



MÓDULO DE CONTROL PRINCIPAL PARA CUBESAT

Martín Ardao, Mariángel Fernández, Mercedes Mato

Tutor: Ing. Juan Pechiar

Instituto de Ingeniería Eléctrica

Facultad de Ingeniería

Universidad de la República



MÓDULO DE CONTROL PRINCIPAL PARA CUBESAT

Martín Ardao, Mariángel Fernández, Mercedes Mato

Tutor: Ing. Juan Pechiar

Instituto de Ingeniería Eléctrica

Facultad de Ingeniería

Universidad de la República

Resumen

El propósito de este trabajo es diseñar un sistema de control principal para un nanosatélite CubeSat, y desarrollar una posible implementación en software y hardware de las tareas más importantes que lleva a cabo el módulo.

En la documentación se presenta una descripción exhaustiva de la arquitectura elegida para la implementación de las distintas funciones, entre las cuales se encuentran la comunicación entre módulos del sistema, la comunicación con Tierra, y la gestión de salud del satélite.

El estudio se extiende al diseño e implementación de un circuito impreso capaz de poner en funcionamiento todo lo desarrollado por software, para luego culminar con un detalle de las pruebas realizadas en la etapa de testeo del sistema construido.

Finalmente, se exponen los resultados obtenidos y se dejan planteadas futuras líneas de trabajo, que pueden ayudar a una implementación posterior del sistema, más robusta y compleja.

Tabla de contenidos

Tabla de contenidos	1
Agradecimientos	4
1. Introducción	5
2. Conceptos generales y antecedentes	6
2.1. Cubesat	6
2.2. Módulo de lanzamiento, P-POD	8
2.3. Vehículos de lanzamiento (LV)	9
2.4. CubeSats lanzados	10
3. Objetivos generales	12
4. Especificación funcional	13
4.1. Descripción general	14
4.2. Ejecución de tareas periódicas y tareas asíncronas	15
4.3. Monitoreo y control del sistema	16
4.4. Comunicación entre la estación terrena y el satélite	17
5. Diseño e implementación del sistema	19
5.1. Diseño del software	19
5.1.1. Scheduler	20
5.1.1.1. Ejecución de las tareas periódicas	21
5.1.1.2. Ejecución de tareas asíncronas	23
5.1.2. Mediciones de temperatura	25
5.1.3. Reloj de tiempo real	28
5.1.4. Protocolo de comunicación entre módulos	31
5.1.4.1. Protocolo de comunicación del satélite	31
5.1.4.2. Protocolo I2C	33
5.1.4.3. Módulo I2C del MSP430F5438A	34
5.1.4.4. Diagrama de flujo de la comunicación	42

5.1.5.	Modo de bajo consumo	47
5.1.6.	Housekeeping	52
5.1.7.	Watchdog	54
5.1.8.	Protocolo de comunicación entre la estación terrena y el satélite	56
5.1.8.1.	Estructura de los paquetes de datos	56
5.1.8.2.	Encriptado de paquetes de datos	57
5.1.8.3.	Modulación	57
5.1.8.4.	Desencriptado	57
5.1.8.5.	Diagramas de flujo de la comunicación	58
5.2.	Diseño del hardware	60
5.2.1.	Elección del microcontrolador	60
5.2.2.	Conexionado de los módulos	62
5.2.3.	Diseño del circuito impreso de prueba	63
5.2.3.1.	Conexión entre microcontroladores	65
5.2.3.2.	Sensor de temperatura	66
5.2.3.3.	Oscilador externo	68
5.2.3.4.	Interfaz de programación y testeo JTAG	69
5.2.3.5.	Conexiones para fotodiodos	73
5.2.3.6.	Conexiones para Magnetómetro	75
5.2.3.7.	Puentes H y conexiones para magnetotorquers	77
5.2.3.8.	Implementación del circuito impreso	79
6.	Pruebas	82
6.1.	Combinación de rutinas de escritura y lectura en un mismo dispositivo.	84
6.2.	Cambio de sentido del flujo de datos	84
6.3.	Versión final del protocolo	85
6.4.	Pruebas ADC	85
6.5.	Pruebas RTC	86
7.	Trabajo a futuro	87
8.	Conclusiones	90
	Bibliografía	91
	Glosario	94
	Anexo	95

A. Módulos que componen el satélite	97
A.1. P-SAT: Desarrollo del sistema de gestión de energía para un satélite	97
A.2. Proyecto Laí Sistema de Determinación de Actitud	98
A.3. SisComSat: Sistema de comunicación para nanosatélite	99
B. Configuración ADC	101
C. Configuración RTC	103
D. Configuración I2C	104
E. Análisis de costos	110
F. Hojas de datos	114
Índice de tablas	150
Índice de figuras	152

Agradecimientos

A nuestras familias, amigos, tutor y todos aquellos que colaboraron con la realización de este proyecto.

Capítulo 1

Introducción

El Proyecto Laí [1] es un emprendimiento promovido por la Facultad de Ingeniería que consiste en diseñar, construir y operar un satélite, utilizando el estándar CubeSat.

Durante 2008 y 2009, se liberaron cuatro globos sonda con sistemas de telemetría, instrumentación y carga científica, siendo los mismos desarrollados por estudiantes de la Facultad de Ingeniería. Luego, se comenzó con la construcción de los distintos módulos que en un futuro conformarán el nanosatélite, algunos de estos proyectos ya finalizaron.

Originalmente éste proyecto se encontraba comprendido en el marco del Proyecto Laí y actualmente forma parte de un proyecto más ambicioso llamado Antel Sat, cuyos objetivos abarcan los del Proyecto Laí sumados a la posibilidad de tomar imágenes desde el satélite.

El proyecto MCPC, consiste en diseñar e implementar el módulo de control principal de un nanosatélite del tipo CubeSat.

Capítulo 2

Conceptos generales y antecedentes

2.1. Cubesat

CubeSat es un nanosatélite utilizado para la investigación espacial. El estándar CubeSat nace de la colaboración entre California Polytechnic State University (Cal Poly) [2] y Stanford University en 1999. Dentro de sus objetivos se encuentran el generar las especificaciones necesarias para la construcción y lanzamientos de nanosatélites (CubeSat), de forma de permitir a universidades de todo el mundo adentrarse en la ciencia y exploración espacial disminuyendo los costos involucrados. Los satélites CubeSat están pensados para ser construidos con componentes y materiales comerciales accesibles al público en general.

El estándar limita el volumen de un módulo CubeSat a un cubo de 10cm x 10cm x 10cm. De la misma forma limita el peso de dicho módulo a 1 kilogramo. Este módulo cúbico, es conocido como $1U$ CubeSat. Un CubeSat puede estar formado de hasta 3 módulos, agrupados a lo largo de uno sólo de sus ejes. Estos satélites de más de un módulo son conocidos como $2U$ CubeSat (satélite de 2 módulos) o $3U$ CubeSat (satélite de 3 módulos). En la figura 2.1 se pueden apreciar los módulos antes mencionados.

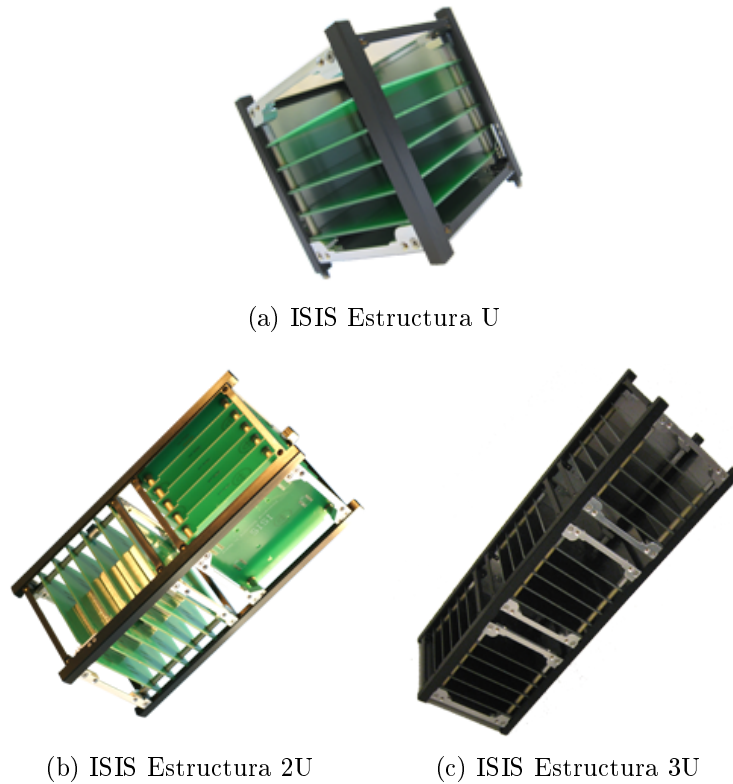


Figura 2.1: Tipos de estructuras

1

Dentro de las especificaciones desarrolladas por Cal Poly, se encuentran definiciones acerca de los materiales a utilizar en este tipo de satélite, por ejemplo, el material de los rieles que estará en contacto con el módulo de lanzamiento. El estándar especifica además como debe comportarse el CubeSat, antes, durante y luego de su despliegue, para asegurar un lanzamiento exitoso del mismo y asegurar la no interferencia con el resto de los satélites desplegados.

La mayor parte de los desarrollos de CubeSat corresponden a universidades de todo el mundo, aunque existen desarrollos de empresas privadas, tales como Boeing, y agencias gubernamentales como la NASA. Actualmente más de 40 universidades del mundo colaboran con el proyecto CubeSat.

¹Imágen disponible en: <http://www.clyde-space.com>

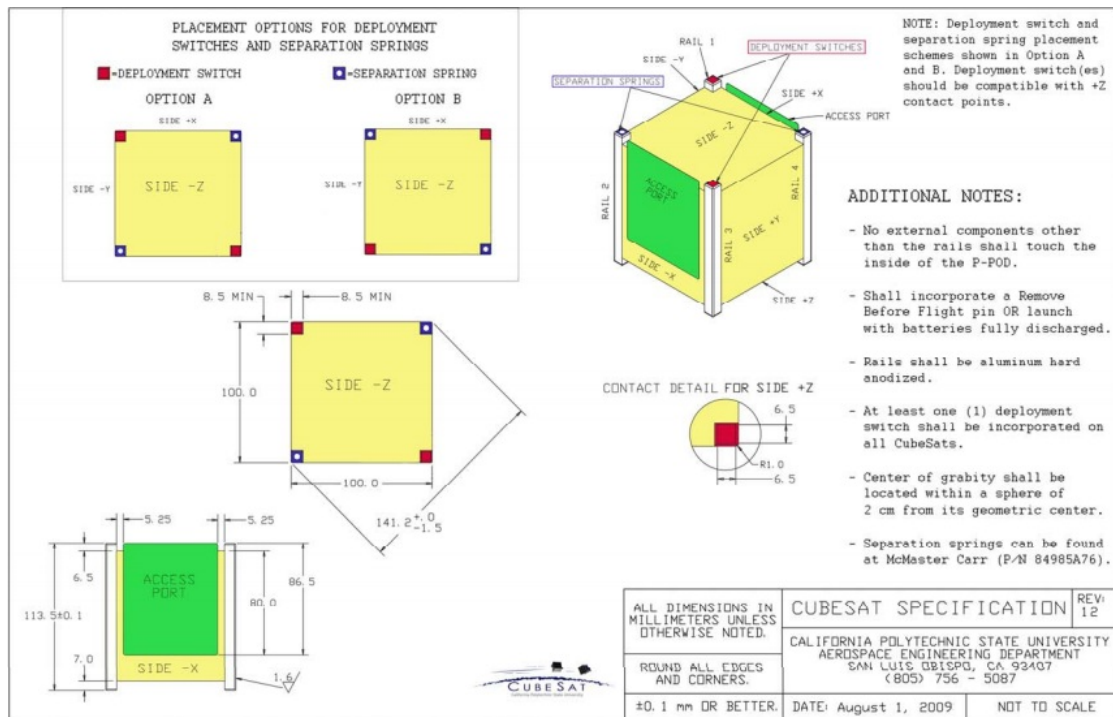


Figura 2.2: Especificaciones para fabricación de CubeSat

2

2.2. Módulo de lanzamiento, P-POD

Con el objetivo de reducir los costos asociados a los lanzamientos de los satélites de tipo CubeSat, Cal Poly desarrolló un módulo de lanzamiento llamado P-POD (Poly-PicoSatellite Orbital Deployer). Este módulo permite el lanzamiento de varios CubeSats de forma simultánea, tanto de $1U$, $2U$ o $3U$, o cualquier combinación de estos, siempre que en su conjunto no se superen las $3U$ de volumen. El P-POD consiste en una estructura tubular de aluminio la cual alberga los módulos CubeSat durante el lanzamiento. La estructura posee cuatro rieles de aluminio y una lanzadera de resorte para permitir el despliegue de los nanosatélites.

Una vez que el vehículo de lanzamiento se encuentra en el espacio, el módulo P-POD, abre su compuerta y mediante la lanzadera de resorte libera los módulos CubeSat. Este tipo de módulo ha lanzado alrededor del 90 % del total de los CubeSAT, y el 100 % de los mismos desde 2006.

²Imágen disponible en: www.cubesat.org

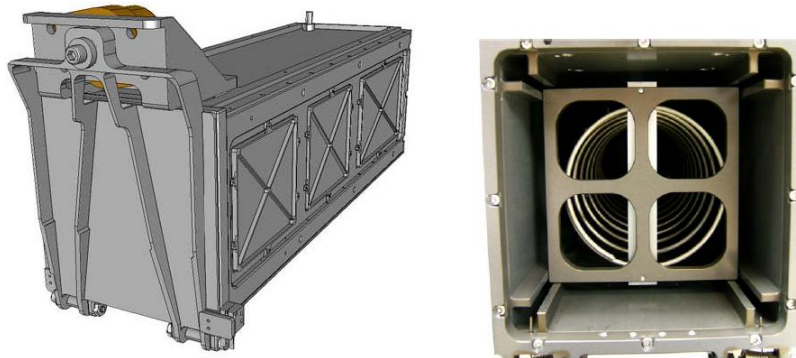


Figura 2.3: Módulo P-POD

3

2.3. Vehículos de lanzamiento (LV)

El transporte de los nanosatélites CubeSat, dentro de los módulo P-POD, se realiza utilizando cohetes como vehículos de lanzamiento. Actualmente ISC Kosmotras y Eurokot son dos de las empresas que ofrecen servicios de lanzamientos.



Figura 2.4: Vehículo de lanzamiento

4

³Imágen disponible en: www.cubesat.org

⁴Imágen disponible en: itunews.itu.int

2.4. CubeSats lanzados

En el cuadro 2.1 se presentan algunos de los CubeSats que han sido lanzados hasta la fecha [3].

Nombre	Tipo	Organización	Misión	Lanzamiento
CanX-1	1U	UTIAS	Pruebas de tecnología	2003-06-30
DTUsat	1U	DTU	Investigación de un tether	2003-06-30
AAU CubeSat	1U	Aalborg University	Captura de imágenes	2003-06-30
QuakeSat	3U	Stanford University	Detección de Terremotos	2003-06-30
CUTE-I (Oscar 55)	1U	Tokyo Institute of Technology	Radio Amateur	2003-06-30
Cubesat XI-IV (Oscar 57)	1U	University of Tokyo	Radio Amateur	2003-06-30
Cubesat XI-V (Oscar 58)	1U	University of Tokyo	Radio Amateur	2005-10-27
nCube-2	1U	ARR/NSC	Radio Amateur	2005-10-27
UWE-1	1U	University of Würzburg	Tecnología y Comunicación	2005-10-27
SACRED	1U	University of Arizona	-	2006-07-26
ION	2U	University of Illinois	-	2006-07-26
Rincon 1	1U	University of Arizona	-	2006-07-26
ICE Cube 1	1U	Cornell University	-	2006-07-26
KUTESat	1U	University of Kansas	-	2006-07-26
NCUBE-1	1U	ARR/NSC	-	2006-07-26
HAUSAT-1	1U	Hankuk Aviation University	-	2006-07-26
SEEDS-1	1U	Nihon University	-	2006-07-26
CP-2	1U	California Polytechnic University	-	2006-07-26
AeroCube 1	1U	Aerospace Corporation	-	2006-07-26
MEROPE	1U	Montana State University	-	2006-07-26
Mea Huakai (Voyager)	1U	University of Hawaii	-	2006-07-26
ICE Cube 2	1U	Cornell University	-	2006-07-26
CP-1	1U	California Polytechnic University	-	2006-07-26
GeneSat-1	3U	NASA-Santa Clara University	Investigación Biológica y Pruebas de Tecnología	2006-12-16
CSTB1	1U	Boeing	-	2007-04-17
AeroCube 2	1U	Aerospace Corporation	-	2007-04-17
CP4	1U	California Polytechnic University	-	2007-04-17
Libertad-1	1U	Sergio Arboleda University	-	2007-04-17
CAPE-1	1U	University of Louisiana at Lafayette	-	2007-04-17
CP-3	1U	California Polytechnic University	-	2007-04-17
MAST	1U	Tethers Unlimited	Tether experiments	2007-04-17
Cute-1.7 + APD II	1U	Tokyo Institute of Technology	Demostración de sistema de separación y experimentos con foto diodos Avalanche	2008-04-28
COMPASS-1	1U	FH Aachen	Demostración de componentes comerciales y captura de imágenes	2008-04-28
AAUSAT-II	1U	University of Aalborg, Denmark	Sistemas ADCS y detección de rayos gamma	2008-04-28
Delfi-C3	3U	Delft University of Technology, The Netherlands	Testes en órbita de celdas solares (TFSC) y sensores solares autónomos e inalámbricos (AWSS), Demostración del primer radio transductor lineal de radio amateur en un CubeSat	2008-04-28
CanX-2	3U	University of Toronto, Canada	Demostración de tecnología de vuelos en formación	2008-04-28

Nombre	Tipo	Organización	Misión	Lanzamiento
SEEDS-2	1U	Nihon University, Japan	Experimentos de Radio Amateur radio y envío de mensajes preprogramados	2008-04-28
PREsat	3U	NASA	Tecnología	2008-08-03
NanoSail-D	3U	NASA	Tecnología	2008-08-03
PharmaSat	3U	NASA Ames Research Center, Santa Clara University	UTMB y medición de efectos antifunguicos en sepas de levaduras en antigravedad	2009-05-19
CP-6	1U	California Polytechnic University	-	2009-05-19
HawkSat I	1U	HISS	-	2009-05-19
AeroCube 3	1U	Aerospace Corporation	-	2009-05-19
SwissCube-1	1U	Ecole Polytechnique Fédérale de Lausanne	Ciencia atmosférica	2009-09-23
BeeSat-1	1U	Berlin Institute of Technology	-	2009-09-23
UWE-2	1U	Universität Würzburg	Demostración de tecnologías ADCS	2009-09-23
ITUpSAT1	1U	Istanbul Technical University	Captura de imágenes y tecnología	2009-09-23
Hayato	1U	Kagoshima University	-	2010-05-20
Waseda-SAT2	1U	Waseda University	-	2010-05-20
Negai	1U	Soka University	-	2010-05-20
TIsat-1	1U	SUPSI	-	2010-07-12
StudSat	1U	StudSat	-	2010-07-12
KySat-1	1U	Kentucky Space	Educativa, pruebas de tecnologías y radio amateur	2011-03-04
Hermes	1U	University of Colorado at Boulder	-	2011-03-04
Explorer-1	1U	MSU	-	2011-03-04
Jugnu	3U	IIT Kanpur	Sistemas de micro imágenes, cámara infra-roja para observar vegetación y GPS para localización	2011-10-12
ROBUSTA	1U	Université Montptellier 2	Efectos de la radiación sobre circuitos basados en transistores bipolares	2012-02-13
e-stor	1U	Politecnico di Torino	Desarrollo y testeos de ADCS activos, Testeos de COTS	2012-02-13
MaSat-1	1U	BME	Pruebas de tecnologías	2012-02-13
Xatcobeo	1U	University of Vigo	Verificación de un nuevo sistema de medida de la cantidad de radiación ionizante (RDS)	2012-02-13
Goliat	1U	University of Bucharest Romania	Captura de imágenes terrestres y medición de ambiente espacial	2012-02-13
PW-Sat	1U	Warsaw University of Technology Poland	Experimentos de tecnologías	2012-02-13
UNICubeSAT	1U	Sapienza University of Rome Italy	Investigación atmosférica	2012-02-13

Cuadro 2.1: Lista de Cubesats lanzados

Capítulo 3

Objetivos generales

El objetivo principal del proyecto es diseñar y desarrollar el módulo de control principal de un nanosatélite tipo CubeSat.

Entre las principales tareas a desarrollar se encuentran:

- el gerenciamiento y registro de toda la información intercambiada con los demás módulos del CubeSat.
- relevamiento de los registros de eventos y variables de funcionamiento del CubeSat, como por ejemplo, voltajes, temperaturas, eventos y errores.
- toma de decisiones de funcionamiento en base a los datos relevados.
- intercambio de información y datos de estado con tierra a través del módulo de comunicación.

Se deben implementar en software y hardware todas las tareas descritas anteriormente.

Capítulo 4

Especificación funcional

Este capítulo tiene como objetivo una breve descripción de cómo funcionan las principales tareas del módulo desarrollado durante este trabajo y la forma en que estos servicios son suministrados para un correcto funcionamiento del sistema operativo del satélite.

Se comenzará con una descripción general de cuáles son los sistemas con los que interactúa el módulo, y la forma en que se lleva a cabo esta interacción, para luego detallar las características de hardware y software del sistema implementado.

A continuación, se explica brevemente lo que representa el entorno de ejecución que implementa el módulo principal para ejecutar cada una de las tareas que tiene a su cargo de forma eficiente. En este trabajo se denomina *cola de ejecución de tareas*, *despachador de tareas*, o *scheduler*. Cada una de las tareas a ejecutar se clasifica según el intervalo de tiempo en el que se ejecuta, como es explicado en esta sección.

Seguidamente se resume la tarea de monitoreo y control del sistema, una de las tareas llevada a cabo por el control principal del satélite, y que es también llamada *Housekeeping*. Esta función se encarga de llevar un registro de aquellos valores imprescindibles para el correcto funcionamiento y estabilidad del sistema total que compone el satélite. La aplicación correcta de esta tarea condiciona la continuidad del funcionamiento del sistema y la toma de medidas a tiempo de cada uno de los factores que pueden generar errores que comprometan el funcionamiento del satélite de forma transitoria e incluso permanente.

Para terminar, se hace una referencia a otra de las funciones principales que lleva a cabo el módulo principal que es la comunicación entre la estación

terrena y el satélite. Aquí se da cuenta de el tipo de datos que se intercambia entre ambos extremos, los protocolos utilizados y los métodos de encriptado empleados.

4.1. Descripción general

El módulo de control principal, de aquí en adelante el MCP, es el encargado de procesar toda la información proveniente de los dispositivos externos. El microcontrolador utilizado para el desarrollo es el MSP430f5438A [4] de Texas Instruments y se comunica con los demás módulos del sistema por medio de un bus serie I2C. Los periféricos que maneja el MCP son los siguientes:

- Sistema de gestión de energía (SGE)
- Módulo de control y detección de actitud (ADCS)
- Payload (PAY)
- Módulo de comunicación (COM)

El SGE es quien administra el sistema de recolección y suministro energético del satélite para garantizar su vida útil.

El ADCS por su lado se encarga de determinar y controlar la actitud del satélite a fin de optimizar la comunicación con Tierra.

El COM es el encargado de llevar a cabo la comunicación con la estación terrena. Toda transmisión y recepción de datos pasa siempre por este módulo. El PAY es también llamado carga útil o científica del satélite, y es donde se encuentra todo lo necesario para llevar a cabo la misión que tiene el satélite. En el caso del AntelSat, el objetivo es la adquisición de imágenes desde el espacio exterior y la carga útil es una cámara fotográfica y todo lo necesario para el procesamiento de estas imágenes.

El MCP debe controlar los distintos subsistemas, monitorear la salud de todo el satélite, tomar decisiones autónomamente, mantener un registro en memoria de estados y fallas, procesar información que llega desde Tierra, entre otras tareas.

El software que se implementa en el microcontrolador se divide en dos grandes partes: la ejecución de las tareas fijas o periódicas y la ejecución de tareas asíncronas. Las tareas fijas se encolan en el despachador de tareas o scheduler y deben ser ejecutadas invariablemente cada cierto tiempo. Las

tareas asíncronas son procesos que se intercalan con los ya encolados en el scheduler, y su ejecución depende del flujo de eventos que van ocurriendo durante el desarrollo del programa principal.

Este trabajo se focaliza en la comunicación principal entre los módulos presentes en el sistema. Cabe destacar que además existen otras vías de comunicación entre algunos de los módulos con el objetivo de repartir la carga de datos, evitar la saturación de los buses, y optimizar el tiempo en el que los buses están disponibles para comunicación. Por ejemplo, existe otro bus que conecta directamente el módulo PAY con el COM. Esta comunicación soporta el mayor flujo de datos de todo el sistema por ser la encargada de transmitir hacia el COM la información de las imágenes tomadas, para luego ser enviadas a Tierra. En este tipo de comunicaciones no interviene el MCP y por lo tanto quedan fuera del enfoque del presente trabajo.

4.2. Ejecución de tareas periódicas y tareas asíncronas

El sistema guarda, en una estructura de cola FIFO, la lista de tareas que deben ir ejecutándose, y reparte así el tiempo del microprocesador entre todos los procesos disponibles para ejecución. Se hace primero una verificación de si existen elementos en la cola. Si la cola no está vacía, se debe tomar la primera tarea y llevarla a cabo. Este procedimiento se repite hasta que la cola no tenga más elementos.

Las tareas periódicas cuentan con un contador interno que se va decrementando con los períodos de reloj. Cuando el contador llega a cero significa que la tarea debe ejecutarse. En ese momento el sistema la encola en la lista de tareas y vuelve a setear el contador a su valor original. Algo similar ocurre cuando en medio del flujo normal de funcionamiento del programa principal se debe hacer un llamado a una tarea no periódica. También en este caso el sistema encola esa función como un elemento más de la cola de tareas. De esta manera, los elementos que componen la cola FIFO pueden ser de ambos tipos.

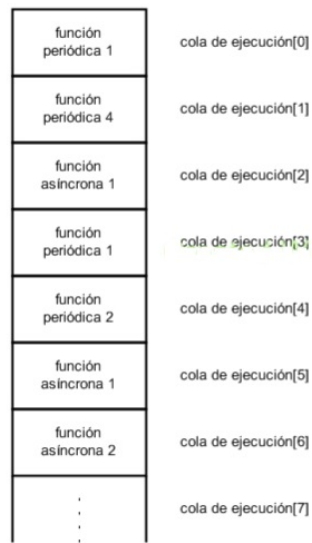


Figura 4.1: Ejemplo de cola de ejecución de tareas del MCP

4.3. Monitoreo y control del sistema

El MCP lleva un registro de aquellas variables que resulten de especial interés para el mantenimiento saludable de todo el sistema. Para ello realiza consultas periódicas a todos los módulos externos, y une esos valores recolectados con otros del propio MCP para generar reportes de estado del sistema. Periódicamente, el satélite envía este registro a Tierra.

El reporte solamente guarda los valores de la última consulta. Con cada nueva consulta los valores del registro se sobrescriben. Para poder mantener un historial de aquellos datos más relevantes, se realiza con cada consulta un resumen de la misma y se guarda en una tabla junto con una estampa de tiempo. Este resumen es el que se manda cuando hay un pedido de actualización de estado del sistema desde la estación terrena.

Algunos de los datos que se relevan en las consultas deben ser procesados previo a su almacenamiento. Entre estos datos se encuentran:

- temperatura de cada cara del satélite
- temperatura interna del microprocesador
- desviaciones del valor correcto del reloj RTC

4.4. Comunicación entre la estación terrena y el satélite

El objetivo es crear un protocolo de comunicación a fin de establecer un conjunto de reglas normalizadas para una transmisión de datos clara y eficiente entre la estación terrena y el satélite.

Los datos pueden ser codificados tanto en la estación terrena como en el módulo de control principal, y siempre pasan a través del módulo de comunicación del satélite.

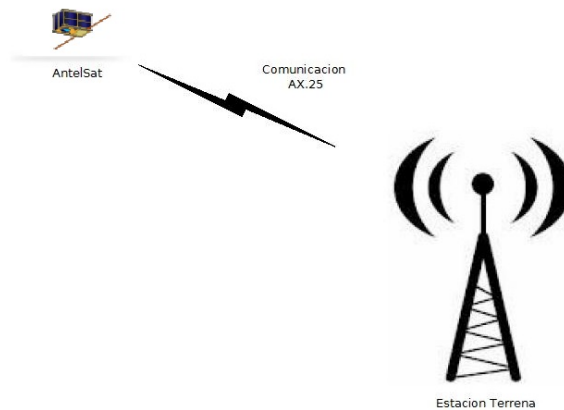


Figura 4.2: diagrama de comunicación entre la estación terrena y el satélite

Los paquetes de información pueden contener tres tipos de datos:

- paquetes que contienen datos de información enviada entre la estación terrena y el satélite.
- paquetes con datos de instrucciones entre la estación terrena y el satélite.
- repeticiones de paquetes provenientes de estaciones terrenas cualesquiera alrededor del mundo.

El envío de datos entre ambos extremos de la comunicación se realiza mediante el protocolo AX.25 [5], [6]. Cuando las tramas llegan al satélite, el módulo de comunicación se encarga de verificar que los datos no lleguen

corruptos y luego pasa la información al MCP. Éste descripta primero los datos de la trama AX.25 y luego separa la carga útil de datos. Los datos llegan desde la estación terrena encriptados mediante el protocolo TEA (Tiny Encryption Algorithm) [7]. El MCP descripta los paquetes y obtiene los datos deseados.

Los paquetes de datos que no provienen de la estación terrena, no llegan encriptados mediante TEA, y por lo tanto el MCP sabe que se trata de un pedido de repetición de paquetes AX.25, también llamado digipeater. En un caso como éste, los datos vuelven a ser transformados en tramas AX.25 y son enviados al módulo de comunicación para ser transmitidos.

Capítulo 5

Diseño e implementación del sistema

En este capítulo se realiza una descripción exhaustiva de la construcción del sistema en términos de hardware y software.

5.1. Diseño del software

En esta sección se traducen los requerimientos del sistema en una representación de software que se ajuste adecuadamente con los requisitos establecidos. Con el fin de reducir la complejidad y la redundancia del software se elabora una arquitectura modular, en donde cada uno de los ocho módulos implementados cumple una tarea específica dentro del funcionamiento integral del sistema.

En primer lugar, se describe la implementación del módulo del Scheduler. Como se explicó anteriormente, el Scheduler es quien determina qué tarea se cumple y en qué momento, asemejándose fuertemente al funcionamiento de un sistema operativo.

Luego se describen los funcionamientos de los módulos que llevan a cabo las mediciones de temperatura y el reloj de tiempo real RTC.

Más adelante se detalla el módulo que implementa el protocolo de comunicación entre las distintas unidades que componen el satélite. Este módulo presenta la particularidad de ser necesario en todas estas unidades mencionadas, por lo que se construye específicamente de forma genérica, con el fin de poder embeberlo en el software de cada una de ellas sin necesidad de realizar

grandes cambios.

La siguiente sección abarca al módulo encargado de los distintos modos de bajo consumo en los que funciona el control principal. Este módulo de software presenta una gran importancia ya que, como fue explicado anteriormente, es sumamente necesario para el funcionamiento del satélite el poder reducir al máximo el consumo de energía.

Seguidamente, se hace referencia al módulo que implementa la tarea de Housekeeping, encargada del relevamiento de todos los datos imprescindibles para el correcto funcionamiento y estabilidad del sistema.

Se continúa con una descripción del módulo que implementa el mecanismo de reseteo del sistema en caso de ocurrir un bloqueo del mismo, llamado Watchdog.

Y para finalizar se tiene una sección que detalla el funcionamiento del módulo encargado de llevar a cabo la comunicación entre satélite y estación terrena, junto al protocolo utilizado en la implementación del mismo.

5.1.1. Scheduler

Como se mencionó anteriormente, el MCP cuenta con un dispatcher o scheduler de tareas que asegura la ejecución las tareas solicitadas. Existe una lista de tareas periódicas y una lista de tareas no periódicas. Estas tareas se van encolando en una cola FIFO a medida que se desarrolla el programa. El scheduler es el encargado de ejecutar la tarea que se encuentra en el primer lugar de la cola. En la figura 5.1 se puede observar un diagrama que representa el flujo principal del MCP.

Lista de tareas periódicas:

- Housekeeping.
- Envío del housekeeping por el beacon [8].
- Control de actitud.

Lista de tareas asíncronas:

- Inicialización.
- Comunicación I2C.

- Envío y pedido de datos por I2C.
- Envío del resumen de housekeeping.

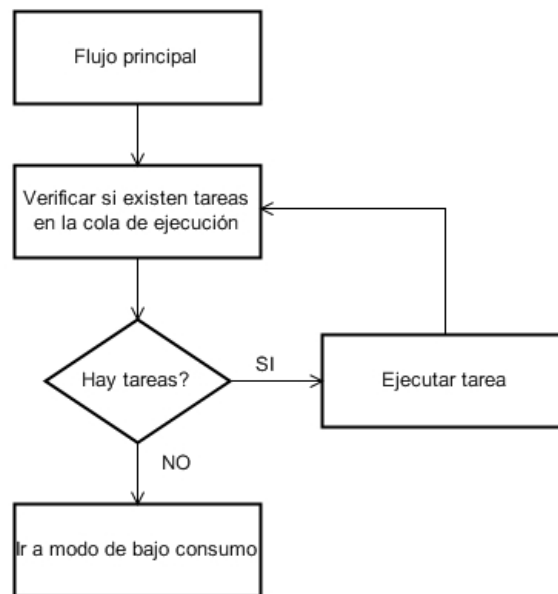


Figura 5.1: Diagrama de flujo principal

5.1.1.1. Ejecución de las tareas periódicas

Cada tarea periódica tiene asociado un contador, cuyo valor inicial es igual al período de ejecución de la tarea. Con cada período del TIMER B, el microprocesador genera una interrupción. La rutina de atención a esta interrupción se encarga de recorrer la lista de tareas periódicas y decrementar en uno el contador asociado. Si el contador llega a cero, la rutina de interrupción encola la tarea en la lista de tareas a ejecutar y vuelve a setear el contador con el valor del período. En la figura 5.2 se muestra el diagrama de flujo correspondiente a esta función.

El funcionamiento es el mismo cuando la cola de tareas a ejecutar está llena y no puede ser encolada la función. El contador vuelve a setearse y la tarea deberá esperar un nuevo período para que sea encolada.

La idea de encolar las tareas en una cola de ejecución, y no ejecutarlas cuando el contador se hace cero, es la de acortar los tiempos de las rutinas

de atención a la interrupción, y por ende, maximizar los tiempos en los que el sistema está alerta con sus interrupciones habilitadas.

Los dos puntos anteriores pueden ser implementados de esta manera debido a que todas las tareas que realiza el microcontrolador tienen períodos de ejecución relativamente cortos y no implican demoras importantes para el sistema general.

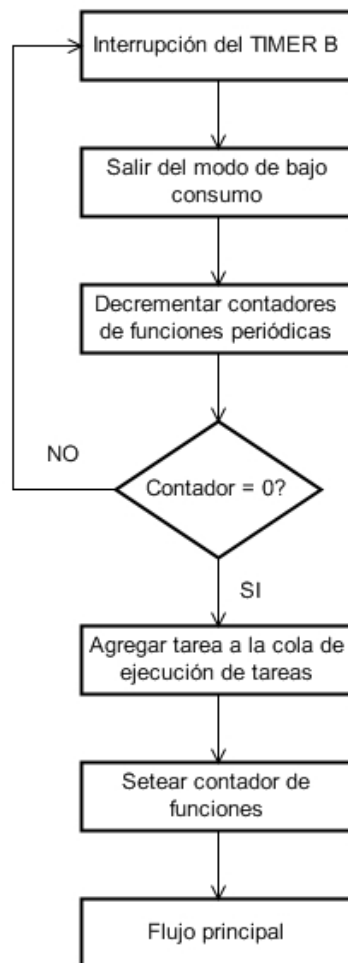


Figura 5.2: Diagrama de flujo del encolamiento de tareas

5.1.1.2. Ejecución de tareas asíncronas

La ejecución de tareas asíncronas surge a medida que se va desarrollando el programa principal. Son subrutinas llamadas por las tareas periódicas mientras éstas se están ejecutando. Un ejemplo de esto puede ser cuando llegan datos al MCP. En este caso, la rutina que maneja la comunicación I2C debe asegurarse de no perder los datos que acaban de llegar. Debe entonces encolar en la lista de tareas a ejecutar una nueva tarea que procese los datos recibidos. Eventualmente, el flujo principal del programa podrá atender a esta tarea y procesar la información que le fue enviada.

En el cuadro 5.1 se muestran las funciones de software que manejan al scheduler.

Cuadro 5.1: Funciones del módulo Scheduler

Nombre de la función	Descripción
<code>void periodica_init(tarea_periodica t[])</code>	Función que inicializa el contador de las funciones de la lista de tareas periódicas.
<code>int decrementar_contador(tarea_periodica t)</code>	Función que decrementa el contador de una tarea específica. Devuelve 1 si el contador llega a cero.
<code>int restablecer_periodo(tarea_periodica t)</code>	Función que restablece el contador de una función específica. Lo setea en el valor del período de la función. Devuelve 1 si el período pudo restablecerse.
<code>void atencion_interrupcion()</code>	Rutina de atención a la interrupción del Timer B. Con cada interrupción del Timer B, la rutina recorre la lista de tareas periódicas y decrementa en uno el contador. Cuando el contador de una tarea llega a cero se manda la tarea a la lista de tareas a ejecutar y restablece su contador.
<code>void mcpc_scheduler_start()</code>	Función que comienza la ejecución de tareas.
<code>void mcpc_scheduler_isr_ie()</code>	Función que habilita la ISR del Timer B.
<code>void mcpc_scheduler_isr_id()</code>	Función que deshabilita la ISR del Timer B.
<code>void mcpc_scheduler_config_init()</code>	Función que inicializa la configuración del scheduler.
<code>void init_queue()</code>	Inicializa la cola de tareas.
<code>void add_element(ptr_tarea e)</code>	Agrega un elemento en la cola.
<code>ptr_tarea remove_element()</code>	Extrae un elemento del principio de la cola. Devuelve el puntero a la tarea que se encuentra al inicio de la cola.
<code>int is_empty()</code>	Devuelve 1 si la cola esta vacía.

5.1.2. Mediciones de temperatura

El microcontrolador cuenta con dos módulos de conversión Analógico-Digital, uno de diez y otro de doce bits. En este proyecto se opta por trabajar con el segundo. La elección se basa en que con éste se obtiene una mayor resolución. Como el nombre lo sugiere, el módulo convierte un voltaje analógico en una de representación digital de doce bits, la cual se almacena en un registro. Accediendo al contenido de ese registro, y realizando las operaciones necesarias, se puede interpretar el valor obtenido.

Para el funcionamiento se deben usar dos voltajes seleccionables y programables V_{-R} y V_R a los que se les llama voltajes de referencia, los cuales son necesarios para definir los límites de conversión. Tanto los canales de entrada, como los voltajes mencionados anteriormente, se definen en la memoria de control-conversión. Ésta y otras configuraciones se verán con mas detalle en el Anexo B.

El módulo permite realizar mediciones periódicas mediante el uso de una de las fuentes de reloj o el oscilador interno. En este caso se opta por realizar mediciones simples cuando la tarea housekeeping los requiere. De esta forma se evita mantener encendido el módulo ADC permanentemente, reduciendo el consumo de energía.

Como se menciona en el capítulo 5.11, el módulo ADC es usado para obtener los valores de temperatura de las caras, la placa del MCP y la del microcontrolador.

Temperatura de caras y placa Tanto para las caras como para la placa, se utilizan sensores analógicos de precisión LM20 del fabricante *National Semiconductors* [9]. La salida de cada uno de estos sensores es conectada a los puertos del módulo, y la conversión se realiza utilizando uno de los voltajes internos de referencia. La configuración de los parámetros necesarios para el funcionamiento se verá con detalle en el Anexo B.

El modo de operación utilizado en esta instancia es el llamado *Sequence-of-channels*, el mismo consiste en realizar la conversión de un número determinado de canales de entrada, de a uno por vez. Una vez iniciada la conversión, del primer canal el sistema entra al modo de bajo consumo LPM3. En el momento en que finaliza la conversión del último canal se genera una interrupción, en cuya rutina de atención se consulta a los registros que almacenan los resultados de las conversiones. Los valores son copiados a en un buffer y

se sale del LPM.

Para obtener el valor final del voltaje es necesario realizar la siguiente operación:

$$N_{ADC} = 4096 \times \frac{V_{IN} - V_{-R}}{V_R - V_{-R}}$$

Donde V_{IN} corresponde al voltaje de salida del sensor LM20.

Luego de obtener ese voltaje y mediante la función de transferencia suministrada por el fabricante se obtienen los valores de temperatura, los cuales son almacenados en las variables correspondiente.

$$T = -1481,96 + \sqrt{2,1962 \times 10^6 + \frac{1,8639 - V_O}{3,88 \times 10^{-6}}}$$

Si bien en la hoja de datos se presenta una aproximación lineal de la función de transferencia, se sugiere utilizar la aproximación parabólica para mayor exactitud en todo el rango de temperaturas.

Temperatura del microcontrolador El MSP430F5438A cuenta con un sensor de temperatura incluido en el chip, al cual se accede mediante uno de los registros de conversión del módulo ADC. La configuración de los parámetros necesarios para su funcionamiento se verá con detalle en el Anexo B.

Como se especifica en la hoja de datos [4], el microcontrolador debe permanecer dentro de el rango de temperatura de operación para que funcione de forma correcta y evitar que se dañe. Otros parámetros tales como el consumo en LPM también dependen de la temperatura de trabajo. En la figura 5.3 se puede observar una curva típica del comportamiento del sensor interno del MSP430F5438A.

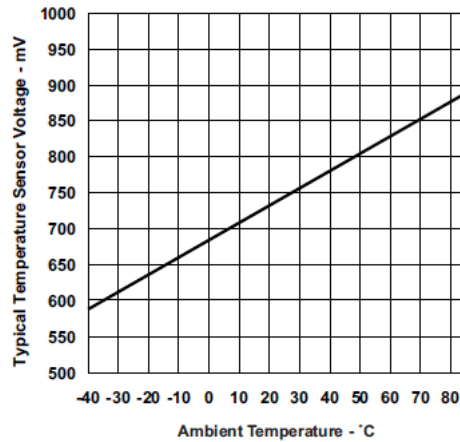


Figura 5.3: Curva de temperatura del sensor interno

Al igual que cuando se mide la temperatura de las caras del satélite, una vez iniciada la conversión se entra al modo de bajo consumo LPM3, en el que se permanece hasta que finaliza la misma. En este momento se genera una interrupción y el dato leído del registro de conversión correspondiente es almacenado para ser procesado.

Mediante el uso de la siguiente expresión se obtiene el valor de la temperatura en grados Celsius.

$$T(^{\circ}C) = \frac{1}{TC_{sensor}} \times \left(\frac{A_{10}}{4096} \times V_{REF} \right) - V_{sensor}$$

En donde $V_{sensor} = 680mV$, $TC_{sensor} = 2,25 \frac{mV}{^{\circ}C}$, y $V_{REF} = 1500mV$.

Como se menciona en el capítulo 5.11, cada lectura de los sensores tiene asociada una estampa de tiempo. Esto hace posible relacionar los valores obtenidos con otros parámetros de forma de verificar el funcionamiento del módulo. En el cuadro 5.2 se muestran las funciones de software para la medición de temperatura.

Cuadro 5.2: Funciones del módulo ADC

Nombre de la función	Descripción
<code>void antelsat_temp_caras ()</code>	Se habilita la interrupción en cuya rutina de atención se consultan los registros de conversión y se almacenan las temperaturas de las caras
<code>void antelsat_temp_micro ()</code>	Se habilita la interrupción en cuya rutina de atención se consulta el registro de conversión y se almacena la temperatura del microcontrolador
<code>void antelsat_temp_caras_config ()</code>	Configuración de los registros del módulo ADC para la lectura de temperatura de las caras y placa
<code>void antelsat_temp_caras_off ()</code>	Deshabilita las conversiones del módulo ADC
<code>void antelsat_temp_micro_config ()</code>	Configuración de los registros del módulo ADC para lectura de temperatura del microcontrolador
<code>void antelsat_temp_micro_off ()</code>	Deshabilita el sensor interno de temperatura y el módulo de referencias del microcontrolador
<code>void adc_antelsat_tv1 ()</code>	Consulta de los registros TLV para calibración del sensor interno de temperatura

5.1.3. Reloj de tiempo real

Para el funcionamiento del satélite, resulta fundamental contar con un reloj de tiempo real (RTC). El motivo fundamental reside en que el vector tiempo es uno de los parámetros usados en la determinación de actitud, por lo que es necesario que estos datos se encuentren disponibles cada vez que el módulo ADCS lo solicite.

Otra razón importante es el hecho de que cada vez que se realiza la tarea de housekeeping, los datos recabados se asocian a una estampa de tiempo que indica fecha y hora en que los datos fueron obtenidos.

El microcontrolador cuenta con el módulo RTC_A, que provee un reloj de tiempo real y una función de calendario que cuenta con los siguientes registros:

- Segundos
- Minutos
- Horas
- Día de la semana
- Día del mes
- Mes
- Año

Esto simplifica, en forma significativa, el código a implementar. Si no se tiene en cuenta la configuración, la que se puede ver en detalle en el Anexo C, el código se limita a la consulta y modificación de los registros correspondientes.

Es importante que las consultas a los registros sean realizadas cuando la información que contienen se encuentra estable. Para esto se cuenta con una bandera que indica que se puede realizar una lectura. La misma es usada para generar una interrupción. La rutina que atiende esta interrupción almacena el contenido de los registros en los campos de una variable global del tipo `time_s` que se describe en el siguiente extracto de código.

```
typedef struct{
    int anio_h;
    int anio_l;
    int mes;
    int dia;
    int hora;
    int min;
    int seg;
} time_s;
```

Si bien el RTC mantiene registro de la hora, es necesario realizar actualizaciones en forma periódica debido a posibles derivas de los osciladores. Para ésto se implementa un función que carga en los registros los valores enviados desde tierra con los nuevos valores del tiempo en forma atómica. Para la actualización de estos valores las interrupciones son deshabilitadas hasta

que se finaliza de modificar los registros. Los nuevos valores se comparan con el contenido de los registros y se almacena la diferencia en una variable llamada **drift**. Éste es uno de los datos que almacena la tarea housekeeping del MCP y se considera de mucha importancia ya que en caso de diferencias significativas puede generar resultados erróneos en el algoritmo de detección de actitud [10].

El módulo RTC es alimentado por la fuente de reloj ACLK la que a su vez se alimenta, por defecto, por un oscilador interno del microcontrolador de 32768 Hz. Esta frecuencia es la necesaria para el correcto funcionamiento del módulo. En forma opcional, se puede utilizar un cristal oscilador de 32768 Hz de alta precisión para alimentar la fuente ACLK. Cabe destacar que esta fuente no es deshabilitada en el modo LPM3.

Al momento de la elección del cristal, es necesario considerar la variación de frecuencia del cristal con la temperatura. En este proyecto se utiliza el cristal incluido en el kit de desarrollo del microcontrolador [11].

En el modo calendario, el módulo cuenta con un registro para la calibración del oscilador con el fin de reducir las diferencias entre la frecuencia obtenida y la deseada. De ser habilitada, es realizada en forma periódica y permite un ajuste de + 4 ppm o - 2 ppm.

El cristal utilizado tiene una precisión de 20 ppm y una deriva térmica de 0.035 ppm/°C. Esto significa una diferencia de $\pm 1,7$ s por día, sin considerar la deriva térmica que puede llegar a aportar aproximadamente -2 ppm, resultando en una diferencia diaria máxima de -1,9 s.

Antes de realizar la elección del cristal, es necesario analizar hasta qué punto se puede apartar el vector tiempo usado por el algoritmo de determinación de actitud y el real, sin que se vea afectado el funcionamiento. En el cuadro 5.3 se muestran las funciones implementadas para el módulo RTC.

Cuadro 5.3: Funciones del módulo RTC

Nombre de la función	Descripción
<code>void rtc_actualizo ()</code>	El reloj se detiene, se cargan los registros con los valores actualizados de la fecha y la hora y se deja correr nuevamente
<code>void rtc_consulto ()</code>	Los datos en los registros del RTC son almacenados en una variable del tipo <code>time_s</code>
<code>void rtc_config ()</code>	Función de configuración de RTC. Se cargan los registros de control del módulo RTC del microcontrolador
<code>__interrupt void RTC_ISR ()</code>	Rutina de atención a interupción del RTC

5.1.4. Protocolo de comunicación entre módulos

5.1.4.1. Protocolo de comunicación del satélite

El correcto funcionamiento del satélite implica transferencias de información entre los diferentes módulos que lo componen. Para lograr que la comunicación sea eficiente, es importante elaborar un protocolo que establezca una serie de reglas a seguir para la transmisión y recepción de datos. De esta forma, sin importar qué módulos se encuentren involucrados en la transferencia, el orden será el mismo. Como consecuencia directa, cuando un módulo necesite enviar o leer datos de cualquier otro módulo, bastará con que invoque una única función.

El dispositivo que inicia la comunicación es el maestro, y es su responsabilidad indicarle al dispositivo esclavo qué tipo de intercambio de datos se llevará a cabo. Por este motivo se establece como parte del protocolo que cada comunicación comience con el envío de un número de bytes por parte del maestro, conteniendo información sobre la transferencia a realizarse.

El orden para el envío de datos se define como sigue:

- Primer byte para COMANDO. En un principio este byte es utilizado únicamente para indicarle al esclavo si se trata de una escritura de datos o una lectura de uno de sus registros.

- Segundo byte para REGISTRO. Mediante el uso de una tabla que contiene las direcciones de los diferentes registros que pueden querer ser accedidos por diferentes maestros, junto a la interpretación de la información contenida en el segundo byte recibido, se obtiene el puntero al registro deseado.

Es importante mencionar que todos los dispositivos deben contar con un buffer donde almacenar los datos recibidos. El comando que identifica la transferencia debe ser almacenado en el primer lugar de buffer para que con la ayuda de una función, se analicen los datos recibidos.

Función de atención a los datos recibidos

El objetivo principal de esta función es la de liberar el buffer de entrada de I2C de forma de que se pueda recibir una nueva transferencia.

Se tienen dos casos en los que un dispositivo recibe datos en su buffer de entrada: cuando el dispositivo realiza una lectura, y cuando otro dispositivo le envía datos. En ambos casos se invoca a la función de *Callback*. En el primero, la función es invocada a continuación de la función que realiza la lectura de los datos. En el segundo, la función se encola en la cola de tareas a ejecutar inmediatamente después de recibir la condición de Stop.

En ambas situaciones la información en el buffer de entrada se organiza de la forma 5.6. Antes de comenzar con el procesamiento de datos, se verifica que la bandera que indica que el buffer de entrada se encuentra ocupado permanece seteada, de esta manera se evita que se procesen los mismos datos dos veces. Utilizando el comando guardado en el primer lugar del buffer de entrada, la función de callback puede determinar cuantos son los bytes que forman el mensaje y dónde se deben almacenar. Es necesario que exista un registro en memoria, con la lista de todos los comandos y parámetros asociados a ellos, con el cual se pueda comparar el primer byte y en base al resultado tomar una acción.

Las transferencias por I2C se realizan de a un byte y muchas veces los datos que se transmiten, por ejemplo en el relevamiento de housekeeping, son enteros de dos bytes. Las acciones que se tomen luego de una función *callback* debe tener en cuenta estos casos al momento de almacenar los datos

transferidos en el buffer correspondiente. En el cuadro 5.7 se muestran las funciones implementadas en el módulo I2C.

5.1.4.2. Protocolo I2C

La comunicación entre módulos se realiza mediante el uso del protocolo I2C [12]. Éste fue desarrollado por Philips en la década del 80' y, desde entonces, se han ido publicando nuevas versiones. Consiste en una transferencia de datos serial y síncrona. Para ésto, los dispositivos utilizan tres líneas:

- datos (SDA)
- reloj (SCL)
- referencia (GND)

La transferencia de datos se establece entre dos dispositivos, uno de los cuales (maestro) suministra la señal de reloj y se encarga de controlar el flujo de bytes, y otro (esclavo) que recibe o envía datos según se le solicite. La comunicación comienza con una condición de Start, que consiste en una transición del nivel alto al bajo de la línea SDA mientras la línea SCL se mantiene en nivel alto, y finaliza con una condición de Stop, que consiste en una transición del nivel bajo al nivel alto de la línea SDA mientras la línea SCL se mantiene en nivel alto.

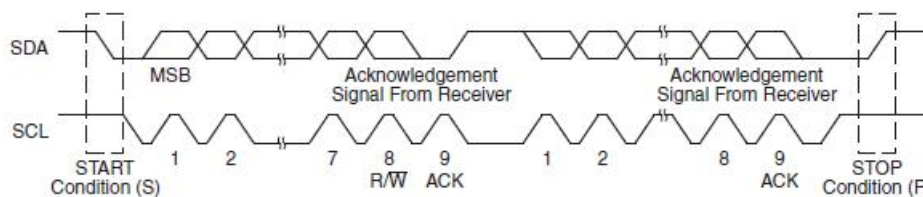


Figura 5.4: Comportamiento de las líneas SDA y SCL.

Como se muestra en la figura 5.4, los datos en la línea SDA deben permanecer estables mientras la línea SCL se encuentra en el nivel alto. De otra forma, pueden detectarse como condiciones de Stop o Start.

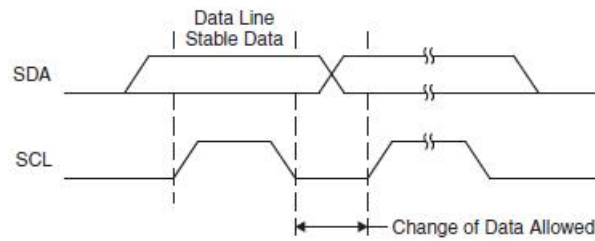


Figura 5.5: Transiciones válidas del SDA

A continuación de la condición de Start, el maestro pone en la línea SDA 7 bits correspondientes a la dirección del esclavo y completa el byte con un bit que indica si la transferencia a realizarse es una lectura o una escritura. Cuando un dispositivo detecta una condición de Start, compara la dirección puesta en la línea SDA por el maestro con la propia y envía una señal ACK, en caso de que coincida. A partir de este momento y hasta que el maestro envíe la condición de Stop, se realiza la transferencia de datos. El envío de cada byte debe ser seguido por el envío del ACK correspondiente por parte del dispositivo receptor. En la figura 5.6 se muestra un ejemplo de la secuencia de transmisión I2C.

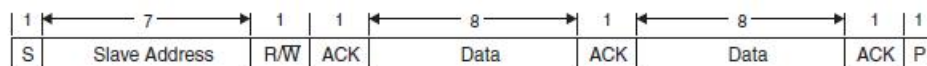


Figura 5.6: Ejemplo de secuencia de transmisión I2C

5.1.4.3. Módulo I2C del MSP430F5438A

El microcontrolador cuenta con un módulo I2C el cual resuelve muchas de las etapas del protocolo en forma automática. Un ejemplo de esto es cuando el dispositivo funciona como esclavo y un maestro inicia una comunicación. En esta situación la comparación de la dirección enviada luego de una condición de Start se realiza en forma automática, así como el envío del ACK correspondiente en caso de que coincida. Otro ejemplo es que a medida que

van llegando los bits, el módulo los almacena en un registro hasta la llegada del bit que completa el byte, momento en el cual pasa la información recibida a otro registro accesible por el usuario generando una interrupción. A partir de ese instante la rutina principal debe atender la llegada de datos. Hasta ese momento el microcontrolador puede realizar otras tareas o permanecer en modo bajo consumo.

Con excepción de la generación de la condición de Start, la demás acciones como la generación de condición de Stop, cargar el registro de salida con el byte a enviar o almacenar el byte recibido en el registro de entrada, se realizan dentro de la rutina de atención a interrupción del módulo I2C. La misma consiste en una estructura switch-case en donde, evaluando el valor de cierto registro se determina de que evento se trata y se realizan las acciones correspondientes.

El protocolo establece dos velocidades de transferencia de datos, 100 KHz (*standard mode*) y 400 KHz (*fast mode*). En este caso se trabaja en *standard mode* debido a que hasta el momento no existe ninguna tarea que implique la necesidad de una transferencia más rápida de datos.

A continuación se detallan las principales características del módulo I2C integrado en el microcontrolador MSP430F5438A.

- Conformidad con la especificación de I2C para semiconductores de Philips v2.1
- Modos de direccionamiento de 7 bits y de 10 bits
- Llamado general
- START/RESTART/STOP
- Modo Multi-master receptor/transmisor
- Modo Esclavo receptor/transmisor
- Soporta modo standard hasta 100 Kbps y modo rápido hasta 400 Kbps
- Frecuencia de reloj programable en modo Maestro
- Diseñado para bajo consumo
- Detección de condición de Start en modo Esclavo para salida de modo de bajo consumo LPMx

- Operación de esclavo en LPM4

En el Anexo D se puede encontrar con mas detalle el funcionamiento del protocolo I2C para el microcontrolador elegido, con ejemplos de los registros usados.

Antes de analizar las diferentes situaciones es importante mencionar que un pedido de lectura a un registro específico del esclavo no implica dos comunicaciones diferentes. Mediante el uso de una *repeated start condition* se logra cambiar la dirección de la transferencia sin liberar el bus de datos, evitando así que el maestro quede a la espera de los datos. En la figura 5.7 se muestra un ejemplo de una *repeated start condition*.

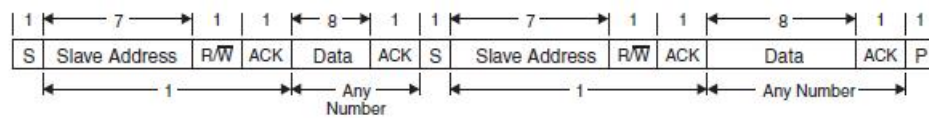


Figura 5.7: Ejemplo de una *repeated start condition* de I2C

El MCP forma parte de un sistema Multi-Maestro junto con los demás módulos del satélite. En esta configuración el dispositivo puede ser él mismo quien inicie una lectura o escritura, pasando a funcionar como maestro, así como también ser el esclavo al que se le escribe o del cual se leen datos.

Por este motivo es necesario considerar dos casos dentro de la rutina de atención a una solicitud de comunicación, cuando el dispositivo funciona como maestro y cuando el dispositivo funciona como esclavo. La descripción detallada de las funciones de envío, configuración, y de atención a la interrupción se encuentran al final de este capítulo.

Para describir el protocolo de forma clara se puede separar cuatro situaciones de funcionamiento diferentes para un mismo dispositivo:

1. Maestro envía un determinado número de bytes

Para realizar una transferencia se invoca a una función que recibe como parámetro de entrada una estructura que contiene todos los datos

necesarios para la comunicación. Entre otros campos, esta estructura contiene una instrucción, un número de bytes a transmitir y una dirección I2C. Luego del envío de una condición de Start, se inicia un timer el que, transcurrido cierto tiempo, y en caso de que no se haya generado una condición de Stop, resetea el registro de status del módulo I2C . Esto sirve para asegurarse que el bus quede desocupado. El motivo se menciona más adelante cuando se estudien los casos particulares de la comunicación I2C.

El primer byte transmitido consiste en un comando el cual sirve para que el dispositivo que recibe los datos pueda determinar cómo procesarlos. Debido a que se trata de una escritura el segundo byte no cumple ninguna función, por lo que se envía un byte previamente estipulado, en este caso el 00000000. Seguido al envío del segundo byte y con la ayuda del campo **número de bytes** se procede a mandar los bytes restantes.

La rutina de atención a la interrupción generada por el envío de bytes es la encargada de incrementar el contador hasta llegar al número de bytes que se desea transmitir, momento en el cual se genera una condición de Stop para dar por terminada la transferencia. En el cuadro 5.4 se muestra la secuencia de bytes enviados por un maestro.

S	Destino	W	ACK	Comando	ACK	0x00	ACK	byte 1	...	n-ésimo byte	ACK
---	---------	---	-----	---------	-----	------	-----	--------	-----	--------------	-----

Cuadro 5.4: Maestro envía un determinado número de bytes

2. Maestro lee un determinado número de bytes

Al igual que en la situación anterior, se invoca a la función que recibe como parámetro de entrada la misma estructura que contiene todos los datos necesarios para la transmisión. La diferencia está, en que en este caso sí se utiliza el campo **registro** de la estructura.

Luego del envío de una condición de Start, se inicia un timer el que, transcurrido cierto tiempo, y en caso de que no se haya generado una condición de Stop, resetea el registro de status del módulo I2C . Esto sirve para asegurarse que el bus quede desocupado. El motivo se

menciona cuando se estudien los casos particulares de la comunicación I2C. Nuevamente, el primer byte enviado consiste en un comando. El segundo byte enviado consiste en el contenido del campo **registro** de la estructura de entrada. Este dato es utilizado por el dispositivo esclavo para establecer, mediante una tabla de direcciones, qué registro es el que el maestro desea leer.

El comando se guarda en el primer lugar del buffer de entrada para luego ser analizado al finalizar la transferencia. Cada transferencia comienza con el envío de ciertos bytes. Por este motivo es necesario cambiar el sentido de la comunicación sin perder control sobre el bus de datos. Enviando una condición de Start repetida, con la misma dirección de esclavo pero con el bit que indica que se trata de una lectura se logra el objetivo. A partir de este momento y con la ayuda del número de bytes asociado al comando se lee la información del esclavo y se guarda en el buffer de entrada a continuación del mismo. En el cuadro 5.5 se muestra este tipo de transferencia.

S	Destino	W	ACK	Comando	ACK	Registro a leer	ACK
S	Dirección del esclavo	R	Byte 1	ACK	...	n-ésimo byte	P

Cuadro 5.5: Maestro lee un determinado número de bytes

Para finalizar la transferencia el maestro envía una condición de Stop y libera al bus I2C. La principal diferencia con respecto al caso anterior es que ahora es necesario invocar a una función que interprete o almacene los datos recibidos en los buffers definitivos. Esta función no se agrega a la lista de tareas a ejecutar (cola de ejecución), sino que se invoca luego de la lectura.

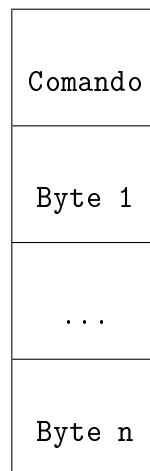
Para evitar la pérdida de datos se considera el uso de una bandera que indica si los datos del buffer de entrada fueron procesados o no. Ésta es seteada luego de que se recibe información y reseteada durante la ejecución de la función que procesa los datos recibidos (*callback*). En caso de estar seteada, no se permite otra escritura y se notifica mediante el envío de un *NACK*.

La función utilizada para el envío de datos es la misma que se emplea

para la lectura. Ésta devuelve un entero cuyo valor sirve para determinar si la transferencia es realizada exitosamente.

3. Esclavo recibe un determinado número de bytes

Este caso se corresponde con el caso en que el maestro envía un determinado número de bytes. En esta situación el dispositivo esclavo no tiene control sobre el flujo de bytes que recibe o que envía. Por este motivo sólo se limita a almacenar el comando enviado por el maestro en el primer byte, no siendo necesario procesar el segundo byte recibido. El esclavo almacena los datos de la misma forma en que lo hace el maestro cuando realiza una lectura a otro dispositivo. En el cuadro 5.6 se muestra el buffer del esclavo cuando recibe un determinado número de bytes.



Cuadro 5.6: Buffer del esclavo cuando recibe un determinado número de bytes

La transferencia continúa hasta que el dispositivo maestro genera una condición de Stop. Cuando el dispositivo esclavo recibe esta señal, agrega en la cola de ejecución la tarea responsable de procesar los datos recibidos, llamada función de *Callback*.

4. Esclavo envía un determinado número de bytes

Este caso se corresponde con la situación en la que el maestro lee un determinado número de bytes. Al igual que en el caso anterior, por tratarse de un dispositivo funcionando como esclavo, éste no tiene control sobre el flujo de bytes.

Si bien se menciona en el estudio del maestro, se vuelve a aclarar que la transferencia comienza cuando el maestro envía un byte de comando seguido de otro byte que el esclavo utiliza para determinar el registro del cual el dispositivo maestro solicita los datos.

Cada dispositivo cuenta con una tabla en la cual se almacenan punteros a los registros que pueden ser leídos por los diferentes maestros. El byte que se recibe indica qué lugar de esta tabla de memoria debe leerse. En este momento el maestro genera una nueva condición de Start indicando que se trata de una lectura. En este punto, el esclavo ya está preparado para enviar los datos desde el registro solicitado. Los datos son enviados hasta que una condición de Stop es recibida.

Envío de condición de start a esclavo no presente en el bus I2C

En la descripción del funcionamiento del protocolo I2C se puede apreciar que todas las comunicaciones son iniciadas por un maestro mediante la generación de una condición de Start.

Originalmente, y con el fin de minimizar el consumo, el microcontrolador entraba a uno de los modos de bajo consumo con interrupciones habilitadas luego de iniciar una transferencia. La salida de este modo se forzaba en la rutina de atención a la interrupción generada por la llegada del ACK del esclavo, avisando del envío o la lectura exitosa del último byte transmitido. El problema que surge con esta arquitectura es que no se toma en cuenta qué es lo que sucede si el módulo al que se quiere acceder no se encuentra accesible.

Si bien el sistema entra a uno de los modos de bajo consumo, las interrupciones quedan habilitadas por lo que el microcontrolador sigue siendo interrumpido por aquellos módulos cuyo reloj no haya sido deshabilitado. Como se menciona en la sección 5.1.1.1, el TIMER B es el encargado de decrementar los contadores de las funciones periódicas del scheduler. Para realizar esta tarea se toma la precaución de previamente abandonar el modo de bajo consumo, resolviendo el problema anterior.

Sin embargo, surge un nuevo problema. El sistema logra retornar al estado activo de la CPU, pero no se llega a ejecutar nunca la línea en la que se genera la condición de Stop. La bandera que indica que el bus está ocupado nunca se resetea, bloqueando la posibilidad de establecer nuevas comunica-

ciones con el dispositivo.

Esperando un tiempo mayor al que lleva la transferencia de datos más larga que pueda haber en el sistema y luego generando un reset de las banderas del módulo I2C puede evita la situación antes mencionada y el dispositivo queda habilitado para una nueva transferencia. Este reset se realiza utilizando el Timer A del microcontrolador. Debido a que no hay esclavo presente, el número de bytes correspondientes a la instrucción no será igual al número de bytes transferidos por lo que la bandera que indica que la transferencia fue realizada de forma correcta no va a ser seteada, permitiendo que el flujo principal del programa pueda verificar que la transacción no pudo ser realizada y de esta forma encolar la tarea nuevamente.

Cuadro 5.7: Funciones del módulo I2C

Nombre de la función	Descripción
<code>int i2c_antelsat_envio(param_i2c par)</code>	Función de envío o de lectura de información a otro dispositivo. Recibe como parámetro de entrada una dato del tipo estructura <code>param_i2c</code> donde se especifica la dirección del esclavo, el tipo de transferencia, el número de bytes asociado y el registro a leer, si corresponde. Devuelve uno en caso de éxito y cero en caso de fracaso
<code>void i2c_antelsat_config()</code>	Configuración de los registros internos del módulo I2C del microcontrolador
<code>void i2c_antelsat_inicio_inst()</code>	Cargado de las variables del tipo estructura <code>param_i2c</code> , que se pasan como parámetro de entrada a la función <code>i2c_antelsat_envio()</code>
<code>void i2c_antelsat_callback()</code>	Función que almacena los datos recibidos en el buffer de entrada de I2C y los almacena en los lugares de memoria definitivos. Invoca o encola funciones en caso de que corresponda
<code>void i2c_msp_stat_tx()</code>	Se modifica un registro interno del microcontrolador para generar una condición de Start para realizar una escritura
<code>void i2c_antelsat_stop()</code>	Se modifica un registro interno del microcontrolador para generar una condición de Stop
<code>void i2c_antelsat_dir_esclavo(int dir)</code>	Se modifica un registro interno del microcontrolador para cargar la dirección I2C del dispositivo con el que se quiere realizar la transferencia
<code>__interrupt void USCI_B0_ISR(void)</code>	Vector de atención a interrupción del módulo I2C.

5.1.4.4. Diagrama de flujo de la comunicación

A continuación se exponen los diagramas de flujo correspondientes a la comunicación

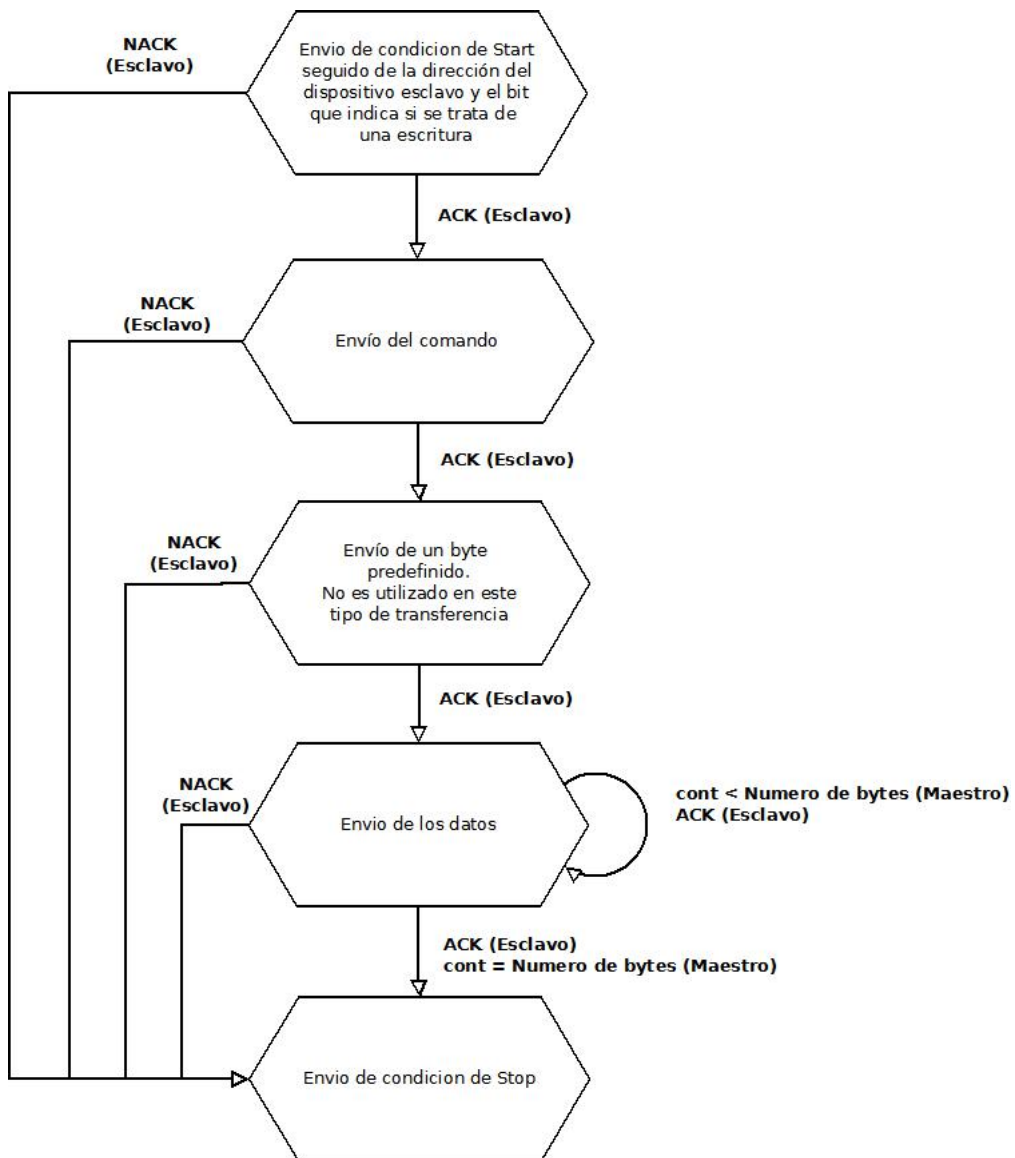


Figura 5.8: Envío de datos del maestro

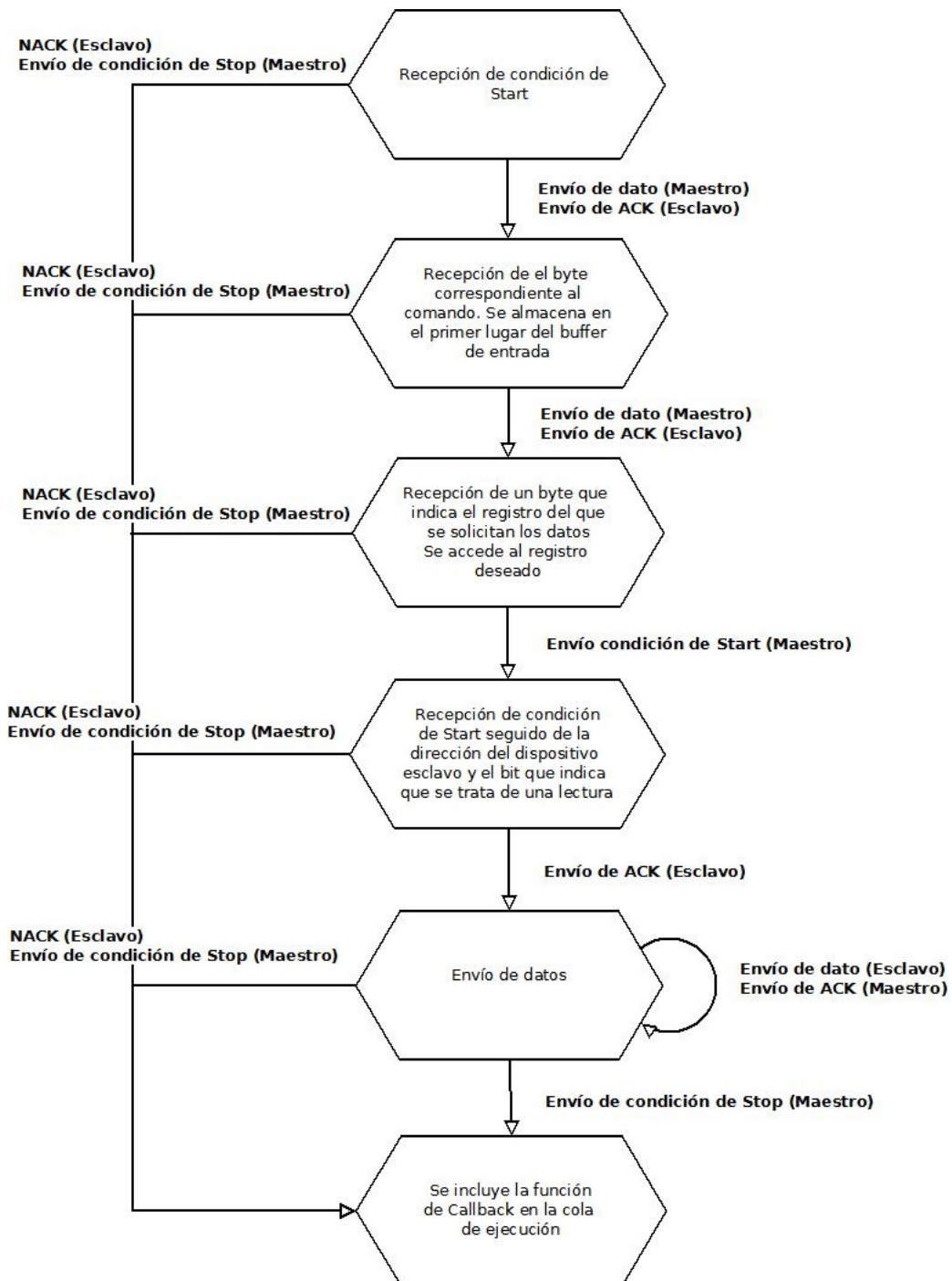


Figura 5.9: Envío de datos del esclavo

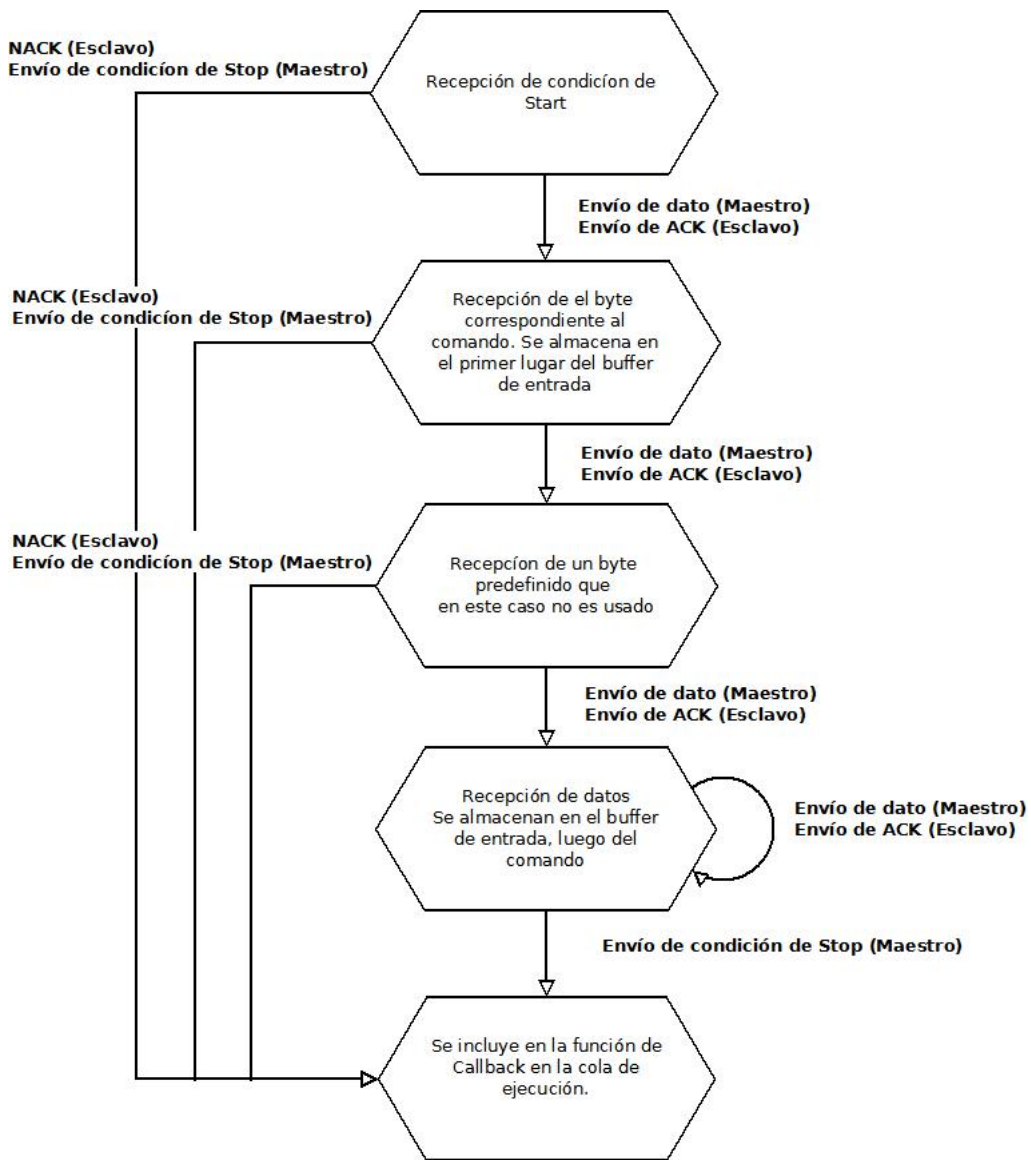


Figura 5.10: Lectura de datos del esclavo

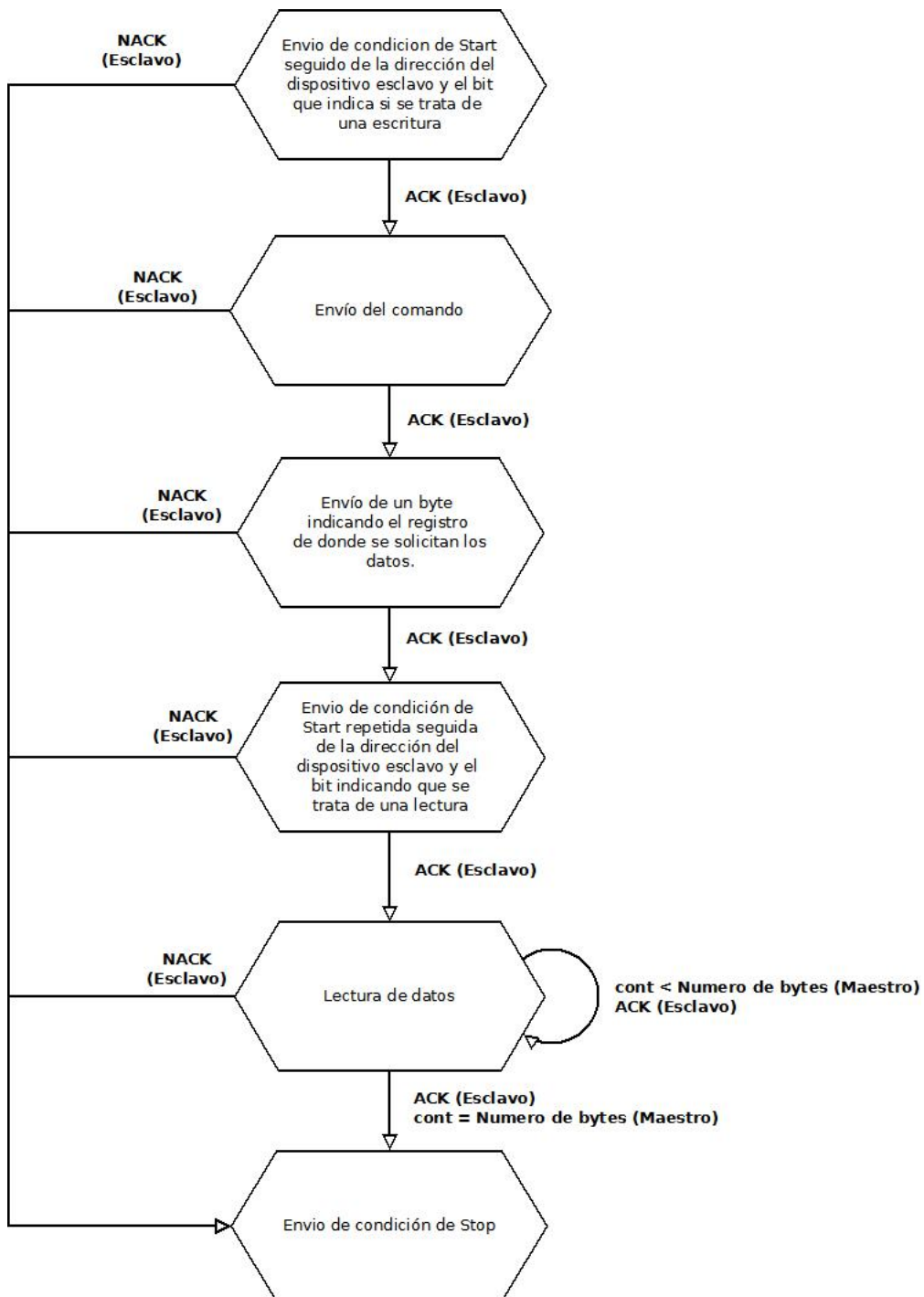


Figura 5.11: Lectura de datos del maestro

5.1.5. Modo de bajo consumo

En un proyecto en donde la fuente de energía consiste en un grupo de baterías, es relevante minimizar el consumo de energía. Para abordar el problema, son útiles consideraciones como el uso de interrupciones que controlen el flujo del programa, alimentar los periféricos solamente cuando necesitan ser usados, y evitar invocaciones frecuentes a subrutinas y funciones. Uno de los motivos que llevaron a la elección del MSP430F5438A es el hecho de que cuenta con modos de bajo consumo. A continuación se detallan las opciones con las que se cuenta al momento de seleccionar uno de los modos.

El fabricante del chip sugiere que se mantenga al microcontrolador el mayor tiempo posible LPM3 o LPM4 para disminuir el consumo. En el cuadro 5.8 se puede observar que los módulos utilizados en el proyecto hacen uso de diferentes fuentes de reloj, algunas de las cuales son deshabilitadas al entrar en estos modos. El hecho de deshabilitar una fuente de reloj implica que el módulo que la utiliza deje de funcionar. El módulo USCI (Universal Serial Communication Interface) de I2C cuenta con una activación automática de reloj para el trabajo con los modos de bajo consumo. Ésto significa que el dispositivo es capaz de recibir mensajes aún cuando el sistema se encuentre en un LPM en donde la fuente de reloj utilizada para la comunicación I2C esté deshabilitada. Por este motivo no existen limitaciones al momento de la elección del LPM en el cual trabajar.

Cuadro 5.8: Modos de bajo consumo

Modos de bajo consumo	Int. Habilitadas	ACLK	CPU/MCLK	FLL	Vcore	SMCLK
LPM0	SI	ON	OFF	ON	ON	OPT
LPM1	SI	ON	OFF	OFF	ON	OPT
LPM2	SI	ON	OFF	OFF	ON	OFF
LPM3	SI	ON	OFF	OFF	ON	OFF
LPM4	SI	OFF	OFF	OFF	ON	OFF

Lo que se debe hacer, es agregar una sentencia en el código de la rutina

que se ejecuta con la llegada de un mensaje, a fin de garantizar la salida del modo de bajo consumo en el que se encuentra y poder procesar los datos recibidos.

Low-Power Mode Supply Currents (Into V_{CC}) Excluding External Current

over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)^{(1) (2)}

PARAMETER	V_{CC}	PMMCOREVx	-40°C		25°C		60°C		85°C		UNIT
			TYP	MAX	TYP	MAX	TYP	MAX	TYP	MAX	
$I_{LPM0.1MHz}$ Low-power mode 0 ^{(3) (4)}	2.2 V	0	69	93	69	93	69	93	69	93	μA
	3.0 V	3	73	100	73	100	73	100	73	100	
I_{LPM2} Low-power mode 2 ^{(5) (4)}	2.2 V	0	11	15.5	11	15.5	11	15.5	11	15.5	μA
	3.0 V	3	11.7	17.5	11.7	17.5	11.7	17.5	11.7	17.5	
$I_{LPM3.XT1LF}$ Low-power mode 3, crystal mode ^{(6) (4)}	2.2 V	0	1.4		1.7		2.6		6.6		μA
		1	1.5		1.8		2.9		9.9		
		2	1.5		2.0		3.3		10.1		
	3.0 V	0	1.8		2.1	2.4	2.8		7.1	13.6	
		1	1.8		2.3		3.1		10.5		
		2	1.9		2.4		3.5		10.6		
$I_{LPM3.VLO}$ Low-power mode 3, VLO mode ^{(7) (4)}	3.0 V	3	2.0		2.3	2.6	3.9		11.8	14.8	
		0	1.0		1.2	1.42	2.0		5.8	12.9	
		1	1.0		1.3		2.3		6.0		
		2	1.1		1.4		2.8		6.2		
I_{LPM4} Low-power mode 4 ^{(8) (4)}	3.0 V	3	1.2		1.4	1.62	3.0		6.2	13.9	
		0	1.1		1.2	1.35	1.9		5.7	12.9	
		1	1.2		1.2		2.2		5.9		
		2	1.3		1.3		2.6		6.1		
$I_{LPM4.5}$ Low-power mode 4.5 ⁽⁹⁾	3.0 V		0.10		0.10	0.13	0.20		0.50	1.14	μA

Figura 5.12: Tabla de los modos de bajo consumo

Como se puede apreciar en la figura 5.12, la diferencia entre la corriente de consumo de los diferentes modos es significativa. Queda claro entonces el por qué de la sugerencia del fabricante al momento de la elección.

Si bien el microcontrolador cuenta con la posibilidad de entrar en modos de bajo consumo, es importante destacar que el ahorro de energía depende del diseño, es decir de la forma en la que se utilice este recurso.

El criterio de diseño que se adopta es el conocido como *Greedy* y consiste en que si no hay una tarea para ejecutar se entra en un modo de bajo consumo. Si bien esta elección optimiza el consumo frente a la situación en la que el microcontrolador permanece siempre encendido, el grado de ahorro depende mucho del ciclo de trabajo, dado que entrar y salir de un modo de bajo consumo implica un costo energético y un tiempo determinado.

Se considera la siguiente situación:

- Tarea 1 con duración T_1 y consumo P_1
- Tarea 2 con duración T_2 y consumo P_2
- Tarea 3 con duración T_3 y consumo P_3
- Un modo de bajo consumo LPM con consumo P_{LPM} y consumos de transición P_{ON-LPM} y $P_{LPM-OFF}$
- Un sistema basado en encolado de funciones

Para simplificar el ejemplo se supone que el consumo y duración de todas las tareas es el mismo.

Si se estuviese en el caso en que el microcotrolador se encuentra siempre encendido se obtienen los resultados que se muestran en la figura 5.13.

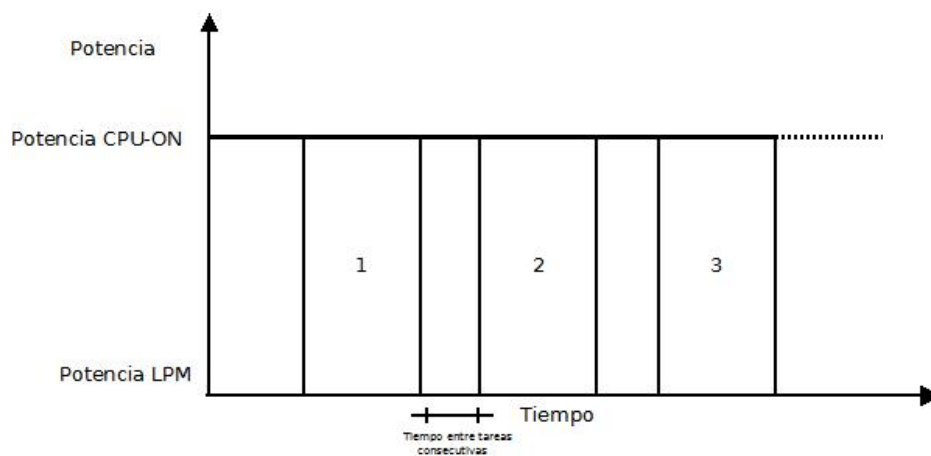


Figura 5.13: Consumo sin LPM

Por otro lado, dos situaciones similares que se basan en el criterio *Greedy* son las siguientes:

- Entrar en un LPM una vez que se termina de ejecutar una tarea. Es decir, se saca la función de la cola y una vez ejecutada se entra en el LPM. En la figura 5.14 se muestra este modo de funcionamiento.

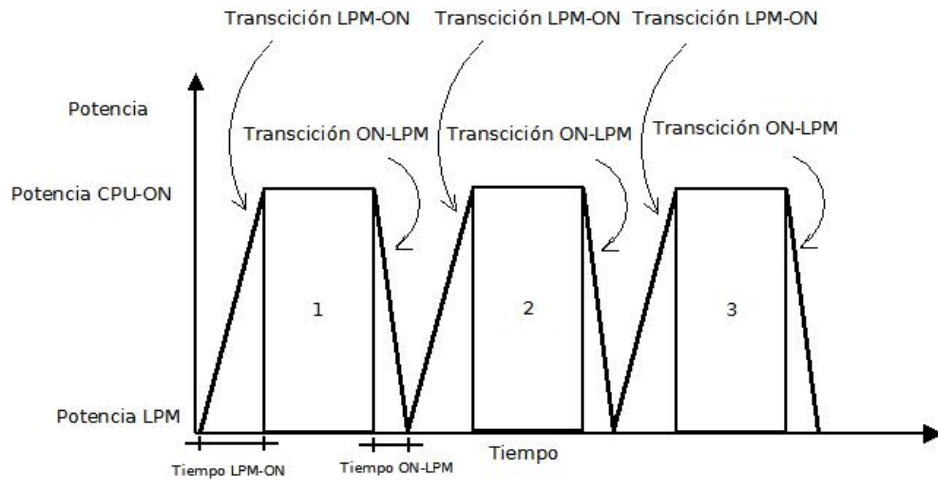
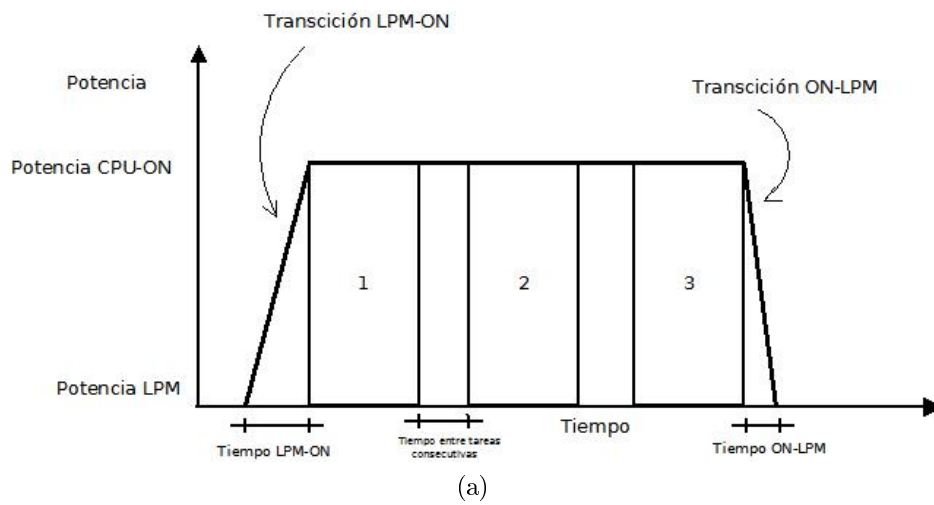
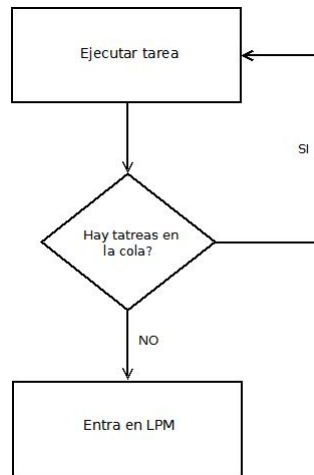


Figura 5.14: Entrada al LPM al finalizar la ejecución de una tarea

- Entrar en un LPM una vez que se finaliza de ejecutar una tarea sólo si no hay más tareas para ejecutar en la cola. De esta forma, se evita entrar en LPM entre tareas consecutivas. La figura 5.15 muestra este modo de funcionamiento.



(a)



(b)

Figura 5.15: Entrada al LPM una vez que finaliza la ejecución de todas las tareas

Para que se pueda tomar una decisión entre los dos criterios anteriores, es necesario medir el consumo en ambas situaciones. En este ejemplo simplificado la evaluación se limita a comparar el mantener el micro encendido el intervalo de tiempo comprendido entre la ejecución de tareas consecutivas, con la suma de los consumos correspondientes a las transiciones de modo activo-LPM y LPM-modo activo.

En este proyecto se utiliza la segunda situación. El cuadro 5.9 muestra la

función implementada por el módulo LPM.

Cuadro 5.9: Funciones del módulo LPM

Nombre de la función	Descripción
<code>void low_power_in_antelsat()</code>	Se invoca a una función propia del microcontrolador que lo lleva al modo de bajo consumo LPM3

5.1.6. Housekeeping

El sistema lleva dos tipos de registros con el estado funcional de cada módulo que compone el satélite: una consulta con los últimos valores registrados y un resumen con los valores más representativos de cada consulta.

El MCP realiza periódicamente un pedido de housekeeping a cada módulo y guarda esos valores relevados en memoria. Con cada pedido de housekeeping estos valores se sobrescriben. Para evitar perder datos importantes, los valores más relevantes son guardados en un resumen y éste es enviado a Tierra cada vez que la estación terrena lo requiera. Aparte de esto, existe una rutina periódica en el scheduler que envía los valores del último pedido de housekeeping al sistema por el *beacon* [8] del satélite.

Cada pedido de housekeeping a un módulo viene acompañado de una consulta RTC para generar una estampa de tiempo con cada dato. De esta manera se mantiene un registro temporal de las variables.

El sistema reserva un máximo de 1Kbyte de datos en el resumen. Guardando 5 datos de housekeeping por módulo con sus estampas de tiempo correspondientes, se puede llegar a tener 17 registros por resumen. Se debe tener en cuenta que este número puede variar en función de la cantidad de datos que se desean incluir en el resumen.

Un ejemplo de las variables que se relevan durante el housekeeping se puede ver en la tabla 5.10. En este caso se muestran las variables correspondientes al módulo ADCS. En el cuadro 5.11 se detallan las funciones implementadas en el módulo de Housekeeping.

Cuadro 5.10: Datos housekeeping

Módulo	Registro	Tipo	Descripción
ADCS	MODE	R	Indica el modo de funcionamiento activo del módulo
ADCS	MAGX	R	Medida del magnetómetro X
ADCS	MAGY	R	Medida del magnetómetro Y
ADCS	MAGZ	R	Medida del magnetómetro Z
ADCS	SUN1	R	Medida de fotodiodo 1
ADCS	SUN2	R	Medida de fotodiodo 2
ADCS	SUN3	R	Medida de fotodiodo 3
ADCS	SUN4	R	Medida de fotodiodo 4
ADCS	SUN5	R	Medida de fotodiodo 5
ADCS	SUN6	R	Medida de fotodiodo 6
ADCS	ERROR_1	R	Vector de errores 1
ADCS	ERROR_2	R	Vector de errores 2
ADCS	Q1_LSB	R	Parámetros de los cuaterniones de rotación
ADCS	Q1_MSB	R	Parámetros de los cuaterniones de rotación
ADCS	Q2_LSB	R	Parámetros de los cuaterniones de rotación
ADCS	Q2_MSB	R	Parámetros de los cuaterniones de rotación
ADCS	Q3_LSB	R	Parámetros de los cuaterniones de rotación
ADCS	Q3_MSB	R	Parámetros de los cuaterniones de rotación
ADCS	Q4_LSB	R	Parámetros de los cuaterniones de rotación
ADCS	Q4_MSB	R	Parámetros de los cuaterniones de rotación
ADCS	WX_LSB	R	Velocidades angulares
ADCS	WX_MSB	R	Velocidades angulares
ADCS	WY_LSB	R	Velocidades angulares
ADCS	WY_MSB	R	Velocidades angulares
ADCS	WZ_LSB	R	Velocidades angulares
ADCS	WZ_MSB	R	Velocidades angulares
ADCS	ON_OFF_STATUS	R	Vector indicando status
ADCS	B_DOT_GAIN_X	R/W	Ganancia del controlador Bdot
ADCS	B_DOT_GAIN_Y	R/W	Ganancia del controlador Bdot
ADCS	B_DOT_GAIN_Z	R/W	Ganancia del controlador Bdot
ADCS	UMBRAL_SPIN	R/W	Umbral de definición de modo spin_DET ACT
ADCS	UMBRAL_B	R/W	Umbral_DET_ACT
ADCS	SUN_MOD	R	Último salida del modelo sol
ADCS	B_MOD	R	Último estimado del modelo magnético
ADCS	MAG_GAIN_LSB	R/W	Ganancia para la calibración del magnetómetro
ADCS	MAG_GAIN_MSB	R/W	Ganancia para la calibración del magnetómetro
ADCS	SUN_1_GAIN_LSB	R/W	Ganancia para la calibración del magnetómetro
ADCS	SUN_1_GAIN_MSB	R/W	Ganancia para la calibración del magnetómetro
ADCS	SUN_2_GAIN_LSB	R/W	Ganancia para la calibración del magnetómetro
ADCS	SUN_2_GAIN_MSB	R/W	Ganancia para la calibración del magnetómetro
ADCS	SUN_3_GAIN_LSB	R/W	Ganancia para la calibración del magnetómetro
ADCS	SUN_3_GAIN_MSB	R/W	Ganancia para la calibración del magnetómetro
ADCS	SUN_4_GAIN_LSB	R/W	Ganancia para la calibración del magnetómetro
ADCS	SUN_4_GAIN_MSB	R/W	Ganancia para la calibración del magnetómetro
ADCS	SUN_5_GAIN_LSB	R/W	Ganancia para la calibración del magnetómetro
ADCS	SUN_5_GAIN_MSB	R/W	Ganancia para la calibración del magnetómetro
ADCS	SUN_6_GAIN_LSB	R/W	Ganancia para la calibración del magnetómetro
ADCS	SUN_6_GAIN_MSB	R/W	Ganancia para la calibración del magnetómetro
ADCS	ADCS_CAUSAS_RESET	R	Errores estimación entre medidas sensores y salidas de los modelos
ADCS	ADCS_ERRORRES	R	Errores estimación entre medidas sensores y salidas de los modelos

Todos los registros tienen un largo de 16 bits.

Cuadro 5.11: Funciones del módulo HK

Nombre de la función	Descripción
<code>void realizar_housekeeping()</code>	Función general para el llamado a realizar housekeeping.
<code>void inicializo_resumen()</code>	Inicializa la cola del resumen de hk.
<code>void resumir_hk()</code>	Realiza el resumen de housekeeping.

5.1.7. Watchdog

El watchdog resulta una herramienta importante en el funcionamiento del satélite. Se puede configurar para generar interrupciones a intervalos de tiempos determinados, o disparar un reinicio del sistema una vez transcurrido cierto tiempo.

Ésto último resulta de mucha utilidad cuando se quiere evitar que el sistema quede ocioso debido a problemas en el software. Durante la ejecución del sistema principal y de forma periódica, el contador del watchdog debe ser reiniciado para eliminar los resets indeseados en medio de un flujo correcto.

El hecho de que el módulo MCP se encuentre en una etapa de desarrollo, hace difícil determinar si el intervalo máximo disponible para el reseteo del watchdog es suficiente para la ejecución de todas las tareas. En caso de contar con una tarea cuyo tiempo de ejecución sea mayor al período seleccionado, se puede considerar reiniciar el contador mas de una vez dentro de la misma tarea.

El reinicio del contador se realiza invocando a la función `wdt_reset_antelsat()`. Actualmente es invocada al principio de cada función que se ejecuta, y en la rutina de atención a interrupción del Timer B. El último punto evita un reinicio del sistema cuando el micro se encuentra en LPM3, reseteando el watchdog con cada interrupción.

A continuación se muestra un extracto de código de la rutina de interrupción del TIMER B:

```

    #pragma vector = TIMERB0_VECTOR
    __interrupt void Timer_B (void){
    __bic_SR_register_on_exit(LPM3_bits);
    % Sale del modo de bajo consumo LPM3
    wdt_reset_antelsat(); Reinicio del conatdor del watchdog
    __disable_interrupt(); Deshabilita interrupciones
    atencion_interrupcion();
    % Invoca a la función atencion_interrupcion()
    __enable_interrupt()}; Habilita interrupciones

```

El siguiente es un extracto de código de la rutina de envío de datos por I2C:

```

    i2c_antelsat_envio(param_i2c par){
    ** Transferencia de datos por medio del protocolo I2C. El parámetro
    de entrada contiene el comando a ser transmitido, la dirección del
    esclavo, el número de bytes a ser enviados o leídos.
    int exito;
    maestro = MAESTRO;
    bytes = par.num_bytes;
    inst_i2c = par.instruccion;
    count = CERO;
    % Se inicializa el contador
    ptxdata = (unsigned char*)par.data_out;
    offset = par.buffer_offset;
    wdt_reset_antelsat();
    % Reinicio del conatdor del watchdog
    i2c_antelsat_dir_esclavo(par.dir_esclavo);
    % Se carga la direccion del esclavo
    .
    .
    .
    .
    }

```

En el cuadro refwd se muestra las funciones implementadas en el módulo Watchdog.

Cuadro 5.12: Funciones del módulo WDT

Nombre de la función	Descripción
<code>void wdt_config_antelsat()</code>	Configuración de los registros internos del módulo de WDT timer del microcontrolador
<code>void wdt_reset_antelsat()</code>	Reset del contador del WDT timer para evitar un reinicio del sistema

5.1.8. Protocolo de comunicación entre la estación terrena y el satélite

5.1.8.1. Estructura de los paquetes de datos

El largo máximo de bytes útiles es 128. El MCP cuenta con un buffer de 128 bytes para el almacenamiento de los datos recibidos y su posterior análisis.

Los datos se encapsulan en paquetes de:
1 byte para el largo + 2 bytes para instrucciones + n bytes de datos, con $0 < n < 125$.

Largo (8 bits)	Inst. (16 bits)	Dato1 (8)	. . .	DatoN (8)
----------------	-----------------	-----------	-------	-----------

La variable largo indica el número n de bytes que vienen inmediatamente después de los dos bytes correspondiente a la instrucción. Debido a que el número máximo de bytes de datos a enviar por AX.25 es 128 y hay 3 bytes reservados para su configuración, el valor de n sólo puede variar entre 0 y 125.

Al momento de concluir este trabajo no existía todavía una lista definitiva de las instrucciones válidas a ser enviadas entre el satélite y la estación terrena. Por tal motivo es que se decide reservar dos bytes para las instrucciones, y de esta manera asegurar la posibilidad de definir en el futuro un gran rango de instrucciones si se desea.

La única restricción que se impone es que todas las intrucciones deben empezar y terminar con un bit 1. El MCP realiza esta verificación antes de

procesar cualquier dato. Si esta condición se cumple, entonces se considera que la trama viene de una fuente confiable. En caso contrario, el MCP lo procesa como un pedido de repetición de paquetes AX.25 de alguna radio cualquiera y lo reenvía al COM para su transmisión.

Cuando la estación terrena desea hacer un pedido de housekeeping al satélite, debe mandar la siguiente instrucción: «1010 1010 1010 1011». Los bit más y menos significativos corresponden a un envío desde la estación terrena, mientras que la serie «010 1010 1010 101» es la instrucción que indica un pedido de housekeeping. Una vez que llega este tipo de pedido, el MCP procede a enviar el resumen de datos que ya fuera mencionado en la sección 5.11.

5.1.8.2. Encriptado de paquetes de datos

Para el encriptado de datos se emplea el protocolo TEA. Los únicos datos que se encriptan son los de uplink, por lo tanto, el encriptado se realiza solamente en la estación terrena para luego ser desencriptados en el satélite. Los datos de downlink se envían en estado puro, sin ningún tipo de cifrado mas que el de la modulación AX.25.

5.1.8.3. Modulación

Se utiliza el protocolo AX.25 para establecer la conexión entre el equipo terminal y el receptor. Cuando el paquete proveniente de la estación terrena llega al satélite, el módulo de comunicación hace una verificación primaria en la que controla que los bits de datos no estén corruptos. En caso de que exista una corrupción, se descarta el dato y no se realiza ninguna operación. Si los datos son correctos, se eliminan los encabezados y bytes específicos de AX.25 y se toman los bytes con la información de la estación terrena.

5.1.8.4. Desencriptado

El MCP recibe el conjunto de bytes y realiza el desencriptado TEA para recuperar el mensaje de Tierra. Los primeros tres bytes son analizados para poder determinar si el mensaje proviene de la estación terrena. El módulo de verificación se fija en las siguientes dos características:

1. El primer byte es un entero entre 0 y 125.

2. El segundo y tercer byte coinciden con instrucciones previamente definidas y que sólo la estación Terrena y el MCP conocen.

Los datos que efectivamente provienen de la estación terrena se interpretan y pasan a ejecutarse, mientras que los datos que no cumplen alguna de las condiciones anteriores son interpretados como peticiones de repetición de paquetes AX.25 desde estaciones terrenas cualesquiera, y se envían nuevamente al módulo de comunicación para que éste los envíe a Tierra.

5.1.8.5. Diagramas de flujo de la comunicación

En las figuras 5.16 y 5.17 se muestran los diagramas de flujo correspondientes a la comunicación.

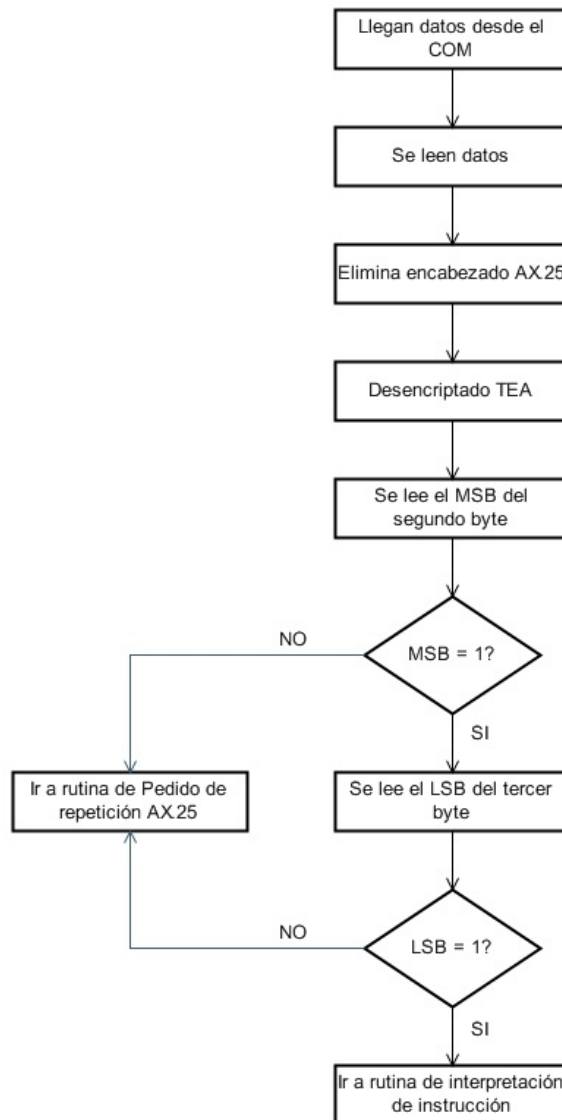


Figura 5.16: Diagrama de flujo de la atención a la llegada de datos del COM

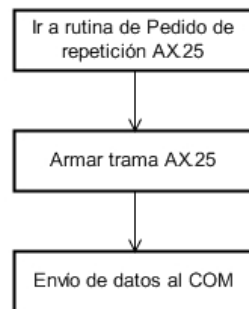


Figura 5.17: Diagrama de flujo de la atención a un pedido de repetición de paquetes AX.25

5.2. Diseño del hardware

5.2.1. Elección del microcontrolador

Inicialmente, el módulo del MCP iba a ser implementado en el mismo microcontrolador en el que estaba implementado el módulo ADCS, el MSP430F5438A de Texas Instruments. Durante el desarrollo del proyecto se decide separar ambos módulos en diferentes microcontroladores, quedando el MCP en el microcontrolador original ya que cuenta con los requerimientos necesarios para su correcto funcionamiento.

Los factores considerados al momento de seleccionar el microcontrolador son los siguientes:

1. Bajo consumo de potencia:

El satélite funciona mediante energía solar recolectada por medio de paneles solares ubicados en sus caras. La alimentación es proporcionada por una batería que almacena la energía durante el ciclo de luz solar. Es de suma importancia que cada uno de los módulos alimentados por la batería consuman la menor cantidad de energía posible. El microcontrolador debe permitir recurrir a modos de bajo consumo.

2. Posibilidad de integración con otro hardware:

El módulo MCP debe interactuar con otros módulos implementados en otros microcontroladores. Al mismo tiempo, debe contar con la posibilidad de conexión a sensores e integrados necesarios para el funcionamiento del módulo.

3. RTC:

Es necesario que el microcontrolador cuente con un Reloj de Tiempo Real (RTC) para tener conocimiento instantáneo de la fecha y la hora a fin de llevar un registro temporal de los eventos que van sucediendo y los estados de salud del sistema.

4. Posibilidad de comunicación I2C:

La comunicación con los demás módulos del sistema se realiza a través del bus serie I2C. El microcontrolador debe soportar este tipo de comunicación.

5. Velocidad de procesamiento:

Si bien no existen funciones que requieran gran velocidad de funcionamiento, es necesario que el MCP ejecute sus tareas de forma rápida ya que es quien controla el funcionamiento total del satélite.

6. Memoria interna:

El software del MCP y los logs que mantiene ocupan poco espacio en memoria por lo que este factor no resulta determinante al momento de elegir el microcontrolador.

7. Voltaje de alimentación:

El microcontrolador debe funcionar con el nivel de alimentación de 3.3V provisto por el SGE.

8. Rango de temperaturas soportadas:

El rango de temperatura que debe soportar es de -40°C a $+65^{\circ}\text{C}$.

También se estudiaron los microcontroladores utilizados por otros proyectos similares con el objetivo de comparar características. En el cuadro 5.13 se presentan los satélites comparados [17], [18], [19].

Cuadro 5.13: Satélites comparados

	MICRO	ROM	RAM	FLASH	BUS	PAYLOAD
AAU Danish Cube	C161PI	256KB	8x512KB	no tiene	I2C	Cámara
SwissCube	ATMEL ARM AT91M55800A + MSP430f164	512KB	512KB	4MB	I2C	Cámara
ION	Hitachi 7045	256kB	1MB	8MB	-	Tubo fotomultiplicador + Cámara

5.2.2. Conexionado de los módulos

El Control Principal debe integrar los diferentes módulos y elementos del satélite, desarrollados en diferentes proyectos. Los elementos que debe integrar son los que figuran a continuación:

- Módulo de Detección de Actitud.
- Magnetómetros.
- Magnetotorquers.
- Sensores de Temperatura.
- Fotodiodos.
- Módulo de Comunicación.
- Módulo de Energía.
- Módulo de Payload

En la figura 5.18 se muestra el conexionado entre los módulos.

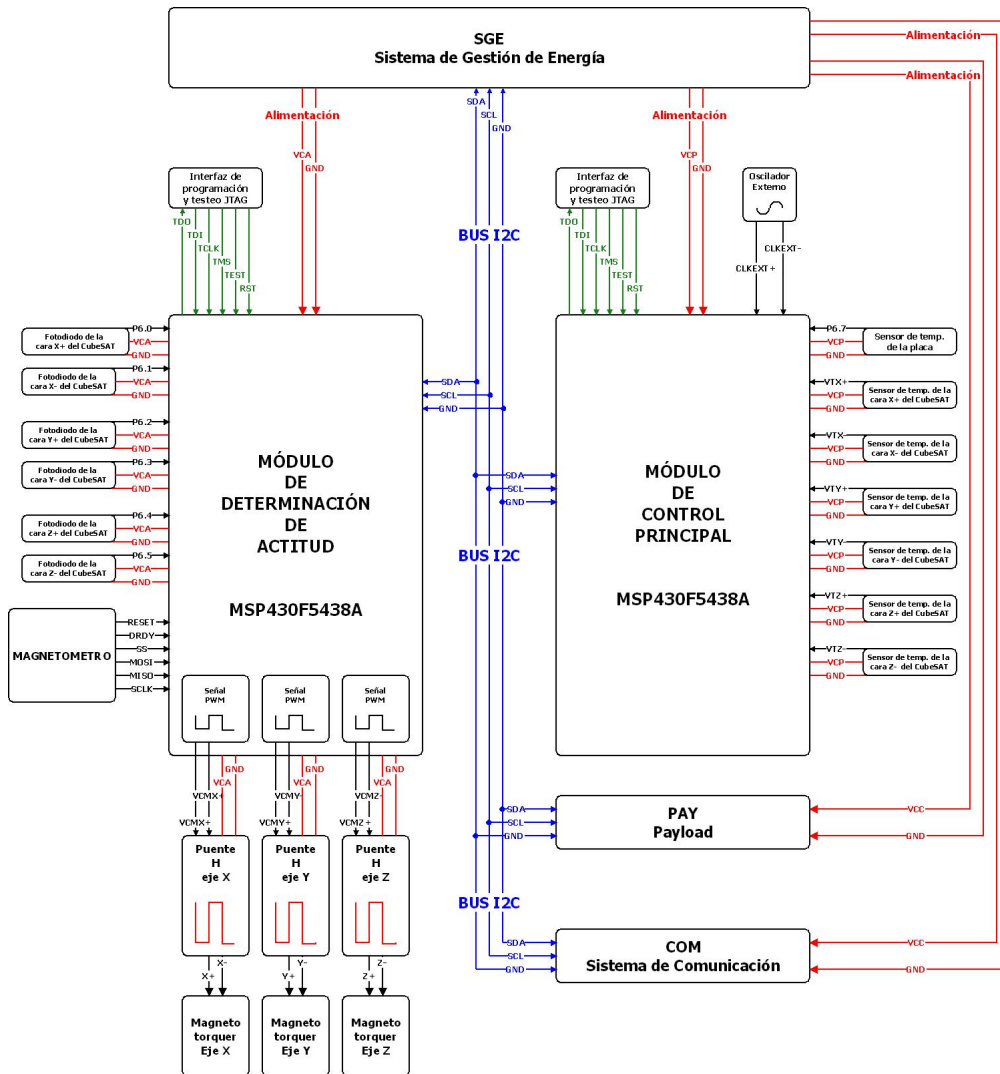


Figura 5.18: Conexión de módulos

5.2.3. Diseño del circuito impreso de prueba

En el diseño del circuito impreso de prueba se incluyen dos MSP430F5438A. Uno de los microcontroladores está destinado al control principal y otro para la determinación y control de actitud.

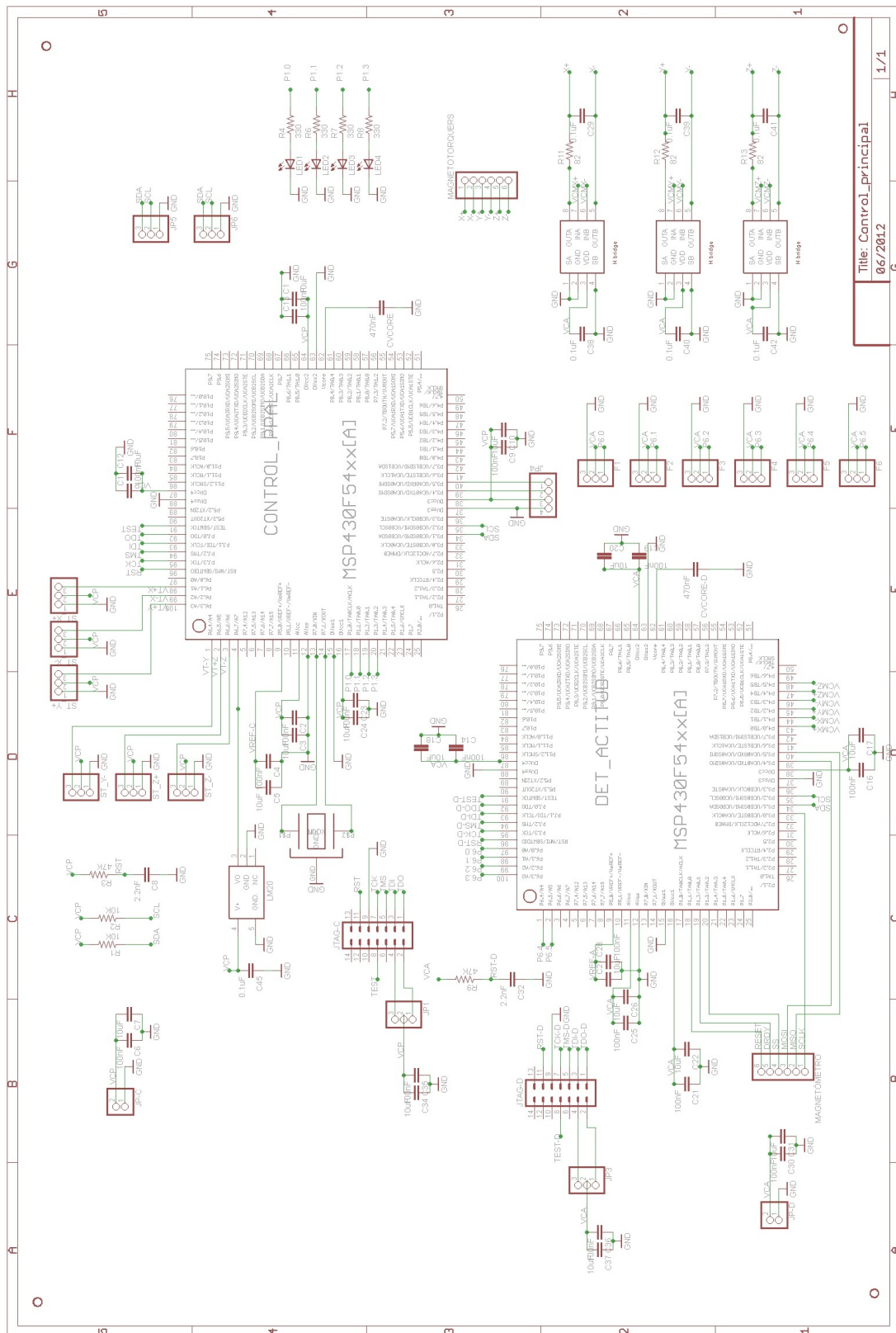


Figura 5.19: Esquemático de la placa de prueba

Dentro del circuito de prueba se integraron, para las diferentes funciones del control principal y determinación y control de actitud, los siguientes elementos.

- Conexión entre microcontroladores, bus I2C.
- Sensor de temperatura, para medir la temperatura interna del circuito impreso.
- Oscilador externo.
- Conexiones para sensores de temperatura externos, para medir la temperatura de las caras del CubeSat.
- Interfaz de de programación y testeo JTAG.
- Conexiones para fotodiodos.
- Conexiones para Magnetómetro.
- Puentes H y conexiones para magnetotorquers.

5.2.3.1. Conexión entre microcontroladores

La conexión entre los microcontroladores se realiza mediante el bus I2C. Para la misma se utilizan las líneas SDA, SCL y GND como referencia. Como las líneas SDA y SCL son open-drain es necesario poner resistencias de pull-up. En la figura 5.20 se muestra la conexión entre los microcontroladores.

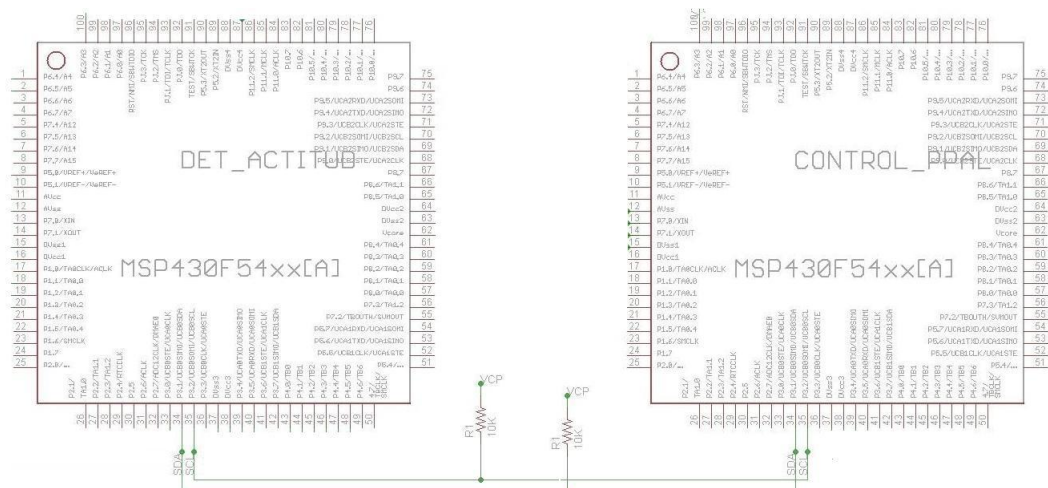


Figura 5.20: Conexión entre microcontroladores

Los pines utilizados en las conexión se detallan en la tabla 5.14.

Cuadro 5.14: Asignación de pines de sensores de temperatura

Pin micro. Principal	Señal	Pin micro. Determinación
34	SDA	34
35	SCL	35

5.2.3.2. Sensor de temperatura

Para medir la temperatura de las caras del CubeSat se utiliza el sensor de temperatura LM20 [14], la hoja de datos de este componente se puede ver en el Anexo F [9]. El sensor LM20 trabaja en un rango de voltaje entre 2.4 y 5.5V, un rango de temperatura entre -55°C y 130°C , y presenta un consumo menor a 10uA. Por este motivo se consideró adecuada su utilización para realizar la medición de temperatura de la placa.

En la figura 5.21 se presenta su función de transferencia, la cual es predominantemente lineal con leves no linealidades en los extremos.

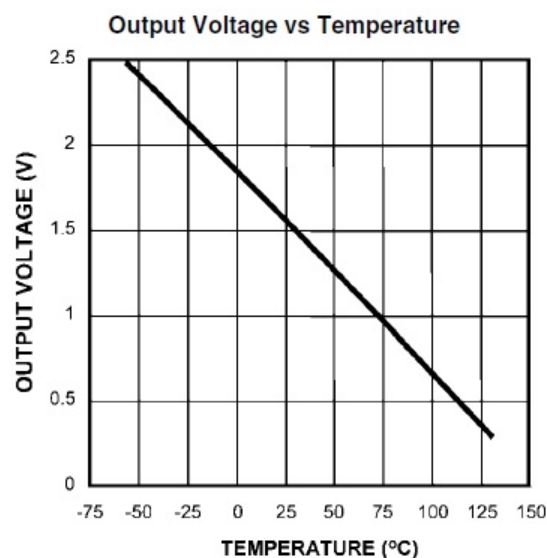


Figura 5.21: Función de transferencia del LM20

pinos para estos sensores.

Cuadro 5.15: Asignación de pines de sensores de temperatura

Pin del MSP430	Puerto del MSP430	Sensor
98	P6.1/A1	1(VT+X)
99	P6.2/A2	2(VT-X)
100	P6.3/A3	3(VT+Y)
1	P6.4/A4	4(VT-Y)
2	P6.5/A5	5(VT+Z)
3	P6.6/A6	6(VT-Z)
4	P6.7/A7	placa

5.2.3.3. Oscilador externo

Para obtener mayor precisión en la fuente de reloj usada por el módulo RTC se utiliza un cristal externo al microcontrolador. El oscilador utilizado es el MS1V-T1K, la hoja de datos de este componente se puede ver en el Anexo F [11]. En la figura 5.24 se observa la conexión entre el oscilador y el microcontrolador.

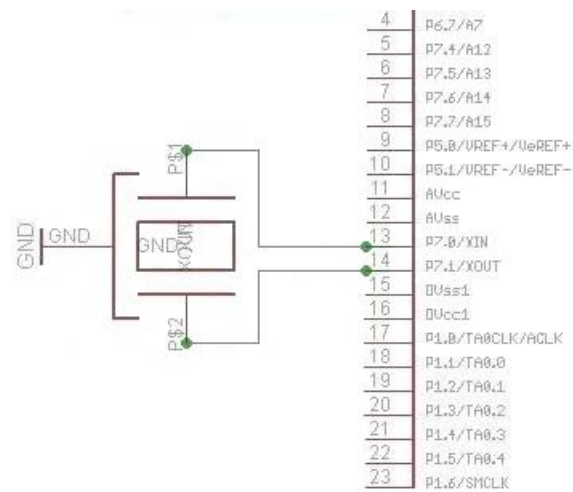


Figura 5.24: Conexión de oscilador externo.

En la tabla 5.16 se muestra la asignación de pines para el oscilador externo.

Cuadro 5.16: Asignación de pines de oscilador externo

Pin del MSP430	Puerto del MSP430	Oscilador
13	P7.0/XIN	Pin
14	P7.1/XOUT	Pout

5.2.3.4. Interfaz de programación y testeo JTAG

La familia MSP430 es compatible con la interfaz estándar JTAG, la cual es utilizada para la programación y testeo. Para efectuar estas tareas, la misma cuenta con las siguientes señales [15].

- TDI, entrada de datos.
- TDO, salida de datos.
- TCK, entrada de reloj.
- TMS, habilita pines JTAG(Selector de Modo de Testeo).
- TRST, reset externo (opcional).

El microcontrolador MSP430F5438A cuenta con las siguientes interfaces I/O para la conexión de un dispositivo JTAG [4].

- PJ.3 como entrada de reloj TCK.
- PJ.2 como entrada de TMS para el control de estado.
- PJ.1 como entrada de datos TDI.
- PJ.0 como salida de datos TDO.
- RST como entrada externa de reset.

Adicionalmente el microcontrolador cuenta con un pin de TEST que es necesario conectar a la interfaz de programación JTAG para habilitar las señales JTAG en los puertos del microcontrolador, dado que éstos son compartidos con puertos digitales de entrada-salida de propósitos generales [16].

La figura 5.25 muestra las conexiones entre el conector JTAG de 14 pines y el MSP430F5438 necesarias para la programación del microcontrolador que sugiere el fabricante.

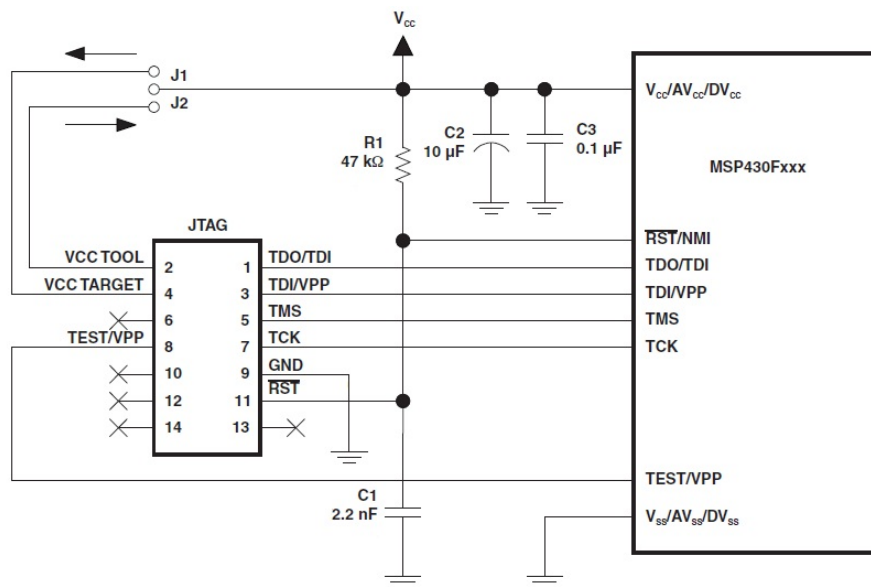


Figura 5.25: Conexiones de las señales del JTAG

1

¹Imágen disponible en: <http://www.ti.com>

La interfaz JTAG permite seleccionar la fuente con la cual se alimenta el circuito que se está programando o testeando. El circuito puede ser alimentado mediante una fuente externa o mediante la fuente de alimentación del dispositivo JTAG. En la figura 5.25 se puede apreciar que al conectar J1, la alimentación es suministrada por una fuente externa. Al conectar J2 el circuito utiliza la alimentación proveniente de la interfaz JTAG. Al elegir el modo de funcionamiento se debe tener especial cuidado en no conectar la fuente externa y J2 simultáneamente.

Para las pruebas se utiliza el MSP430 USB-Debug-Interface que se muestra en la figura 5.26.



Figura 5.26: MSP430 USB-Debug-Interface

2

En la placa de prueba se integran dos JTAGs, uno para cada uno de los microcontroladores. El conexionado es idéntico en los dos casos y se puede ver en la figura 5.27.

²Imágen disponible en: <http://www.ti.com>

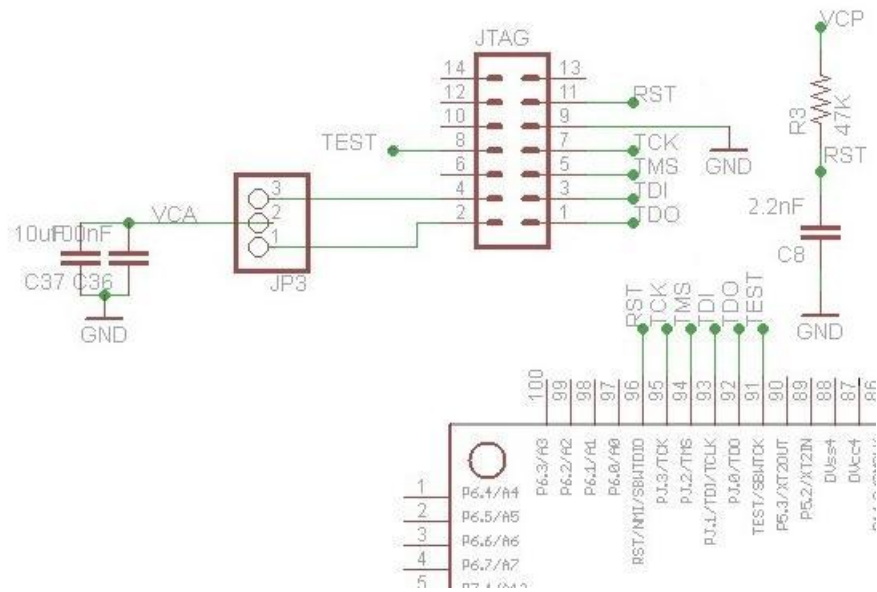


Figura 5.27: Conexión JTAGs.

Cada JTAG está conectado a un microcontrolador. En el cuadro 5.17 se muestra la asignación de pines para los JTAGs.

Cuadro 5.17: Asignación de pines de JTAG

Microcontrolador	Pin del MSP430	Puerto del MSP430	Pin JTAG
Control Principal	92	PJ.0/TDO	TDO
Control Principal	93	PJ.1/TDI/TCK	TDI
Control Principal	94	PJ.2/TMS	TMS
Control Principal	95	PJ.3/TCK	TCK
Control Principal	96	RST/NMI/SBWDIO	RST
Determinación	92	PJ.0/TDO	TDO
Determinación	93	PJ.1/TDI/TCK	TDI
Determinación	94	PJ.2/TMS	TMS
Determinación	95	PJ.3/TCK	TCK
Determinación	96	RST/NMI/SBWDIO	RST

5.2.3.5. Conexiones para fotodiodos

La placa de prueba incluye la conexión para los seis fotodiodos ubicados en las caras del CubeSat. Los mismos son utilizados para obtener la dirección del sol. Los fotodiodos que fueron seleccionados por un proyecto anterior [10] son los BPW34, la hoja de datos de este componente se puede ver en el Anexo F [20]. Los fotodiodos BPW34 requieren un circuito de acondicionamiento de señal previo a ser conectados al microcontrolador. En el circuito impreso diseñado se prevee la entrada de la señal proveniente de los fotodiodos ya acondicionada. Los fotodiodos son conectados al microcontrolador destinado a la determinación de actitud. En la figura 5.28 se muestran las conexiones previstas.

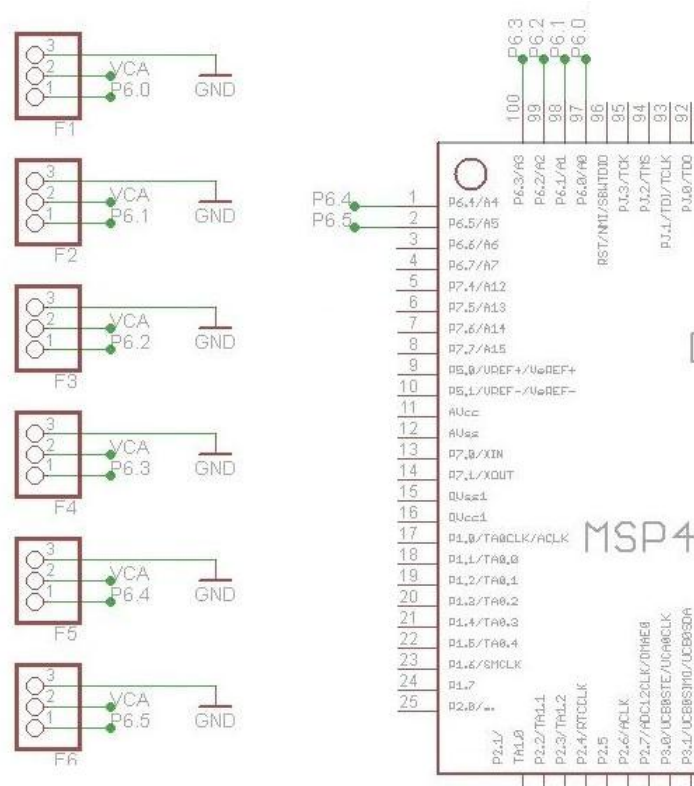


Figura 5.28: Conexiones para Fotodiodos

Los pines utilizados para la conexión de los mismos pueden observarse en el cuadro 5.18.

Cuadro 5.18: Asignación de pines para fotodiodos

Pin del MSP430	Puerto del MSP430	Fotodiodo
97	P6.0/A0	F1
98	P6.1/A1	F2
99	P6.2/A2	F3
100	P6.3/A3	F4
1	P6.4/A4	F5
2	P6.5/A5	F6

5.2.3.6. Conexiones para Magnetómetro

La placa de prueba incluye conexiones para el magnetómetro el cual provee la dirección y el módulo del campo magnético. El magnetómetro utilizado es el Micromag3 el cual fue elegido en un proyecto previo [10], la hoja de datos de este componente se puede ver en el Anexo F [21]. La conexión entre el Micromag3 y el microcontrolador de detección de actitud se realiza en forma directa, sin necesidad de componentes adicionales. En la figura 5.29 se muestran las conexiones entre el magnetómetro y el microcontrolador.

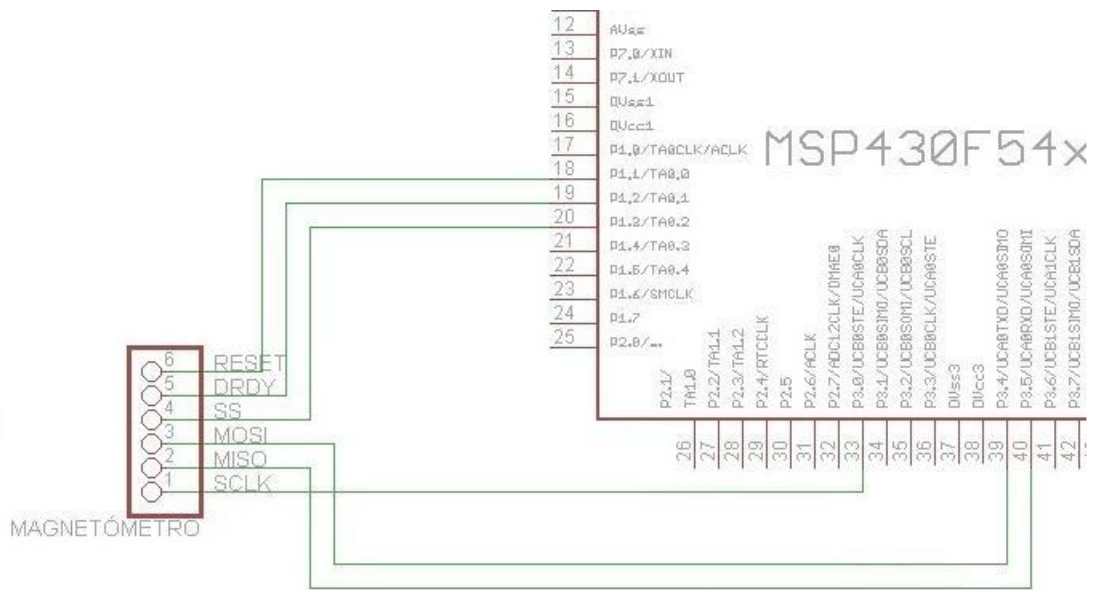


Figura 5.29: Conexiones para magnetómetro

Los pines utilizados para la conexión de los mismos pueden observarse en el cuadro 5.19.

Cuadro 5.19: Asignación de pines para magnetómetro

Pin del MSP430	Puerto del MSP430	Pin Micromag3	Señal
17	P1.1/TA0.0	6	Reset
18	P1.2/TA0.1	5	DRDY
19	P1.3/TA0.2	4	SS
33	P3.0/UCBOSTE/UCA0CLK	1	SCLK
39	P3.4/UCATXD/UCASIMO	3	MOSI
40	P3.5/UCARXD/UCASOMI	2	MISO

5.2.3.7. Puentes H y conexiones para magnetotorquers

Para realizar las funciones de estabilización y posicionamiento del CubeSat se utilizan magnetotorquers. Al hacer circular una corriente por los magnetorquers, éstos generan un campo magnético, el cual interactúa con el campo magnético terrestre, haciendo posible el cambio de posición o dirección del Satélite.

Para conectar los magnetotorquers al microcontrolador responsable de la determinación de actitud, es necesaria la utilización de puentes-H. Estos últimos se utilizan para convertir las señales de baja intensidad, provenientes del microcontrolador, en las corrientes necesarias para que los magnetorquers actúen sobre el campo magnético terrestre.

Las señales enviadas desde el microprocesador serán PWM. Dicha señales actuarán sobre el puente H, logrando, mediante la apertura y cierre de las compuertas internas del componente, generar, a partir de la fuente de alimentación proveniente del SGE, VCA, una señal PWM que sea capaz de suministrar la corriente necesaria a los magnetorquers.

El cubeSat contará con tres magnetorquers para poder realizar una rotación en cualquiera de los 3 ejes del satélite. Por esto, para el diseño de la placa, se utilizaron tres puentes H "Si9987 H-Bridges", la hoja de datos de estos componentes se puede ver en el Anexo F [22]. Los mismos tienen un rango de voltaje de entrada de 3,8 a 13,2 V y operan en un rango de temperatura que va desde -40 °C a los 85 °C. Pese a que el voltaje que suministra el sistema de energía SGE es de 3,3 V y se encuentra fuera del rango de operación recomendado por el fabricante de este dispositivo, existen estudios realizados por el proyecto Swiss Cube [23] que indican que los Si9987 H-Bridges funcionan correctamente con fuentes de voltaje de hasta 2V. Es por eso que se decidió, en una primera instancia, la utilización de este componente. En la figura 5.30 se muestra el esquemático de conexión.

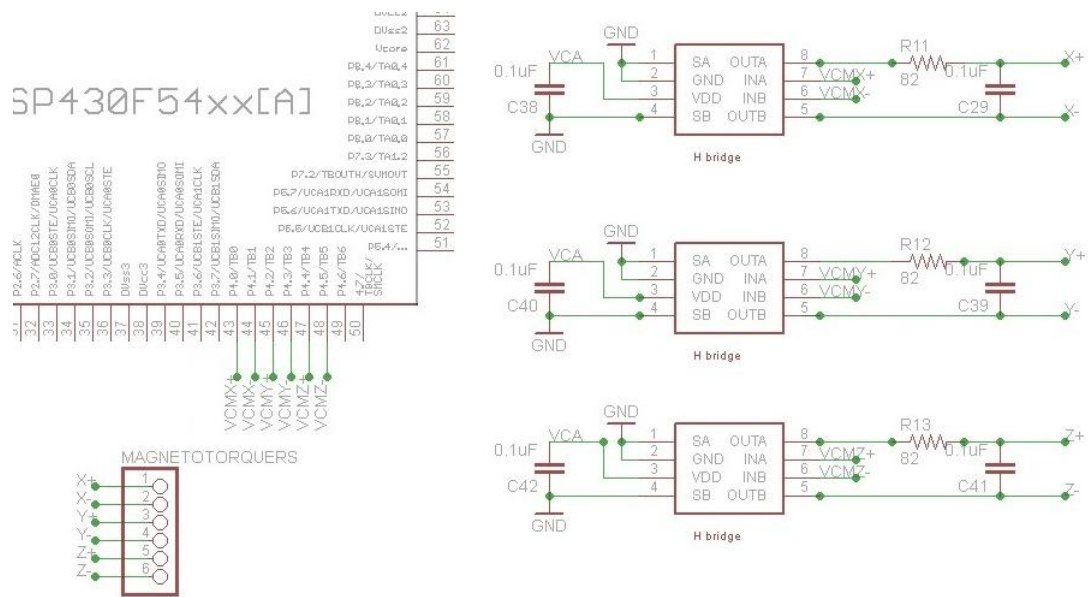


Figura 5.30: Conexiones para magnetotorquers

Los pines utilizados para la conexión de los mismos pueden observarse en el cuadro 5.20.

Cuadro 5.20: Asignación de pines para Puentes-H

Pin del MSP430	Puerto del MSP430	Señal Si9987
43	P4.0/TB0.0	XINa
44	P4.1/TB0.1	XINb
45	P4.2/TB0.2	YINa
46	P4.3/TB0.3	YINb
47	P4.4/TB0.4	ZINa
48	P4.5/TB0.5	ZINb

5.2.3.8. Implementación del circuito impreso

Para la realización del esquemático y el ruteo del circuito impreso de prueba se utilizó el software Eagle [24]. El ruteo de la placa se presenta en las figuras 5.31 y 5.32.

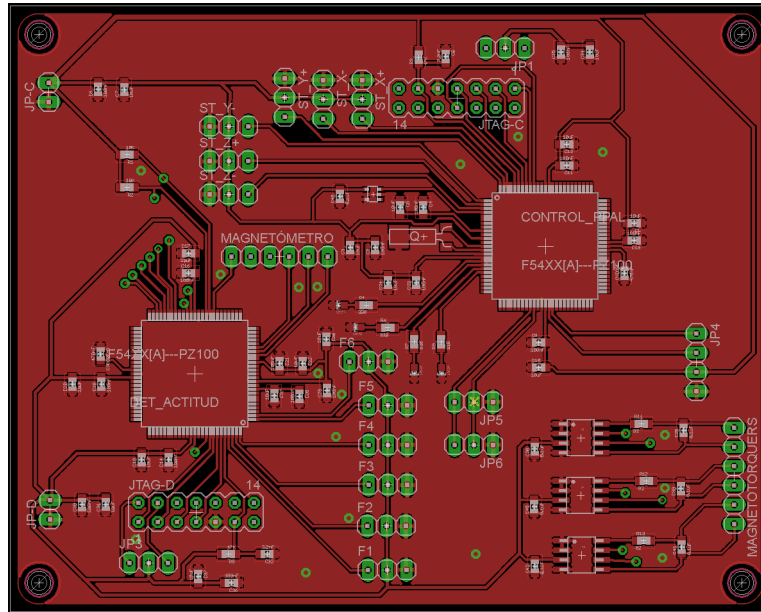


Figura 5.31: Capa superior de la placa de prueba

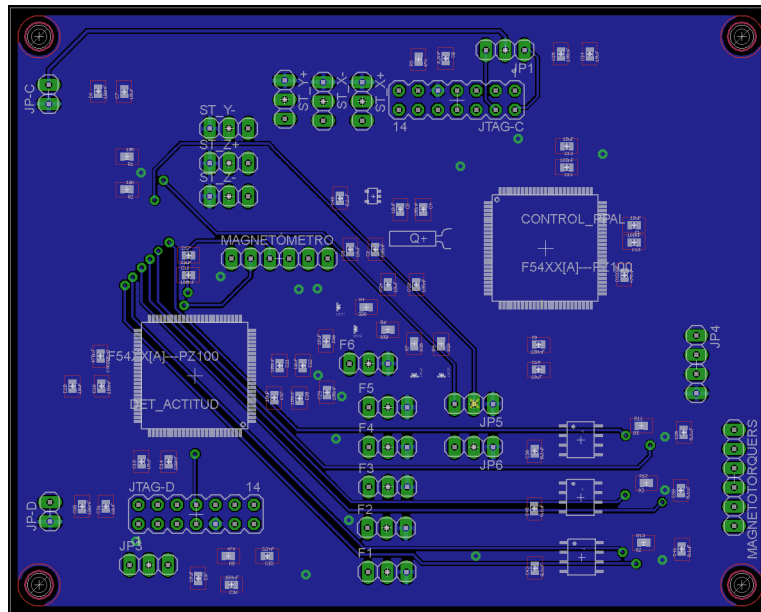


Figura 5.32: Capa Inferior de la placa de prueba

En la figura 5.33 se muestra una fotografía del módulo construido. La placa tiene integrados los componentes del módulo de control principal y cuenta con las huellas de los componentes del módulo de control de actitud.

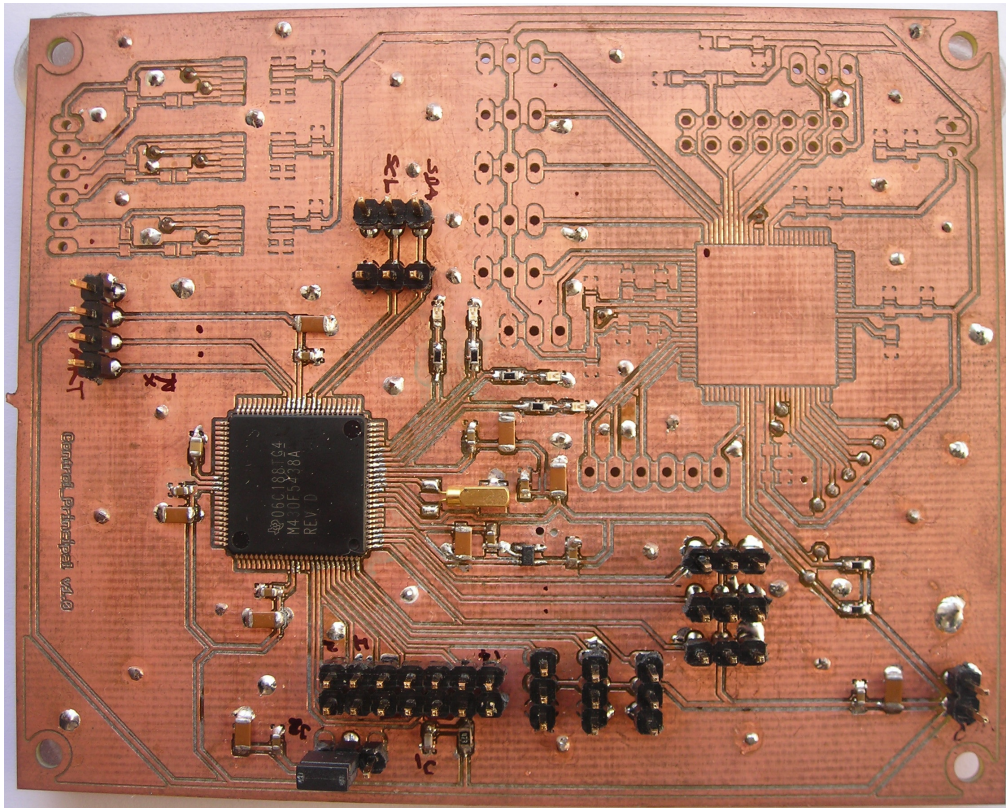


Figura 5.33: Foto de la placa de prueba

Capítulo 6

Pruebas

La implementación del código, el depurado y las pruebas se realizaron en el entorno de desarrollo IAR [25] mediante el uso del Fet Debugger y la placa MSP-TS430PZ5X-100 [26]. El entorno cuenta con numerosas herramientas, muchas de las cuales no se llegaron a utilizar. Un ejemplo de las herramientas empleadas, es el acceso al contenido de los registros del microcontrolador y el monitoreo de variables, lo que resultó de mucha ayuda para la corrección de errores y comprensión del funcionamiento.

Las comunicaciones por medio del protocolo I2C son parte fundamental del proyecto MCP. Llegar al resultado con el que hoy se cuenta, implicó varias etapas de estudio y pruebas en donde fue necesario superar algunos obstáculos. Si bien puede parecer que la implementación no reviste gran complejidad, significó varias horas de dedicación.

A continuación se resume el trabajo realizado en las diferentes etapas y se mencionan las pruebas realizadas:

El siguiente cuadro muestra algunos ejemplos de códigos [27] suministrados por el fabricante (Texas Instruments).

Cuadro 6.1: Ejemplos de código del fabricante

Ejemplos de Código de Texas Instruments - I2C	Descripción
MSP430x54x_uscib0_i2c_04.c	Maestro lee un byte del esclavo. Código del maestro
MSP430x54x_uscib0_i2c_05.c	Esclavo envía un byte al maestro. Código del esclavo
MSP430x54x_uscib0_i2c_06.c	Maestro envía un byte al esclavo. Código del maestro
MSP430x54x_uscib0_i2c_07.c	Esclavo recibe un byte del maestro. Código del esclavo
MSP430x54x_uscib0_i2c_08.c	Maestro envía varios bytes al esclavo. Código del maestro
MSP430x54x_uscib0_i2c_09.c	Esclavo recibe varios bytes del maestro. Código del esclavo
MSP430x54x_uscib0_i2c_10.c	Maestro lee varios bytes del esclavo. Código del maestro
MSP430x54x_uscib0_i2c_11.c	Esclavo recibe varios bytes del maestro. Código del esclavo

Como se puede observar, los ejemplos se corresponden en pares de la siguiente forma:

- Ejemplos 4 y 5
- Ejemplos 6 y 7
- Ejemplos 8 y 9
- Ejemplos 10 y 11

Luego de estudiar el funcionamiento del protocolo I2C se pasó a trabajar con los microcontroladores y los códigos suministrados por el fabricante. El hecho de que se tratara de nuestra primera incursión en el funcionamiento del IAR, sumado a que los códigos usados tenían errores, provocó que esta etapa llevara un tiempo considerable.

Al final de esta etapa se logró realizar las siguientes transferencias de datos entre dos microcontroladores:

- Envío por parte del maestro (uno y varios bytes) – Recepción del esclavo
- Lectura por parte del maestro (uno y varios bytes) – Envío del esclavo

6.1. Combinación de rutinas de escritura y lectura en un mismo dispositivo.

Habiendo superado la primera etapa, se pasó a implementar las rutinas para el esclavo y el maestro, de forma que éste último realice una escritura de datos en el esclavo y que a continuación realice la lectura de esos mismos.

Para lograr el objetivo, fue necesario profundizar en el estudio de la configuración de los registros, ver puntos en común entre los dos tipos de transferencias y trabajar sobre las diferencias para implementar las rutinas. Los numerosos intentos sirvieron para familiarizarse con detalles que hacen al protocolo.

Fue en este proceso en el que se adquirió la mayor parte de las herramientas usadas para lograr el producto final. También se realizaron transferencias de 1000 bytes para estudiar el comportamiento del bus en transferencias largas.

6.2. Cambio de sentido del flujo de datos

Pensando en diseño final del protocolo, se procedió a la implementación de rutinas que realizaran un cambio dirección en el flujo de datos. Luego de varias pruebas sin éxito, se logró que el maestro enviara dos bytes y leyera de un buffer en una sola comunicación.

El primer byte consistía en un comando, al segundo no se le dio uso en esta etapa. En primera instancia se optó por realizar parte del procesamiento del comando dentro de la rutina de interrupción ISR, lo que generó diversos problemas. Cualquier error que implicara no llegar al envío de la condición de Stop por parte del maestro significaba que la transferencia no se llevara a cabo y que fuese necesario reiniciar los dos microcontroladores involucrados

para realizar nuevas pruebas.

En esta etapa se decidió simplificar la ISR y dar paso a una función de callback para procesar los datos. Esto nos obligó a definir ciertas reglas que no se habían establecido. Como resultado, se establece un orden para almacenar los datos que posteriormente serán procesados por la función de callback, se avanza sobre la implementación de esta función y se implementa una única función de comunicación I2C que recibe como parámetros el comando a transmitir y la dirección del esclavo. La función realiza el envío del comando y dependiendo del valor de éste procede a enviar más datos o a realizar una lectura del esclavo.

6.3. Versión final del protocolo

La versión implementada en la etapa anterior resultó muy similar a la final, con la salvedad de algunos detalles de funcionamiento que se terminan de definir en esta sección.

Se implementan funciones que realizan las modificaciones en los registros de forma de separar hardware y software, y se procede a la incorporación de las funciones implementadas al flujo principal del programa que se estaba desarrollando en paralelo.

Nuevamente, partiendo de los códigos de ejemplo de Texas Instruments, se implementan programas para simular las diferentes instancias de transmisión y así testear el funcionamiento del programa principal en su conjunto.

6.4. Pruebas ADC

Para la implementación de las funciones relacionadas al ADC, también se parte de ejemplos de código de Texas Instruments.

Cuadro 6.2: Ejemplos de código del fabricante

Ejemplos de Código de Texas Instruments - ADC	Descripción
msp430x54x_adc12_10.c	Cálculo de la temperatura por medio del sensor incorporado en el microcontrolador
msp430x54x_adc12_10.c	Conversión de una secuencia de canales

El ejemplo del sensor de temperatura incorporado en el microcontrolador, fue útil para comprender el funcionamiento del módulo ADC. Si bien con el código implementado se obtienen valores razonables de temperatura, es necesario verificar el funcionamiento ya que el fabricante aclara que se puede tener un offset significativo.

En lo que respecta al ejemplo de la conversión de una secuencia de datos, se puede afirmar que los ejemplos suministrados fueron la base sobre la cual se implementó el código para la medida de temperatura de las caras y la placa. Para verificar el funcionamiento y el algoritmo de cálculo de temperatura se conectó las entradas del ADC a un divisor resistivo, verificando que se medía el voltaje aplicado.

6.5. Pruebas RTC

El código de ejemplo suministrado por el fabricante fue usado para entender lo referente a la configuración de los registros. El código fue probado con éxito en el programa principal.

Cuadro 6.3: Ejemplos de código del fabricante

Ejemplos de Código de Texas Instruments - RTC	Descripción
RTC_Sample.c	Asignación de valores a los registros involucrados en el modo calendario

Capítulo 7

Trabajo a futuro

El software y hardware desarrollado durante el proyecto cumple en general con los requerimientos básicos de un sistema de control principal. Sin embargo, la solución planteada se considera todavía en etapa de desarrollo y lejos está de ser una versión definitiva.

Antes de concluir, se desean dejar varios ejemplos e ideas que puedan aportar a futuros estudios que continúen este trabajo, agregándole valor al producto final.

Algunas de las mejoras a incluir en el trabajo futuro se listan a continuación según el módulo al que pertenecen:

- **WDT**

1. Ajustar los resets del watchdog según los tiempos de ejecución de tareas, una vez que se definan todas las tareas a ejecutarse.

- **RTC**

1. Calibración del cristal oscilador para RTC.
2. Rutina que genere un evento de error cuando el drift es mayor a cierto valor.

- **ADC**

1. Rutina que compare las temperaturas de la placa y los dos microcontroladores embebidos y que tome acciones al respecto.
2. Rutina que genere un evento de error cuando la temperatura salga de determinados rangos.

■ Scheduler

1. Redireccionamiento del flujo principal en caso de fallar alguno de los módulos.
2. Definir con más detalle los períodos de ejecución de cada tarea.
3. Definir acciones a tomar en caso de repetidas fallas de encolamiento de tarea.

■ Housekeeping

1. Hacer un lista definitiva de todas aquellos datos que deseen relevarse por módulo.
2. Definir cuales son los datos importantes que se guardan en el resumen.
3. Mantener un registro más extenso de los últimos resúmenes de housekeeping (hoy en día sólo guarda los últimos 17 registros).
4. Generar alarmas si algún valor no se encuentra dentro de los rangos esperados.

■ Comunicación I2C

1. Optimizar tiempos de espera antes de una condición de Stop.
2. Respuesta frente a fallas repetidas en el envío o la recepción de datos.
3. Aislar los dispositivos del bus I2C que se encuentren apagados. De lo contrario el resto de los dispositivos no podrá realizar transferencias

- **Comunicación con Tierra**

1. Rutina que interprete comandos que llegan desde Tierra y tome las acciones pertinentes.
2. Testing exhaustivo del módulo de envío y recepción de datos desde y hacia Tierra.
3. Agregar complejidad y seguridad al protocolo de intercambio de datos.

Como último punto, se desea aclarar que el diseño actual carece de un sistema formal de gestión de contingencias que garantice la continuidad y la recuperación total frente a fallas del satélite.

Algunos aspectos fueron considerados en el diseño, pero otros tantos deberán implementarse en futuras versiones si se pretende que el módulo forme parte del sistema embebido del primer satélite experimental que Uruguay envíe al espacio.

Capítulo 8

Conclusiones

El principal resultado a destacar de este trabajo es la formalización de lo que sería un módulo de control principal de un nanosatélite. Este formalismo se valida mediante la construcción de un sistema autónomo, que se comunica con los demás módulos del satélite, que realiza un conjunto de tareas específicas, y que es capaz de comunicarse con Tierra.

Muchas de las funcionalidades, que en un principio se pensaban integrar al sistema, debieron ser eliminadas por restricciones de tiempo, pero sobre todo por ciertas incertidumbres al momento de definir las tareas que el satélite iba a cumplir. Este punto resultó ser la complicación más grande encontrada durante todo el proceso.

Sin embargo, se obtuvo un producto altamente adaptable y capaz de adecuarse a posibles cambios en los requerimientos del sistema, lo que satisface en gran medida los objetivos propuestos.

Bibliografía

- [1] Proyecto Lai. Disponible en: iee.fing.edu.uy/investigacion/grupos/lai/
- [2] The CubeSat Program, CubeSat Design Specifications Rev.12, California Polytechnic State University, 2009. Disponible en: www.cubesat.org/images/developers/cds_rev12.pdf
- [3] Lista de CubeSats lanzados. Disponible en: en.wikipedia.org/wiki/List_of_CubeSats
- [4] Texas Instruments, MSP430f5438A. Disponible en: www.ti.com
- [5] W. A. Beech, D. E. Nielsen, J. Taylor, *AX.25 Link Access Protocol for Amateur Packet Radio*, 1998. Disponible en: www.tapr.org
- [6] AX.25 Requirements www.ax25.net
- [7] D. J. Wheeler, R. M. Needham. *TEA, a Tiny Encryption Algorithm*. England: Cambridge University. Disponible en: <http://www.cix.co.uk>
- [8] G. Gutiérrez, F. Lanzari, *SisComSat: Sistema de comunicación para nanosatélite*. Montevideo, Universidad de la República, 2012
- [9] National Semiconductor, LM20. Disponible en: www.national.com/ds/LM/LM20.pdf
- [10] S. Zito, A. Castro, V. Alonsopérez, *Proyecto Lai Sistema de Determinación de Actitud*. Montevideo, Universidad de la República, 2011

- [11] Microcrystal, MS1V-T1K. Disponible en: www.microcrystal.com/CMSPages/GetFile.aspx?nodeguid=ad6a56e5-81d2-4720-93ff-e8d2cb178c48
- [12] I2C2P Official Homepage. Disponible en: i2c2p.twibright.com
- [13] Texas Instruments, *MSP430F543xA, MSP430F541xA MIXED SIGNAL MICROCONTROLLER*. Disponible en: www.ti.com/lit/ds/slas655b/slas655b.pdf
- [14] I. de León, J. Ramos, G. Sotta, *Desarrollo del sistema de gestión de energía para un satélite, P-SAT*. Montevideo, Universidad de la República, 2011
- [15] Descripción JTAG. Diponible en: en.wikipedia.org/wiki/Joint_Test_Action_Group
- [16] *MSP430x5xx / MSP430x6xx Family User's Guide*, SLAU208J, 2011. Disponible en: www.ti.com/lit/ug/slau278j/slau278j.pdf
- [17] AAU CubeSat. Aalborg University. Disponible en: www.space.aau.dk/cubesat/
- [18] B. S. Gregory, *Attitude Control System design for ION, The Illinois observing nanosatellite*. Marquette University, 2001
- [19] Swiss Cube <http://ctsgepc7.epfl.ch/>
- [20] Vishay, BPW. Disponible en: <http://www.vishay.com/docs/81521/bpw34.pdf>
- [21] Sparkfun Electronics, MicroMag 3-Axis Magnetometer, PNI. Disponible en: <http://www.sparkfun.com/datasheets/Sensors/MicroMag3%20Data%20Sheet.pdf>
- [22] Vishay, Si9987. Disponible en: <http://www.vishay.com/docs/70864/si9987.pdf>
- [23] H. Péter-Contesse, M. Borgeaud, M. Noca, *ADCS System Engineer - ADCS System and Hardware*. EPFL Lausanne Switzerland, 2007. Disponible en: http://ctsgepc7.epfl.ch/06%20-%20Attitude%20control/S3-B-C-ADCS-1-3-ADCS_HW_and_System.pdf

- [24] CadSoft, Software para diseño de PCB EAGLE. Disponible en: www.cadsoftusa.com/
- [25] IAR Embedded Workbench IDE online help. Disponible en: www.iar.com/Products/IAR-Embedded-Workbench/
- [26] Texas Instruments, Fet Debugger y placa MSP-TS430PZ5X-100. Disponible en: www.tij.co.jp/tool/jp/msp-ts430pz5x100
- [27] Texas Instruments, Ejemplos de código. Disponible en: www.tij.co.jp/tool/jp/msp-ts430pz5x1009

Glosario

ADC Conversor Analógico-Digital

ADCD Módulo de control y detección de actitud

AX.25 Amateur Packet Radio Link Layer Protocol

Beacon Baliza

Cal Poly California Polytechnic State University

COM Módulo de Comunicación

FIFO First in, first out

I2C Bus de comunicación serie (Inter-Integrated Circuit)

IAR Entorno de desarrollo

ISR Rutina de atención a interrupción

LPM Modo de bajo consumo

MCP Módulo de control principal

PAY Payload, carga útil o científica

P-POD Poly Picosatellite Orbital Deployer

PWM Pulse width modulation

RCT Real Time Clock

RF Radiofrecue

SCL I2C Clock Line

SDA I2C Data Line

SPI Interfaz de protocolo serial

TEA Tiny Encryption Algorithm

TLE Two Line Element

USCI Universal Serial Communication Interface

UTC Tiempo Universal Coordinado

Anexos

Anexo A

Módulos que componen el satélite

A.1. P-SAT: Desarrollo del sistema de gestión de energía para un satélite

El Sistema de Gestión de Energía o SGE [14] tiene como función principal la administración de la energía solar recolectada por medio de celdas fotovoltaicas. Es quien se encarga de energizar a los demás módulos, controlar las fallas de consumo que puedan ocurrir, administrar el correcto funcionamiento de las baterías y proteger los diferentes subsistemas de posibles errores causados por fallas eléctricas.

Dentro de las tareas que tiene el SGE, también se encuentra la de energizar al resto del satélite y desplegar sus antenas un tiempo determinado luego del lanzamiento del mismo a su órbita. Ésto es con el fin de evitar posibles interferencias eléctricas o de RF con el vehículo de lanzamiento.

Con respecto al control de posibles fallas eléctricas que tiene en cuenta el SGE, los tipos considerados son los siguientes:

- Sobrecarga
- Cortocircuito
- Latch UP

El módulo preve un sistema ágil de respuesta frente a fallas de este tipo para que no se destruya la electrónica del satélite o se drene toda su energía. La primer medida que toma es cortando por hardware el suministro de energá al periférico que presente la falla para luego restaurársela. En el caso en que al restaurar la energía se vuelva a detectar una gran corriente por alguno de los puertos de alimentación, el sistema corta de forma permanente el suministro energético a ese periférico.

El SGE cuenta con una unidad de procesamiento de datos y de toma de decisiones implementada en un PIC16F887 desarrollado por la compañía MICROCHIP. Para la toma de decisiones, el módulo sensa y mantiene un registro de todas aquellas magnitudes que puedan ser de interés para mantener el correcto funcionamiento del satélite. Dentro de estas magnitudes se encuentran mediciones de las corrientes y tensiones consumidas por cada módulo, el rango de temperaturas en que se encuentren funcionando las baterías, y corrientes y tensiones de los paneles fotovoltaicos.

Los valores mencionados en el párrafo anterior, son los que MCP va a relevar y almacenar mediante un pedido de housekeeping al SGE a través del bus I2C.

A.2. Proyecto Laí Sistema de Determinación de Actitud

Este trabajo implementa y prueba un subsistema que determina la actitud de un CubeSat [10].

El movimiento de un satélite en el espacio queda completamente descrito mediante su posición, velocidad, actitud y velocidad angular. La posición y velocidad indican el movimiento de traslación del cuerpo, mientras que la actitud y velocidad angular indican el movimiento de rotación respecto a su centro de masa. La determinación de actitud es el proceso de cálculo de la orientación del satélite respecto a un sistema de referencia inercial o un objeto específico, como pueden ser la Tierra o el Sol. Algunos vectores de referencia comúnmente utilizados son el vector de campo magnético terrestre y vectores en dirección al Sol. El

proceso involucra distintos tipos de sensores junto con un sistema de procesamiento de datos.

Para poder trabajar de forma óptima, el módulo necesita de un vector TLE, que representa la diferencia entre una posición específica de un objeto y su posición esperada y el tiempo UTC. El sistema no recibe un TLE hasta que no se comunica por primera vez con la estación terrena. Por lo tanto, durante este tiempo el módulo funciona de acuerdo a valores estimados e incluso no puede ejecutar algunas de sus funciones.

Una vez funcionando en régimen, para determinar la posición del satélite en un instante de tiempo dado, el sistema debe conocer el tiempo transcurrido desde la recepción del último TLE. Por este motivo, además del tiempo UTC, el módulo lleva un registro actualizado RTC que mediante ciertos cálculos permite obtener la diferencia entre ambas medidas.

El módulo desarrollado por este proyecto deberá ser complementado por un módulo de control de actitud con el fin de obtener un sistema completo de detección y control de la posición del satélite en el espacio. Una de las tareas periódicas que se lleva a cabo en la rutina principal del MCP es controlar la actitud del satélite mediante un pedido de detección de actitud al ADCS. En caso de ser necesaria alguna corrección, el MCP le pide al ADCS que se encargue de ajustar la actitud.

Al momento de finalizar este trabajo se encontraba desarrollado únicamente el módulo de detección y es por este motivo que sólo se describe su funcionamiento en esta sección.

A.3. SisComSat: Sistema de comunicación para nanosatélite

Diseña e implementa un prototipo de módulo de telemetría y telemando para CubeSats, para enviar y recibir datos entre el satélite y la Tierra. [8]

El trabajo hace una descripción detallada de los procesos de diseño, frecuencias y modos de operación, cálculos de enlaces de comunicación, y la especificación del hardware diseñado.

Al momento de finalizar este trabajo se encontraba desarrollado únicamente el módulo de radio, faltando complementar con un módulo que contenga toda la lógica de procesamiento de datos.

Una vez finalizado, el módulo del COM contará con un sistema transmisor, uno receptor, y un modem. A través de éste módulo el MCP intercambia instrucciones, comandos y datos de estado con Tierra para su posterior procesamiento.

El COM debe asegurar los siguientes requerimientos:

- Establecer un canal de comunicación sin fallas.
- Permitir pausar y retomar la comunicación dependiendo si se está, o no, dentro de la huella de comunicación.
- Priorizar entre distintos tipos de información.
- Transmitir una baliza.
- Trabajar de manera compatible con la comunidad amateur de radio.

Anexo B

Configuración ADC

El conversor analógico digital empleado es el ADC12_A que viene integrado al MSP5430F5438A. A continuación se hace referencia a los registros y variables necesarios para su correcta configuración.

Cuadro B.1: Configuración ADC

Valor	Descripción
int lectura[6]	Almacenamiento de las conversiones de los ADC correspondiente a la temperatura de cada cara del satélite
int tempC[6]	Almacenamiento de las temperaturas de las caras expresadas en grados Celsius
long temp	Almacenamiento de la conversión del ADC correspondiente a temperatura del microcontrolador
int tempDegC	Almacenamiento de la temperatura del microcontrolador expresada en grados Celsius
ADC12MCTLx	Los bits ADC12MCTL1 a ADC12MCTL6 se utilizan para guardar los resultados de la medición de temperaturas de las caras. Luego se pasan a lectura en la rutina de interrupción del ADC
ADC12CTL	Registros de control del ADCS
PUERTO_6_ADC	Se configuran los pines del puerto 6 (PSEL6) para la entrada del ADC
ADC12IE	Bit que habilita o deshabilita los pedidos de interrupción del ADC
ADC12ENC	Bit que Bit que habilita o deshabilita la conversión ADC
ADC12SC	Comienza el muestreado y la conversión ADC
ADC12TCOFF	Bit que apaga el sensor de temperatura
REFCTL0	Registro de control de referencia
REFON	Habilita la referencia
ADC12SHP	Habilita o deshabilita el <i>sample timer</i>
ADC12SREF	3 bits para seleccionar el nivel de la referencia. En este caso está entre $VR+ = V_{REF+}$ y $VR- = AVSS$
ADC12INCH	Selecciona el canal de entrada, en este caso la temperatura del diodo
ADCCAL	Calibración del ADC
ADC12_VECTOR	Vector de interrupción del ADC
ADC12IV	Generador del vector de interrupción del ADC. Se utiliza para determinar qué fuente genera la interrupción
ADC12IFGx	Registro que se setea cuando su correspondiente ADC12MEMx se carga
ADC12MEMx	Lugar de memoria donde se registran los cambios

Anexo C

Configuración RTC

La siguiente tabla menciona los registros y variables utilizados para la configuración del RTC del microcontrolador. Se recuerda que como fuente de reloj se utiliza un cristal externo al microcontrolador de 32768 KHz.

Cuadro C.1: Configuración RTC

Valor	Descripción
time_s time_stamp	Estampa de tiempo que indica la fecha en el formato AA-M-D-H-M-S
time_s time_drift	Diferencia entre tiempo actualizado por Tierra y el del RTC
int time_actual[]	Variable donde se almacena la hora actualizada desde Tierra
RTCCTLxx	Registro de control del RTC. En este caso se setea con el valor RTCTEVIE + RTCMODE + RTCTEV_0
RTCTEVIE	Interrupciones habilitadas o deshabilitadas para eventos de tiempo
RTCMODE	Modo del RTC. 0 para 32-bit counter mode
RTCTEV	Eventos de tiempo del RTC
RTCRDYIE	Habilita o deshabilita las interrupciones del RTC
RTCYEARH	Registro en donde se carga la parte alta del año
RTCYEARL	Registro en donde se carga la parte baja del año
RTCMON	Registro en donde se carga el mes
RTCDAY	Registro en donde se carga el día
RTCHOUR	Registro en donde se carga la hora
RTCMIN	Registro en donde se carga el minuto
RTCSEC	Registro en donde se carga el segundo
RTCHOLD	Es 0 cuando el RTC está operacional
RTC_VECTOR	Vector de interrupción del RTC

Anexo D

Configuración I2C

El módulo USCI (Universal Serial Communication Interface) es utilizado para diferentes tipos de comunicación serial, entre los que encontramos UART, SPI e I2C. La configuración se realiza invocando a la función `i2c_antelsat_config()`. A continuación se presenta una lista de los registros usados para la configuración del módulo I2C del MSP430F5438A.

Cuadro D.1: Registros de configuración I2C

Registro	Nombre	Descripción
P3SEL	—	Se asignan los pines al módulo USCIB0
UCBOCTL0	Registro de control 0	Se selecciona modo I2C, modo síncrono, maestro, multi-maestro
UCBOCTL1	Registro de control 1	Se controla el software reset, y se selecciona la fuente de reloj
UCBOBR0	Registro de control de ratio de baudios 0	$UCxxBR0 + UCxxBR1 \times 256$
UCBOBR1	Registro de control de ratio de baudios 0	$UCxxBR0 + UCxxBR1 \times 256$
UCBOIE	Registro de habilitación de interrupciones	Transmisión, recepción, start, stop, y nack
UCBOIFG	Registro de banderas de interrupciones	Transmisión, recepción, start, stop, y nack
UCBORXBUF	Buffer de recepción	Recepción del byte enviado
UCBOTXBUF	Buffer de transmisión	Se copia el byte a transmitir
UCBOI2COA	Registro de dirección propia	Dirección I2C del dispositivo
UCBOI2CSA	Registro de dirección del esclavo	Dirección I2C del dispositivo esclavo

A continuación se muestra los cuatro casos de comunicación, con los

registros involucrados.

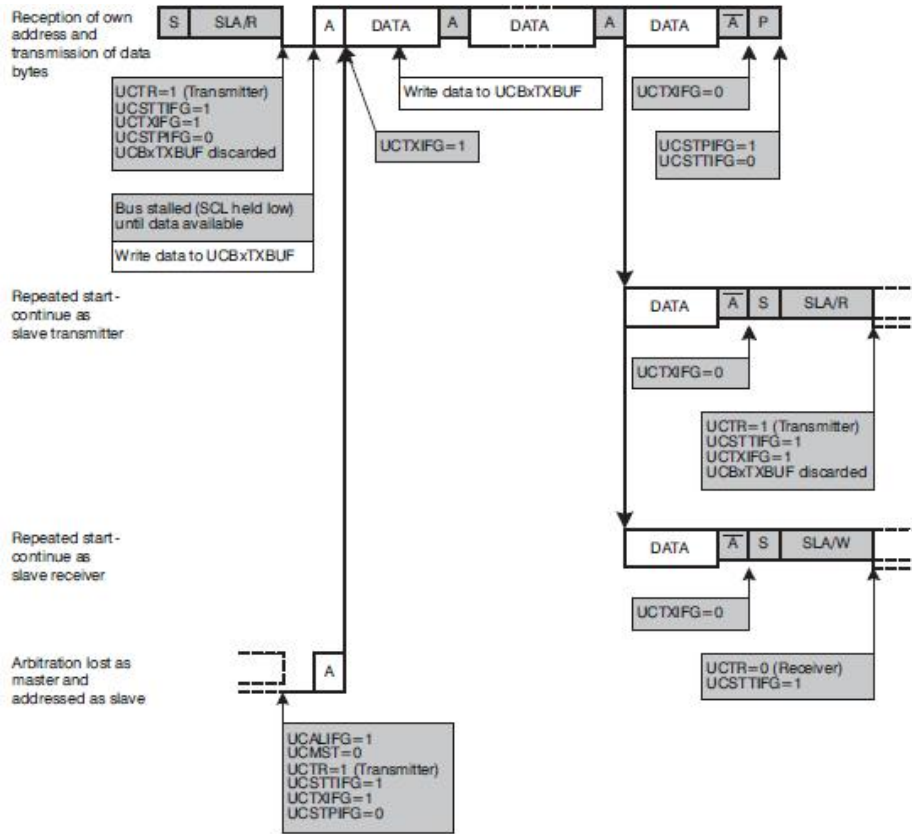


Figura D.1: Modo esclavo transmisor

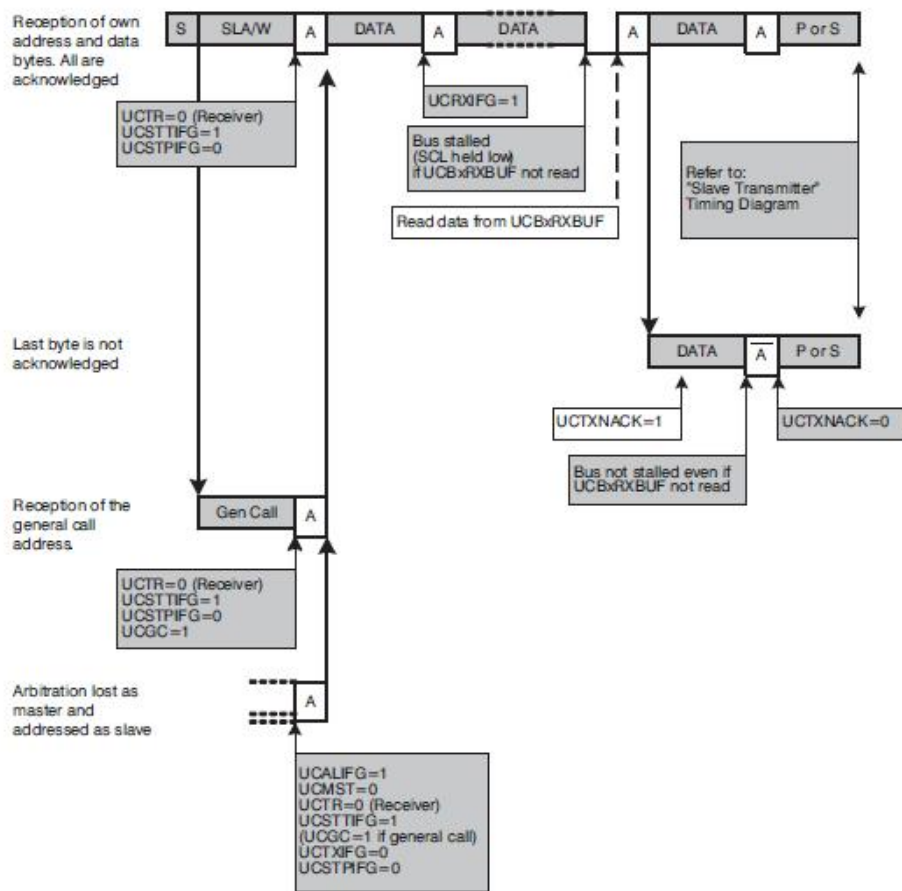


Figura D.2: Modo esclavo receptor

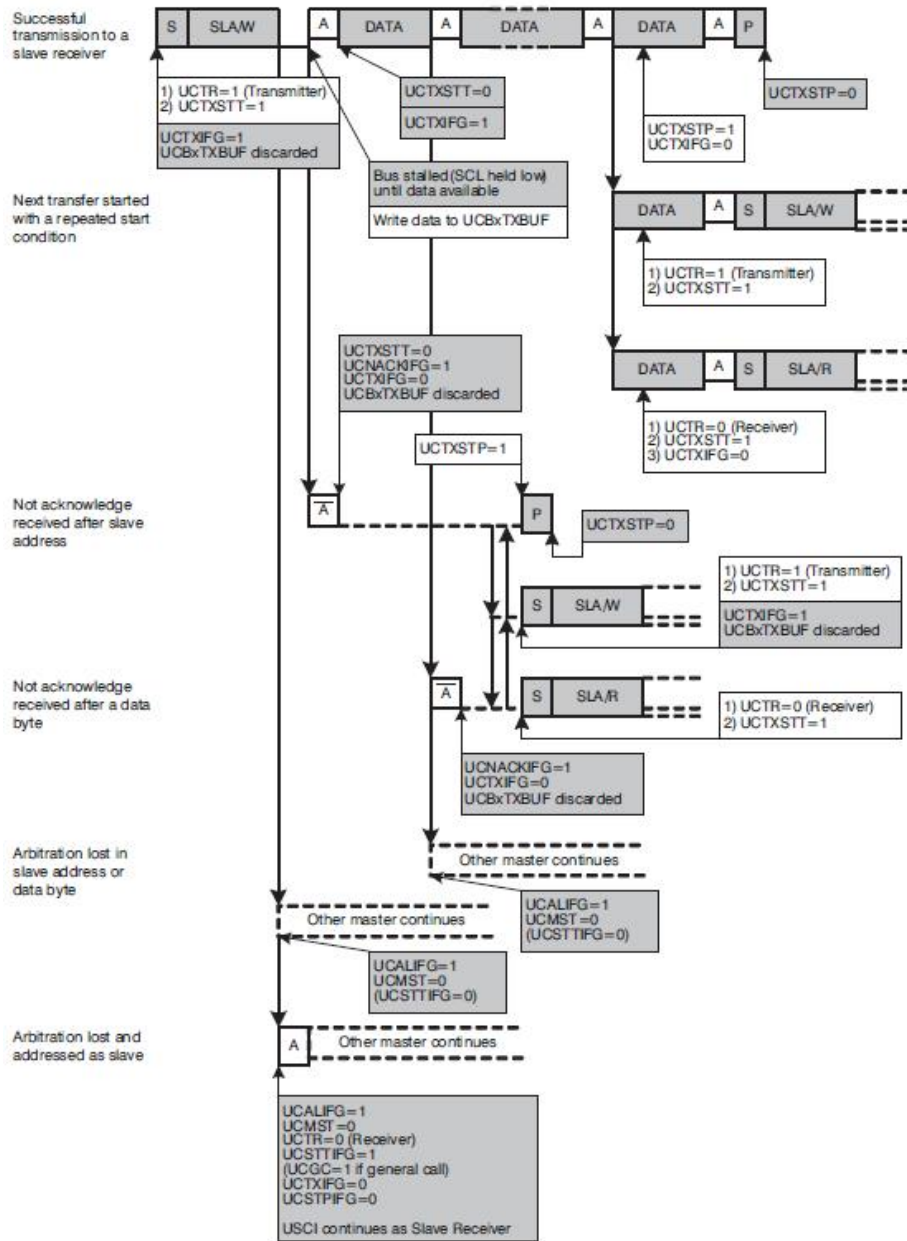


Figura D.3: Modo maestro transmisor

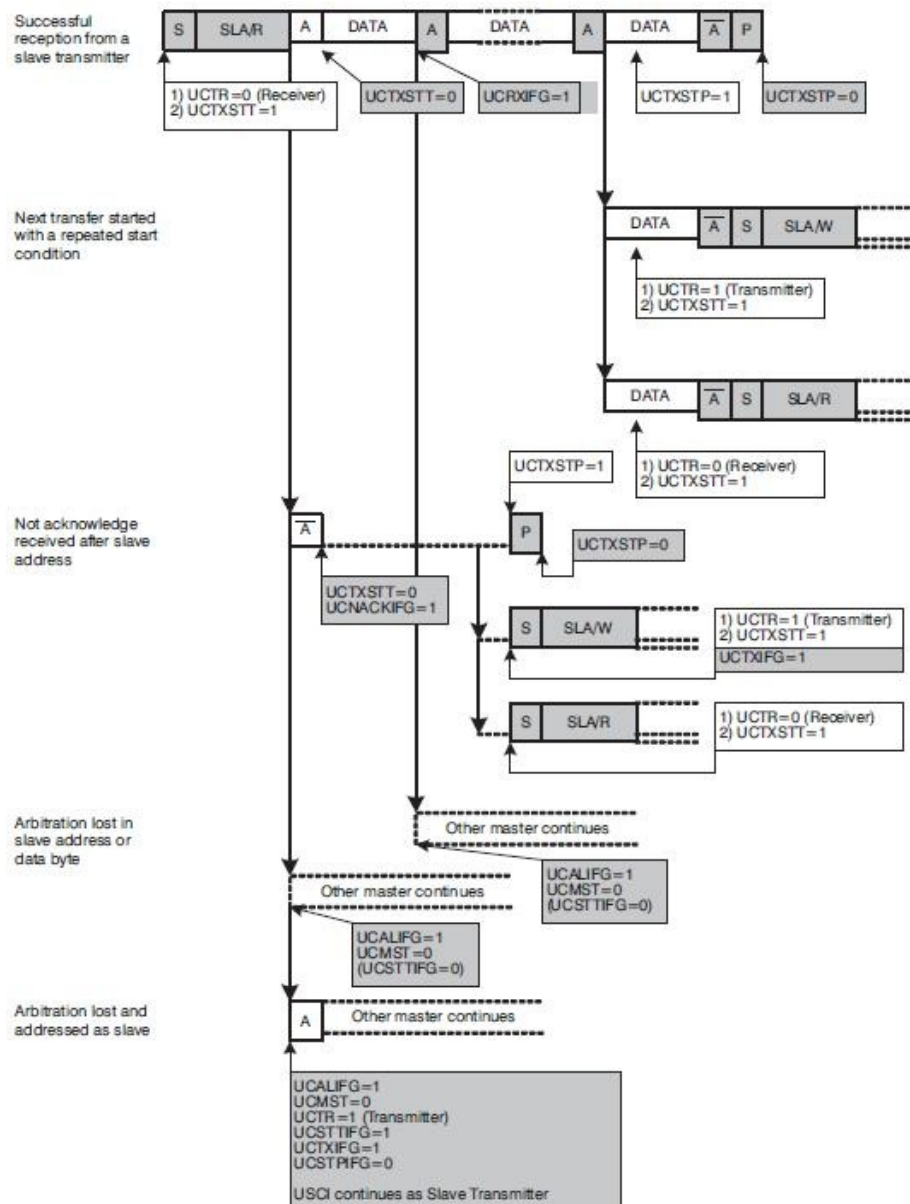


Figura D.4: Modo maestro receptor

En caso de que dos maestros intenten hacer uso del bus en forma simultánea el procedimiento de arbitraje es invocado. El maestro que ponga el byte de menor valor binario en el bus es que el gana el arbitraje. En la figura ?? se puede observar un ejemplo de arbitraje.

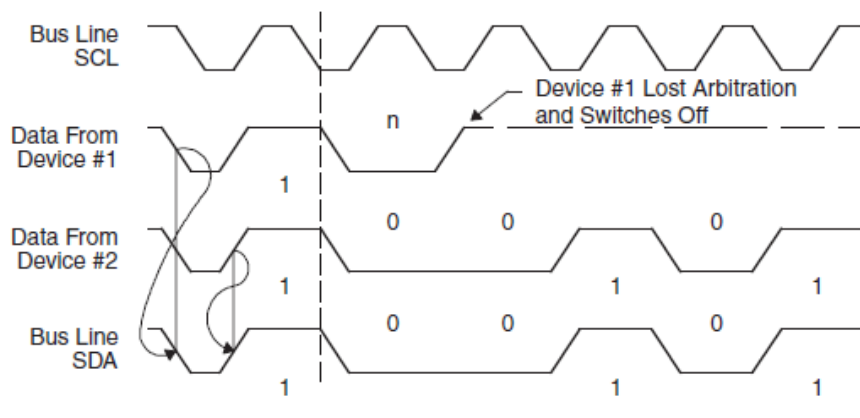


Figura D.5: Modo maestro receptor

Anexo E

Análisis de costos

En esta sección se hace un análisis del tiempo y costo que insumió la realización del proyecto.

Para el estudio, se realiza un conteo de las horas invertidas y de los costos de los componentes utilizados. Se agregan las comparaciones entre datos estimados y datos reales con el objetivo de obtener una noción de cuánto se alejó el proyecto de lo esperado.

Costos

La siguiente tabla hace una comparación entre los costos estimados y los costos reales del proyecto.

Cuadro E.1: Cálculo de costos en materiales

	Ítem	Real (U\$S)	Estimada (U\$S)
Costos fijos	Kits de desarrollo	298	570
	Licencias SW	340	
Costos variables	Placa	90	90
	Componentes	60	80
	Importación	-	80
	Imprevistos (5% total)	N/C	15
Total		788	835

Tiempos

El tiempo inicial estimado para el desarrollo del proyecto era de 12 meses, desde abril de 2011 hasta abril de 2012. Finalmente, en marzo de 2012, se solicita prorrogar la entrega final y el proyecto termina concluyendo 14 meses luego de su comienzo, desde abril de 2012 a junio de 2012.

La solicitud de la prórroga se fundamentó en diversas causas que resultaron en un atraso crítico con respecto a la planificación.

En primer lugar, como uno de los principales motivos que llevaron a una diferencia tan grande con el cronograma, se tiene que dos de los tres integrantes del grupo aumentaron significativamente su horario laboral, resultando directamente en una reducción importante en las horas hombre dedicadas al estudio del proyecto. Esto ocurrió inmediatamente después de la fecha del primer hito, y generó un gran desbalance entre

las horas dedicadas a la primera y segunda mitad del proyecto.

En segundo lugar, se considera que no se procedió de la mejor manera al momento de tomar decisiones sobre factores que aún no tenían una especificación clara y completa. En lugar de pasar a la acción, en más de una ocasión se tomó una postura pasiva, demorando así tareas que podrían haberse resuelto en menor tiempo.

El propio avance del trabajo, sumado al ingreso de Antel como colaborador en plena ejecución del proyecto, y detalles que no habían sido definidos de antemano, llevaron al replanteamiento de ciertos aspectos del alcance que no estaban previstos por el grupo, el cual tuvo serios problemas al resolver esas incertidumbres. Como resultado, gran parte de los tiempos de buffer considerados para la amortiguación de impactos fue consumido.

Este punto se menciona en las conclusiones como la mayor dificultad encontrada durante el desarrollo del proyecto. El tercer motivo que se expone es el gran componente de desarrollo de software que resultó teniendo el proyecto, y la falta de experiencia de los integrantes del equipo en esta área. Si bien se tuvo en cuenta al momento de la planificación, el tiempo que llevó adaptarse a esta tarea fue sustancialmente mayor y no permitió alcanzar objetivos según el cronograma. Ésto resultó en un atraso en las siguientes etapas del proyecto que, sumado a los motivos anteriores, terminó por hacer diverger por completo los tiempos reales con la planificación inicial.

Todo lo expuesto anteriormente, sumado a una falta de experiencia del equipo en proyectos de este porte llevaron a una mala planificación y por ende una mala administración de los tiempos. La primera acción que se tomó fue replantear el alcance y eliminar la tarea *Control de Actitud* de los objetivos iniciales. La segunda fue definir con más claridad las entregas intermedias con el tutor, de manera de otorgarle tiempos más reales a cada una de las tareas restantes. La tercera es extender el tiempo de finalización del proyecto para poder cumplir con los compromisos asumidos. Y por último, el aumento de la dedicación horaria prevista.

La siguiente tabla muestra los valores de la dedicación horaria real y estimada.

Cuadro E.2: Cálculo de horas invertidas

Fase del proyecto	Real(hs)	Estimado(hs)
Estudio	1029	936
Desarrollo	1287	1170
Documentación	257	234
Total	2573	2340

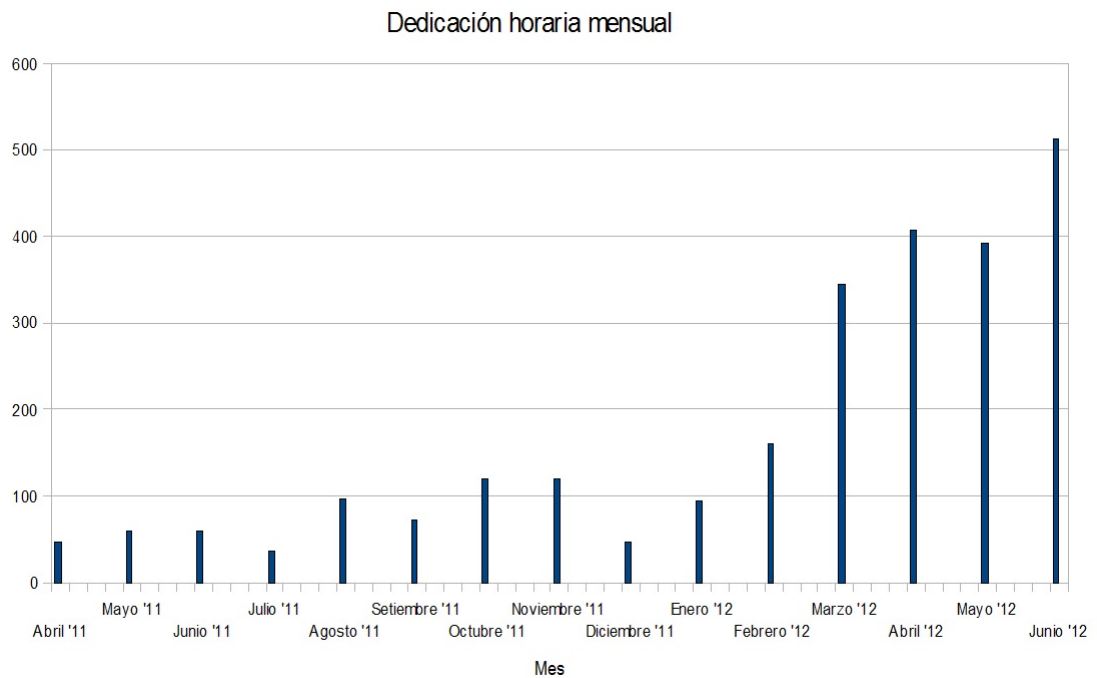


Figura E.1: Carga horaria mensual durante el proyecto

Anexo F

Hojas de datos

LM20

2.4V, 10µA, SC70, micro SMD Temperature Sensor

General Description

The LM20 is a precision analog output CMOS integrated-circuit temperature sensor that operates over a -55°C to $+130^{\circ}\text{C}$ temperature range. The power supply operating range is $+2.4\text{ V}$ to $+5.5\text{ V}$. The transfer function of LM20 is predominately linear, yet has a slight predictable parabolic curvature. The accuracy of the LM20 when specified to a parabolic transfer function is $\pm 1.5^{\circ}\text{C}$ at an ambient temperature of $+30^{\circ}\text{C}$. The temperature error increases linearly and reaches a maximum of $\pm 2.5^{\circ}\text{C}$ at the temperature range extremes. The temperature range is affected by the power supply voltage. At a power supply voltage of 2.7 V to 5.5 V the temperature range extremes are $+130^{\circ}\text{C}$ and -55°C . Decreasing the power supply voltage to 2.4 V changes the negative extreme to -30°C , while the positive remains at $+130^{\circ}\text{C}$.

The LM20's quiescent current is less than $10\text{ }\mu\text{A}$. Therefore, self-heating is less than 0.02°C in still air. Shutdown capability for the LM20 is intrinsic because its inherent low power consumption allows it to be powered directly from the output of many logic gates or does not necessitate shutdown at all.

Applications

- Cellular Phones
- Computers
- Power Supply Modules
- Battery Management

- FAX Machines
- Printers
- HVAC
- Disk Drives
- Appliances

Features

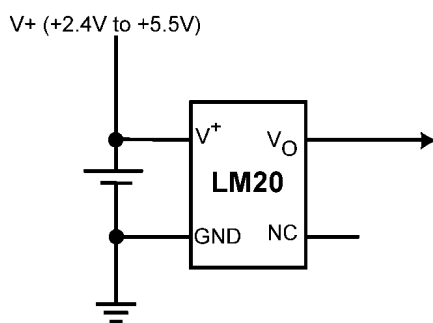
- Rated for full -55°C to $+130^{\circ}\text{C}$ range
- Available in an SC70 and 2 micro SMD packages
- Predictable curvature error
- Suitable for remote applications

Key Specifications

■ Accuracy at $+30^{\circ}\text{C}$	± 1.5 to $\pm 4^{\circ}\text{C}$ (max)
■ Accuracy at $+130^{\circ}\text{C}$ & -55°C	± 2.5 to $\pm 5^{\circ}\text{C}$ (max)
■ Power Supply Voltage Range	$+2.4\text{V}$ to $+5.5\text{V}$
■ Current Drain	$10\text{ }\mu\text{A}$ (max)
■ Nonlinearity	$\pm 0.4\%$ (typ)
■ Output Impedance	$160\text{ }\Omega$ (max)
■ Load Regulation	-2.5 mV (max)
	$0\text{ }\mu\text{A} < I_L < +16\text{ }\mu\text{A}$

Typical Application

Full-Range Celsius (Centigrade) Temperature Sensor (-55°C to $+130^{\circ}\text{C}$)
Operating from a Single Li-Ion Battery Cell

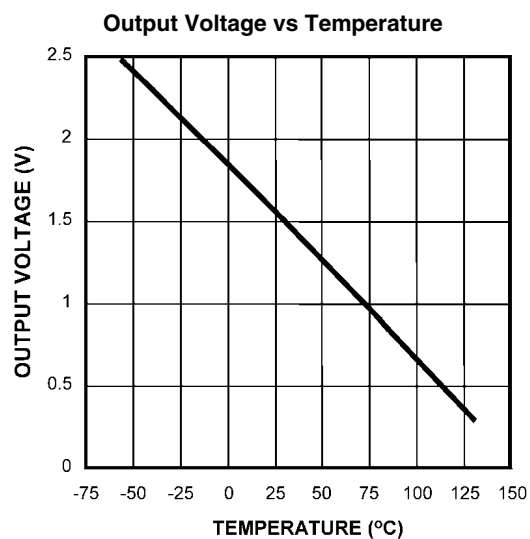


$$V_O = (-3.88 \times 10^{-6} \times T^2) + (-1.15 \times 10^{-2} \times T) + 1.8639 \quad 10090802$$

$$T = -1481.96 + \sqrt{2.1962 \times 10^6 + \frac{(1.8639 - V_O)}{3.88 \times 10^{-6}}}$$

where:

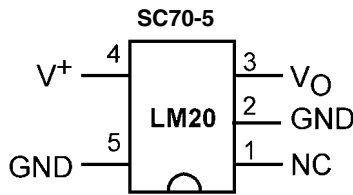
T is temperature, and V_O is the measured output voltage of the LM20.



10090824

Temperature (T)	Typical V _O
+130°C	+303 mV
+100°C	+675 mV
+80°C	+919 mV
+30°C	+1515 mV
+25°C	+1574 mV
0°C	+1863.9 mV
-30°C	+2205 mV
-40°C	+2318 mV
-55°C	+2485 mV

Connection Diagrams

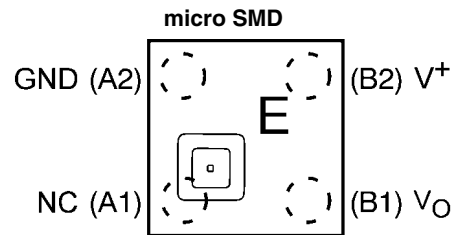


10090801

Note:

- GND (pin 2) may be grounded or left floating. For optimum thermal conductivity to the pc board ground plane pin 2 should be grounded.
- NC (pin 1) should be left floating or grounded. Other signal traces should not be connected to this pin.

Top View
See NS Package Number MAA05A



10090832

Note:

- Pin numbers are referenced to the package marking text orientation.
- Reference JEDEC Registration MO-211, variation BA
- The actual physical placement of package marking will vary slightly from part to part. The package marking will designate the date code and will vary considerably. Package marking does not correlate to device type in any way.

Top View
See NS Package Number BPA04DDC, TPA04EEA and TLA04ZZA

Ordering Information

Order Number	Temperature Accuracy	Temperature Range	NS Package Number	Device Marking	Transport Media
LM20BIM7	±2.5°C	-55°C to +130°C	MAA05A	T2B	1000 Units on Tape and Reel
LM20BIM7X	±2.5°C	-55°C to +130°C	MAA05A	T2B	3000 Units on Tape and Reel
LM20CIM7	±5°C	-55°C to +130°C	MAA05A	T2C	1000 Units on Tape and Reel
LM20CIM7X	±5°C	-55°C to +130°C	MAA05A	T2C	3000 Units on Tape and Reel
LM20SIBP	±3.5°C	-40°C to +125°C	BPA04DDC	Date Code	250 Units on Tape and Reel
LM20SIBPX	±3.5°C	-40°C to +125°C	BPA04DDC	Date Code	3000 Units on Tape and Reel
LM20SITL	±3.5°C	-40°C to +125°C	TLA04ZZA	Date Code	250 Units on Tape and Reel
LM20SITLX	±3.5°C	-40°C to +125°C	TLA04ZZA	Date Code	3000 Units on Tape and Reel
LM20SITP	±3.5°C	-40°C to +125°C	TPA04EEA	Date Code	250 Units on Tape and Reel
LM20SITPX	±3.5°C	-40°C to +125°C	TPA04EEA	Date Code	3000 Units on Tape and Reel

Absolute Maximum Ratings (Note 1)

Supply Voltage	+6.5V to -0.2V
Output Voltage	(V+ + 0.6 V) to -0.6 V
Output Current	10 mA
Input Current at any pin (Note 2)	5 mA
Storage Temperature	-65°C to +150°C
Maximum Junction Temperature (T _{JMAX})	+150°C
ESD Susceptibility (Note 3) :	
Human Body Model	2500 V
Machine Model	250 V

Soldering process must comply with National's Reflow Temperature Profile specifications. Refer to www.national.com/packaging. (Note 4)

Operating Ratings (Note 1)

Specified Temperature Range:	$T_{MIN} \leq T_A \leq T_{MAX}$
LM20B, LM20C with 2.4 V ≤ V+ ≤ 2.7 V	-30°C ≤ T _A ≤ +130°C
LM20B, LM20C with 2.7 V ≤ V+ ≤ 5.5 V	-55°C ≤ T _A ≤ +130°C
LM20S with 2.4 V ≤ V+ ≤ 5.5 V	-30°C ≤ T _A ≤ +125°C
LM20S with 2.7 V ≤ V+ ≤ 5.5 V	-40°C ≤ T _A ≤ +125°C
Supply Voltage Range (V+)	+2.4 V to +5.5 V
Thermal Resistance, θ _{JA} (Note 5)	
SC-70	415°C/W
micro SMD	340°C/W

Electrical Characteristics

Unless otherwise noted, these specifications apply for V+ = +2.7 V_{DC}. **Boldface limits apply for T_A = T_J = T_{MIN} to T_{MAX}**; all other limits T_A = T_J = 25°C; Unless otherwise noted.

Parameter	Conditions	Typical (Note 6)	LM20B	LM20C	LM20S	Units (Limit)
			Limits (Note 7)	Limits (Note 7)	Limits (Note 7)	
Temperature to Voltage Error V _O = (-3.88×10 ⁻⁶ ×T ²) + (-1.15×10 ⁻² ×T) + 1.8639V (Note 8)	T _A = +25°C to +30°C		±1.5	±4.0	±2.5	°C (max)
	T _A = +130°C		±2.5	±5.0		°C (max)
	T _A = +125°C		±2.5	±5.0	±3.5	°C (max)
	T _A = +100°C		±2.2	±4.7	±3.2	°C (max)
	T _A = +85°C		±2.1	±4.6	±3.1	°C (max)
	T _A = +80°C		±2.0	±4.5	±3.0	°C (max)
	T _A = 0°C		±1.9	±4.4	±2.9	°C (max)
	T _A = -30°C		±2.2	±4.7	±3.3	°C (min)
	T _A = -40°C		±2.3	±4.8	±3.5	°C (max)
T _A = -55°C		±2.5	±5.0		°C (max)	
Output Voltage at 0°C		+1.8639				V
Variance from Curve		±1.0				°C
Non-Linearity (Note 9)	-20°C ≤ T _A ≤ +80°C	±0.4				%
Sensor Gain (Temperature Sensitivity or Average Slope) to equation: V _O = -11.77 mV/°C × T + 1.860V	-30°C ≤ T _A ≤ +100°C	-11.77	-11.4 -12.2	-11.0 -12.6	-11.0 -12.6	mV/°C (min) mV/°C (max)
Output Impedance	0 μA ≤ I _L ≤ +16 μA (Note 11, Note 12)		160	160	160	Ω (max)
Load Regulation (Note 10)	0 μA ≤ I _L ≤ +16 μA (Note 11, Note 12)		-2.5	-2.5	-2.5	mV (max)
Line Regulation	+2.4 V ≤ V+ ≤ +5.0V		+3.3	+3.7	+3.7	mV/V (max)
	+5.0 V ≤ V+ ≤ +5.5 V		+11	+11	+11	mV (max)
Quiescent Current	+2.4 V ≤ V+ ≤ +5.0V	4.5	7	7	7	μA (max)
	+5.0 V ≤ V+ ≤ +5.5 V	4.5	9	9	9	μA (max)
	+2.4 V ≤ V+ ≤ +5.0V	4.5	10	10	10	μA (max)
Change of Quiescent Current	+2.4 V ≤ V+ ≤ +5.5 V	+0.7				μA

Parameter	Conditions	Typical (Note 6)	LM20B	LM20C	LM20S	Units (Limit)
			Limits (Note 7)	Limits (Note 7)	Limits (Note 7)	
Temperature Coefficient of Quiescent Current		-11				nA/°C
Shutdown Current	$V_+ \leq +0.8\text{ V}$	0.02				μA

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is functional, but do not guarantee specific performance limits. For guaranteed specifications and test conditions, see the Electrical Characteristics. The guaranteed specifications apply only for the test conditions listed. Some performance characteristics may degrade when the device is not operated under the listed test conditions.

Note 2: When the input voltage (V_i) at any pin exceeds power supplies ($V_i < \text{GND}$ or $V_i > V_+$), the current at that pin should be limited to 5 mA.

Note 3: The human body model is a 100 pF capacitor discharged through a 1.5 k Ω resistor into each pin. The machine model is a 200 pF capacitor discharged directly into each pin.

Note 4: Reflow temperature profiles are different for lead-free and non-lead-free packages.

Note 5: The junction to ambient thermal resistance (θ_{JA}) is specified without a heat sink in still air using the printed circuit board layout shown in [Figure 1](#).

Note 6: Typical values are at $T_J = T_A = 25^\circ\text{C}$ and represent most likely parametric norm.

Note 7: Limits are guaranteed to National's AOQL (Average Outgoing Quality Level).

Note 8: Accuracy is defined as the error between the measured and calculated output voltage at the specified conditions of voltage, current, and temperature (expressed in $^\circ\text{C}$).

Note 9: Non-Linearity is defined as the deviation of the calculated output-voltage-versus-temperature curve from the best-fit straight line, over the temperature range specified.

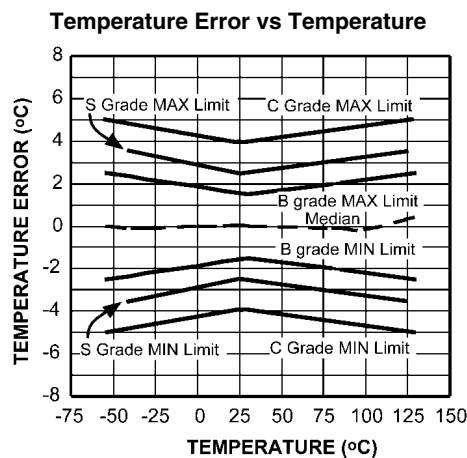
Note 10: Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the internal dissipation by the thermal resistance.

Note 11: Negative currents are flowing into the LM20. Positive currents are flowing out of the LM20. Using this convention the LM20 can at most sink $-1\ \mu\text{A}$ and source $+16\ \mu\text{A}$.

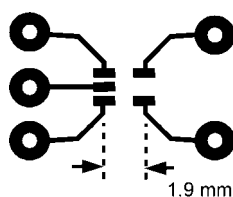
Note 12: Load regulation or output impedance specifications apply over the supply voltage range of $+2.4\text{V}$ to $+5.5\text{V}$.

Note 13: Line regulation is calculated by subtracting the output voltage at the highest supply input voltage from the output voltage at the lowest supply input voltage.

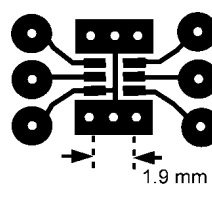
Typical Performance Characteristics



PCB Layouts Used for Thermal Measurements



a) Layout used for no heat sink measurements.



b) Layout used for measurements with small heat hink.

FIGURE 1. PCB Lyouts used for thermal measurements.

1.0 LM20 Transfer Function

The LM20's transfer function can be described in different ways with varying levels of precision. A simple linear transfer function, with good accuracy near 25°C, is

$$V_O = -11.69 \text{ mV/}^\circ\text{C} \times T + 1.8663 \text{ V}$$

Over the full operating temperature range of -55°C to +130°C, best accuracy can be obtained by using the parabolic transfer function

$$V_O = (-3.88 \times 10^{-6} \times T^2) + (-1.15 \times 10^{-2} \times T) + 1.8639$$

solving for T:

$$T = -1481.96 + \sqrt{2.1962 \times 10^6 + \frac{(1.8639 - V_O)}{3.88 \times 10^{-6}}}$$

A linear transfer function can be used over a limited temperature range by calculating a slope and offset that give best results over that range. A linear transfer function can be calculated from the parabolic transfer function of the LM20. The slope of the linear transfer function can be calculated using the following equation:

$$m = -7.76 \times 10^{-6} \times T - 0.0115,$$

where T is the middle of the temperature range of interest and m is in V/°C. For example for the temperature range of $T_{\min} = -30$ to $T_{\max} = +100^\circ\text{C}$:

$$T = 35^\circ\text{C}$$

and

$$m = -11.77 \text{ mV/}^\circ\text{C}$$

The offset of the linear transfer function can be calculated using the following equation:

$$b = (V_{OP}(T_{\max}) + V_{OP}(T) - m \times (T_{\max} + T))/2$$

where:

- $V_{OP}(T_{\max})$ is the calculated output voltage at T_{\max} using the parabolic transfer function for V_O
- $V_{OP}(T)$ is the calculated output voltage at T using the parabolic transfer function for V_O .

Using this procedure the best fit linear transfer function for many popular temperature ranges was calculated in [Figure 2](#). As shown in [Figure 2](#) the error that is introduced by the linear transfer function increases with wider temperature ranges.

Temperature Range		Linear Equation $V_O =$	Maximum Deviation of Linear Equation from Parabolic Equation (°C)
T_{\min} (°C)	T_{\max} (°C)		
-55	+130	$-11.79 \text{ mV/}^\circ\text{C} \times T + 1.8528 \text{ V}$	± 1.41
-40	+110	$-11.77 \text{ mV/}^\circ\text{C} \times T + 1.8577 \text{ V}$	± 0.93
-30	+100	$-11.77 \text{ mV/}^\circ\text{C} \times T + 1.8605 \text{ V}$	± 0.70
-40	+85	$-11.67 \text{ mV/}^\circ\text{C} \times T + 1.8583 \text{ V}$	± 0.65
-10	+65	$-11.71 \text{ mV/}^\circ\text{C} \times T + 1.8641 \text{ V}$	± 0.23
+35	+45	$-11.81 \text{ mV/}^\circ\text{C} \times T + 1.8701 \text{ V}$	± 0.004
+20	+30	$-11.69 \text{ mV/}^\circ\text{C} \times T + 1.8663 \text{ V}$	± 0.004

FIGURE 2. First order equations optimized for different temperature ranges.

2.0 Mounting

The LM20 can be applied easily in the same way as other integrated-circuit temperature sensors. It can be glued or cemented to a surface. The temperature that the LM20 is sensing will be within about +0.02°C of the surface temperature to which the LM20's leads are attached to.

This presumes that the ambient air temperature is almost the same as the surface temperature; if the air temperature were much higher or lower than the surface temperature, the actual temperature measured would be at an intermediate temperature between the surface temperature and the air temperature.

To ensure good thermal conductivity the backside of the LM20 die is directly attached to the pin 2 GND pin. The temperatures of the lands and traces to the other leads of the LM20 will also affect the temperature that is being sensed.

Alternatively, the LM20 can be mounted inside a sealed-end metal tube, and can then be dipped into a bath or screwed into a threaded hole in a tank. As with any IC, the LM20 and accompanying wiring and circuits must be kept insulated and dry, to avoid leakage and corrosion. This is especially true if the circuit may operate at cold temperatures where condensation can occur. Printed-circuit coatings and varnishes such as Humiseal and epoxy paints or dips are often used to ensure that moisture cannot corrode the LM20 or its connections.

The thermal resistance junction to ambient (θ_{JA}) is the parameter used to calculate the rise of a device junction temperature due to its power dissipation. For the LM20 the equation used to calculate the rise in the die temperature is as follows:

$$T_J = T_A + \theta_{JA} [(V^+ I_Q) + (V^+ - V_O) I_L]$$

where I_Q is the quiescent current and I_L is the load current on the output. Since the LM20's junction temperature is the actual temperature being measured care should be taken to minimize the load current that the LM20 is required to drive.

The tables shown in Figure 3 summarize the rise in die temperature of the LM20 without any loading, and the thermal resistance for different conditions.

	SC70-5 no heat sink		SC70-5 small heat sink	
	θ_{JA} (°C/W)	$T_J - T_A$ (°C)	θ_{JA} (°C/W)	$T_J - T_A$ (°C)
Still air	412	0.2	350	0.19
Moving air	312	0.17	266	0.15

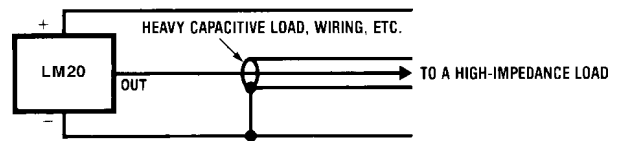
See Figure 1 for PCB layout samples.

	micro SMD no heat sink		micro SMD small heat fin	
	θ_{JA} (°C/W)	$T_J - T_A$ (°C)	θ_{JA} (°C/W)	$T_J - T_A$ (°C)
Still air	340	0.18	TBD	TBD
Moving air	TBD	TBD	TBD	TBD

FIGURE 3. Temperature Rise of LM20 Due to Self-Heating and Thermal Resistance (θ_{JA})

3.0 Capacitive Loads

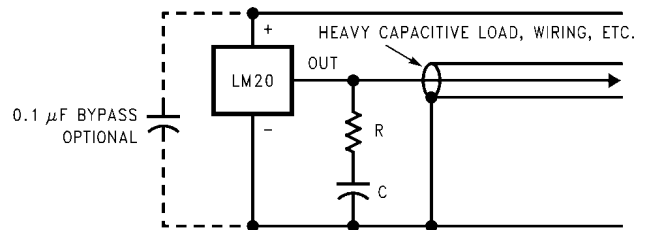
The LM20 handles capacitive loading well. Without any precautions, the LM20 can drive any capacitive load less than 300 pF as shown in Figure 4. Over the specified temperature range the LM20 has a maximum output impedance of 160 Ω. In an extremely noisy environment it may be necessary to add some filtering to minimize noise pickup. It is recommended that 0.1 μF be added from V^+ to GND to bypass the power supply voltage, as shown in Figure 5. In a noisy environment it may even be necessary to add a capacitor from the output to ground with a series resistor as shown in Figure 5. A 1 μF output capacitor with the 160 Ω maximum output impedance and a 200 Ω series resistor will form a 442 Hz lowpass filter. Since the thermal time constant of the LM20 is much slower, the overall response time of the LM20 will not be significantly affected.



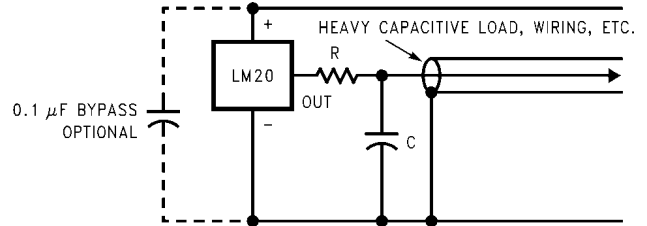
10090815

FIGURE 4. LM20 No Decoupling Required for Capacitive Loads Less than 300 pF.

R (Ω)	C (μF)
200	1
470	0.1
680	0.01
1 k	0.001



10090816



10090833

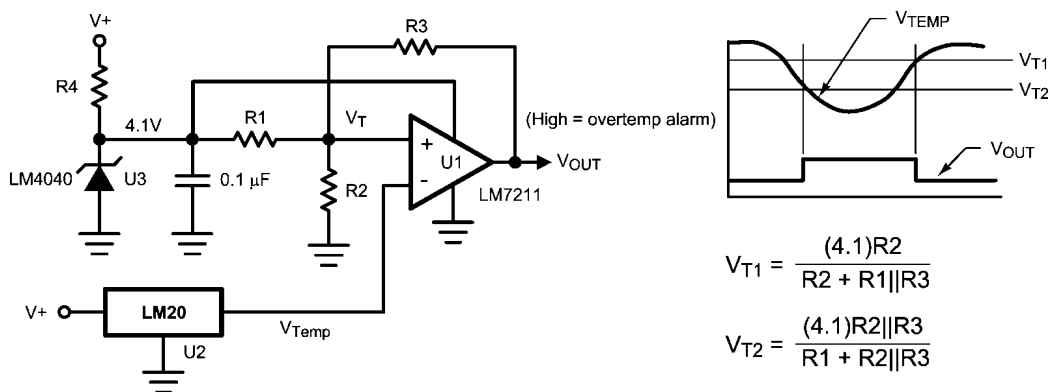
FIGURE 5. LM20 with Filter for Noisy Environment and Capacitive Loading greater than 300 pF. Either placement of resistor as shown above is just as effective.

4.0 LM20 micro SMD Light Sensitivity

Exposing the LM20 micro SMD package to bright sunlight may cause the output reading of the LM20 to drop by 1.5V. In a normal office environment of fluorescent lighting the output voltage is minimally affected (less than a millivolt drop). In either case it is recommended that the LM20 micro SMD be

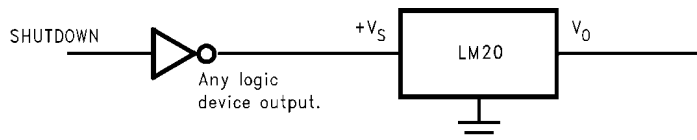
placed inside an enclosure of some type that minimizes its light exposure. Most chassis provide more than ample protection. The LM20 does not sustain permanent damage from light exposure. Removing the light source will cause LM20's output voltage to recover to the proper value.

5.0 Applications Circuits



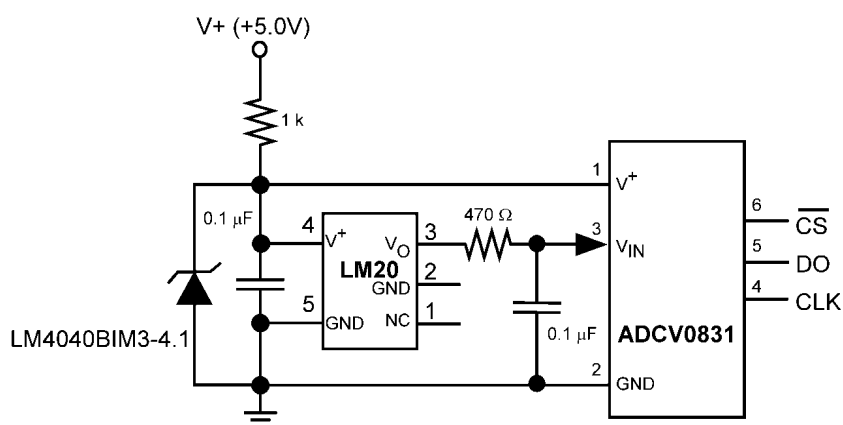
10090818

FIGURE 6. Centigrade Thermostat



10090819

FIGURE 7. Conserving Power Dissipation with Shutdown

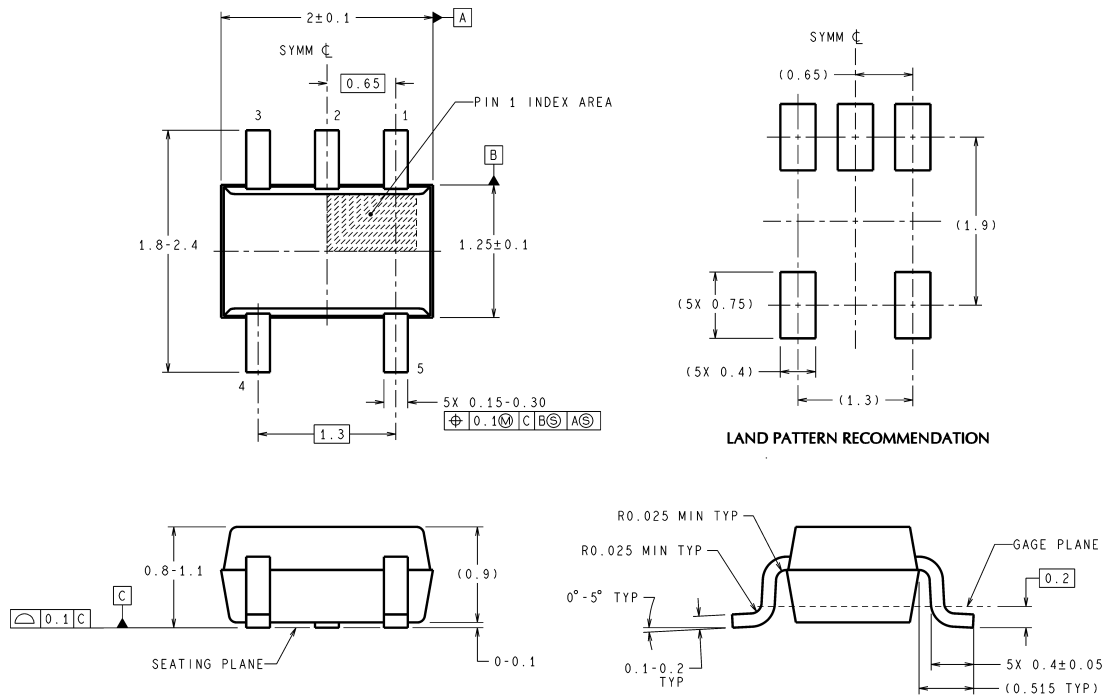


10090828

Most CMOS ADCs found in ASICs have a sampled data comparator input structure that is notorious for causing grief to analog output devices such as the LM20 and many op amps. The cause of this grief is the requirement of instantaneous charge of the input sampling capacitor in the ADC. This requirement is easily accommodated by the addition of a capacitor. Since not all ADCs have identical input stages, the charge requirements will vary necessitating a different value of compensating capacitor. This ADC is shown as an example only. If a digital output temperature is required please refer to devices such as the LM74.

FIGURE 8. Suggested Connection to a Sampling Analog to Digital Converter Input Stage

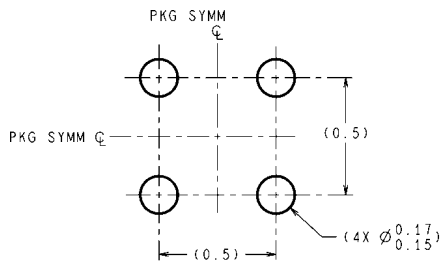
Physical Dimensions inches (millimeters) unless otherwise noted



DIMENSIONS ARE IN MILLIMETERS
DIMENSIONS IN () FOR REFERENCE ONLY

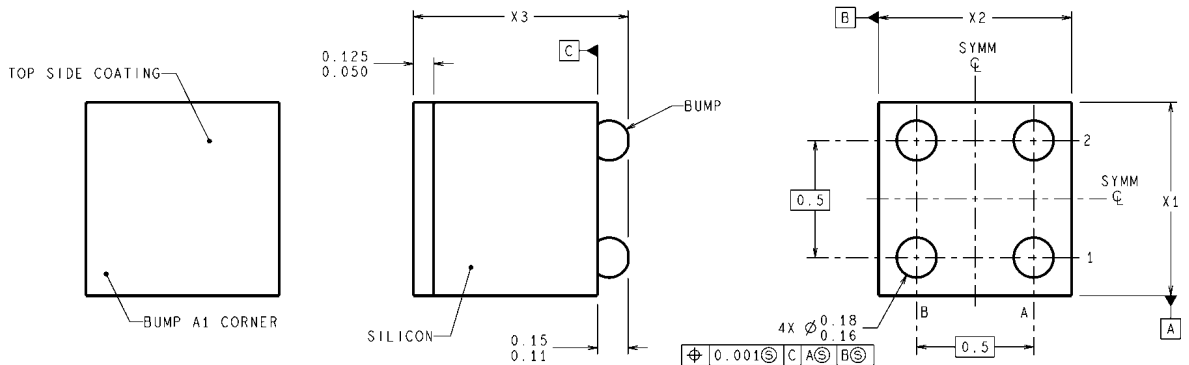
MAA05A (Rev D)

5-Lead SC70 Molded Package
Order Number LM20BIM7 or LM20CIM7X
NS Package Number MAA05A



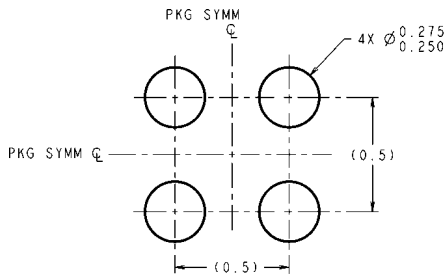
DIMENSIONS ARE IN MILLIMETERS
DIMENSIONS IN () FOR REFERENCE ONLY

LAND PATTERN RECOMMENDATION



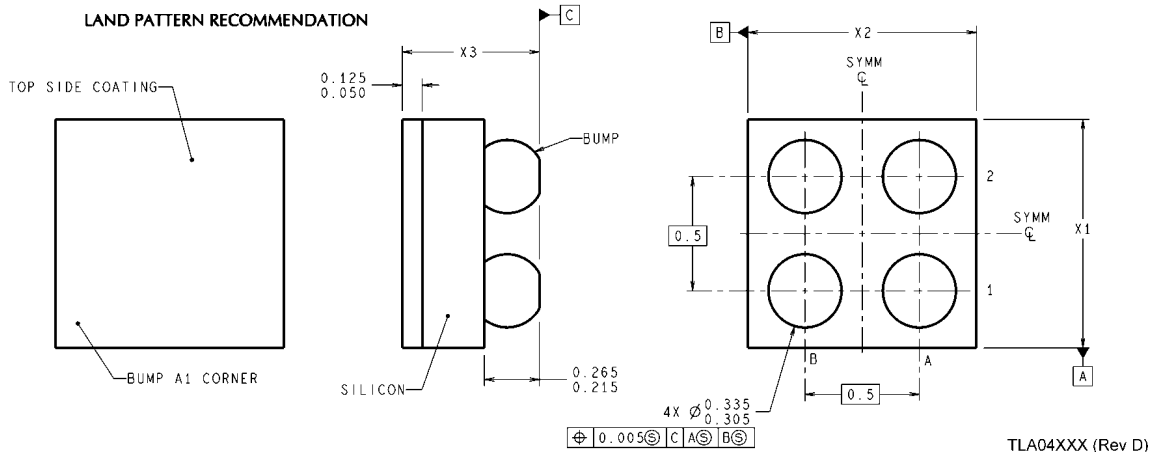
BPA04XXX (Rev E)

4-Bump micro SMD Ball Grid Array Package (Small Bump)
Order Number LM20SIBP or LM20SIBPX
NS Package Number BPA04DDC
The following dimensions apply to the BPA04DDC package
shown above: X1=X2 = 853µm ±30µm, X3= 900µm ±50µm



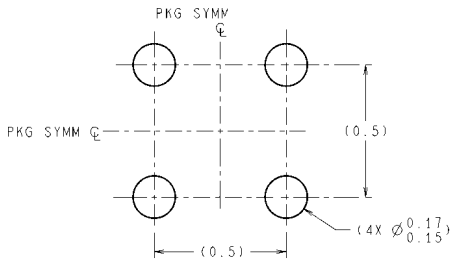
DIMENSIONS ARE IN MILLIMETERS
DIMENSIONS IN () FOR REFERENCE ONLY

LAND PATTERN RECOMMENDATION



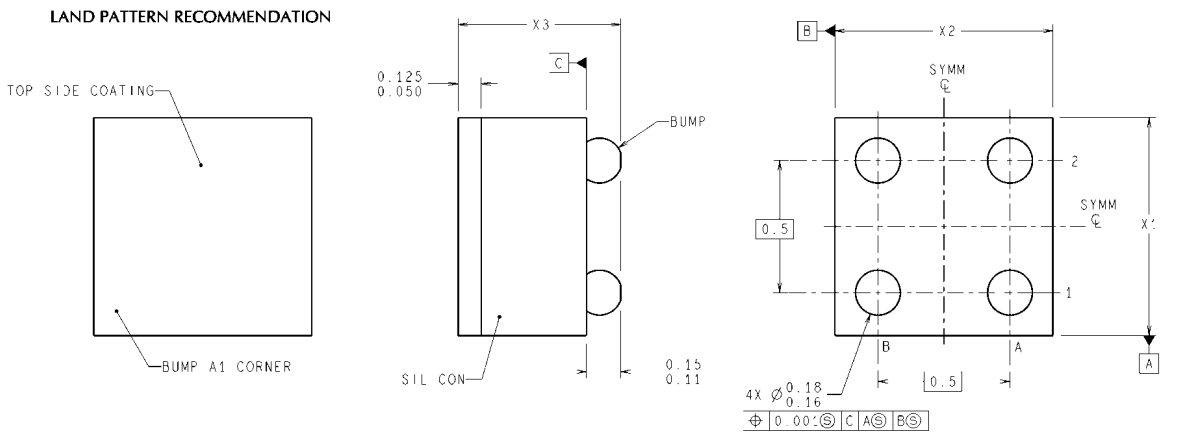
4-Bump micro SMD Ball Grid Array Package (Large Bump)
Order Number LM20SITL or LM20SITLX
NS Package Number TLA04ZZA

The following dimensions apply to the TLA04ZZA package shown above: $X1 = X2 = 963\mu\text{m} \pm 30\mu\text{m}$, $X3 = 600\mu\text{m} \pm 75\mu\text{m}$



DIMENSIONS ARE IN MILLIMETERS

LAND PATTERN RECOMMENDATION



4-Bump Thin micro SMD Ball Grid Array Package (Small Bump)
Order Number LM20SITP or LM20SITPX
NS Package Number TPA04EEA

The following dimensions apply to the TPA04EEA package shown above: $X1 = X2 = 879\mu\text{m} \pm 30\mu\text{m}$, $X3 = 500\mu\text{m} \pm 75\mu\text{m}$

Notes

For more National Semiconductor product information and proven design tools, visit the following Web sites at:
www.national.com

Products		Design Support	
Amplifiers	www.national.com/amplifiers	WEBENCH® Tools	www.national.com/webench
Audio	www.national.com/audio	App Notes	www.national.com/appnotes
Clock and Timing	www.national.com/timing	Reference Designs	www.national.com/refdesigns
Data Converters	www.national.com/adc	Samples	www.national.com/samples
Interface	www.national.com/interface	Eval Boards	www.national.com/evalboards
LVDS	www.national.com/lvds	Packaging	www.national.com/packaging
Power Management	www.national.com/power	Green Compliance	www.national.com/quality/green
Switching Regulators	www.national.com/switchers	Distributors	www.national.com/contacts
LDOs	www.national.com/ldo	Quality and Reliability	www.national.com/quality
LED Lighting	www.national.com/led	Feedback/Support	www.national.com/feedback
Voltage References	www.national.com/vref	Design Made Easy	www.national.com/easy
PowerWise® Solutions	www.national.com/powerwise	Applications & Markets	www.national.com/solutions
Serial Digital Interface (SDI)	www.national.com/sdi	Mil/Aero	www.national.com/milaero
Temperature Sensors	www.national.com/tempsensors	SolarMagic™	www.national.com/solarmagic
PLL/VCO	www.national.com/wireless	PowerWise® Design University	www.national.com/training

THE CONTENTS OF THIS DOCUMENT ARE PROVIDED IN CONNECTION WITH NATIONAL SEMICONDUCTOR CORPORATION ("NATIONAL") PRODUCTS. NATIONAL MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS PUBLICATION AND RESERVES THE RIGHT TO MAKE CHANGES TO SPECIFICATIONS AND PRODUCT DESCRIPTIONS AT ANY TIME WITHOUT NOTICE. NO LICENSE, WHETHER EXPRESS, IMPLIED, ARISING BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT.

TESTING AND OTHER QUALITY CONTROLS ARE USED TO THE EXTENT NATIONAL DEEMS NECESSARY TO SUPPORT NATIONAL'S PRODUCT WARRANTY. EXCEPT WHERE MANDATED BY GOVERNMENT REQUIREMENTS, TESTING OF ALL PARAMETERS OF EACH PRODUCT IS NOT NECESSARILY PERFORMED. NATIONAL ASSUMES NO LIABILITY FOR APPLICATIONS ASSISTANCE OR BUYER PRODUCT DESIGN. BUYERS ARE RESPONSIBLE FOR THEIR PRODUCTS AND APPLICATIONS USING NATIONAL COMPONENTS. PRIOR TO USING OR DISTRIBUTING ANY PRODUCTS THAT INCLUDE NATIONAL COMPONENTS, BUYERS SHOULD PROVIDE ADEQUATE DESIGN, TESTING AND OPERATING SAFEGUARDS.

EXCEPT AS PROVIDED IN NATIONAL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, NATIONAL ASSUMES NO LIABILITY WHATSOEVER, AND NATIONAL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO THE SALE AND/OR USE OF NATIONAL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE CHIEF EXECUTIVE OFFICER AND GENERAL COUNSEL OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

National Semiconductor and the National Semiconductor logo are registered trademarks of National Semiconductor Corporation. All other brand or product names may be trademarks or registered trademarks of their respective holders.

Copyright© 2010 National Semiconductor Corporation

For the most current product information visit us at www.national.com



**National Semiconductor
Americas Technical
Support Center**
Email: support@nsc.com
Tel: 1-800-272-9959

**National Semiconductor Europe
Technical Support Center**
Email: europe.support@nsc.com

**National Semiconductor Asia
Pacific Technical Support Center**
Email: ap.support@nsc.com

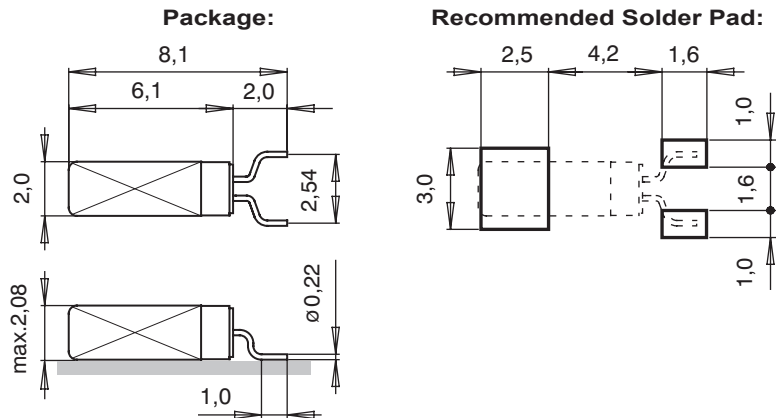
**National Semiconductor Japan
Technical Support Center**
Email: jpn.feedback@nsc.com

MS1V-T1K

Tuning Fork Crystal 30 kHz – 200 kHz



DIMENSIONS



All dimensions in mm typical

Small SMT package
Package height 2.08 mm max.
Formed leads for SMD mounting
Industrial temp. range
Very price competitive
High stability
Low aging
Low power consumption
High shock and vibration resistance

DESCRIPTION:

The MS1V-T1K is a square-bodied 2.0 x 8.1 x 2.0 mm metal-can package with formed leads, intended for surface mounting and reflow soldering. Within the package is a high-quality tuning fork quartz crystal resonator, manufactured with a precision high-volume photolithographic process and operating in the fundamental mode.

Due to highly automated, large-volume production, the MS1V-T1K is very price competitive and ideal for any consumer and industrial applications.

For pick-and-place equipment, the crystals are supplied in 16 mm tape:
 7" (178 mm) reel with 2'000 crystals
 13" (330 mm) reel with 8'000 crystals

ELECTRICAL CHARACTERISTICS AT 25°C:

Frequency Range	F_L	30–200	kHz
Load capacitance ¹⁾	C_L	7.0 / 9.0 / 12.5	pF
Frequency tolerance ²⁾	$\Delta F/F$	+/-20	ppm
	$\Delta F/F$	+/-100	ppm
Drive level max.	P	1.0	μW
Insulation resistance min.	R_i	500	M Ω
Aging first year max.	$\Delta F/F_0$	+/-3	ppm

Frequency Example ³⁾	F_L	32.000	32.768	76.800	kHz
Series resistance typ./max.	R_s	45 / 60	45 / 60	18 / 30	k Ω
Motional capacitance typ.	C_1	2.3	2.1	1.35	fF
Static capacitance typ.	C_0	0.9	0.9	0.9	pF
Turnover temperature	T_0	25 +/-5	25 +/-5	22 +/-5	$^{\circ}C$
Frequency vs. temperature	$\Delta F/F_0$	$-0.035 \text{ ppm}/^{\circ}C^2 (T - T_0)^2 \pm 10\%$			ppm

1) For standard and other load capacitances consult factory.

2) Tighter and wider frequency tolerances on request.

3) Values for other frequencies consult factory.

STANDARD FREQUENCIES:

Frequency kHz				
32.000	32.768	38.400	40.000	60.000
75.000	76.800	77.500		
Other frequencies on request				

ENVIRONMENTAL CHARACTERISTICS:

		Conditions	Max. Dev.
Storage temperature range		-55 to +85°C	
TA Operating temperature range		-40 to +85°C	
Shock resistance	ΔF/F	5000 g, 0.3 ms, ½ sine	+/-5 ppm
Vibration resistance	ΔF/F	20 g / 10–2000 Hz	+/-5 ppm

TERMINATIONS AND PROCESSING:

Type	Termination	Processing
MS1V-T1N	Can: Au flashed Leads: Straight leads Au flashed	For trough-hole mounting 260°C / 20 sec. max
MS1V-T1K	Can: Au flashed Leads: Formed leads Au flashed	For SMD mounting Reflow soldering 260°C / 20 sec. max

PRODUCT DESCRIPTION AND ORDERING INFORMATION:

MS1V-T1K 32.768kHz 12.5pF +/-20ppm TA QC

Frequency f_0
 C_L Load Capacitance
 7.0 pF
 9.0 pF
 12.5 pF
Frequency tolerance
 +/- 20 ppm
 +/- 100 ppm

Temperature Range
 TA = -40 to 85°C (Standard)

Optional:
 if not defined,
 'Standard' will apply

Qualification
 QC = Commercial (Standard)

A unique part number will be generated for each product specification, i.e:
 20xxxx-ML02 2'000 pcs (in 16mm tape on 7" reel)
 20xxxx-PL08 8'000 pcs (in 16mm tape on 13" reel)
 Please contact us.

All specifications subject to change without notice.

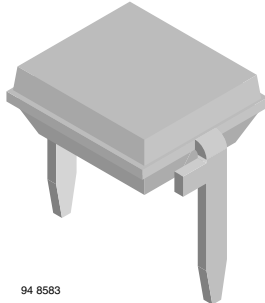


Micro Crystal AG
 Mühlestrasse 14
 CH-2540 Grenchen
 Switzerland

Tel. +41 32 655 82 82
 Fax +41 32 655 82 83
 sales@microcrystal.com
 www.microcrystal.com



Silicon PIN Photodiode



94 8583

FEATURES

- Package type: leaded
- Package form: top view
- Dimensions (L x W x H in mm): 5.4 x 4.3 x 3.2
- Radiant sensitive area (in mm²): 7.5
- High photo sensitivity
- High radiant sensitivity
- Suitable for visible and near infrared radiation
- Fast response times
- Angle of half sensitivity: $\phi = \pm 65^\circ$
- Compliant to RoHS Directive 2002/95/EC and in accordance to WEEE 2002/96/EC



DESCRIPTION

BPW34 is a PIN photodiode with high speed and high radiant sensitivity in miniature, flat, top view, clear plastic package. It is sensitive to visible and near infrared radiation. BPW34S is packed in tubes, specifications like BPW34.

Note

** Please see document "Vishay Material Category Policy": www.vishay.com/doc?99902

APPLICATIONS

- High speed photo detector

PRODUCT SUMMARY			
COMPONENT	I_{ra} (μA)	ϕ (deg)	$\lambda_{0.1}$ (nm)
BPW34	50	± 65	430 to 1100
BPW34S	50	± 65	430 to 1100

Note

- Test condition see table "Basic Characteristics"

ORDERING INFORMATION			
ORDERING CODE	PACKAGING	REMARKS	PACKAGE FORM
BPW34	Bulk	MOQ: 3000 pcs, 3000 pcs/bulk	Top view
BPW34S	Tube	MOQ: 1800 pcs, 45 pcs/tube	Top view

Note

- MOQ: minimum order quantity

ABSOLUTE MAXIMUM RATINGS ($T_{amb} = 25^\circ C$, unless otherwise specified)				
PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
Reverse voltage		V_R	60	V
Power dissipation	$T_{amb} \leq 25^\circ C$	P_V	215	mW
Junction temperature		T_j	100	°C
Operating temperature range		T_{amb}	- 40 to + 100	°C
Storage temperature range		T_{stg}	- 40 to + 100	°C
Soldering temperature	$t \leq 3$ s	T_{sd}	260	°C
Thermal resistance junction/ambient	Connected with Cu wire, 0.14 mm ²	R_{thJA}	350	K/W



BASIC CHARACTERISTICS ($T_{amb} = 25\text{ }^{\circ}\text{C}$, unless otherwise specified)						
PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT
Breakdown voltage	$I_R = 100\text{ }\mu\text{A}$, $E = 0$	$V_{(BR)}$	60			V
Reverse dark current	$V_R = 10\text{ V}$, $E = 0$	I_{ro}		2	30	nA
Diode capacitance	$V_R = 0\text{ V}$, $f = 1\text{ MHz}$, $E = 0$	C_D		70		pF
	$V_R = 3\text{ V}$, $f = 1\text{ MHz}$, $E = 0$	C_D		25	40	pF
Open circuit voltage	$E_e = 1\text{ mW/cm}^2$, $\lambda = 950\text{ nm}$	V_o		350		mV
Temperature coefficient of V_o	$E_e = 1\text{ mW/cm}^2$, $\lambda = 950\text{ nm}$	TK_{V_o}		- 2.6		mV/K
Short circuit current	$E_A = 1\text{ klx}$	I_k		70		μA
	$E_e = 1\text{ mW/cm}^2$, $\lambda = 950\text{ nm}$	I_k		47		μA
Temperature coefficient of I_k	$E_e = 1\text{ mW/cm}^2$, $\lambda = 950\text{ nm}$	TK_{I_k}		0.1		%/K
Reverse light current	$E_A = 1\text{ klx}$, $V_R = 5\text{ V}$	I_{ra}		75		μA
	$E_e = 1\text{ mW/cm}^2$, $\lambda = 950\text{ nm}$, $V_R = 5\text{ V}$	I_{ra}	40	50		μA
Angle of half sensitivity		ϕ		± 65		deg
Wavelength of peak sensitivity		λ_p		900		nm
Range of spectral bandwidth		$\lambda_{0.1}$		430 to 1100		nm
Noise equivalent power	$V_R = 10\text{ V}$, $\lambda = 950\text{ nm}$	NEP		4×10^{-14}		$\text{W}/\sqrt{\text{Hz}}$
Rise time	$V_R = 10\text{ V}$, $R_L = 1\text{ k}\Omega$, $\lambda = 820\text{ nm}$	t_r		100		ns
Fall time	$V_R = 10\text{ V}$, $R_L = 1\text{ k}\Omega$, $\lambda = 820\text{ nm}$	t_f		100		ns

BASIC CHARACTERISTICS ($T_{amb} = 25\text{ }^{\circ}\text{C}$, unless otherwise specified)

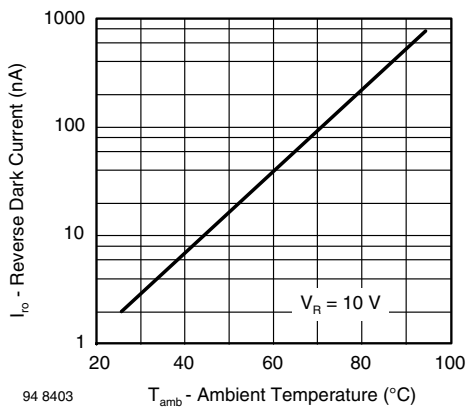


Fig. 1 - Reverse Dark Current vs. Ambient Temperature

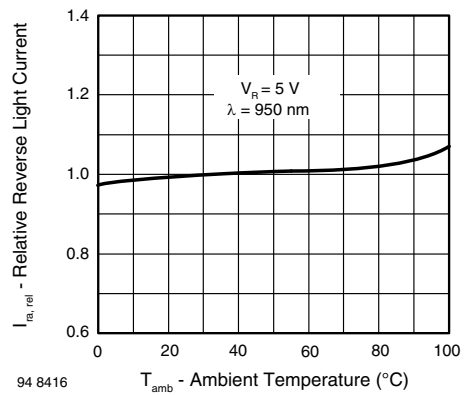


Fig. 2 - Relative Reverse Light Current vs. Ambient Temperature

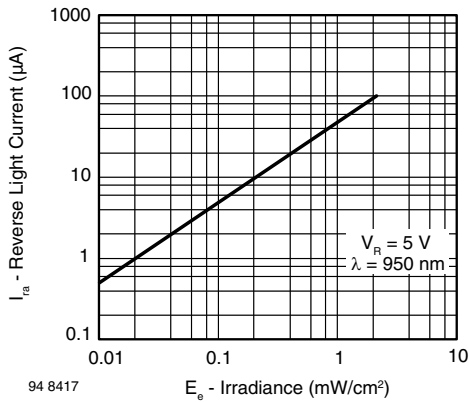


Fig. 3 - Reverse Light Current vs. Irradiance

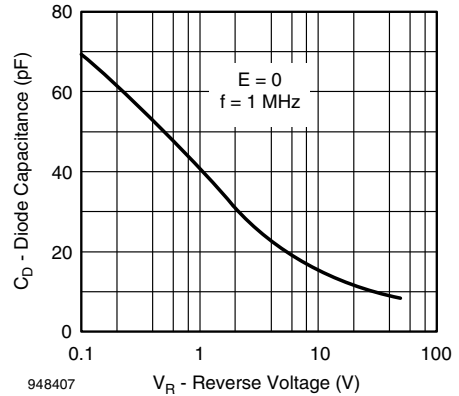


Fig. 6 - Diode Capacitance vs. Reverse Voltage

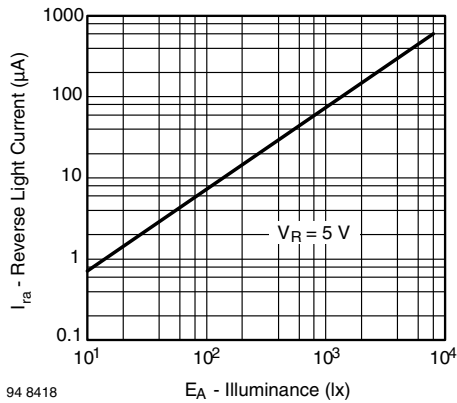


Fig. 4 - Reverse Light Current vs. Illuminance

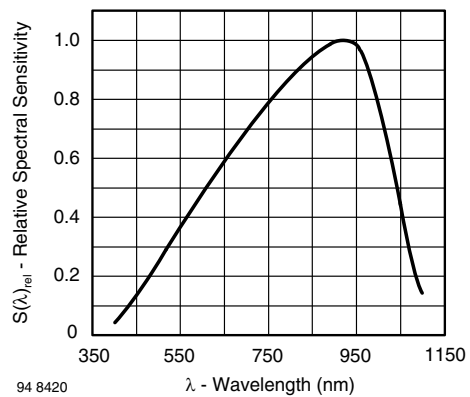


Fig. 7 - Relative Spectral Sensitivity vs. Wavelength

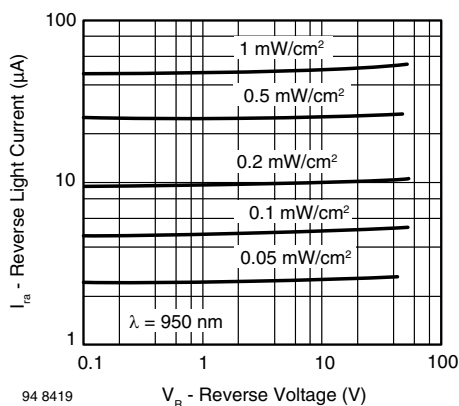


Fig. 5 - Reverse Light Current vs. Reverse Voltage

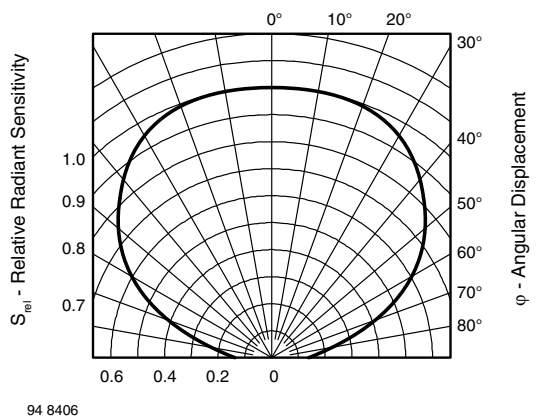
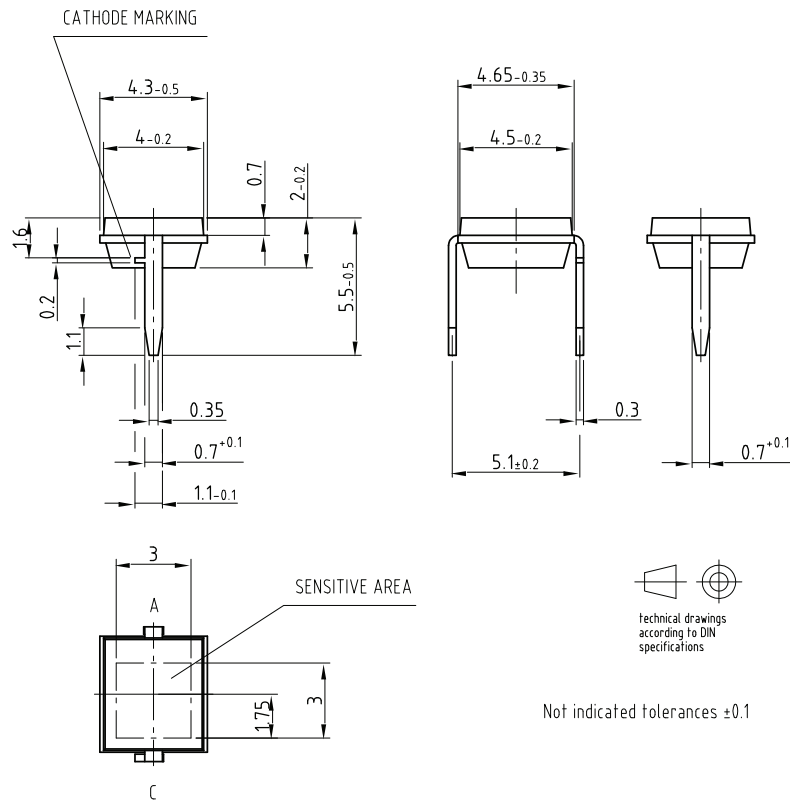


Fig. 8 - Relative Radiant Sensitivity vs. Angular Displacement

PACKAGE DIMENSIONS in millimeters



Drawing-No.: 6.544-5315.01-4
Issue: 1; 19.10.07
96 12186

TUBE PACKAGING DIMENSIONS in millimeters

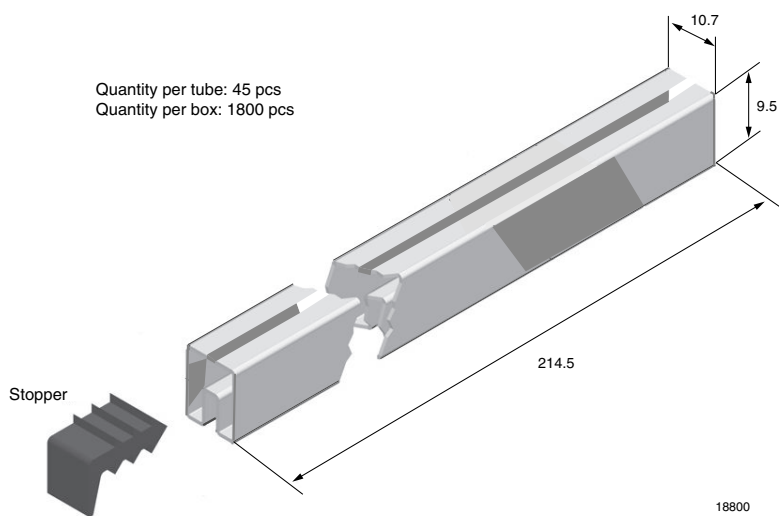


Fig. 9 - Drawing Proportions not scaled



Disclaimer

ALL PRODUCT, PRODUCT SPECIFICATIONS AND DATA ARE SUBJECT TO CHANGE WITHOUT NOTICE TO IMPROVE RELIABILITY, FUNCTION OR DESIGN OR OTHERWISE.

Vishay Intertechnology, Inc., its affiliates, agents, and employees, and all persons acting on its or their behalf (collectively, "Vishay"), disclaim any and all liability for any errors, inaccuracies or incompleteness contained in any datasheet or in any other disclosure relating to any product.

Vishay makes no warranty, representation or guarantee regarding the suitability of the products for any particular purpose or the continuing production of any product. To the maximum extent permitted by applicable law, Vishay disclaims (i) any and all liability arising out of the application or use of any product, (ii) any and all liability, including without limitation special, consequential or incidental damages, and (iii) any and all implied warranties, including warranties of fitness for particular purpose, non-infringement and merchantability.

Statements regarding the suitability of products for certain types of applications are based on Vishay's knowledge of typical requirements that are often placed on Vishay products in generic applications. Such statements are not binding statements about the suitability of products for a particular application. It is the customer's responsibility to validate that a particular product with the properties described in the product specification is suitable for use in a particular application. Parameters provided in datasheets and/or specifications may vary in different applications and performance may vary over time. All operating parameters, including typical parameters, must be validated for each customer application by the customer's technical experts. Product specifications do not expand or otherwise modify Vishay's terms and conditions of purchase, including but not limited to the warranty expressed therein.

Except as expressly indicated in writing, Vishay products are not designed for use in medical, life-saving, or life-sustaining applications or for any other application in which the failure of the Vishay product could result in personal injury or death. Customers using or selling Vishay products not expressly indicated for use in such applications do so at their own risk and agree to fully indemnify and hold Vishay and its distributors harmless from and against any and all claims, liabilities, expenses and damages arising or resulting in connection with such use or sale, including attorneys fees, even if such claim alleges that Vishay or its distributor was negligent regarding the design or manufacture of the part. Please contact authorized Vishay personnel to obtain written terms and conditions regarding products designed for such applications.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document or by any conduct of Vishay. Product names and markings noted herein may be trademarks of their respective owners.

Material Category Policy

Vishay Intertechnology, Inc. hereby certifies that all its products that are identified as RoHS-Compliant fulfill the definitions and restrictions defined under Directive 2011/65/EU of The European Parliament and of the Council of June 8, 2011 on the restriction of the use of certain hazardous substances in electrical and electronic equipment (EEE) - recast, unless otherwise specified as non-compliant.

Please note that some Vishay documentation may still make reference to RoHS Directive 2002/95/EC. We confirm that all the products identified as being compliant to Directive 2002/95/EC conform to Directive 2011/65/EU.



MicroMag3

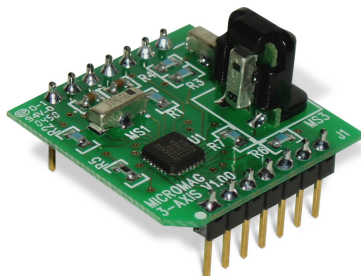
3-Axis Magnetic Sensor Module

General Description

The MicroMag3 is an integrated 3-axis magnetic field sensing module designed to aid in evaluation and prototyping of PNI Corporation's technology. The MicroMag3 combines PNI Corporation's patented Magneto-Inductive (MI) sensors and measurement circuit technology for unparalleled cost effectiveness and performance. The MI sensors change inductance by 100% over its field measurement range. This variable inductance property is used in a cost and space efficient ASIC (PNI 11096) which incorporates a patented temperature and noise stabilized oscillator/counter circuit with an SPI interface. The microprocessor compatible SPI interface allows easy access to the MicroMag3's measurement parameters and resulting field measurement data.

Advantages include 3V operation for compatibility with new systems, low power consumption, large signal noise immunity under all conditions, and a large dynamic range. Resolution and field measurement range are software configurable for a variety of applications. The measurement is very stable over temperature and inherently free from offset drift.

These advantages make PNI Corporation's MicroMag3 the choice for prototyping, education and low volume applications. A reference design is also available for use in high volume applications.



Features

- Low power: draws < 500 μ A at 3 VDC
- Small size: 25.4 x 25.4 x 19 mm
- Large field measurement range: $\pm 1100 \mu$ T (± 11 Gauss)
- High resolution field measurement: 0.015 μ T (0.00015 Gauss)
- Fast sample rate: up to 2000 samples/second
- Operation: 3.0 VDC
- Fully digital interface: SPI protocol at 3 V

Applications

- High-performance magnetic field sensing
- High-performance solid state navigation equipment for automotive, marine, and aeronautic applications
- Fast prototyping tool for PNI Corporation's Magneto-Inductive (MI) sensors

Ordering Information

Name	Part Number	Package
MicroMag3	12349	Each



Absolute Maximum Ratings

Symbol	Parameter	Maximum
V _{DD}	DC Supply Voltage	5.25 VDC
V _{IN}	Input Pin Voltage	V _{DD} + 0.3 VDC
I _{IN}	Input Pin Current	10.0 mA at 25 °C
T _{STRG}	Storage Temperature	125 °C

Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. These are stress ratings only. Functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

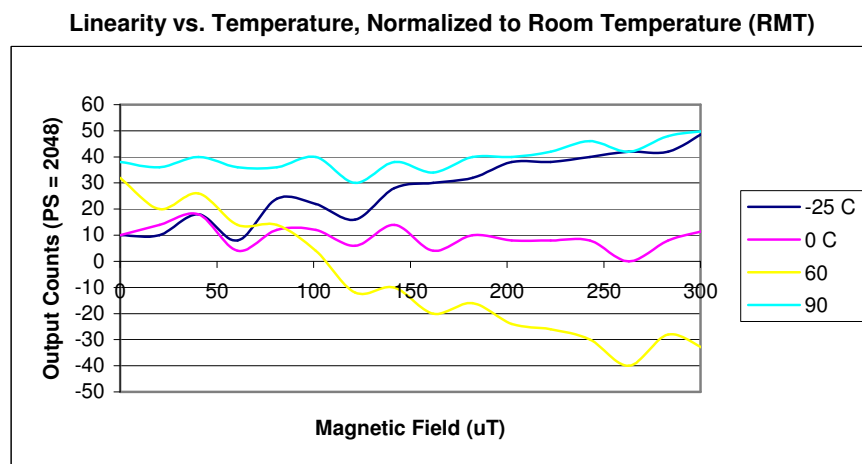
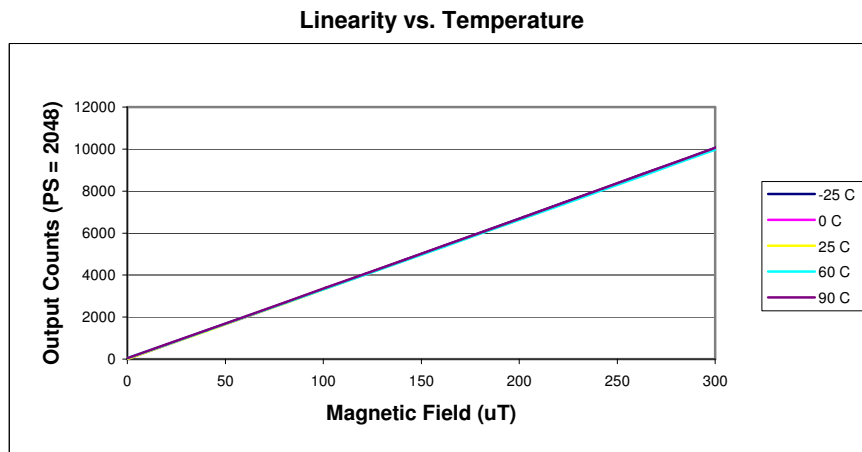
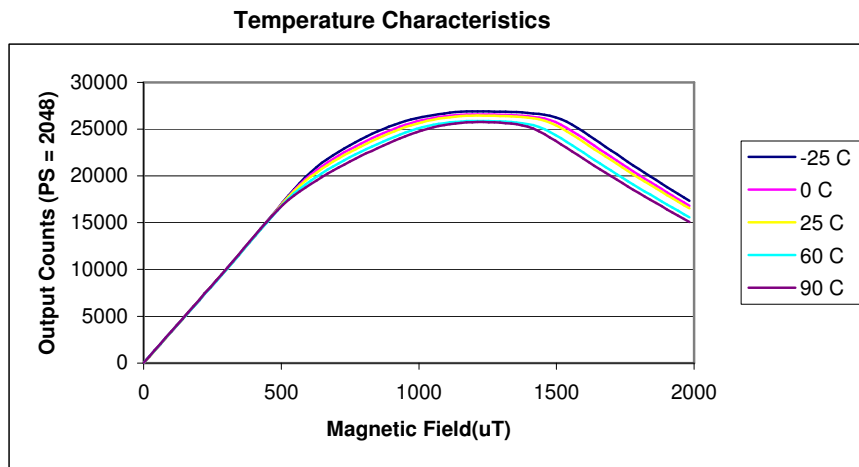
Module Characteristics

Parameter	Min	Typ	Max	Units
Idle current, measured at ASIC V _{CC} (3 VDC, R _b =43 Ω)		< 0.1		mA RMS
Continuous current, measured at ASIC V _{CC} (3 VDC, R _b =43 Ω)		0.4	0.5	mA RMS
Field Measurement Range ⁽¹⁾ (3 VDC, R _b =43 Ω)	-1100		1100	μT
Gain ⁽²⁾ (3 VDC, R _b =43 Ω)		31.24		Count/μT
Linearity (Error from best fit straight line at ±300 μT)		0.6	1	%
Resolution		1/Gain		μT
Sensor Frequency (3 VDC, R _b =43 Ω. Within free Earth's magnetic field.)		175		KHz
Operating Temperature	-20		70	°C
Storage Temperature	-40		125	°C

(1) Field Measurement Range is defined as the monotonic region of the output characteristic curve.

(2) Gain is defined as the change in the number of counts from the ASIC, when the period select is set to 2048, per change in the magnetic field in μT. For situations requiring higher gain and less field measurement range, the gain and resolution can be increased by a factor of 2 by setting the ASIC period select to 4096. When setting higher period selects, be aware that the ASIC counter can overflow if the field is strong enough drive the count beyond a signed 16-bit integer. Period select set to 2048 is the highest setting where it is impossible to overflow the counter. In practical magnetometer applications, a sensor gain calibration is normally performed when the sensor module is in the host system. Gain will change in response to a 5 VDC supply voltage or with the use of alternate bias resistor values. PNI recommends changing the R_b value to 75 Ω for 5 VDC operations.

Typical Operating Characteristics: MicroMag3 (3 VDC; Rb = 43 Ω)



Output Counts is defined as the Period Select (PS) setting for the 11096 ASIC. See “PNI-11096, 3-Axis Magneto-Inductive Sensor Driver and Controller with SPI Serial Interface” data sheet for more information.

Host Processor Interface

All accesses to and from the MicroMag3 are through a hardware handshaking, synchronous serial interface that adheres to the Motorola SPI protocol. The interface consists of six signals; SCLK, MOSI, MISO, SSNOT, RESET and DRDY. See “PNI-11096, 3-Axis Magneto-Inductive Sensor Driver and Controller with SPI Serial Interface” data sheet for detailed information on the interface.

SPI Port Line Descriptions

MOSI (Master Out Slave In)

The data sent from the master to the MicroMag3. Data is transferred most significant bit first. The MOSI line will accept data once the SPI is enabled by taking the SSNOT low. Valid data must be presented at least 100 nS before the rising edge of the clock, and remain valid for 100 nS after the edge. New data may be presented to the MOSI pin on the falling edge of SCLK.

SSNOT (Slave Select Line)

Selects the MicroMag3 as the operating slave device. The SSNOT line must be low prior to data transfer and must stay low during the entire transfer. Once the command byte is received by the MicroMag3, and the MicroMag3 begins to execute the command, the SSNOT line can be deselected until the next SPI transfer.

SCLK (Serial Clock)

Used to synchronize both the data in and out through the MISO and MOSI lines. SCLK is generated by a master device. SCLK should be 1 MHz or less. The MicroMag3 is configured to run as a slave device, making it an input. One byte of data is exchanged over eight clock cycles. Data is captured by the master device on the rising edge of SCLK. Data is shifted out and presented to the MicroMag3 on the MOSI pin on the falling edge of SCLK.

MISO (Master In Slave Out)

The data sent from the MicroMag3 to the master. Data is transferred most significant bit first. The MISO line is placed in a high impedance state if the slave is not selected (SSNOT = 1).

Hardware Handshaking Line Descriptions

RESET

RESET is usually low. RESET must be toggled from low-high-low.

DRDY (Data Ready)

DRDY is low after a RESET; after a command has been received and the data is ready, DRDY will be high. It is recommended that the DRDY line always be used to ensure that the data is clocked out of the MicroMag3 only when it is available. If it is determined that the DRDY line cannot be used due to lack of I/O lines to the host processor, then the times listed in the table below can be used to set open-loop wait times. The values listed are the maximum delays from the end of the SCLK command until the rise of the DRDY at each period select setting. The maximum delay occurs when the sensor being sampled is in a zero field

Period Select	Maximum Delay
/32	500 μ S
/64	1.0 mS
/128	2.0 mS
/256	4.0 mS
/512	7.5 mS
/1024	15 mS
/2048	35.5 mS
/4096	60 mS

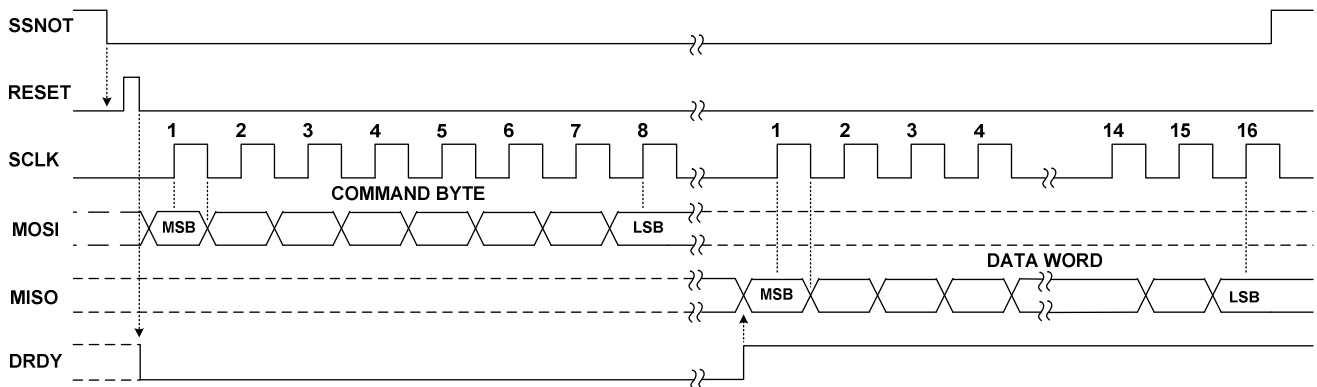
Operation

Basic operation will follow these steps. Refer to the timing diagrams on the following page.

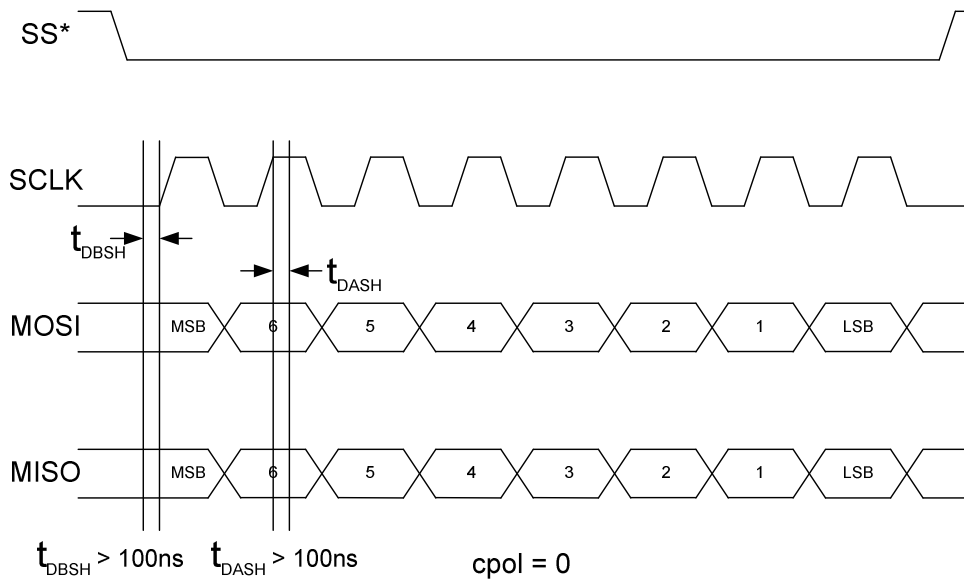
1. SSNOT is brought low.
2. Pulse RESET high (return to low state). You must RESET the MicroMag3 before every measurement.
3. Data is clocked in on the MOSI line. Once eight bits are read in, the MicroMag3 will execute the command.
4. The MicroMag3 will make the measurement. A measurement consists of forward biasing the sensor and making a period count; then reverse biasing the sensor and counting again; and finally, taking the difference between the two bias directions.
5. At the end of the measurement, the DRDY line is set to high indicating that the data is ready. In response to the next 16 SCLK pulses, data is shifted out on the MISO line.

If you need to make another measurement, go to Step 2. You can send another command after the reset. In this case, keep SSNOT low. If you will not be using the MicroMag3, set SSNOT to high to disable the SPI port.

SPI Port Full Timing Sequence (cpol = 0)



SPI Port Timing Parameters (cpol = 0)



SPI Port Usage Tips

A SPI port can be implemented using different clock polarity options. The clock polarity used with the MicroMag3 must be normally low, (***cpol = 0***). *SPI Port Timing Parameters* graphically shows the timing sequence (***cpol = 0***). Data is always considered valid while the SCLK is high ($t_{DASH} = \text{Time, Data After SCLK High}$). When SCLK is low, the data is in transition ($t_{DBSH} = \text{Time, Data Before SCLK High}$). When implementing a SPI port, whether it is a dedicated hardware peripheral port, or a software implemented port using general purpose I/O (also known as *Bit-Banging*) the timing parameters given in *SPI Port Timing Parameters* must be met to ensure reliable communications. The clock set-up and hold times, t_{DBSH} and t_{DASH} must be greater than 100 nS.

Idle Mode

The MicroMag3 does not initialize in the idle mode at power-up. The MicroMag3 must be in a data-ready state for the idle mode to occur. After power-up the MicroMag3 can be brought to the data-ready state by following these steps for sending a read command to the MicroMag3.

1. Set SSNOT low.
2. Pulse the RESET line.
3. Send a command to the MicroMag3 to measure one of the sensors.
4. Once the SSNOT pin is set to high again the MicroMag3 will go into the low power idle mode.
5. The DRDY pin will eventually go high signifying that the MicroMag3 is in the data-ready state. The resultant data *does not* have to be read from the MicroMag3.

Magnetic Measurements

The MicroMag3 magnetic sensor operates as an oscillator circuit composed of the internal sensors, bias resistors, digital gates and a comparator. Only one sensor can be measured at a time. The user sends a command byte to the MicroMag3 through the SPI port specifying the sensor axis to be measured. After dividing by the ratio set by PS2-PS0, the MicroMag3 will return the result of a complete forward - reverse measurement of the sensor in a 16-bit 2's complement format (Range: -32768 to 32767).

Command Byte

The operation of the MicroMag3 is controlled by the data received into the SPI port. The command byte syntax is as follows:

Position	7	6	5	4	3	2	1	0
Bit	DHST	PS2	PS1	PS0	ODIR	MOT	ASI	ASO
RESET	0	0	0	0	0	0	0	0

ASO and AS1 (Axis Select)

Determines which axis is being measured.

Function	ASI	AS0
X axis	0	1
Y axis	1	0
Z axis	1	1

MOT (Magnetic Oscillator Test)

When set, causes the magnetic oscillator selected by AS0 and AS1 in the directions selected by ODIR to run continuously until the MicroMag3 is reset.

ODIR (Oscillator Direction)

Determines the magnetic oscillator direction if MOT is set to 1. It has no effect on direction when the MOT bit is set to 0. *This is used for debug purposes only, and will not be set in normal operation.*

DHST (High Speed Oscillator Test)

When high, the internal high speed clock is set to drive the DHST pin of the ASIC at ½ the clock speed. When low, the DHST pin is set to DVDD. *This is used for debug purposes only, and will not be set in normal operation.*

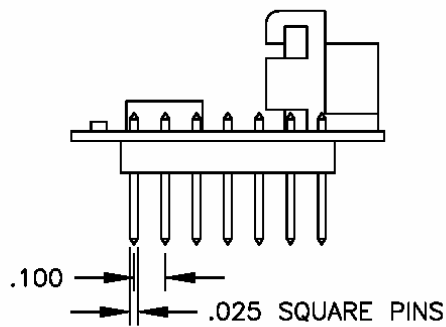
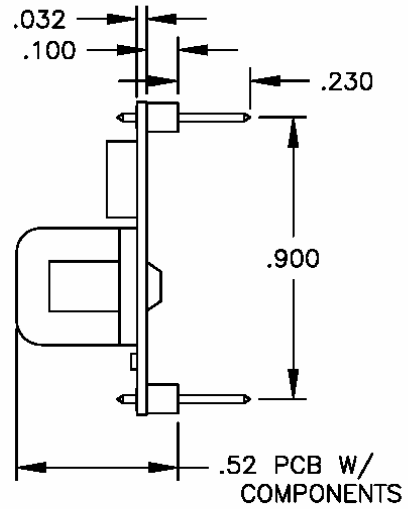
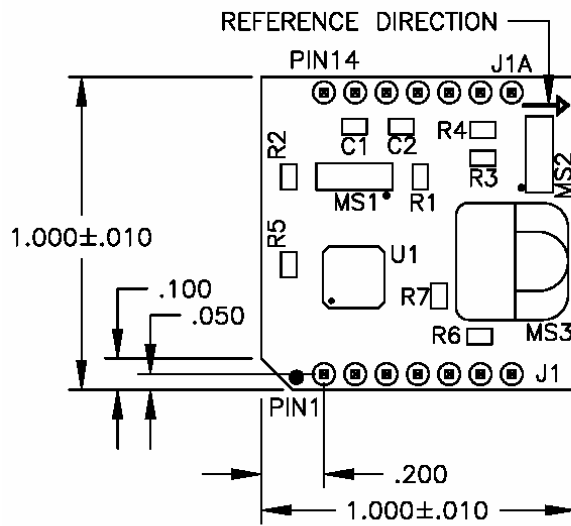
PS0, PS1, and PS2 (Period Select)

Selects the division ratio applied to the L/R oscillator output to set the period being measured.

PS2	PS1	PS0	Ratio
0	0	0	/32
0	0	1	/64
0	1	0	/128
0	1	1	/256
1	0	0	/512
1	0	1	/1024
1	1	0	/2048
1	1	1	/4096

Mechanical Information

All dimensions are in inches.



UNLESS OTHERWISE SPECIFIED
ALL DIMENSIONS ARE MAX

PIN	FUNCTION	PIN	FUNCTION
1	SCLK	8	N/C
2	MISO	9	N/C
3	MOSI	10	N/C
4	SS	11	N/C
5	DRDY	12	VDD
6	RESET	13	N/C
7	GND	14	GND

Buffered H-Bridge

FEATURES

- 1.0-A H-Bridge
- 500-kHz Switching Rate
- Shoot-Through Limited
- TTL Compatible Inputs
- 3.8- to 13.2-V Operating Range
- Surface Mount Packaging

APPLICATIONS

- VCM Driver
- Brushed Motor Driver
- Stepper Motor Driver
- Power Converter
- Optical Disk Drives
- Power Supplies
- High Performance Servo

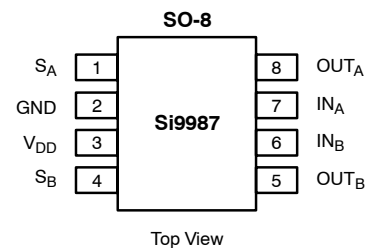
DESCRIPTION

The Si9987 is an integrated, buffered H-bridge with TTL compatible inputs and the capability of delivering a continuous 1.0 A @ $V_{DD} = 5.0\text{ V}$ (room temperature) at switching rates up to 500 kHz. Internal logic prevents the upper and lower outputs of either half-bridge from being turned on simultaneously. Unique input codes allow both outputs to be forced low (for braking) or

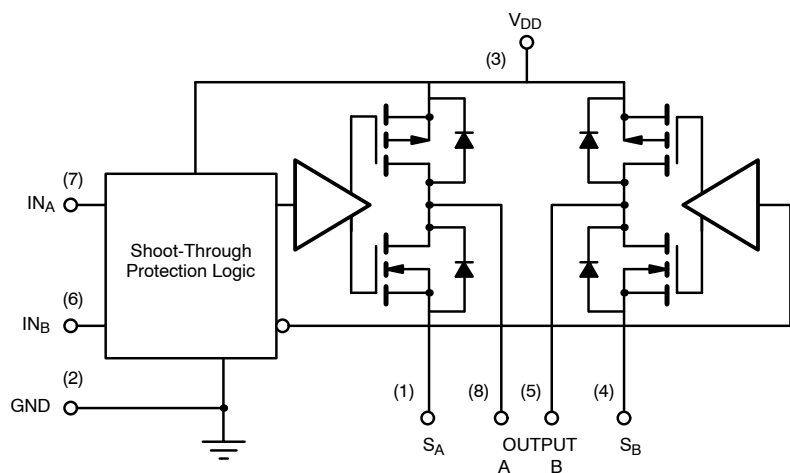
forced to a high impedance level.

The Si9987 is available in an 8-Pin SOIC package, specified to operate over a voltage range of 3.8 V to 13.2 V, and the commercial temperature range of 0 to 70°C (C suffix) and -40 to 85°C (D suffix). The Si9987 is available in lead free.

FUNCTIONAL BLOCK DIAGRAM, PIN CONFIGURATION AND TRUTH TABLE



TRUTH TABLE			
IN _A	IN _B	OUT _A	OUT _B
1	0	1	0
0	1	0	1
0	0	0	0
1	1	HiZ	HiZ



ORDERING INFORMATION		
Part Number	Temperature Range	Package
Si9987CY-T1	0 to 70°C	Tape and Reel
Si9987DY-T1	-40 to 85°C	
Si9987CY-T1—E3	0 to 70°C	Lead Free Tape and Reel
Si9987DY-T1—E3	-40 to 85°C	
Si9987CY	0 to 70°C	Bulk (tubes)
Si9987DY	-40 to 85°C	



ABSOLUTE MAXIMUM RATINGS^a

Voltage on any pin with respect to ground	-0.3 V to $V_{DD} + 0.3$ V
Voltage on pins 5, 8 with respect to GND	-1 V to $V_{DD} + 1$ V
Voltage on pins 1, 4	-0.3 V to GND +1 V
Maximum V_{DD}	15 V
Peak Output Current	1.5 A
Storage Temperature	-65 to 150°C
Maximum Junction Temperature (T_J)	150°C
Power Dissipation ^b	1 W
θ_{JA}	100°C/W

Continuous I_{OUT} Current ($T_J = 135^\circ\text{C}$)^c

$T_A = 25^\circ\text{C}$	± 1.02 A
$T_A = 70^\circ\text{C}$	± 0.75 A
$T_A = 85^\circ\text{C}$	± 0.65 A

Operating Temperature Range

Si9987CY	0 to 70°C
Si9987DY	-40 to 85°C

Notes

- Device mounted with all leads soldered or welded to PC board.
- Derate 10 mW/°C above 25°C.
- $T_J = T_A + (P_D \times \theta_{JA})$, P_D = Power Dissipation .

RECOMMENDED OPERATING RANGE

V_{DD}	3.8 V to 13.2 V
Maximum Junction Temperature (T_J)	135°C

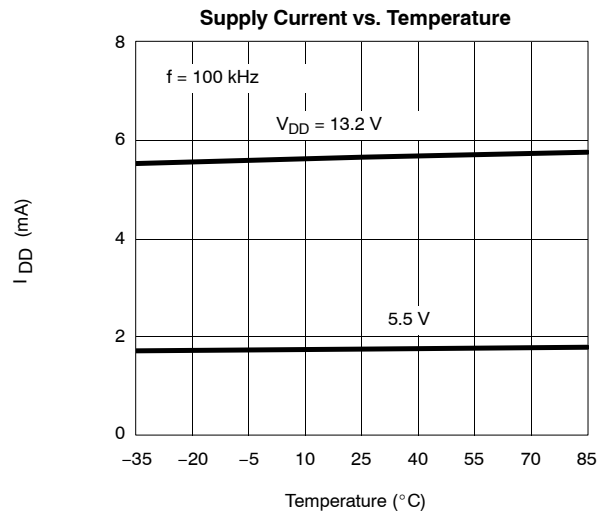
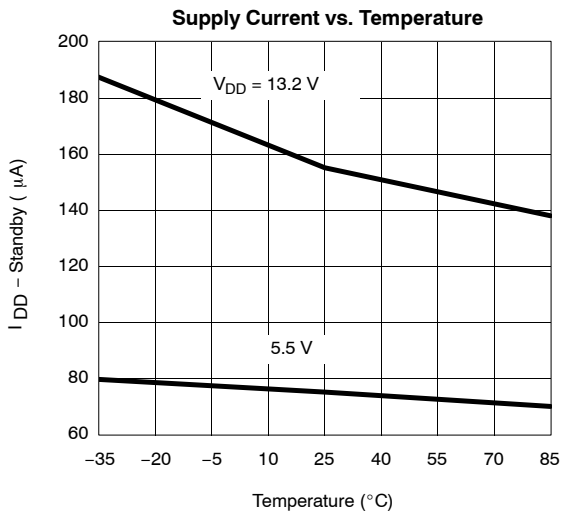
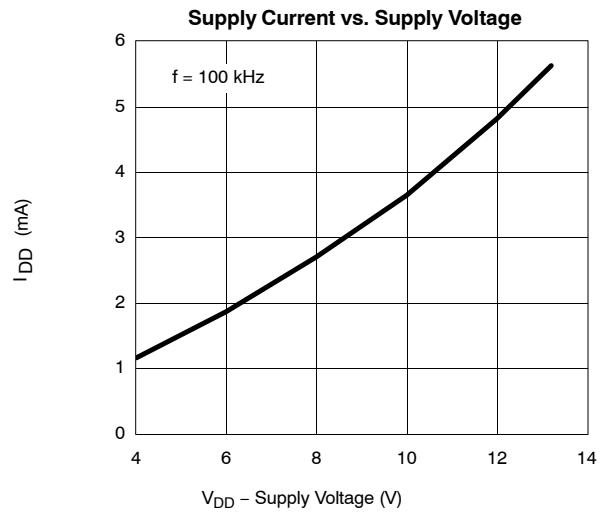
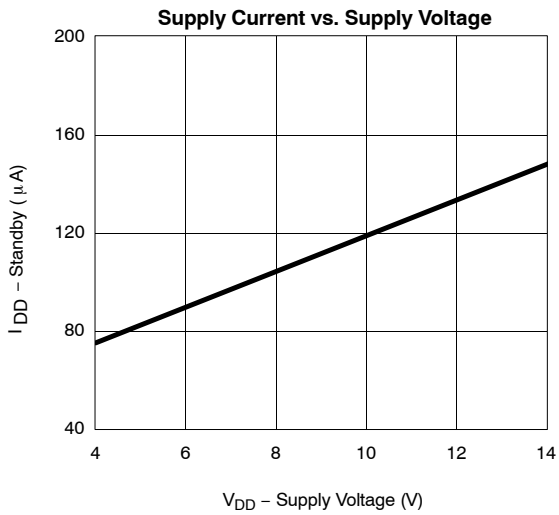
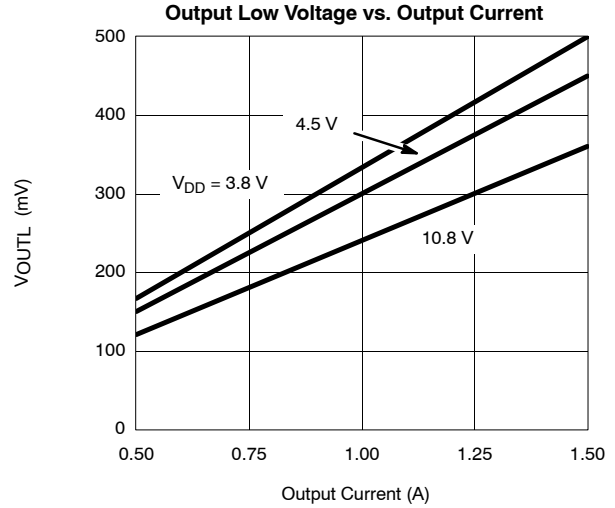
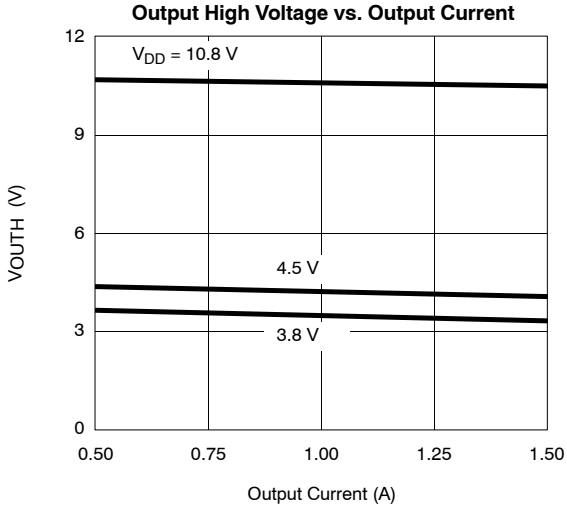
SPECIFICATIONS						
Parameter	Symbol	Test Conditions Unless Specified $V_{DD} = 3.8$ to 13.2 V S_A @ GND, S_B @ GND	Limits			Unit
			Min ^a	Typ ^b	Max ^a	
Input						
Input Voltage High	V_{INH}		2			V
Input Voltage Low	V_{INL}				1	V
Input Current with Input Voltage High	I_{INH}	$V_{IN} = 2$ V			1	μA
Input Current with Input Voltage Low	I_{INL}	$V_{IN} = 0$ V	-1			μA
Output						
Output Voltage High ^c	V_{OUTH}	$I_{OUT} = -1$ A	$V_{DD} = 10.8$ V	10.40	10.56	V
			$V_{DD} = 4.5$ V	4.00	4.20	
		$I_{OUT} = -500$ mA	$V_{DD} = 10.8$ V	10.60	10.68	
$V_{DD} = 4.5$ V	4.25		4.35			
		$I_{OUT} = -300$ mA, $V_{DD} = 3.8$ V	3.63	3.70		
Output Voltage Low ^c	V_{OUTL}	$I_{OUT} = 1$ A	$V_{DD} = 10.8$ V		0.24	
			$V_{DD} = 4.5$ V		0.30	0.50
		$I_{OUT} = 500$ mA	$V_{DD} = 10.8$ V		0.12	0.20
			$V_{DD} = 4.5$ V		0.15	0.25
		$I_{OUT} = 300$ mA, $V_{DD} = 3.8$ V		0.10	0.17	
Output Leakage Current Low	I_{OLL}	$I_{NA} = I_{NB} \geq 2$ V, $V_{OUT} = V_{DD} = 13.2$ V		0	10	μA
Output Leakage Current High	I_{OLH}	$V_{OUT} = 0$, $V_{DD} = 13.2$ V	-10	0		μA
Output V Clamp High	V_{CLH}	$I_{NA} = I_{NB} \geq 2$ V	$I_{OUT} = 100$ mA		$V_{DD} + 0.7$	$V_{DD} + 0.9$
Output V Clamp Low	V_{CLL}		$I_{OUT} = -100$ mA	-0.9	-0.7	
Supply						
V_{DD} Supply Current	I_{DD}	$I_N = 100$ kHz, $V_{DD} = 5.5$ V		1.8	2.5	mA
		$I_{NA} = I_{NB} = 4.5$ V, $V_{DD} = 5.5$ V		75	125	μA
Dynamic						
Propagation Delay Time	T_{PLH}	$V_{DD} = 5$ V		300		nS
	T_{PHL}			100		

Notes

- The algebraic convention whereby the most negative value is a minimum and the most positive a maximum, is used in this data sheet.
- Typical values are for DESIGN AID ONLY, not guaranteed nor subject to production testing.
- Maximum value measured at $T_J = 135^\circ\text{C}$. Typical value measured at $T_J = T_A = 25^\circ\text{C}$ (pulse width ≤ 300 μsec , duty cycle $\leq 2\%$).

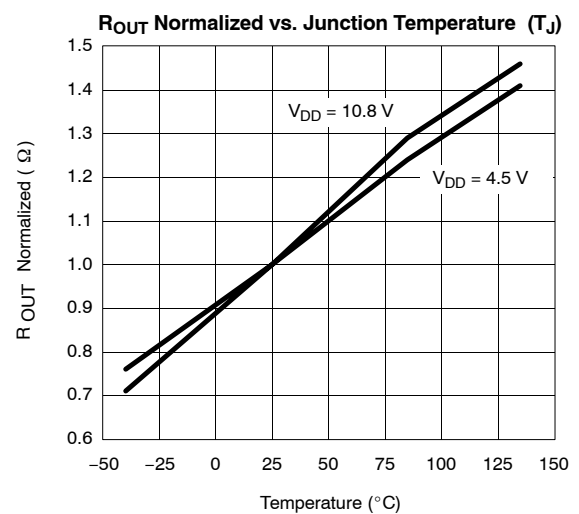
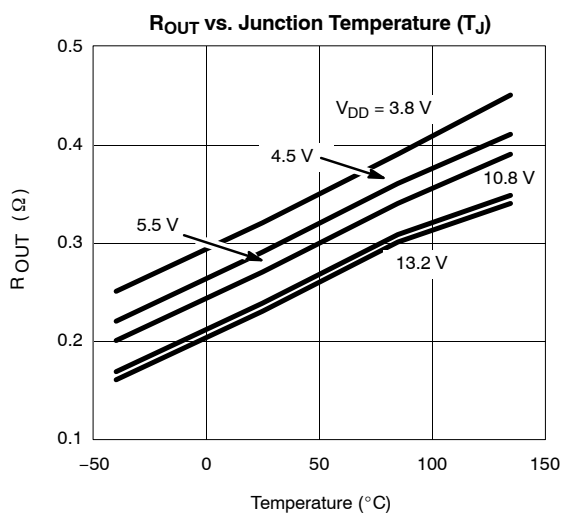
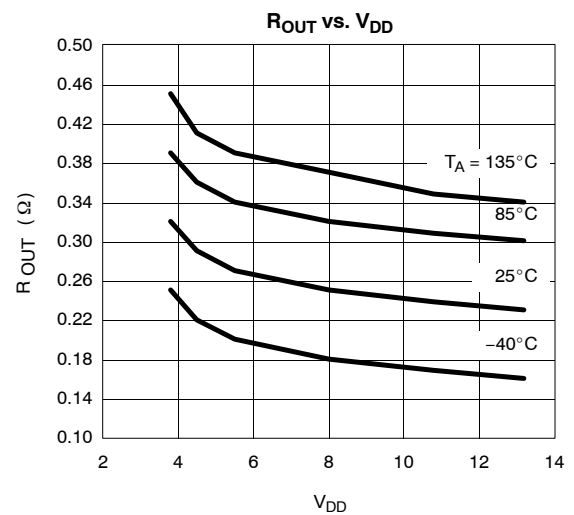
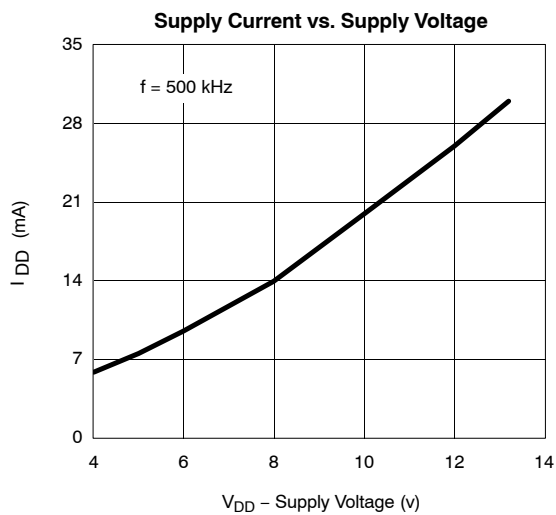
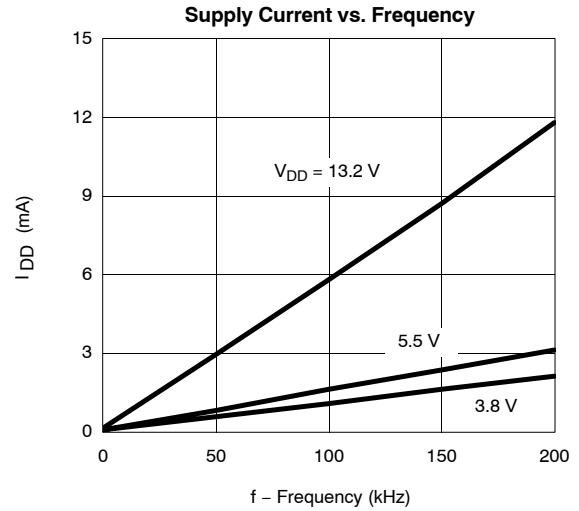
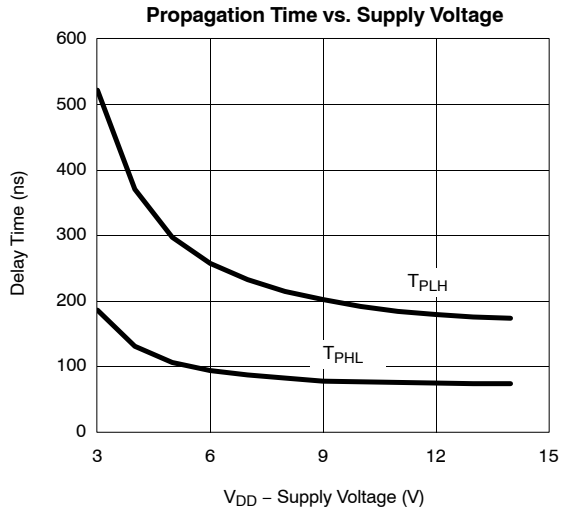


TYPICAL CHARACTERISTICS (25 °C UNLESS NOTED)





TYPICAL CHARACTERISTICS (25 °C UNLESS NOTED)





Notice

Specifications of the products displayed herein are subject to change without notice. Vishay Intertechnology, Inc., or anyone on its behalf, assumes no responsibility or liability for any errors or inaccuracies.

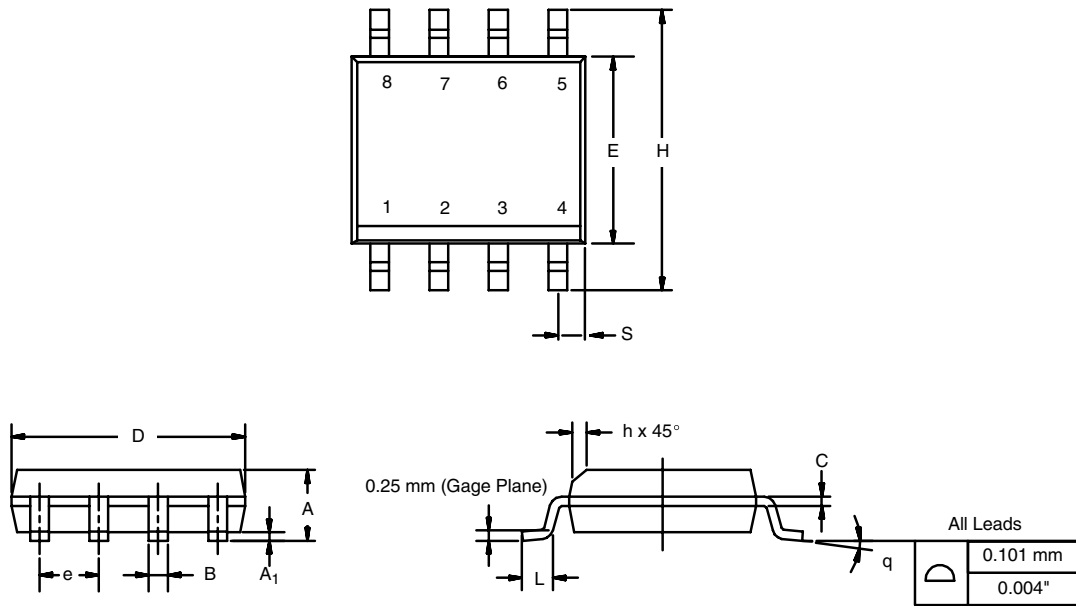
Information contained herein is intended to provide a product description only. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Vishay's terms and conditions of sale for such products, Vishay assumes no liability whatsoever, and disclaims any express or implied warranty, relating to sale and/or use of Vishay products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright, or other intellectual property right.

The products shown herein are not designed for use in medical, life-saving, or life-sustaining applications. Customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Vishay for any damages resulting from such improper use or sale.



SOIC (NARROW): 8-LEAD

JEDEC Part Number: MS-012



DIM	MILLIMETERS		INCHES	
	Min	Max	Min	Max
A	1.35	1.75	0.053	0.069
A ₁	0.10	0.20	0.004	0.008
B	0.35	0.51	0.014	0.020
C	0.19	0.25	0.0075	0.010
D	4.80	5.00	0.189	0.196
E	3.80	4.00	0.150	0.157
e	1.27 BSC		0.050 BSC	
H	5.80	6.20	0.228	0.244
h	0.25	0.50	0.010	0.020
L	0.50	0.93	0.020	0.037
q	0°	8°	0°	8°
S	0.44	0.64	0.018	0.026
ECN: C-06527-Rev. I, 11-Sep-06				
DWG: 5498				



Disclaimer

ALL PRODUCT, PRODUCT SPECIFICATIONS AND DATA ARE SUBJECT TO CHANGE WITHOUT NOTICE TO IMPROVE RELIABILITY, FUNCTION OR DESIGN OR OTHERWISE.

Vishay Intertechnology, Inc., its affiliates, agents, and employees, and all persons acting on its or their behalf (collectively, "Vishay"), disclaim any and all liability for any errors, inaccuracies or incompleteness contained in any datasheet or in any other disclosure relating to any product.

Vishay makes no warranty, representation or guarantee regarding the suitability of the products for any particular purpose or the continuing production of any product. To the maximum extent permitted by applicable law, Vishay disclaims (i) any and all liability arising out of the application or use of any product, (ii) any and all liability, including without limitation special, consequential or incidental damages, and (iii) any and all implied warranties, including warranties of fitness for particular purpose, non-infringement and merchantability.

Statements regarding the suitability of products for certain types of applications are based on Vishay's knowledge of typical requirements that are often placed on Vishay products in generic applications. Such statements are not binding statements about the suitability of products for a particular application. It is the customer's responsibility to validate that a particular product with the properties described in the product specification is suitable for use in a particular application. Parameters provided in datasheets and/or specifications may vary in different applications and performance may vary over time. All operating parameters, including typical parameters, must be validated for each customer application by the customer's technical experts. Product specifications do not expand or otherwise modify Vishay's terms and conditions of purchase, including but not limited to the warranty expressed therein.

Except as expressly indicated in writing, Vishay products are not designed for use in medical, life-saving, or life-sustaining applications or for any other application in which the failure of the Vishay product could result in personal injury or death. Customers using or selling Vishay products not expressly indicated for use in such applications do so at their own risk and agree to fully indemnify and hold Vishay and its distributors harmless from and against any and all claims, liabilities, expenses and damages arising or resulting in connection with such use or sale, including attorneys fees, even if such claim alleges that Vishay or its distributor was negligent regarding the design or manufacture of the part. Please contact authorized Vishay personnel to obtain written terms and conditions regarding products designed for such applications.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document or by any conduct of Vishay. Product names and markings noted herein may be trademarks of their respective owners.

Material Category Policy

Vishay Intertechnology, Inc. hereby certifies that all its products that are identified as RoHS-Compliant fulfill the definitions and restrictions defined under Directive 2011/65/EU of The European Parliament and of the Council of June 8, 2011 on the restriction of the use of certain hazardous substances in electrical and electronic equipment (EEE) - recast, unless otherwise specified as non-compliant.

Please note that some Vishay documentation may still make reference to RoHS Directive 2002/95/EC. We confirm that all the products identified as being compliant to Directive 2002/95/EC conform to Directive 2011/65/EU.

Índice de tablas

2.1. Lista de Cubesats lanzados	11
5.1. Funciones del módulo Scheduler	24
5.2. Funciones del módulo ADC	28
5.3. Funciones del módulo RTC	31
5.4. Maestro envía un determinado número de bytes	37
5.5. Maestro lee un determinado número de bytes	38
5.6. Buffer del esclavo cuando recibe un determinado número de bytes	39
5.7. Funciones del módulo I2C	42
5.8. Modos de bajo consumo	47
5.9. Funciones del módulo LPM	52
5.10. Datos housekeeping	53
5.11. Funciones del módulo HK	54
5.12. Funciones del módulo WDT	56
5.13. Satélites comparados	61
5.14. Asignación de pines de sensores de temperatura	66
5.15. Asignación de pines de sensores de temperatura	68
5.16. Asignación de pines de oscilador externo	69
5.17. Asignación de pines de JTAG	73
5.18. Asignación de pines para fotodiodos	75
5.19. Asignación de pines para magnetómetro	76
5.20. Asignación de pines para Puentes-H	78

<i>ÍNDICE DE TABLAS</i>	151
6.1. Ejemplos de código del fabricante	83
6.2. Ejemplos de código del fabricante	86
6.3. Ejemplos de código del fabricante	86
B.1. Configuración ADC	102
C.1. Configuración RTC	103
D.1. Registros de configuración I2C	104
E.1. Cálculo de costos en materiales	111
E.2. Cálculo de horas invertidas	113

Índice de figuras

2.1. Tipos de estructuras	7
2.2. Especificaciones para fabricación de CubeSat	8
2.3. Módulo P-POD	9
2.4. Vehículo de lanzamiento	9
4.1. Ejemplo de cola de ejecución de tareas del MCP	16
4.2. diagrama de comunicación entre la estación terrena y el satélite	17
5.1. Diagrama de flujo principal	21
5.2. Diagrama de flujo del encolamiento de tareas	22
5.3. Curva de temperatura del sensor interno	27
5.4. Comportamiento de las líneas SDA y SCL.	33
5.5. Transiciones válidas del SDA	34
5.6. Ejemplo de secuencia de transmisión I2C	34
5.7. Ejemplo de una <i>repeated start condition</i> de I2C	36
5.8. Envío de datos del maestro	43
5.9. Envío de datos del esclavo	44
5.10. Lectura de datos del esclavo	45
5.11. Lectura de datos del maestro	46
5.12. Tabla de los modos de bajo consumo	48
5.13. Consumo sin LPM	49
5.14. Entrada al LPM al finalizar la ejecución de una tarea	50

5.15. Entrada al LPM una vez que finaliza la ejecución de todas las tareas	51
5.16. Diagrama de flujo de la atención a la llegada de datos del COM	59
5.17. Diagrama de flujo de la atención a un pedido de repetición de paquetes AX.25	60
5.18. Conexionado de módulos	63
5.19. Esquemático de la placa de prueba	64
5.20. Conexión entre microcontroladores	65
5.21. Función de transferencia del LM20	66
5.22. Conexión básica LM20	67
5.23. Conexión de sensores de temperatura.	67
5.24. Conexión de oscilador externo.	69
5.25. Conexiones de las señales del JTAG	70
5.26. MSP430 USB-Debug-Interface	71
5.27. Conexión JTAGs.	72
5.28. Conexiones para Fotodiodos	74
5.29. Conexiones para magnetómetro	76
5.30. Conexiones para magnetotorquers	78
5.31. Capa superior de la placa de prueba	79
5.32. Capa Inferior de la placa de prueba	80
5.33. Foto de la placa de prueba	81
D.1. Modo esclavo transmisor	105
D.2. Modo esclavo receptor	106
D.3. Modo maestro transmisor	107
D.4. Modo maestro receptor	108
D.5. Modo maestro receptor	109
E.1. Carga horaria mensual durante el proyecto	113