MHz, correspondientes al AT91SAM7SE512 y al MCP2515, cuidando que la pista sea lo más recta y corta posible. Una vez hecho esto se utilizó el autoruteador configurado para que la capa superior sea ruteada horizontalmente y la inferior verticalmente.

Las figuras 7.2 y 7.3 muestran la capa superior e inferior, respectivamente, del diseño del circuito impreso de la CV. En las tablas 7.1 y 7.2 se muestran los componentes integrados en ambas capas de la CV.

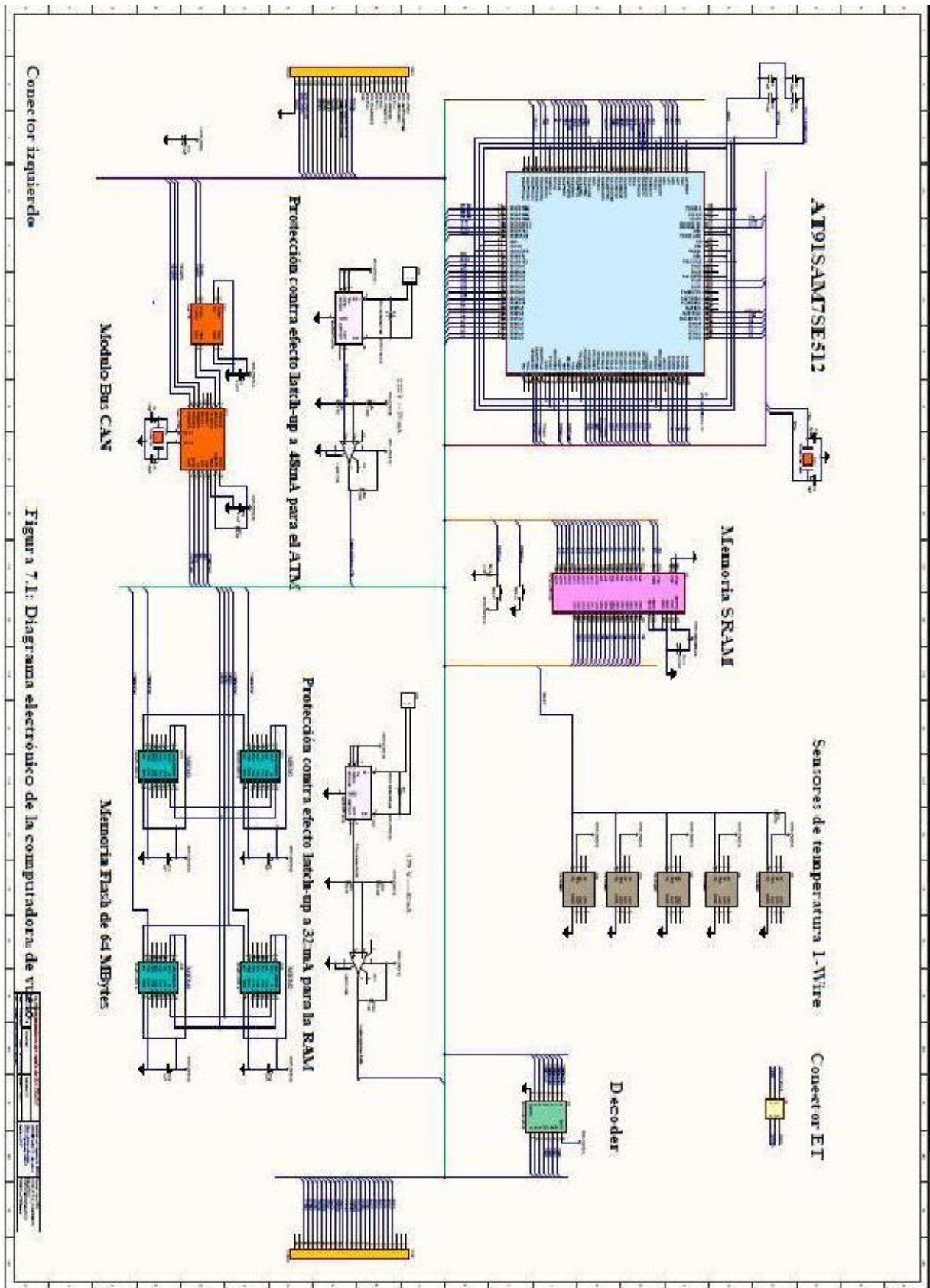


Figura 7.1 Diagrama electrónico de la nueva CV

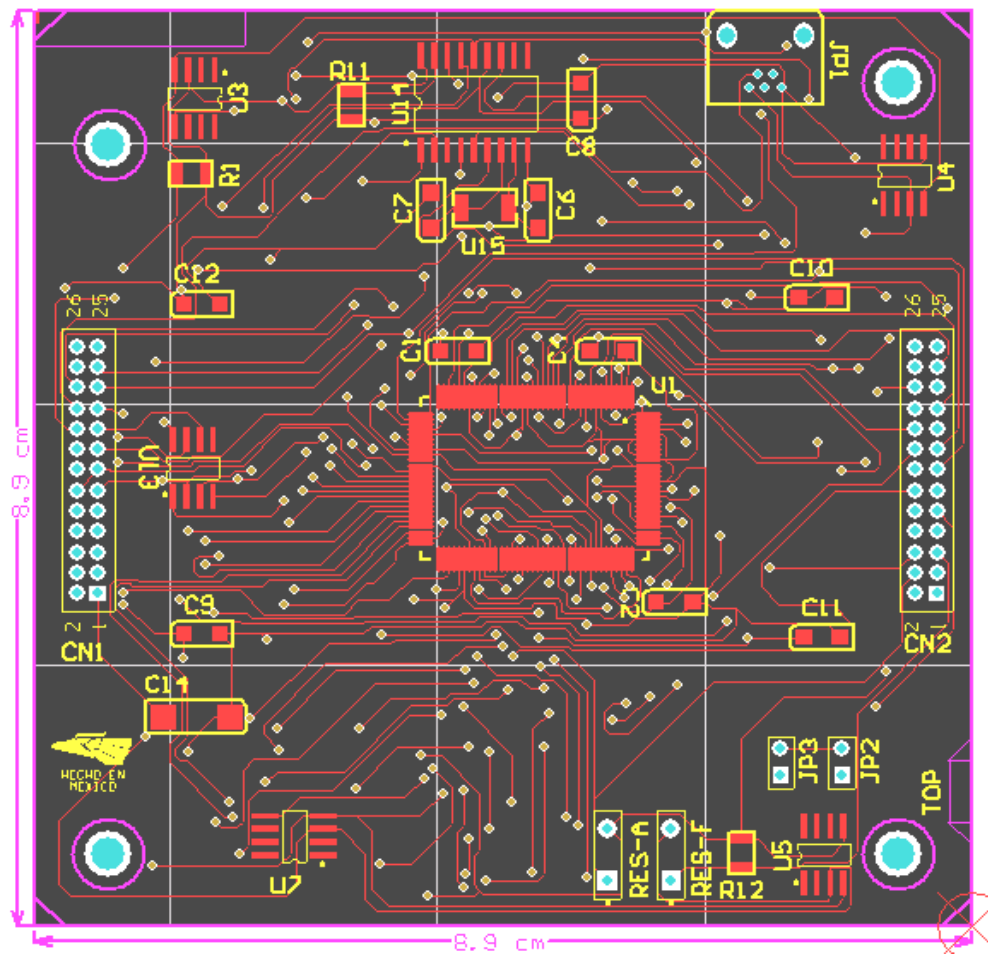


Figura 7.2 Capa superior de la nueva CV

Designación	Elemento
U1	Microcontrolador AT91SAM7SE512
U3, U4, U5, U7	Sensor de temperatura
U13	Transceiver CAN
U14	Controlador CAN
U15	Cristal Osc CAN
C1, C2, C4	Cap AT91SAM7SE512
C6, C7	Cap cristal Osc CAN
C8	Cap controlador CAN
C9, C10, C11, C12	Cap memoria SPI
C14	Fuente 5V
CN1, CN2	Conector izquierdo y derecho
JP1, JP2, JP3	Conector USB y jumpers
R1	Res Sensor de Temperatura
R11	Res Controlador CAN
R12	Res Reset de AT91SAM7SE512
RES-A	Reset de AT91SAM7SE512
RES-B	Reset de memoria Flash

Tabla 7.1 Componentes en capa superior de la nueva CV

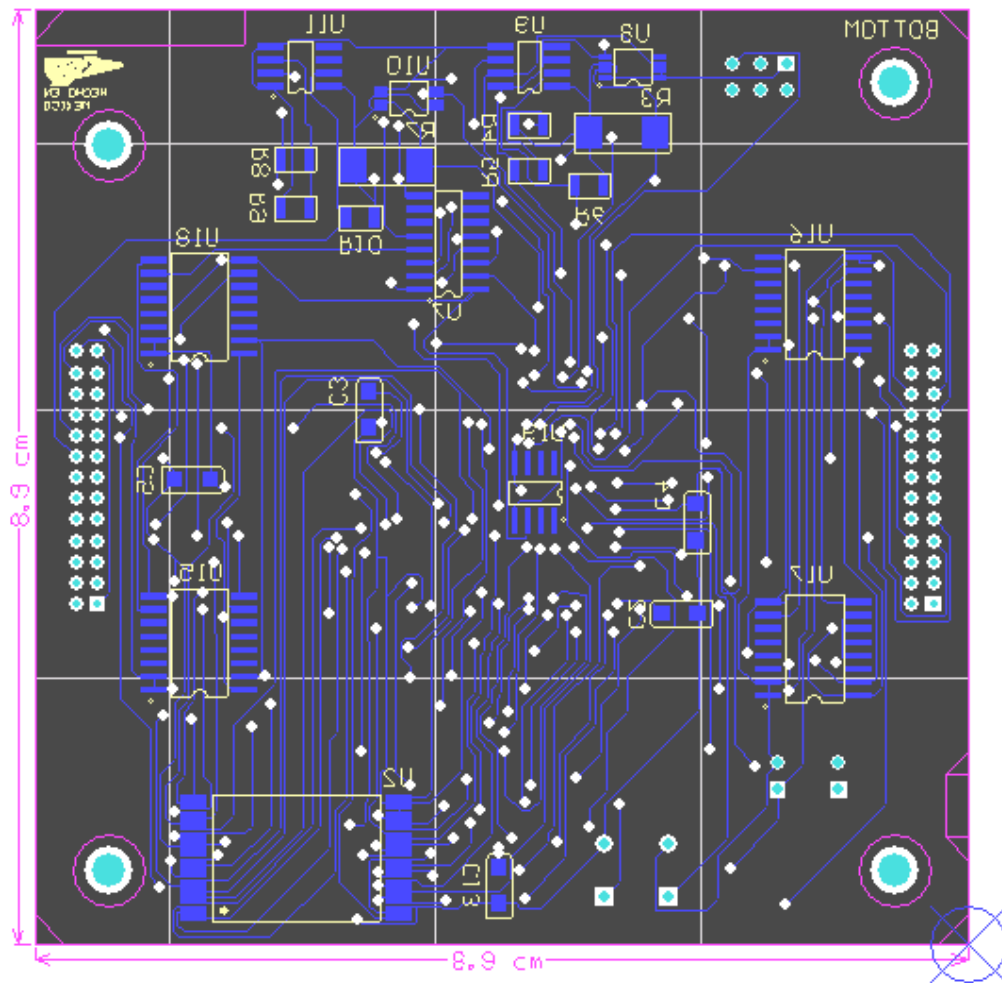


Figura 7.3 Capa inferior de la nueva CV

Designación	Elemento
U2	Memoria RAM
U6	Sensor de Temperatura
U8, U10	Sensor de corriente
U9, U11	Comparador
U12	Decoder 4 a 10
U16, U17, U18, U19	Memoria SPI
U20	Cristal Osc AT91SAM7SE512
C3	Cap AT91SAM7SE512
C5	Cap Transceiver CAN
C13	Cap Memoria RAM
C15, C16	Cap Cristal Osc AT91SAM7SE512
R3, R7	Res Sensor de corriente
R4, R5, R6, R8, R9, R10	Res Comparador

Tabla 7.2 Componentes en capa inferior de la nueva CV

7.4 Análisis de resultados del diseño de la CV

La CV anterior de SATEDU utiliza 67 componentes electrónicos que incluyen: microcontroladores, memoria RAM, memoria SPI, protecciones efecto latch-up, sensores de temperatura, señales de reloj, circuitos de lógica digital y conectores, principalmente.

La nueva CV utiliza 54 componentes, 13 menos que la anterior CV. Además de utilizar menos componentes en el diseño de la nueva CV, se obtienen mejoras como: ampliación de memoria RAM externa de 256 kBytes a 2 MBytes, ampliación de memoria de almacenamiento SPI de 32 a 64 MBytes, sustitución de multiplexores (8 canales seriales) por un controlador y receptor-transmisor de bus CAN capaz de soportar hasta 100 nodos conectados, 5 sensores de temperatura, cambio de conectores de 40 a conectores de 26 pines, cambio de conector RJ11 por conector mini-USB.

Además cabe señalar que mientras la anterior CV utiliza 161 redes (nets) de conexión, la nueva emplea solo 117. Esto es de vital importancia en un sistema donde las restricciones de dimensiones, tarjetas de 8.9x8.9 cm, obligan a buscar la reducción de componentes y líneas en un nuevo diseño sin sacrificar la funcionalidad del sistema.

Por otro lado la nueva CV, bajo condiciones de operación normal, ejecución de código en memoria flash interna y habilitados todos los periféricos y sus señales de reloj, funciona con una corriente de 31 [mA]. Con la memoria RAM externa habilitada funciona con 68 [mA]. La anterior CV de SATEDU trabaja con una corriente de 260 [mA], 100 [mA] del SAB80C166 y 80 [mA] por cada uno de los dos módulos de memoria RAM externa. Esto representa una reducción de 192 [mA] de corriente de operación en la nueva CV. Sin embargo el ahorro de energía es mayor considerando que el sistema de alimentación se cambió de 5 a 3.3 [V] en la nueva CV. En términos de potencia, la anterior CV funciona con 1.3 [W] mientras la actual con 0.22 [W] que representan un 83% de ahorro.

De lo anterior se concluye que el nuevo diseño de la CV cumple con las expectativas planteadas a lo largo de su desarrollo.

Las figuras 7.4, 7.5, 7.6 y 7.7 muestran una vista en tercera dimensión de las capas superior e inferior del circuito impreso, tanto de la anterior como la nueva CV, permitiendo hacer de manera visual una comparación entre ambos diseños y verificar los resultados obtenidos en esta tesis.

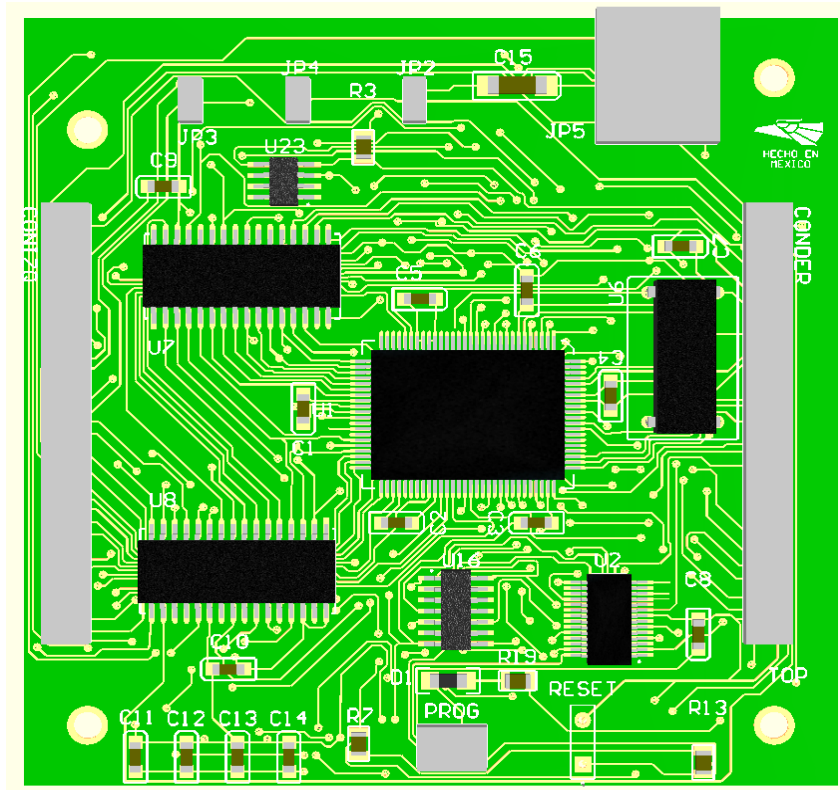


Figura 7.4 Capa superior de la CV anterior de SATEDU

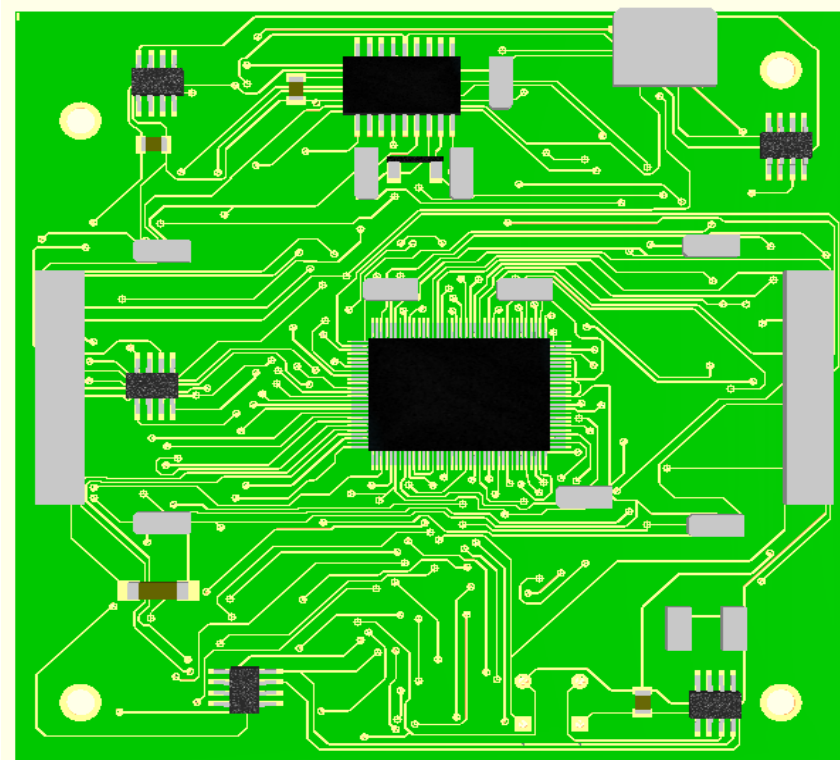


Figura 7.5 Capa superior de la nueva CV

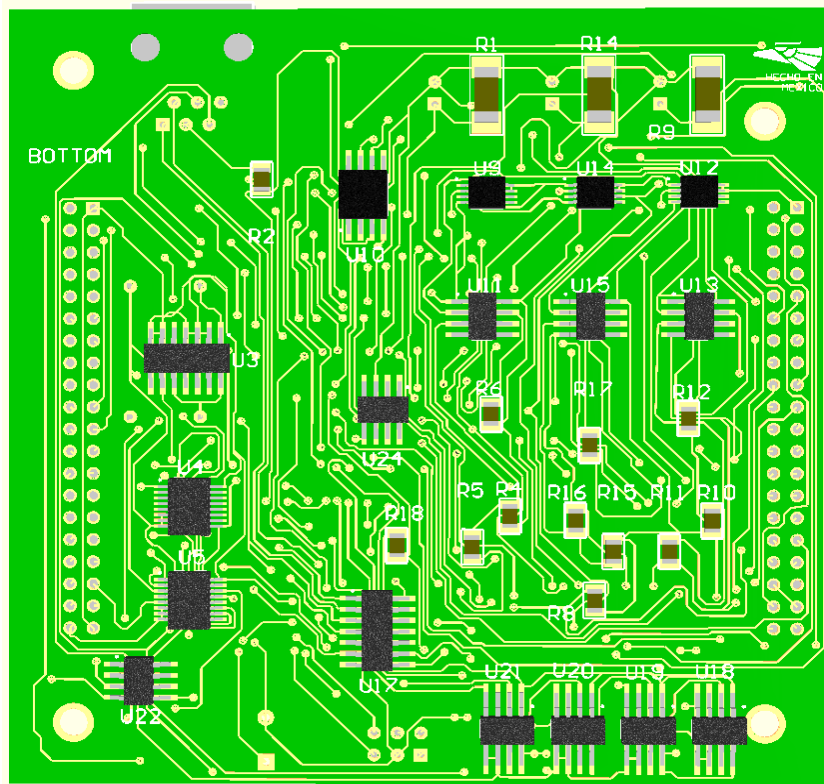


Figura 7.6 Capa inferior de la CV anterior de SATEDU

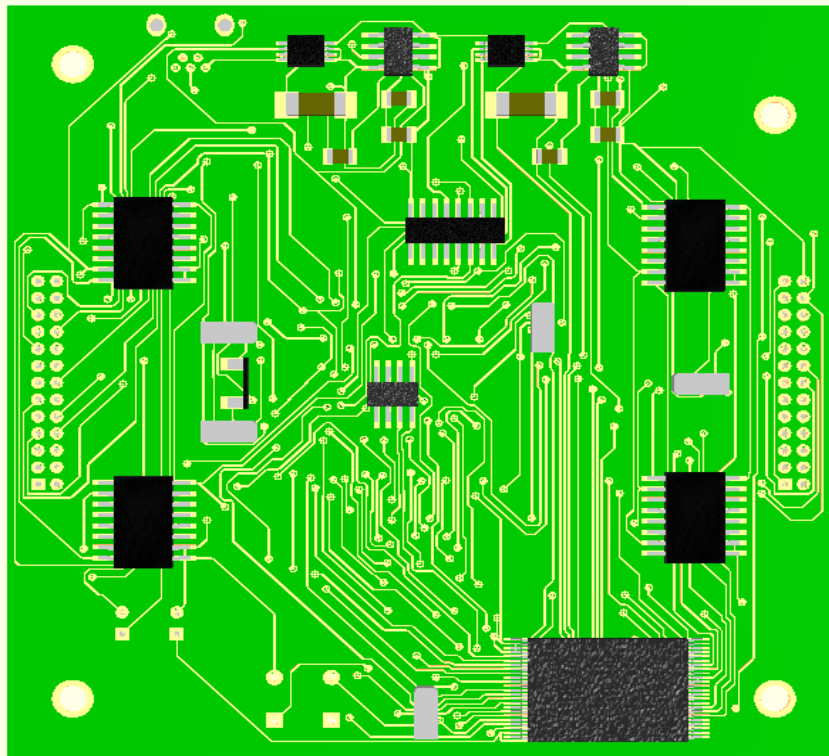


Figura 7.7 Capa inferior de la nueva CV

Capítulo 8 Conclusiones y recomendaciones

8.1 Conclusiones

Del trabajo desarrollado se concluye lo siguiente.

- Se efectuó el diseño de la nueva computadora de vuelo de SATEDU cumpliendo los objetivos planteados al inicio de su desarrollo. Se obtuvieron resultados en el ahorro de energía, ampliación de recursos y compatibilidad con los subsistemas de SATEDU.
- Se sustituyó la unidad de procesamiento SAB80C166 por el microcontrolador AT91SAM7SE512 de 32 bits con conjunto de instrucciones RISC, memoria interna flash de 512 kBytes y 32 kBytes de SRAM, interfaz de bus externo, sistemas de comunicación SPI, I2C y USART, convertidor A/D de 10 bits y sistema de alimentación de 3.3 V.
- Se amplió la memoria externa SRAM de 256kBytes a 2Mbytes a través de un módulo CY62167DV30 Cypress con un bus de direcciones de 20 bits y uno de datos de 16 bits.
- Se amplió la memoria externa SPI de 32 Mbytes a 64 Mbytes por medio de 4 módulos S25FL128P Spansion.
- Se sustituyeron los canales de comunicación serial RS232 por un sistema basado en el protocolo Bus CAN. Esto se realizó con un controlador de bus CAN MCP2515 Microchip y un transmisor y receptor MCP2551 Microchip.
- Se integraron 5 sensores de temperatura 1-Wire DS18S20 Maxim con rango de temperaturas de -55 a 125 °C.
- Se añadieron sistemas de protección de efecto latch-up tanto para la unidad de procesamiento como para la memoria externa SRAM. Esto se logra con sensores de corriente MAX4071 MAXIM y comparadores de voltaje LM6511 National Semiconductor.
- Con la sustitución de los componentes electrónicos en la nueva computadora de vuelo, se logró un ahorro de hasta un 83 % en el consumo de energía en comparación con el anterior diseño. Es decir, en términos de potencia, mientras la anterior computadora de vuelo utiliza 1.3 [W], la actual CV emplea 0.22 [W].

8.2 Recomendaciones

Algunas recomendaciones para la mejora de la computadora de vuelo son las siguientes.

- En caso de utilizar sistemas operativos, para el desarrollo de aplicaciones más complejas, es recomendable comenzar con un sistema operativo de tiempo real (RTOS) por su relativa sencillez en comparación con otros. Existen en el mercado sistemas operativos de tiempo real que incluyen el manejo de protocolos de red, sistemas de archivos y herramientas que administran de forma eficiente los recursos de la unidad de procesamiento.
- Es recomendable sustituir la interfaz de comunicación, entre STEDU y la PC de estación terrena, serial por una interfaz USB. Las actuales computadoras no disponen de un conector RS232 por su escaso uso en la actualidad. La unidad de procesamiento de la nueva CV integra un módulo de comunicación USB que se encuentra disponible para cuando se requiera su uso.

Apéndice A

A.1 Señales del AT91SAM7SE512

Las tablas siguientes muestran las señales del AT91SAM7SE512 utilizadas en el diseño de la CV.

Línea I/O	Periférico	Descripción	Correspondencia
PA0	NLB	Lower Byte Select	Byte Low Enable SRAM
PA1	A0	Bus de Dirección	SRAM
PA2	A1	Bus de Dirección	SRAM
PA3	TWD	Two-Wire Serial Data	Conector Izq
PA4	TWCK	Two-Wire Serial Clock	Conector Izq
PA5	RXD0	USART0 Receive Data	Conector Izq
PA6	TXD0	USART0 Transmit Data	Conector Izq
PA7	RTS0	USART0 Request to Send	Conector Der
PA8	CTS0	USART0 Request to Clear	Conector Der
PA9	DRXD	Debug Receive Data	Conector Izq
PA10	DTXD	Debug Transmit Data	Conector Izq
PA11	CS#0	SPI Chip Select 0	Bus SPI
PA12	MISO	SPI Master In Slave Out	Bus SPI
PA13	MOSI	SPI Master Out Slave In	Bus SPI
PA14	SPCK	SPI Serial Clock	Bus SPI
PA15	1W-ST	1-Wire Sensor Temperatura	Bus 1-Wire
PA16	PA16	Línea I/O	Conector Der
PA17	AD0	CA/D	V-Corriente-ATM
PA18	AD1	CA/D	V-Corriente-RAM
PA19	AD2	CA/D	Conector Der
PA20	AD3	CA/D	Conector Der
PA21	PA21	Línea I/O	No utilizada
PA22	PA22	Línea I/O	No utilizada
PA23	NUB	Upper Byte Select	Byte High Enable SRAM
PA24	PA24	Línea I/O, TX0RTS	Transmit Buffer Request to Send CAN
PA25	PA25	Línea I/O, TX1RTS	Transmit Buffer Request to Send CAN
PA26	PA26	Línea I/O, TX2RTS	Transmit Buffer Request to Send CAN
PA27	PA27	Línea I/O	No utilizada
PA28	PA28	Línea I/O	No utilizada
PA29	PA29	Línea I/O	No utilizada
PA30	IRQ1	Interrupción Externa	INT/CAN
PA31	CS#1	SPI Chip Select 1	Bus SPI

Tabla A.1 Señales del AT91SAM7SE512 en la CV (puerto A).

Línea I/O	Periférico	Descripción	Correspondencia
PB0	TIOA0	Timer-Counter	Conector Izq
PB1	TIOB0	Timer-Counter	Conector Izq
PB2	SCK0	USART0 Serial Clock	Conector Izq
PB3	A2	Bus de Dirección	SRAM
PB4	A3	Bus de Dirección	SRAM
PB5	A4	Bus de Dirección	SRAM
PB6	A5	Bus de Dirección	SRAM
PB7	A6	Bus de Dirección	SRAM
PB8	A7	Bus de Dirección	SRAM
PB9	A8	Bus de Dirección	SRAM
PB10	A9	Bus de Dirección	SRAM
PB11	A10	Bus de Dirección	SRAM
PB12	A11	Bus de Dirección	SRAM
PB13	A12	Bus de Dirección	SRAM
PB14	A13	Bus de Dirección	SRAM
PB15	A14	Bus de Dirección	SRAM
PB16	A15	Bus de Dirección	SRAM
PB17	A16	Bus de Dirección	SRAM
PB18	PCK2	Programmable Clock Output	Conector Der
PB19	FIQ	Fast Interrupt Input	No utilizada
PB20	IRQ0	External Interrupt Input	No utilizada
PB21	PCK1	Programmable Clock Output	Conector Der
PB22	CS#3	Chip Select 3	Bus SPI
PB23	PWM0	PWM	Conector Der
PB24	PWM1	PWM	Conector Der
PB25	PWM2	PWM	Conector Der
PB26	PB26	Línea I/O	No utilizada
PB27	PB27	Línea I/O	No utilizada
PB28	TCLK1	External Clock Input	Conector Der
PB29	PB29	Línea I/O	No utilizada
PB30	CS#2	Chip Select 2	Bus SPI
PB31	PB31	Línea I/O	Conector Der

Tabla A.2 Señales del AT91SAM7SE512 en la CV (puerto B).

Línea I/O	Periférico	Descripción	Correspondencia
PC0	D0	Bus de Datos	SRAM
PC1	D1	Bus de Datos	SRAM
PC2	D2	Bus de Datos	SRAM
PC3	D3	Bus de Datos	SRAM
PC4	D4	Bus de Datos	SRAM
PC5	D5	Bus de Datos	SRAM
PC6	D6	Bus de Datos	SRAM
PC7	D7	Bus de Datos	SRAM

Tabla A.3 Señales del AT91SAM7SE512 en la CV (puerto C).

Línea I/O	Periférico	Descripción	Correspondencia
PC8	D8	Bus de Datos	SRAM
PC9	D9	Bus de Datos	SRAM
PC10	D10	Bus de Datos	SRAM
PC11	D11	Bus de Datos	SRAM
PC12	D12	Bus de Datos	SRAM
PC13	D13	Bus de Datos	SRAM
PC14	D14	Bus de Datos	SRAM
PC15	D15	Bus de Datos	SRAM
PC16	A17	Bus de Dirección	SRAM
PC17	A18	Bus de Dirección	SRAM
PC18	A19	Bus de Dirección	SRAM
PC19	A20	I/O, Bus de direcciones	No utilizado
PC20	A21	I/O, Bus de direcciones	No utilizado
PC21	NWE	Write enable	Write Enable SRAM
PC22	NRD	Read Signal	Output Enable SRAM
PC23	NCS0	Chip Enable	Chip Enable SRAM

Tabla A.3 Señales del AT91SAM7SE512 en la CV (puerto C) (continuación).

Línea	Pines	Descripción
VDDIN	8	Regulador de voltaje y ADC Input
VDDOUT	7	Regulador de voltaje Output
VDDFLASH	123	Fuente de Flash y USB
VDDIO	71, 46, 21, 95, 120	Fuente de líneas I/O
VDDCORE	23, 47, 72, 96, 118,	Fuente de núcleo
VDDPLL	128	PLL
GND	2, 22, 45, 70, 94, 119, 124	Tierra
XIN	125	Reloj
XOUT	126	Reloj
PLLRC	127	PLL
PCK0-PCK2	49	Reloj de salida PB18
TCK	76	JTAG Test Clock
TDI	65	JTAG Test Data In
TDO	66	JTAG Test Data Out
TMS	77	JTAG Test Mode Select
JTAGSEL	78	JTAG Selection
ERASE	75	Borra la memoria Flash
NRST	73	Resetea el microcontrolador
TST	74	Test Mode Select
ADVREF	1	ADC Reference
AD7-AD4	3,4,5,6	ADC Input

Tabla A.4 Señales del AT91SAM7SE512 en la CV.

Apéndice B

B.1 Programa asociado a la comunicación SPI

El programa que se utilizó para realizar la simulación de la comunicación SPI es el siguiente:

```
#include <16f876a.h>
#fuses XT,NOVDT,NOPROTECT,NOLVP,PUT,BROWNOUT
#use delay(clock=4000000) // 4 MHz
#include <lcd.c>
#define ss pin_A1 // Chip select de la memoria

// Comando WREN 0x06 Habilita la escritura
void habilitarescritura() {
    lcd_gotoxy(1,1);
    printf(lcd_putc,"Comando WREN");
    lcd_gotoxy(1,2);
    printf(lcd_putc,"Admite Escribir");
    delay_ms(1000);
    lcd_init();
    delay_ms(100);
    output_high(ss);
    output_low(ss);
    spi_write(0x06); // Función de escritura spi del compilador
    delay_us(1);
    output_high(ss);
    delay_us(1);
}

// Comando 0x05 RDSR Lee Estatus
byte leerestatus() {
    byte datarec;
    lcd_gotoxy(1,1);
    printf(lcd_putc,"Comando RDSR");
    lcd_gotoxy(1,2);
    printf(lcd_putc,"Lee Estatus");
    delay_ms(1000);
    lcd_init();
    output_low(ss);
    spi_write(0x05);
    datarec=spi_read(0); // Función de lectura spi del compilador
    delay_us(1);
    output_high(ss);
    lcd_gotoxy(1,1);
    printf(lcd_putc,"Estatus ""%X", datarec);
    delay_ms(1000);
    lcd_init();
    return datarec;
}
```

```

// Comando 0x01 WRSR Escribe Estatus
escribirestatus(byte a) {
byte datar;
lcd_gotoxy(1,1);
printf(lcd_putc,"Comando WRSR");
lcd_gotoxy(1,2);
printf(lcd_putc,"Escribe Estatus");
delay_ms(1000);
lcd_init();
output_low(ss);
spi_write(0x01);
spi_write(a);
delay_us(1);
output_high(ss);
}

void main () {
// Inicio de variables
byte i,datar;
byte page[32];
datar;
lcd_init();
setup_spi(spi_master | SPI_L_TO_H); // Modo de operación SPI 1, polaridad y fase de reloj 0

// Generación de pagina de datos
for(i=0; i<32; i++) {
page[i]=i;
}

// Comando WRITE 0x02 Escritura de Página
habilitaescritura();
lcd_gotoxy(1,1);
printf(lcd_putc,"Comando WRITE");
lcd_gotoxy(1,2);
printf(lcd_putc,"Escribe Pagina");
delay_ms(1000);
lcd_init();
output_low(ss);
spi_write(0x02);
spi_write(0x00); //Dirección High, 8 bits más significativos de la dirección 0x0000
spi_write(0x00); //Dirección Low 8 bits menos significativos de la dirección 0x0000
for(i=0;i<32;i++) {
spi_write(page[i]); // Página a escribir
}
delay_us(1);
output_high(ss);
delay_ms(8);

// Comando READ 0x03 Lee Página
lcd_gotoxy(1,1);
printf(lcd_putc,"Comando READ");
lcd_gotoxy(1,2);
printf(lcd_putc,"Lee Pagina");
delay_ms(1000);
lcd_init();

```

```

output_low(ss);
spi_write(0x03);
spi_write(0x00); //Dirección High 8 bits más significativos de la dirección 0x0000
spi_write(0x00); //Dirección Low 8 bits menos significativos de la dirección 0x0000
for(i=0;i<32;i++) {
page[i]=spi_read(0); // Dato Leído
}
delay_us(1);
output_high(ss);
delay_ms(6);
lcd_init();
for(i=0;i<32; i++) {
lcd_gotoxy(1,1);
printf(lcd_putc,"Leido ""%X", page[i]);
lcd_gotoxy(1,2);
printf(lcd_putc,"Direccion ""%X", i);
delay_ms(1000);
}

habilitarescritura();
escribirestatus(0x08);
datar=leerestatus();
} // Fin de programa

```

B.2 Programa asociado a los sensores de temperatura 1-Wire

El programa utilizado para la simulación de los sensores de temperatura 1-Wire es el siguiente:

```

#include <16f876a.h>
#fuses XT,NOWDT,NOPROTECT,NOLVP,PUT,BROWNOUT
#use delay(clock=4000000) // 4 MHz
#define DQ pin_c0 // Línea asociada al bus 1-Wire
#include <lcd.c>

byte ROM_Code[8][8]; // Arreglo que almacena el ROM Code de los sensores

void inicializar ( ) { // Inicializa el sensor 1-Wire
output_low(DQ);
delay_us(485);
output_float(DQ);
delay_us(65);
delay_us(420); // En DQ debe de haber cero.
}

void enviaUno ( ) { // Envía el bit 1 por el bus 1-Wire
delay_us(10);
output_low(DQ);
delay_us(10);
output_float(DQ);
delay_us(50);
}

void enviaCero ( ) { // Envía el bit 0 por el bus 1-Wire

```



```

delay_us(10);
output_low(DQ);
delay_us(80);
output_float(DQ);
}

void enviaByte (byte dato) { // Envía 1 byte por el bus 1-Wire
byte a;
for (a=0;a<8;a++) {
if(shift_right(&dato,1,0))
enviaUno();
else
enviaCero();
}
}

byte leebyte (byte temp) { // Lee 1 byte por el bus 1-Wire
delay_us(10);
output_low(DQ);
delay_us(2);
output_float(DQ);
delay_us(5);
shift_left(&temp,1,input(DQ));
delay_us(45);
return temp;
}

byte leeByte ( ) { // Lee 1 byte por el bus 1-Wire
byte a,dato;
for(a=0;a<8;a++) {
delay_us(10);
output_low(DQ);
delay_us(2);
output_float(DQ);
delay_us(5);
shift_right(&dato,1,input(DQ));
delay_us(45);
}
return dato;
}

void buscar(int num) { // Busca ROM Codecs

int i, j, k, discrep, lzero, number_bit;
int interior, vdis;
byte temp, result, valor_bit, complemento_bit;

discrep=0; // Inicio de variables y LCD
number_bit=0;
lzero=0;
vdis=0;
temp=0x00;
result=0x00;
lcd_init();

```

```

for(k=0;k<num; k++) { // El número de búsqueda es k+1
number_bit=1;
lzero=0;

inicializar( ); // Inicializar el sensor 1-Wire
enviaByte(0xF0); // Comando Search ROM

for(i=0;i<8;i++) {
interior=0;
for(j=0;j<8;j++){

temp=0x00;
valor_bit=leebit(temp);
temp=0x00;
complemento_bit=leebit(temp);

// Caso de error
if(valor_bit==1 && complemento_bit==1) {
discrep=0;
lcd_init();
lcd_gotoxy(1,1);
printf(lcd_putc,"Error");
lcd_gotoxy(1,1);
printf(lcd_putc,"%X ""%X ""%X ",k,i,number_bit);
delay_ms(5000);
}

// Casos de discrepancia
else if(valor_bit==0x00 && complemento_bit==0x00) {

if(number_bit>discrep){
enviaCero();
bit_clear(result,interior); // Almacena el valor del bit encontrado
lzero=number_bit;
number_bit++;
interior++;
} else if(number_bit==discrep){
enviaUno();
bit_set(result,interior); // Almacena el valor del bit encontrado
number_bit++;
interior++;
} else if(number_bit<discrep && bit_test(ROM_Code[k-1][i], interior)==0){
enviaCero();
bit_clear(result,interior);
lzero=number_bit;
number_bit++;
interior++;
} else if(number_bit<discrep && bit_test(ROM_Code[k-1][i], interior)==1){
enviaUno();
bit_set(result,interior);
number_bit++;
interior++;
}
}

} // Final discrepancia

```

```

// Casos sin discrepancia

else if(valor_bit==0 && complemento_bit==1) {
enviaCero();
bit_clear(result,interior);           // Almacena el valor del bit encontrado
number_bit++;
interior++;
}

else if(valor_bit==1 && complemento_bit==0) {
enviaUno();
bit_set(result,interior);             // Almacena el valor del bit encontrado
number_bit++;
interior++;
}
}
// Final 1er For
ROM_Code[k][i]=result;
result=0x00;
interior=0;
}
// Final 2do for, búsqueda´
discrep=lzero;
}
// Final algoritmo de búsqueda

// Imprimir Datos
for(i=0;i<num;i++) {
for(j=0;j<8;j++) {
lcd_gotoxy(2*j+1,1);
printf(lcd_putc,"%X",ROM_Code[i][7-j]);
}
lcd_gotoxy(1,2);
printf(lcd_putc,"ROM_Code ""%d",i+1);
delay_ms(5000);
}
}

void temperaturas(int num) {          // Toma la lectura de temperaturas
int i,j, valor;
byte t1sb, t5sb;

lcd_init();
lcd_gotoxy(1,1);
printf(lcd_putc,"ROM_Code ");
lcd_gotoxy(1,2);
printf(lcd_putc,"Temp");
delay_ms(2000);
lcd_init();
inicializar( );                      // Inicializar el sensor 1-Wire
enviaByte(0xCC);                      // Comando Skip Rom
enviaByte(0x44);                      // Comando Convert T
delay_ms(1000);                       // Esperar Conversión 800 ms mínimo
for(i=0; i<num; i++) {
lcd_init();
delay_ms(100);
inicializar( );
enviaByte(0x55);                      // Comando Match Rom
}
}

```

```

for(j=0;j<8;j++){
enviaByte(ROM_Code[i][j]);          // Envía ROM Code
lcd_gotoxy(2*j+1,1);
printf(lcd_putc,"%X",ROM_Code[i][7-j]);
}

enviaByte(0xBE);                     // Comando Read Scratchpad
tlsb=leeByte();
tmsb=leeByte();
inicializar( );

if(tmsb==00) {                       // Imprime Valores
lcd_gotoxy(1,2);
valor=tlsb/2;
printf(lcd_putc,"%u",valor);
printf(lcd_putc," C");
delay_ms(5000);
} else {
lcd_gotoxy(1,2);
valor=(256-tlsb)/2;
printf(lcd_putc,"-");
printf(lcd_putc,"%u",valor);
printf(lcd_putc," C");
delay_ms(5000);
lcd_init();
}
}
}

main () {                             // Función principal
int cant, i, j;
for (i=0;i<8;i++) {
for (j=0;j<8;j++) {
ROM_Code[i][j]=0x00;
}
}
cant=5;                               // Es la cantidad de sensores en la CV

buscar(cant);
temperaturas(cant);
}

```

B.3 Programa asociado a los sistemas de protección efecto latch-up

El programa que se empleó en la simulación de los circuitos de protección de efecto latch-up es el siguiente:

```

#include <16f876A.h>
#define adc=8
#define fuses XT,NOWDT,NOPROTECT,NOLVP,PUT,BROWNOUT
#define use delay(clock=4000000)
#include <lcd.c>

```

```

void main () {
float value, result;
lcd_init();
setup_adc_ports(RA0_ANALOG);
set_adc_channel(0);
delay_ms(500);
value=read_adc();           // Conversion A/D
result=value*0.0196;
lcd_gotoxy(1,1);
printf(lcd_putc, "C Energia");
if(result>=3.2){
lcd_gotoxy(1,2);
printf(lcd_putc, "Advertencia");
} else {
lcd_gotoxy(1,2);
printf(lcd_putc, "Exit");
}
}
}

```

Referencias

- [1] Aalborg University. AAU CUBE SAT [en línea]. [s.l.] Aalborg University, 2001. [fecha de consulta noviembre 2008]. Disponible en: <http://www.cubesat.auc.dk/>.
- [2] Aalborg University. AAUSAT II [en línea]. [s.l.] Aalborg University, 2005. [fecha de consulta noviembre 2008]. Disponible en: <http://aausatii.space.aau.dk/homepage/index.php?language=en&page=flash>.
- [3] Aalborg University. AAUSAT3 [en línea]. [s.l.] Aalborg University, 2009. [fecha de consulta enero 2009]. Disponible en: <http://www.aausat3.space.aau.dk/>.
- [4] Atmel Corporation. SAM Boot Assistant (SAM-BA): User Guide [en línea]. Revisión 6132C, octubre 2006. [s.l.] Atmel Corporation, 2006. [fecha de consulta enero 2009]. Disponible en: http://www.replicators.eu/tiki-download_file.php?fileId=19
- [5] Atmel Corporation. AT91 In-system Programmer (ISP): User Guide [en línea]. Revisión 6224D, Julio 2006. [s.l.] Atmel Corporation, 2006. [fecha de consulta febrero 2009]. Disponible en: http://www.datasheetpro.com/584193_download_AT91_datasheet.html
- [6] Atmel Corporation. AT91SAM7SE512 Data Sheet [en línea]. Revisión 6222D, enero 2008. [s.l.] Atmel Corporation, 2008. [fecha de consulta septiembre 2008]. Disponible en: http://www.atmel.com/dyn/resources/prod_documents/doc6222.pdf
- [7] Bosch-Robert. CAN Specification [en línea]. Stuttgart, Alemania. Robert Bosch GMBH, 1991. [fecha de consulta enero 2009]. Disponible en: <http://www.semiconductors.bosch.de/pdf/can2spec.pdf>
- [8] Cypress Semiconductor Corporation. CY62167DV30 MoBL [en línea]. San Jose, CA. Cypress Semiconductor Corporation, 2006. [fecha de consulta enero 2009]. Disponible en: http://download.cypress.com.edgesuite.net/design_resources/datasheets/contents/cy62167dv30ll_8.pdf
- [9] D. Jeff Dionea. Embedded Linux/Microcontroller Project [en línea]. [s.l.] D. Jeff Dionea, 1998. [fecha de consulta noviembre 2008]. Disponible en: <http://www.uclinux.org/>.
- [10] EyasSat, The Classroom Satellite. [fecha de consulta noviembre 2008]. Disponible en: <http://www.eyassat.com/>.
- [11] Ginsberg-Gerald L. *Printed Circuit Design*. EEUU. McGraw – Hill. 1991. pp 113-132.
- [12] Gómez Islas- Francisco J. Computadora de vuelo para un sistema de capacitación de recursos humanos en el manejo de satélites. Tesis (licenciatura en ingeniería). México DF. Universidad Nacional Autónoma de México. 2007. pp 76-115.

- [13] Infineon Technologies AG. XE164, 16-Bit Single-Chip Real Time Signal Controller [en línea]. Edición 2008-08. 81726 Munich, Alemania. Infineon Technologies AG, 2008. [fecha de consulta noviembre 2008]. Disponible en: http://www.infineon.com/dgdl/xe164x_ds_v2.1_2008_08.pdf?folderId=db3a3043156fd5730115b3a665650d23&fileId=db3a3043156fd57301161007ff671b4d
- [14] INSTITUTO DE INGENIERÍA, UNAM. SATEDU: Satélite Educativo Mexicano [en línea]. México DF. INSTITUTO DE INGENIERÍA, 2009. [fecha de consulta junio 2009]. Disponible en: <http://proyectos.iingen.unam.mx/satedu/Default.htm>.
- [15] López, E. Protocolo SPI(Serial Peripheral Interface) [en línea]. México DF. Ingeniería en Microcontroladores. [fecha de consulta febrero 2009]. Disponible en: <http://www.i-micro.com/pdf/articulos/spi.pdf>
- [16] Maxim Integrated Products. DS18S20, High-Precision 1-Wire Digital Thermometer [en línea]. Revisión 04/22/08. 120 San Gabriel Drive, Sunnyvale, CA. Maxim Integrated Products, 2008. [fecha de consulta marzo 2009]. Disponible en: <http://datasheets.maxim-ic.com/en/ds/DS18S20.pdf>
- [17] Maxim Integrated Products. MAXIM Bidirectional, High-Side, Current-Sense Amplifiers with Reference MAX4069–MAX4072 [en línea]. Revision 3. 120 San Gabriel Drive, Sunnyvale, CA. Maxim Integrated Products, 2008. [fecha de consulta abril 2009]. Disponible en: <http://datasheets.maxim-ic.com/en/ds/MAX4069-MAX4072.pdf>
- [18] Microchip Technology Incorporated. 25AA080A/B, 25LC080A/B, 8K SPI™ Bus Serial EEPROM [en línea]. [s.l.] Microchip Technology Incorporated, 2003. [fecha de consulta julio 2009]. Disponible en: <http://ww1.microchip.com/downloads/en/DeviceDoc/21808D.pdf>
- [19] Microchip Technology Incorporated. MCP2515, Stand-Alone CAN Controller With SPI Interface [en línea]. Revisión noviembre 2007. [s.l.] Microchip Technology Incorporated, 2007. [fecha de consulta mayo 2009]. Disponible en: <http://ww1.microchip.com/downloads/en/DeviceDoc/21801e.pdf>
- [20] Microchip Technology Incorporated. MCP2551, High-Speed CAN Transceiver [en línea]. Revisión enero 2007. [s.l.] Microchip Technology Incorporated, 2007. [fecha de consulta mayo 2009]. Disponible en: <http://ww1.microchip.com/downloads/en/DeviceDoc/21667E.pdf>
- [21] Microchip Technology Incorporated. PIC24HJXXXGPX06/X08/X10 Data Sheet [en línea]. Revisión marzo 2009. [s.l.] Microchip Technology Incorporated, 2009. [fecha de consulta abril 2009]. Disponible en: <http://ww1.microchip.com/downloads/en/DeviceDoc/70175H.pdf>
- [22] National Semiconductor Corporation. LM6511 180 ns 3V Comparator [en línea]. National Semiconductor Corporation, 2003. [fecha de consulta febrero 2009]. Disponible en: <http://www.national.com/ds/LM/LM6511.pdf>

[23] Renesas Technology Corp. H8SX/1544 Group Hardware Manual [en línea]. Revisión 2007. Ote-machi, Chiyoda-Ku, Tokio, Japon. Renesas Technology Corp, 2007. [fecha de consulta diciembre 2008]. Disponible en:

http://documentation.renesas.com/eng/products/mpumcu/rej09b0381_h8sx1544hm.pdf

[24] Siemens AG. Microcomputer Components SAB 80C166/83C166 16 - Bit CMOS Single -Chip Microcontrollers for Embedded Applications [en línea]. Edición 06.90 / 08.97. Balanstraße 73, 81541 München. Siemens AG, 1997. [fecha de consulta octubre 2008].

Disponible en:

<http://www.infineon.com/dgdl/m166.pdf?folderId=db3a304412b407950112b415bf961eaf&fileId=db3a304412b407950112b41db27d30b8>

[25] Spansion Incorporated. S25FL128P, 128 Megabit CMOS 3.0 Volt Flash Memory with 104-MHz SPI (Serial Peripheral Interface) Bus [en línea]. Revision mayo 2009 Spansion Incorporated, 2009. [fecha de consulta junio 2009]. Disponible en:

http://www.spansion.com/Products/Support/Datasheets/S25FL128P_00_07_e.pdf

[26] Texas Instruments Incorporated. SN54HC42, SN74HC42, 4-LINE TO 10-LINE DECODERS [en línea]. Revision septiembre 2003. Texas Instruments Incorporated, 2003.

[fecha de consulta enero 2009]. Disponible en:

<http://focus.ti.com/lit/ds/scls091d/scls091d.pdf>

[27] The CubeSat Program California Polytechnic State University. CubeSat Design Specification [en línea]. Revisión 11, 10/02/08. San Luis Obispo, CA 93407. The CubeSat Program California Polytechnic State University, 2008. [fecha de consulta diciembre 2008]. Disponible en: http://cubesat.atl.calpoly.edu/media/CDS_rev11.pdf

[28] The Xmodem Protocol Specification [en línea]. [fecha de consulta enero 2009].

Disponible en: http://www.menie.org/georges/embedded/xmodem_specs.txt.

[29] Torres Portero-Manuel. *Diseño e Ingeniería Electrónica asistida con Protel DXP*. México. Alfaomega. 2005. pp 1-294.

[30] Universidad Würzburg. UWE-1 [en línea]. Am Hubland, 97074 Würzburg. Universidad Würzburg, 2009. fecha de consulta diciembre 2008]. Disponible en:

http://www7.informatik.uni-wuerzburg.de/en/research/space_exploration/projekte/cubesat/uwe-1/.

[31] Universidad Würzburg. UWE-1 [en línea]. Am Hubland, 97074 Würzburg. Universidad Würzburg, 2009. fecha de consulta diciembre 2008]. Disponible en:

http://www7.informatik.uni-wuerzburg.de/en/research/space_exploration/projekte/cubesat/uwe-2/.