



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DISEÑO, CONSTRUCCIÓN Y VALIDACIÓN
DEL SISTEMA DE MONITOREO Y
ADQUISICIÓN DE DATOS (HOUSEKEEPING)
PARA EL FOTO DETECTOR A BORDO DE
LA MISIÓN EUSO-BALLOON

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO ELÉCTRICO ELECTRÓNICO

(ÁREA: ELECTRÓNICA)

PRESENTAN:

De la Cruz Martínez Adrián

López López César Alan

Rojas Garcés Javier

Silvarán Guerrero Sócrates Alberto



DIRECTOR DE TESIS: M. I. LAURO SANTIAGO CRUZ

MÉXICO, D.F.

2013

AGRADECIMIENTOS

Al Dr. Gustavo Adolfo Medina Tanco, gracias por su liderazgo, visión, dirección y confianza durante el desarrollo del proyecto, por los apoyos económicos brindados y por la aportación del equipo con el que desarrollamos este trabajo, además de que su imprescindible participación como coordinador de proyecto, ya que fue de gran ayuda para alcanzar las metas propuestas en esta tesis.

Al M. I. Lauro Santiago Cruz, por su valiosa e invaluable dirección en este proyecto. Con su experiencia y dedicación se convirtió en nuestro principal apoyo durante el desarrollo del proyecto y del trabajo de tesis, siendo un ejemplo a seguir en el ámbito profesional, laboral y académico.

Al Instituto de Ciencias Nucleares, que nos brindó sus instalaciones y el equipo necesario para el desarrollo del proyecto, de manera especial agradecemos al M. C. Enrique Patiño, por su asesoría técnica en el manejo del equipo de laboratorio.

Al Instituto de Ingeniería, por permitirnos realizar nuestro servicio social en sus instalaciones, así como contribuir en partes esenciales en el desarrollo de este trabajo. Al Dr. Frederic Trillaud, por su constante colaboración, consejos brindados, así como su valiosa aportación.

A la colaboración internacional de JEM-EUSO, que dio la oportunidad a México para participar dentro de este proyecto tan importante, y así ayudar a la formación de recursos humanos en el país.

Al apoyo de becas de CONACYT, las cuales fueron una gran ayuda para nosotros durante el desarrollo del proyecto.

A los sinodales que revisaron este trabajo, gracias por sus valiosas observaciones.

Adrián, César, Javier, Sócrates

A **Dios** por permitirme terminar mi carrera, por no dejarme solo en esta larga travesía y darme la oportunidad de dar esta gran satisfacción a mis padres y hermanos.

A mis padres **Ramón y Sonia**, por siempre creer en mí y hacer todo lo posible para que este momento sea posible, sin ustedes no lo habría logrado.

A mis hermanos, por la comprensión y apoyo brindados en todo momento.

A la **Universidad Nacional Autónoma de México** por brindarme la oportunidad de obtener una carrera universitaria y una formación profesional. Así como a la **Facultad de Ingeniería** por brindarme un lugar dentro de sus aulas, en las cuales atravesé por una de las mejores etapas de mi vida.

Al **Instituto de Ciencias Nucleares y de Ingeniería** por brindarme la oportunidad de realizar servicio social y este trabajo de tesis dentro de sus instalaciones.

Al **Doctor Gustavo Adolfo Medina Tanco**, responsable del proyecto JEM-EUSO en México, por darme la oportunidad de participar en este importante proyecto, el cual contribuyó para mi desarrollo profesional en un campo muy competitivo de la ingeniería. Además de brindarme el apoyo necesario para la realización de este trabajo de tesis, permitirme viajar al extranjero y vivir experiencias importantes para mi formación profesional.

A mi director de tesis el **M. I. Lauro Santiago Cruz**, por su valiosa asesoría y dedicación para dirigir este trabajo de tesis, por su entrega para formar profesionales de alta calidad y por brindarme su amistad.

Al **Doctor Frederic Trillaud** por su apoyo en el uso de la cámara de vacío y conocimientos compartidos.

Al **Maestro Enrique Patiño**, por su valiosa asesoría en el uso de la línea de producción de tarjetas de circuitos impresos del Laboratorio de Detectores.

A mis compañeros de tesis **César, Javier y Sócrates** por compartir sus conocimientos y muchas experiencias conmigo, pero en especial por brindarme sus consejos y amistad a lo largo de este importante proceso. Así como a mis

compañeros del Laboratorio de Detectores **Nelly, Greta, Héctor, Mario, Samuel, Juan Carlos, Miguel y Luis**, por brindarme su amistad.

A mis compañeros y amigos de generación por todos esos momentos compartidos, esas largas noches de estudio y proyectos realizados a lo largo de toda la carrera: **Pepas, Javi, Matado, Mamut, Coco, Valera, Cabañas, Cuchillo, Bob, Ángel, Armando, Andrés, Piochas, Hippie, Marco, Preciado, Bambi, Huicho, Gato, Trajas, Tony, Fabio, Polo, Choco, Diego, Edher, Melvin, Richi, Botitas, Franco, Job, Fercho, Emilio, Sandra, Nayelli**. A **German Daniel**, el tío, por su amistad y consejos brindados.

A los amigos que me han acompañado a lo largo de toda la carrera: **Ana Laura, Carmen, Giovanna, Aranda, Edmundo**; mis carnales del CCH **Christian, Sebastian, Marco Antonio; Eduardo Martin** mi gran amigo de secundaria y por su amistad brindada; **Cenobia Jannet** por esos grandes momentos, por su apoyo, amistad y cariño brindado, lucha siempre por tus sueños. Así como a todas las personas que han influido de todas las formas posibles en mi vida personal y profesional.

Adrián De la Cruz Martínez

A toda mi familia, porque siempre me han querido y me han apoyado, agradezco por todos los momentos que hemos compartido.

*A la familia **Palma Ferrer**, porque me enseñaron a ser una persona sencilla y a valerme por mí mismo, gracias por ser siempre una segunda familia para mí.*

*A la familia **Aranda**, gracias siempre por los buenos deseos, por la ayuda que me han dado, por las palabras de aliento y por la amistad que he encontrado en ustedes.*

*A mis compañeros de Tesis: **Adrián de la Cruz, Javier Rojas, Sócrates Guerrero**, gracias por compartir la responsabilidad de trabajo, por acompañarme en las noches de desvelo y de mucha presión y finalmente por su amistad de la cual he aprendido bastante.*

*A **Evelyn Aranda**, por su contribución en las fotografías y recomendaciones que me ha dado para la publicación del trabajo, y por todo el apoyo que me dio en los momentos de duda y desesperación durante el desarrollo del proyecto.*

*A mis amigos: **Evelyn, Jesús, Josué, Migue, MiguelBob, Rosario, Verónica**, gracias por su valiosa amistad, por la confianza y el apoyo que he encontrado en ustedes a lo largo de mi vida, por los consejos y por todos los momentos divertidos que hemos vivido.*

*A las personas que conocí en el laboratorio de Detectores: **Aquiles, Claudia, Christian, Enrique, Héctor, Iván, Juan Carlos, Leonardo**.*

*A mis todos mis amigos de la facultad de Ingeniería, gracias por todos esos momentos divertidos que pasamos juntos, porque sin ustedes mi carrera en la universidad hubiera sido muy aburrida, por las fiestas y por su amistad: **Alejo, Andrés, Ángel, Armas, Armando, Bab, Bandita, Becas, Belem, Cabañas, Camet, Choco, Coco, Cuchi, Diego, Eder, Emilio, Fabio, Franco, Gabriela, Hippie, Javi, Juansal, Karla, Manzana, Marco, Matado, Max, Nayeli, Nómada, Óscar, Osvaldo, Piochas, Polo, Preciado, Richie, Sandra, Tío, Tony, Valera, Yisus**.*

*A las personas que contribuyeron en el desarrollo de la plataforma Pixqui y ayudaron a mi formación como ingeniero: **Ángel, Asahel, Carlos, Fernando, Frederic, Greta, Gustavo, Jorge, Juan Carlos, Lauro, Mario, Nelly, Óscar, Tony, Samuel**.*

A **Gustavo Medina**, por la confianza depositada en mí, para desarrollar mi trabajo en el extranjero, la oportunidad de conocer nuevos horizontes y por los consejos que me ha brindado.

A **Lauro Santiago**, por sus enseñanzas en mi formación profesional, pero principalmente por su amistad y la guía que ha brindado en la realización de este trabajo.

A las personas de la colaboración Internacional de JEM-EUSO con los que tuve la oportunidad de interactuar y aprender mucho de ellos y de su cultura.

A mi Alma mater, la **Universidad Nacional Autónoma de México**, por darme la oportunidad de prepararme profesionalmente como Ingeniero, dentro de la **Facultad de Ingeniería**, así como a todos mis profesores en la carrera, ya que gracias a esta preparación tuve el honor de representar a mi universidad, por medio del desarrollo de este trabajo.

Por mi raza hablará el espíritu.

César Alan López López, Noviembre 2013.

A **Dios** por siempre estar aquí... “*Y me alegro también de las debilidades, los insultos, las necesidades, las persecuciones y las dificultades que sufro por Cristo, porque cuando más débil me siento es cuando más fuerte soy*”. 2 Corintios 12: 10.

A la **Universidad Nacional Autónoma de México**, mi **Alma mater**, por darme la oportunidad de realizar una carrera y por todos los beneficios que de ella recibí.

A la **Facultad de Ingeniería** y a todos mis profesores, gracias por su conocimiento transmitido.

A los **Institutos de Ciencias Nucleares y de Ingeniería** de la **UNAM**, al laboratorio de **Astro Particule and Cosmology** de París, Francia, y al **Istituto Nazionale di Fisica Nucleare** de Nápoles, Italia, por darme las facilidades para realizar este trabajo.

Al **Dr. Gustavo Medina Tanco**, por todo el apoyo que me ha brindado durante mi estancia en los diferentes laboratorios e institutos donde he trabajado, por los consejos personales y profesionales, por la oportunidad de acercarme al área de tecnología espacial y sobre todo por la confianza que ha puesto en mí.

Al **M.I. Lauro Santiago Cruz**, por los conocimientos y consejos que me ha transmitido dentro y fuera del aula, por la confianza que ha depositado en mí, por su paciencia y su amistad.

Al **Dr. Frederic Trillaud Pighi**, por las pláticas y consejos, que me han ayudado en mi formación académica y profesional.

Al **M. en C. Enrique Patiño**, por todas las enseñanzas y conocimiento que me ha transmitido, por los cursos de la línea de producción, eres una parte muy importante en mi formación profesional.

To **Giuseppe, Guillaume, Valentina, Aera, Philippe and his wife, Pierre, Claudio**, thank you for your advice, your support and your friendship. Grazie! Mercie! 감사합니다.

A mis amigos del laboratorio de detectores, **Héctor Silva, Isaí, César Tavera, Héctor Valera, Miguel (Bob), Juan Carlos, Miguel Ángel, Samuel, Mario, Claudia** e **Iván** por todo el conocimiento y la amistad que me brindaron. **Aglae**, gracias por las pláticas que hemos tenido, por los consejos, las oraciones y por la música compartida.

A **César, Adrián** y **Sócrates**, gracias por estos años de trabajo, de desvelo, de enojo, de convivencia y de amistad. Lo hemos logrado.

A **Nestor, Omar, Fidel** y **Héctor Cuchillo**, por su amistad, por sus consejos, por los ratos de convivencia.

A **mis tíos** y **primos**, por su apoyo incondicional a lo largo de la carrera.

A **quienes** caminaron a mi lado en algún momento.

Javier Rojas Garcés

*A la **Universidad Nacional Autónoma de México** y a sus profesores, por la formación, conocimiento y experiencia que me han brindado.*

*Al **Instituto de Ciencias Nucleares**, por brindarme la oportunidad de desarrollar el proyecto en sus instalaciones.*

*Al **Doctor Gustavo Adolfo Medina Tanco**, por permitirme formar parte de este proyecto y por su valioso apoyo para la consecución del mismo, además de su liderazgo y visión como representante de la misión JEM EUSO en México.*

*Al **Maestro Lauro Santiago Cruz**, nuestro Director de Tesis, por su dirección y valiosos consejos durante el desarrollo del proyecto y del trabajo de Tesis, los cuales enriquecieron este trabajo, y por manifestar su completo apoyo para la culminación del mismo.*

*A mis compañeros de Tesis: **Adrián, César, Javier**, por acompañarme durante el desarrollo del proyecto, por compartir conmigo esta importante etapa en mi desarrollo profesional y personal, y en especial por su amistad y por las experiencias que hemos compartido juntos.*

*A mis compañeros y amigos de generación, por acompañarme durante mi trayecto estudiantil: **Adrián, Alejo, Ángel, Armando, Armas, Arturo, Bandita, Belem, Bob, Botas, Camet, Cabañas, Caballero, Charlie, Choco, Chuchín, Coco, Cuchillo, Diego, Ether, Fabio, Franco, Hippie, Ismael, Javi, Jocho, JuanSal, Karla, Mamut, Marco, Max, Nayeli, Nomada, Nestor, Nophal, Omar, Oscar, Piochas, Pipope, Polo, Preciado, Richie, Sandra, Terrazas, Tío, Tony, Valera.***

*A los amigos que conocí en mi estancia en el Instituto de Ciencias Nucleares: **Claudia, Greta, Héctor, Iván, Juan Carlos, Luis, Mario, Miguel, Nelly, Samuel.***

Por último, a todas las personas que alguna vez cruzaron su camino con el mío y dejaron huella.

Sócrates Alberto Silvarán Guerrero

DEDICATORIAS

Este trabajo de tesis está dedicado:

A mis padres **Ramón y Sonia**, por brindarme todo lo necesario para hacer posible mi formación académica, así como su apoyo, cariño y comprensión a lo largo de todo este tiempo; gracias a ustedes este sueño se ha vuelto realidad, gracias por guiarme siempre y hacer de mí una mejor persona.

A mis hermanos **Ramón y Gisela**, por brindarme su cariño y apoyo, ustedes fueron una parte muy importante para terminar este proceso.

A mis sobrinos **Omar y Fernando**, esperando que esta tesis les sirva de ejemplo para que algún día puedan vivir este sueño.

A mis abuelitos **Enrique y Ofelia**, que aunque no estén físicamente con nosotros me han cuidado y guiado desde el cielo con mucho cariño, te he cumplido abuelita.

A mis abuelitos **Faustino y María** por el cariño que me han brindado.

A la familia **De la Cruz Castillo**, mis tíos **Alejandro, Martha, Enrique, Guadalupe, Rosario, Martín** que se encuentran en estos momentos en el cielo cuidando de todos. A mis padrinos **Adriana y Arturo**. A la familia **Martínez Castañeda, Guadalupe, Teresa, Gonzalo, Jesús**.

Adrián De la Cruz Martínez

*A mis padres **Margarita y Arnulfo**, a mi hermana **Elizabeth**, gracias por todos los valores que me han enseñado, por todo el apoyo, la comprensión y el amor que me brindaron durante toda mi vida para llegar a donde estoy ahora, los quiero mucho.*

*A **Evelyn Aranda**, por todo tu amor y confianza, por permitirme ser parte de tu vida, por todos esos momentos que hemos compartido juntos.*

*A **Jesús Palma**, por enseñarme a valorar mi lugar en la universidad, y por tu tenacidad en el estudio, la cual me inspiró a ser un mejor estudiante.*

*A las personas que me vieron empezar mi formación profesional pero por desgracia ya no me acompañan en este momento. En especial a mis abuelos **Raymundo López y Emilia López**, los quiero mucho donde quiera que se encuentren, así como al Sr. **Pedro Aranda**, quien tuve la oportunidad de conocer y fue buena persona conmigo.*

César Alan López López, Noviembre 2013.

A mis padres, Sílvia y Javier.

Gracias por amor, sus consejos, su comprensión y su apoyo incondicional. Hemos caminado juntos, lo hemos logrado y vamos a seguir adelante.

A mi hermana Ayde.

Sigue esforzándote día a día para lograr tus metas personales y profesionales.

A mis abuelos, Estela (+), Joaquín (+), Aurora y Miguel (+)

Gracias por regalarme a mis padres y por brindarme todo su apoyo.

A Diana García.

Gracias por tu amistad, confianza y por caminar juntos.

A Liz.

Por ser parte de mi inspiración para salir adelante.

Javier Rojas Garcés

*A mis padres **Benigno y Leonor**, y a mi hermano **Erik**, quienes me han apoyado de manera incondicional en todo mi trayecto estudiantil y por ser el pilar más importante de mi vida. Agradezco su cariño, comprensión, y todo el esfuerzo y la dedicación que han puesto para sacar adelante a nuestra familia.*

*A la familia **Pineda García**, en especial a mis tíos **José y Martha**, por haberme acogido como un miembro más de su familia, por ser una familia sin igual y por enseñarme que la vida te regala personas únicas que debemos cuidar por siempre.*

Sócrates Alberto Silvarán Guerrero

Contenido

| | |
|---|--------------|
| Índice de figuras | xviii |
| Índice de tablas | xxvii |
| Prólogo | xxx |
| | |
| 1. Introducción | 1 |
| 1.1. Misión JEM-EUSO | 1 |
| 1.2. Misión EUSO-BALLOON | 2 |
| 1.3. Proyecto AEMB-F1 “Pixqui” | 4 |
| 1.4. Alcance del proyecto | 4 |
| | |
| 2. Antecedentes del proyecto | 7 |
| 2.1. Circuitos electrónicos | 7 |
| 2.1.1. Analógicos | 7 |
| 2.1.2. Digitales | 11 |
| 2.2. Sistemas de monitoreo y adquisición de datos | 13 |
| 2.2.1. Acondicionamiento de señales | 15 |
| 2.2.2. Sensores de temperatura | 21 |
| 2.2.3. Conversión analógica-digital | 25 |
| 2.3. Sistemas embebidos | 27 |
| 2.4. Microcontroladores | 28 |
| 2.4.1. Arquitectura de microcontroladores | 30 |
| 2.4.2. Familias de microcontroladores | 31 |
| 2.4.3. Microcontroladores AVR | 32 |
| 2.4.4. Software de desarrollo | 33 |
| 2.5. Comunicación serie | 37 |
| 2.5.1. Comunicación asíncrona | 39 |

| | |
|--|------------|
| 2.5.2. Comunicación síncrona | 44 |
| 2.5.3. Detección de errores en la transmisión de datos | 47 |
| 2.6. Almacenamiento de datos | 48 |
| 2.6.1. Memorias ROM..... | 49 |
| 3. Diseño y desarrollo..... | 50 |
| 3.1. Filosofía de operación | 51 |
| 3.2. Requerimientos y especificaciones..... | 51 |
| 3.2.1. Condiciones de operación..... | 53 |
| 3.2.2. Interfaces de comunicación externas..... | 53 |
| 3.2.3. Requerimientos de potencia y montaje | 59 |
| 3.2.4. Metodología empleada..... | 59 |
| 3.3. Propuesta y diseño del primer prototipo del Housekeeping..... | 60 |
| 3.3.1. Selección del dispositivo principal de control..... | 60 |
| 3.3.2. Diseño de las interfaces de hardware | 64 |
| 3.3.3. Simulación de interfaces externas | 74 |
| 3.3.4. Pruebas preliminares | 80 |
| 3.3.5. Diseño de software | 88 |
| 3.3.6. Desarrollo de prototipos | 95 |
| 3.4. Bancos de pruebas de subsistemas | 115 |
| 3.4.1. Banco de pruebas para CPU y SIREN | 115 |
| 3.4.2. Banco de pruebas SPI..... | 123 |
| 3.4.3. Banco de pruebas de LVPS..... | 135 |
| 3.5. Versión de vuelo del Housekeeping..... | 139 |
| 3.5.1. Medidas de protección contra fallas | 139 |
| 3.5.2. Diseño del software final | 140 |
| 3.5.3. Ensamble del modelo de vuelo | 170 |
| 4. Evaluación y puesta a punto | 173 |
| 4.1. Validación del microcontrolador en condiciones de vacío | 173 |

| | |
|---|------------|
| 4.2. Pruebas al primer prototipo del Housekeeping | 179 |
| 4.2.1. Pruebas de interfaces externas | 179 |
| 4.2.2. Pruebas de vacío y ciclado térmico | 184 |
| 4.3. Pruebas a la versión de vuelo del Housekeeping | 191 |
| 4.3.1. Integración con bancos de prueba | 191 |
| 4.3.2. Integración con las LVPS | 192 |
| 4.3.3. Condiciones de vacío y ciclado térmico | 195 |
| 5. Integraciones internacionales y vuelo con NASA..... | 201 |
| 5.1. Integración del Housekeeping en Data Processor..... | 201 |
| 5.1.1. Primera integración | 201 |
| 5.1.2. Segunda integración | 206 |
| 5.2. Integración entre DP y PDM | 212 |
| 5.3. Vuelo en plataforma de NASA | 215 |
| 5.3.1. Desarrollo de la plataforma suborbital Pixqui | 215 |
| 5.3.2. Pruebas de compatibilidad electromagnética | 218 |
| 5.3.3. Vuelo en globo estratosférico | 221 |
| 5.3.4. Resultados | 224 |
| 6. Resultados y conclusiones..... | 231 |
| 6.1. Resultados logrados | 231 |
| 6.2. Conclusiones | 232 |
| 6.3. Trabajo a futuro | 232 |
| 6.4. Recomendaciones | 233 |
| Apéndices | 237 |
| A. Glosario de términos..... | 238 |
| B. Tabla de CRC..... | 240 |
| C. Constancia de participación en congreso SOMI 2013 | 241 |
| Bibliografía y referencias | 242 |

Índice de figuras

Capítulo 1

| | |
|--|---|
| Figura 1.1. JEM-EUSO en la estación espacial internacional ISS | 2 |
| Figura 1.2. Simulación de globo CNES en la estratósfera | 3 |

Capítulo 2

| | |
|---|----|
| Figura 2.1. Representación de un amplificador operacional | 8 |
| Figura 2.2. Circuito de un amplificador operacional inversor | 9 |
| Figura 2.3. Configuración de un amplificador no inversor | 10 |
| Figura 2.4. Configuración de seguidor de voltaje | 11 |
| Figura 2.5. Simbología y tablas de verdad para las compuertas lógicas | 12 |
| Figura 2.6. Estructura de un circuito decodificador | 12 |
| Figura 2.7. Estructura de un multiplexor | 13 |
| Figura 2.8. Sistema de adquisición de datos | 14 |
| Figura 2.9. Bandas de trabajo de un filtro paso bajas | 16 |
| Figura 2.10. Filtros según su respuesta en frecuencia | 17 |
| Figura 2.11. Amplificador diferencial | 18 |
| Figura 2.12. Esquemático de un amplificador de instrumentación | 19 |
| Figura 2.13. Mejora al amplificador de instrumentación | 20 |
| Figura 2.14. Termopar típico | 22 |
| Figura 2.15. Sensor de temperatura RTD | 22 |
| Figura 2.16. Sensor de temperatura LM35 | 22 |
| Figura 2.17. Termistor típico | 23 |
| Figura 2.18. Curva característica de un termistor | 24 |
| Figura 2.19. Diagrama de bloques básico de un microcontrolador | 29 |
| Figura 2.20. Arquitectura <i>Von Neumann</i> | 30 |
| Figura 2.21. Arquitectura <i>Harvard</i> | 31 |

| | |
|---|----|
| Figura 2.22. Entorno de desarrollo Atmel Studio | 36 |
| Figura 2.23. Conexión tipo <i>Simplex</i> | 37 |
| Figura 2.24. Conexión tipo <i>Full-Duplex</i> | 37 |
| Figura 2.25. Conexión tipo <i>Half-Duplex</i> | 37 |
| Figura 2.26. Topología Punto a Punto | 38 |
| Figura 2.27. Topología <i>Multi-Drop</i> | 38 |
| Figura 2.28. Topología Multi-Punto | 38 |
| Figura 2.29. Protocolo de comunicación serie | 39 |
| Figura 2.30. Interfaz RS-232 | 40 |
| Figura 2.31. Interfaz RS-422 y LVDS | 42 |
| Figura 2.32. Velocidad de transmisión vs longitud del conductor | 43 |
| Figura 2.33. Bus SPI | 45 |
| Figura 2.34. Protocolo SPI | 45 |
| Figura 2.35. Modos de SPI | 46 |

Capítulo 3

| | |
|--|----|
| Figura 3.1. Góndola de la misión EUSO-BALLOON | 50 |
| Figura 3.2. Estructura donde se integran los subsistemas del DP | 52 |
| Figura 3.3. Gabinetes para cada subsistema del DP | 52 |
| Figura 3.4. Diagrama general del DP | 54 |
| Figura 3.5. Arquitectura del microcontrolador | 62 |
| Figura 3.6. Diagrama interno del microcontrolador | 63 |
| Figura 3.7. <i>Housekeeping</i> y sus interfaces externas | 64 |
| Figura 3.8. Propuesta de la comunicación con el estándar RS-422 | 65 |
| Figura 3.9. Propuesta de comunicación SPI | 66 |
| Figura 3.10. Propuesta del control de las señales <i>Chip Select</i> | 66 |
| Figura 3.11. Circuito <i>High Level Command</i> | 67 |
| Figura 3.12. Control de los <i>High Level Command</i> | 67 |
| Figura 3.13. Conexión con relevadores de las LVPS | 68 |

| | |
|--|----|
| Figura 3.14. Propuesta del monitoreo analógico | 69 |
| Figura 3.15. Propuesta para la medición de la temperatura | 69 |
| Figura 3.16. Líneas dedicadas para subsistemas | 70 |
| Figura 3.17. Sensores internos del <i>Housekeeping</i> | 71 |
| Figura 3.18. Cable trenzado blindado | 72 |
| Figura 3.19. Cable trenzado sin blindaje | 73 |
| Figura 3.20. <i>Backshells</i> para cables | 73 |
| Figura 3.21. Simulación de la interfaz RS-422 | 74 |
| Figura 3.22. Diagrama de flujo de la transmisión serie | 75 |
| Figura 3.23. Comunicación microcontrolador - sensor de temperatura | 76 |
| Figura 3.24. Comunicación microcontrolador – reloj de tiempo real | 76 |
| Figura 3.25. Comunicación SPI con dos esclavos | 77 |
| Figura 3.26. Configuración y lectura de sensor en SPI | 78 |
| Figura 3.27. Simulación del control del decodificador | 79 |
| Figura 3.28. Simulación de <i>High Level Command</i> | 79 |
| Figura 3.29. Esquema de la prueba de comunicación <i>full-duplex</i> | 81 |
| Figura 3.30. Prueba de comunicación <i>full-duplex</i> | 81 |
| Figura 3.31. Prueba de comunicación RS-422..... | 82 |
| Figura 3.32. Recepción por RS-422 | 82 |
| Figura 3.33. Transmisión por RS-422 | 82 |
| Figura 3.34. Acelerómetro SPI | 83 |
| Figura 3.35. Datos del acelerómetro | 83 |
| Figura 3.36. Esquema de comunicación SPI y RS-422 | 84 |
| Figura 3.37. Diagrama de flujo de la prueba SPI-RS-422 | 84 |
| Figura 3.38. Diagrama de la configuración de pruebas del ADC | 84 |
| Figura 3.39. Temperatura medida | 85 |
| Figura 3.40. Medición de temperatura | 85 |
| Figura 3.41. Prueba de <i>High Level Command</i> | 86 |
| Figura 3.42. Circuito <i>High Level Command</i> | 87 |

| | |
|---|-----|
| Figura 3.43. Circuito de prueba para <i>High Level Command</i> | 87 |
| Figura 3.44. Prueba de <i>High Level Command</i> | 88 |
| Figura 3.45. Proceso principal del <i>Housekeeping</i> | 89 |
| Figura 3.46. Configuración de inicio del <i>Housekeeping</i> | 90 |
| Figura 3.47. Configuración de dispositivos internos del <i>Housekeeping</i> | 90 |
| Figura 3.48. Configuración de dispositivos externos del <i>Housekeeping</i> | 90 |
| Figura 3.49. Interrupción de USART | 91 |
| Figura 3.50. Instrucción de CPU a <i>Housekeeping</i> | 92 |
| Figura 3.51. Lectura de CCB y CLKB en <i>Housekeeping</i> | 93 |
| Figura 3.52. Descripción de lectura de GPS en <i>Housekeeping</i> | 94 |
| Figura 3.53. Descripción del almacenamiento en microSD | 94 |
| Figura 3.54. Conectores D-sub | 96 |
| Figura 3.55. Señales en conectores con interfaz RS422 | 97 |
| Figura 3.56. Señales en conectores con interfaz SPI | 97 |
| Figura 3.57. Señales en conectores con interfaz SPI | 98 |
| Figura 3.58. Distribución de señales en conectores con LVPS | 98 |
| Figura 3.59. Distribución de señales en conectores con LVPS | 99 |
| Figura 3.60. Distribución de señales en la tarjeta de desarrollo | 101 |
| Figura 3.61. Distribución de interfaces en el <i>Housekeeping</i> | 102 |
| Figura 3.62. <i>Layout</i> de PCB01 | 103 |
| Figura 3.63. <i>Layout</i> de PCB02 | 103 |
| Figura 3.64. <i>Layout</i> de PCB03 | 104 |
| Figura 3.65. <i>Layout</i> de PCB04 | 104 |
| Figura 3.66. <i>Layout</i> de PCB05 | 105 |
| Figura 3.67. Modelo 3D de la PCB01 | 105 |
| Figura 3.68. Modelo 3D de la PCB02 | 105 |
| Figura 3.69. Modelo 3D de la PCB03 | 106 |
| Figura 3.70. Modelo 3D de la PCB04 | 106 |
| Figura 3.71. Modelo 3D de la PCB05 | 106 |

| | |
|--|-----|
| Figura 3.72. Devastado de la cara superior de la PCB04 | 108 |
| Figura 3.73. Platinado de la cara superior de la PCB04 | 109 |
| Figura 3.74. Cara superior de la PCB04 con máscara antisoldante | 109 |
| Figura 3.75. Vista de los <i>pads</i> con el microscopio | 110 |
| Figura 3.76. Colocación de componente | 110 |
| Figura 3.77. Tarjeta en el horno de reflujo | 111 |
| Figura 3.78. Tarjetas de <i>Housekeeping</i> | 111 |
| Figura 3.79. <i>Housekeeping</i> ensamblado | 111 |
| Figura 3.80. Medidas del primer nivel de la estructura del <i>Housekeeping</i> | 112 |
| Figura 3.81. Vistas de las PCB02 a PCB05 | 113 |
| Figura 3.82. Vista de la estructura del <i>Housekeeping</i> | 114 |
| Figura 3.83. Gabinete para el <i>Housekeeping</i> | 114 |
| Figura 3.84. Panel frontal del gabinete del <i>Housekeeping</i> | 115 |
| Figura 4.85. Banco de pruebas de CPU/SIREN | 116 |
| Figura 3.86. Señales en conectores y modelo 3D | 116 |
| Figura 3.87. Banco emulador de CPU y SIREN | 117 |
| Figura 3.88. Ejemplo del cálculo del CRC | 122 |
| Figura 3.89. Estructura del protocolo propuesto | 122 |
| Figura 3.90. Diagrama de bloques del banco de pruebas SPI | 124 |
| Figura 3.91. Cara superior del banco de pruebas SPI | 125 |
| Figura 3.92. Diseño terminado del banco de pruebas SPI | 125 |
| Figura 3.93. Diagrama de flujo del banco de pruebas SPI | 126 |
| Figura 3.94. Diagrama de flujo de interrupción por SPI | 127 |
| Figura 3.95. Diagrama de flujo de interrupción externa y de la USART | 128 |
| Figura 3.96. Interacción entre el <i>Housekeeping</i> y HVPS | 129 |
| Figura 3.97. Conexión entre los bancos de prueba SPI y el <i>Housekeeping</i> | 131 |
| Figura 3.98. Diagrama del banco de pruebas de HVPS | 132 |
| Figura 3.99. Diseño de PCB del banco de pruebas de HVPS | 132 |
| Figura 3.100. Banco de pruebas de HVPS ensamblado | 133 |

| | |
|--|-----|
| Figura 3.101. <i>Software</i> de control del banco de pruebas de HVPS | 134 |
| Figura 3.102. Diagrama del banco de pruebas de LVPS | 136 |
| Figura 3.103. Control de los DACs del banco de pruebas de LVPS | 136 |
| Figura 3.104. Control de los DACs | 137 |
| Figura 3.105. Vista top del banco de pruebas de LVPS | 138 |
| Figura 3.106 Banco de pruebas de LVPS ensamblado | 138 |
| Figura 3.107. Diagrama general de la conexión SPI | 140 |
| Figura 3.108. Librería serial del <i>Housekeeping</i> | 141 |
| Figura 3.109. Librería para calcular el CRC | 142 |
| Figura 3.110. Librería de retardos del <i>Housekeeping</i> | 143 |
| Figura 3.111. Librería para la adquisición analógica del <i>Housekeeping</i> | 143 |
| Figura 3.112. Librería del monitoreo analógico por sistema | 144 |
| Figura 3.113. Librería general de la comunicación SPI | 145 |
| Figura 3.114. Librería para interactuar con el sistema HVPS | 145 |
| Figura 3.115. Librería para almacenamiento y generación alarmas de SPI | 146 |
| Figura 3.116. Librería para almacenamiento en la <i>microSD</i> | 146 |
| Figura 3.117. Librería para control de relevadores de las LVPS | 147 |
| Figura 3.118. Configuración inicial en el <i>Housekeeping</i> | 147 |
| Figura 3.119. Diagrama de almacenamiento en EEPROM del <i>Housekeeping</i> | 149 |
| Figura 3.120. Almacenamiento en EEPROM del <i>Housekeeping</i> | 150 |
| Figura 3.121. Librería para el control del <i>Watchdog</i> en el <i>Housekeeping</i> | 150 |
| Figura 3.122. Captura de la interfaz de comunicación | 160 |
| Figura 3.123. Diagrama de flujo de la interfaz de <i>software</i> | 161 |
| Figura 3.124. Interfaz de depuración del <i>Housekeeping</i> | 163 |
| Figura 3.125. Diagrama de flujo de la operación del <i>Housekeeping</i> | 164 |
| Figura 3.126. Función de configuración de inicio | 165 |
| Figura 3.127. Función de configuración de HVPS | 165 |
| Figura 3.128. Monitoreo fase A | 167 |
| Figura 3.129. Monitoreo fase B | 168 |

| | |
|--|-----|
| Figura 3.130. Control del <i>buffer</i> dinámico para recibir tramas | 169 |
| Figura 3.131. Gabinete para montar el <i>Housekeeping</i> | 170 |
| Figura 3.132. Panel frontal del gabinete del <i>Housekeeping</i> | 171 |
| Figura 3.133. <i>Housekeeping</i> sobre la placa inferior del gabinete | 171 |
| Figura 3.134. <i>Housekeeping</i> dentro del gabinete | 172 |
| Figura 3.135. <i>Housekeeping</i> dentro del gabinete | 172 |

Capítulo 4

| | |
|--|-----|
| Figura 4.1. Circuito en <i>protoboard</i> con los dispositivos de prueba | 174 |
| Figura 4.2. Circuito dentro de la cámara de vacío | 175 |
| Figura 4.3. Pruebas a la primera versión del <i>Housekeeping</i> | 180 |
| Figura 4.4. Diagrama de la prueba para la interfaz RS422 en el <i>Housekeeping</i> | 180 |
| Figura 4.5. Trama enviada por el <i>Housekeeping</i> | 181 |
| Figura 4.6. Esquema de pruebas para <i>High Level Command</i> | 181 |
| Figura 4.7. Pruebas de interfaz SPI LVDS del <i>Housekeeping</i> | 182 |
| Figura 4.8. Archivos generados en la memoria SD | 183 |
| Figura 4.9. Datos obtenidos del subsistema LENSES | 183 |
| Figura 4.10. Cámara de vacío del ICN | 184 |
| Figura 4.11. Perfil de temperatura aplicado para las pruebas | 186 |
| Figura 4.12. <i>Housekeeping</i> sobre la plancha de la cámara de vacío | 187 |
| Figura 4.13. Temperatura en el microcontrolador | 187 |
| Figura 4.14. Temperatura en el regulador de voltaje | 188 |
| Figura 4.15. Temperatura en la superficie de la PCB05..... | 188 |
| Figura 4.16. Temperatura en uno de los soportes del <i>Housekeeping</i> | 189 |
| Figura 4.17. Temperatura en el gabinete sobrepuesto | 189 |
| Figura 4.18. Temperatura en el plato de la cámara de vacío | 190 |
| Figura 4.19. Temperatura de la campana de vacío | 190 |
| Figura 4.20. Conexión entre <i>Housekeeping</i> y los bancos de pruebas | 191 |
| Figura 4.21. <i>Housekeeping</i> y LVPS dentro del <i>rack</i> | 193 |

| | |
|--|-----|
| Figura 4.22. Conexión entre el <i>Housekeeping</i> y las LVPS | 193 |
| Figura 4.23. <i>Housekeeping</i> , módulos LVPS y los bancos de prueba | 194 |
| Figura 4.24. Presión de 3 mbar durante la prueba | 195 |
| Figura 4.25. <i>Housekeeping</i> y demás módulos en la cámara de vacío | 196 |
| Figura 4.26. Temperatura en el microcontrolador | 197 |
| Figura 4.27. Temperatura en el regulador de la tarjeta de desarrollo | 197 |
| Figura 4.28. Temperatura en el gabinete del <i>Housekeeping</i> | 198 |
| Figura 4.29. Temperatura en el interior del <i>rack</i> | 199 |
| Figura 4.30. Temperatura en un punto fuera del <i>rack</i> | 199 |

Capítulo 5

| | |
|--|-----|
| Figura 5.1. Conexión entre <i>Housekeeping</i> , CPU y LVPS | 202 |
| Figura 5.2. Integración del <i>Housekeeping</i> en el <i>rack</i> | 203 |
| Figura 5.3. <i>Housekeeping</i> y LVPS montadas en <i>rack</i> de DP | 207 |
| Figura 5.4. Subsistemas durante la segunda integración | 209 |
| Figura 5.5. Vista superior de la góndola | 209 |
| Figura 5.6. <i>Rack</i> del DP completo | 212 |
| Figura 5.7. Conexión entre PDMB y <i>Housekeeping</i> | 213 |
| Figura 5.8. Conexión entre HVPS y el <i>Housekeeping</i> | 214 |
| Figura 5.9. Estructura de Pixqui | 216 |
| Figura 5.10. Tarjetas montadas en la estructura de Pixqui | 216 |
| Figura 5.11. Prueba de alimentación | 217 |
| Figura 5.12. Prueba de vacío a los gabinetes de Pixqui | 218 |
| Figura 5.13. Antenas de la cámara semi-anecoica | 220 |
| Figura 5.14. Pixqui dentro de la cámara | 220 |
| Figura 5.15. Gráfica resultante de las prueba de EMC | 220 |
| Figura 5.16. Góndola donde fue montado Pixqui | 221 |
| Figura 5.17. Lanzamiento del globo de NASA | 222 |
| Figura 5.18. Recorrido del globo en el vuelo | 222 |

| | |
|---|-----|
| Figura 5.19. Temperatura en el microcontrolador | 224 |
| Figura 5.20. Temperatura del regulador de voltaje | 225 |
| Figura 5.21. Temperatura en la caja 2 del sistema Pixqui | 226 |
| Figura 5.22. Temperatura en el simulador de SPI | 227 |
| Figura 5.23. Temperatura de la PCB01 | 228 |
| Figura 5.24. Simulación de temperaturas máximas esperadas | 229 |
| Figura 5.25. Telemetría cortesía de la NASA | 229 |

Índice de tablas

Capítulo 1

| | |
|--|---|
| Tabla 1.1. Cronograma del desarrollo del <i>Housekeeping</i> | 6 |
|--|---|

Capítulo 2

| | |
|--|----|
| Tabla 2.1. Asignación de terminales en los conectores para RS-232..... | 41 |
|--|----|

Capítulo 3

| | |
|---|-----|
| Tabla 3.1. Comparación de microcontroladores..... | 61 |
| Tabla 3.2. Comparación de precios entre los diferentes tipos de cable | 73 |
| Tabla 3.3. Comparativa de precios para componentes..... | 95 |
| Tabla 3.4. Uso de los puertos de E/S del microcontrolador | 99 |
| Tabla 3.5. Uso de canales analógicos | 100 |
| Tabla 3.6. Uso de los módulos <i>timer</i> | 100 |
| Tabla 3.7. Uso de los módulos de comunicación | 100 |
| Tabla 3.8. Uso de memoria <i>flash</i> y EEPROM..... | 100 |
| Tabla 3.9. Especificaciones de la línea producción LPKF | 107 |
| Tabla 3.10. Comandos entre CPU y <i>Housekeeping</i> | 118 |
| Tabla 3.11. Registro de alarmas | 119 |
| Tabla 3.12. Subsistemas que generan alarmas | 119 |
| Tabla 3.13. Alarmas válidas por subsistema | 120 |
| Tabla 3.14. Tipos de tramas para la comunicación con CPU | 151 |
| Tabla 3.15. Trama para encender el subsistema CLKB..... | 152 |

| | |
|---|-----|
| Tabla 3.16. Tareas que el <i>Housekeeping</i> puede realizar | 152 |
| Tabla 3.17. Lectura de la temperatura 1 del <i>Housekeeping</i> | 153 |
| Tabla 3.18. Parámetros que pueden ser leídos por subsistema | 154 |
| Tabla 3.19. Envío de la lectura de la temperatura 1 del <i>Housekeeping</i> | 154 |
| Tabla 3.20. Configuración del umbral alto con el valor 0x03FFh | 154 |
| Tabla 3.21. Configuración del umbral bajo con el valor 0x0080h | 155 |
| Tabla 3.22. Parámetros de configuración del <i>Housekeeping</i> | 155 |
| Tabla 3.23. Trama para el envío de alarmas a CPU | 156 |
| Tabla 3.24. Enmascaramiento de alarmas de baja prioridad | 157 |
| Tabla 3.25. Trama ejemplo de alarma de alta prioridad | 158 |

Capítulo 4

| | |
|--|-----|
| Tabla 4.1. Mediciones realizadas a temperatura y presión ambiente | 176 |
| Tabla 4.2. Mediciones en vacío con temperatura cercana a 45 °C | 176 |
| Tabla 4.3. Mediciones en vacío con temperatura cercana a -36 °C | 177 |
| Tabla 4.4. Mediciones en alto vacío con temperatura cercana a 21 °C | 177 |
| Tabla 4.5. Mediciones en alto vacío con temperatura cercana a -37 °C | 178 |
| Tabla 4.6. Potencia consumida por el circuito de prueba | 178 |
| Tabla 4.7. Consumos del <i>Housekeeping</i> | 194 |

Capítulo 5

| | |
|---|-----|
| Tabla 5.1. Consumo registrado por la LVPS- <i>Housekeeping</i> | 202 |
| Tabla 5.2. Resultados de la primera integración en el DP con el <i>Housekeeping</i> | 205 |
| Tabla 5.3. Mejoras alcanzadas con los bancos de prueba | 206 |

| | |
|--|-----|
| Tabla 5.4. Resultados de la segunda integración en el DP del <i>Housekeeping</i> | 211 |
| Tabla 5.5. Estándares para EMC | 219 |
| Tabla 5.6. Características del estándar | 219 |
| Tabla 5.7. Distribución de los sensores de temperatura | 223 |

Prólogo

Este trabajo de tesis surge por la participación de México dentro del Consorcio JEM-EUSO (*Extreme Universe Space Observatory on board the Japanese Experiment Module*), que tiene como objetivo el diseño de un telescopio espacial que será usado para detectar partículas de altas energías en la atmósfera. Dentro de la misión, México tiene la responsabilidad de diseñar y construir un sistema de suma importancia para el correcto funcionamiento del telescopio. Este sistema se denomina *Housekeeping*, y está encargado de monitorear el correcto funcionamiento de la electrónica principal del instrumento.

Como un primer acercamiento al instrumento de JEM-EUSO, toda la electrónica y el sistema de detección de partículas del instrumento se redujeron a un sistema mínimo, con el fin de validarlo en condiciones de operación representativas, en estratósfera, acorde a las condiciones a las que operará el instrumento final. Dentro de este sistema al que se redujo el instrumento, se encuentra incluido el *Housekeeping*, del que se abordará en este trabajo de tesis.

La presente tesis se estructura en 6 capítulos, los cuales están organizados de la siguiente manera:

En el primer capítulo se hace una breve explicación de las misiones en las cuales *Housekeeping* está involucrado, hablando de sus características, así como los objetivos que busca cada una de éstas. También se muestra las etapas de diseño del *Housekeeping* y al finalizar este capítulo se presenta un cronograma para el desarrollo de éste.

En el segundo capítulo se abordan conceptos teóricos que contribuyen a entender y justificar la forma en que se diseñó el *Housekeeping*. Se aborda el tema de electrónica analógica para dar entrada al amplificador operacional, posteriormente se proporciona una definición de electrónica digital para explicar brevemente el funcionamiento de los circuitos lógicos combinacionales básicos. En el mismo capítulo se aborda un tema de suma importancia que son los sistemas de monitoreo y la adquisición de datos, al darse los conceptos básicos para la adquisición de señales. Finalmente se habla muy brevemente de los sistemas embebidos para dar entrada a los microcontroladores, para terminar con el tema de comunicación serie.

En el tercer capítulo se presentan los requerimientos de la misión, tanto mecánicos como eléctricos, se explica el funcionamiento y las tareas generales de todos los subsistemas del instrumento, así como la interacción directa de estos con el *Housekeeping* y las condiciones a las que el *Housekeeping* estará en operación. En este capítulo se hace una distinción entre un

primer prototipo de *Housekeeping* y una versión de vuelo del *Housekeeping*. Para el primer prototipo del *Housekeeping* se describe detalladamente el diseño de las interfaces externas en *hardware* y las pruebas que se realizaron para validarlas. También se describe el diseño de software para realizar tareas básicas, que permitan a *Housekeeping* interactuar con el exterior. Al terminar esta parte, se hace mención a la selección del material para el cableado, así como la propuesta de los conectores externos, y la administración de recursos del microcontrolador para realizar las tareas requeridas. Para terminar, se presenta el proceso que se llevó a cabo para el diseño de las PCB que constituyen al *Housekeeping*, el montaje de éste en un gabinete para ser integrado con los subsistemas y el diseño de bancos de prueba de los subsistemas externos para validar las interfaces externas. Para la versión de vuelo se describen algunas medidas que se llevaron a cabo, para mitigar algunos casos que pueden provocar que *Housekeeping* quede fuera de operación. Se aborda una descripción detallada sobre el desarrollo de la versión final del *software* y finalmente las mejoras que se realizaron para un montaje de mejor calidad que el del primer prototipo.

En capítulo 4 se describen las pruebas que fueron necesarias para validar los dispositivos con los que fue construido el *Housekeeping*. Estas pruebas se dividen en tres apartados:

- Pruebas para validar el microcontrolador en condiciones de vacío
- Pruebas al primer prototipo del *Housekeeping*.
- Pruebas a la versión de vuelo del *Housekeeping*

En el primer apartado se presentan las pruebas que se llevaron a cabo con el microcontrolador que se utilizó como núcleo en el desarrollo de *Housekeeping*. Se mencionan las condiciones de vacío a las cuales fue sometido el microcontrolador y el resultado de éstas.

En el segundo apartado de este capítulo se describe las pruebas y los resultados a los que se llegaron, al someter al primer prototipo del *Housekeeping* a condiciones de vacío similares a las condiciones de operación de éste en vuelo.

El tercer y último apartado se llevan a cabo pruebas con la versión de vuelo, en éstas se hacen integraciones con bancos de prueba que fueron diseñados y construidos en el Laboratorio de Detectores del ICN, con las LVPS diseñadas en México, para caracterizarlas.

En el capítulo 5 se presentarán los resultados que se han obtenido con el *Housekeeping*, en las integraciones internacionales realizadas en Europa. Este capítulo está dividido en 4 apartados:

En el primer apartado se muestran los resultados que se lograron en la primera integración en Nápoles, Italia, los cuales dieron lugar a mejoras y cambios en el diseño final de *Housekeeping* en esta etapa, principalmente en *Hardware*. En el segundo apartado se muestran los

resultados de una segunda integración realizada en Nápoles, Italia. En ésta se definió una gran parte de la filosofía de operación del *Housekeeping*, se probaron las interfaces con los demás subsistemas y se desarrolló una versión final de *firmware* que será entregado junto con la versión de vuelo. En el tercer apartado se desarrolla la última participación de Housekeeping en París, Francia, con la integración de todos los subsistemas de la misión. Como un cuarto apartado de este capítulo se describe y se muestran los resultados, de las primeras pruebas del *Housekeeping* en un vuelo en globo estratosférico de la NASA (*National Aeronautics and Space Administration*).

En el capítulo 6 se presentan conclusiones generales sobre los objetivos y alcances del proyecto, además se hacen las sugerencias necesarias para mejorar el diseño del *Housekeeping* para versiones futuras de éste.

Finalmente se presenta la bibliografía consultada, así como un glosario de términos que se presentan a lo largo del desarrollo de este trabajo.

1. Introducción

En este capítulo mencionaremos las misiones para las cuales se está desarrollando el subsistema *Housekeeping*, así como el alcance científico de éstas y el aporte tecnológico que implicará el desarrollo del subsistema para el país. Las misiones son las siguientes:

1. La misión JEM-EUSO [1][2] consiste en un telescopio que detectará un gran número de partículas de energías extremas provenientes del espacio exterior.
2. EUSO-BALLOON [3][4][5] (*Extreme Universe Space Observatory on Balloon*), es un proyecto piloto de la misión JEM-EUSO y consiste en una versión reducida del telescopio, que será montado en un globo estratosférico.
3. Pixqui, que en lengua náhuatl significa guardián, consiste en una primera prueba de vuelo que servirá para someter al *Housekeeping* a las condiciones de operación reales durante su operación en EUSO-BALLOON.

Finalmente se hará la definición del alcance del proyecto.

1.1. Misión JEM-EUSO

JEM-EUSO es un nuevo tipo de observatorio espacial que utilizará la atmósfera de la tierra como un detector. Observará fenómenos luminosos momentáneos, que tienen lugar en la atmósfera de la tierra causados por las partículas procedentes del espacio. El sensor colocado en el telescopio [6] [7] es de un amplio espectro que puede detectar partículas de extrema energía por encima de 3×10^{19} eV.

Los rayos cósmicos que impactan la tierra colisionan con los núcleos de las moléculas de la atmósfera y generan numerosos electrones, mesones y rayos gamma. Las partículas secundarias producen una nueva generación de partículas, a través de su paso en la atmósfera. Al rastro completo que deja a su paso se le llama “cascada atmosférica”. Un evento cósmico de alta energía de 10^{20} eV genera 100 billones de partículas, que caen al suelo en un radio de 3 km. Un electrón en una cascada atmosférica excita las moléculas de nitrógeno N_2 en la atmósfera, y éstas instantáneamente emiten numerosos fotones de fluorescencia ultravioleta a lo largo de su trayectoria. JEM-EUSO capturará esta luz remotamente, y tomará imágenes del rastro de la cascada durante varios microsegundos, como una cámara digital de velocidad extremadamente alta. El incremento y decaimiento de la intensidad de luz a lo largo de una cascada descendiente registra la energía y la dirección de llegada del rayo cósmico.

JEM-EUSO será montado en la Estación Espacial Internacional (ISS), a una altitud de aproximadamente 400 kilómetros. Este telescopio orbitará alrededor de la Tierra cada 90 minutos. En la figura 1.1 se observa una imagen de la ISS con JEM-EUSO.

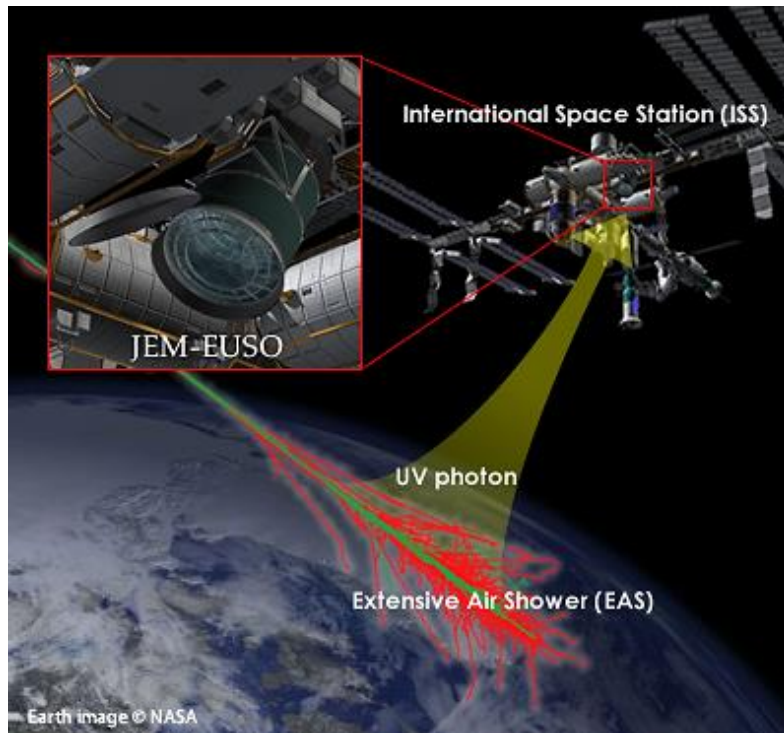


Figura 1.1. JEM-EUSO en la estación espacial internacional ISS.

El interés por el estudio de estas partículas de alta energía ha logrado la coordinación de diversas instituciones y organismos internacionales, las cuales tienen como objetivo implementar proyectos en donde se desarrolla y aplica tecnología para lograr el estudio de dichas partículas.

El consorcio JEM-EUSO está liderado por Japón y a su vez integrado por agencias espaciales de diversas partes del mundo, entre las cuales se encuentra JAXA (Japón), ESA (Europa), ROSCOSMOS (Rusia), NASA (USA) entre otras; además cuenta con alrededor de 280 investigadores de 13 países, entre los que se encuentra México, en donde participa la UNAM, a través de los Institutos de Ciencias Nucleares (ICN) y de Ingeniería (II).

1.2. Misión EUSO-BALLOON

EUSO-BALLOON es un proyecto de colaboración entre el consorcio internacional JEM-EUSO y CNES (*Centre National d'Etudes Spatiales*) de la agencia espacial francesa, que pretende desarrollar un prototipo de telescopio de fluorescencia a bordo de un globo estratosférico de CNES. Este globo se planea sobrevolar en varias campañas de lanzamientos durante 2014.

El objetivo de este tipo de misión es poner a prueba la validez de los conceptos y decisiones técnicas que se toman hoy en día para la misión espacial JEM-EUSO, prevista para 2017, o cualquier otro proyecto posterior encaminado a observar fenómenos luminiscentes inducidos por rayos cósmicos energéticos desde el espacio.

El diseño y la construcción de este telescopio es un verdadero reto tecnológico, ya que implica el uso de nuevas tecnologías desarrolladas en laboratorios tanto de investigación como del área industrial, en campos tan diversos como, la óptica de alta precisión utilizando lentes de Fresnel, técnicas de foto detección altamente sensitivas con alta resolución, e integración de electrónica analógica y digital muy innovadoras.

Los objetivos principales del proyecto EUSO-BALLOON son tres:

1. Probar todas las tecnologías desarrolladas para JEM-EUSO en condiciones muy severas (estratósfera), parcialmente representativas de condiciones que tendría que cumplir cualquier telescopio espacial de luz fluorescente, ya sea acomodado en la ISS o en vuelo libre dentro de un satélite.
2. Medir la intensidad de la luminiscencia atmosférica con una resolución angular apropiada para JEM-EUSO.
3. Poner a prueba la capacidad de un instrumento para detectar lluvias de rayos cósmicos desde el espacio (por encima del nivel de fondo medido a partir de la luminiscencia atmosférica).

En la figura 1.2 se puede ver una captura de una simulación del globo de CNES.

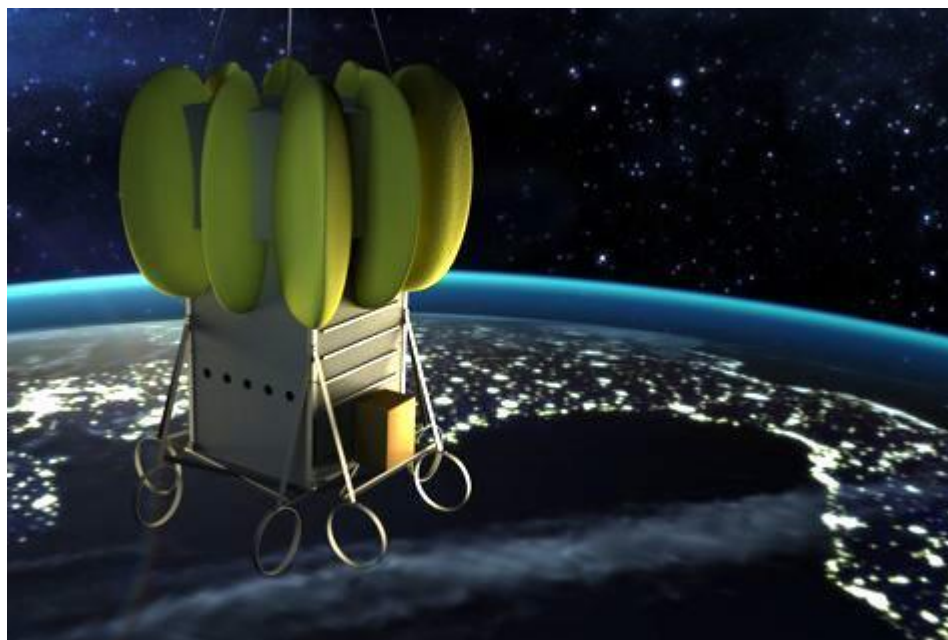


Figura 1.2. Simulación del globo de CNES en la estratósfera.

1.3. Proyecto AEMB-F1 “Pixqui”

Como parte de las pruebas a las que será sometido el *Housekeeping*, la AEM (Agencia Espacial Mexicana) logró un acuerdo de participación con la agencia espacial estadounidense NASA, la cual colabora en el consorcio JEM-EUSO. El acuerdo permitirá probar los prototipos para JEM-EUSO fabricados en México, en una serie de vuelos con globos estratosféricos de la NASA. Estos vuelos se realizarán a una altura cercana a los 30 km sobre la tierra, en condiciones muy parecidas a las del espacio exterior. Para el primer vuelo, México participaría con una carga de prueba en la góndola de la NASA, donde estaría incluido el *Housekeeping*.

En México el desarrollo de esta carga se denominó AEMB-F1, siglas que provienen de (AEM *balloon - flight one*) para posteriormente ser renombrado bajo el nombre de “Pixqui”. Esta carga tiene 2 objetivos principales:

- Someter el prototipo de vuelo del *Housekeeping* a condiciones reales de operación, en comparación a las simuladas en laboratorio, y validar con esto, el correcto funcionamiento del *Housekeeping*, en un vuelo previo al agendado por CNES en 2014.
- Adaptar el prototipo *Housekeeping* para que sirva como base en el desarrollo de una plataforma de usos múltiples para vuelos suborbitales, que beneficiaría la validación de desarrollos nacionales, en vuelos a costos accesibles para el desarrollo en tecnología espacial mexicana.

El desarrollo de Pixqui fue hecho por la UNAM, en concreto por el ICN e II, en colaboración con la Facultad de Ingeniería (FI).

1.4. Alcance del proyecto

Para el desarrollo de la misión EUSO-BALLOON, la gestión del proyecto estará a cargo de la agencia espacial francesa CNES. El proyecto está estructurado para llevarse a cabo a través de integraciones supervisadas por cada sistema alojado en el instrumento. Cada subsistema es desarrollado por las instituciones responsables, utilizando para ello sus propios recursos de diseño, y deberán cumplir con los tiempos señalados por la agencia CNES. En cada subsistema han sido consideradas tres etapas de diseño:

- La etapa de diseño fase A [8] ha sido considerada principalmente conceptual. Está enfocada al proyecto origen de esta misión, JEM-EUSO. En esta etapa se muestra la funcionalidad del sistema, el análisis e investigación de los objetivos científicos de

estudio, además se establecieron las posibles soluciones en el desarrollo del proyecto para justificar la viabilidad de su construcción. Esto se considera el mínimo nivel de diseño necesario para avanzar a la siguiente fase. Esta fase terminó a mediados de 2011, pero los requerimientos para la siguiente fase estuvieron disponibles a partir de septiembre de 2012.

- La etapa de diseño fase B1 se basa en los requisitos generados por la fase A. Consiste en una selección de los componentes principales del sistema y la demostración por análisis y comprobación empírica de que el rendimiento es el esperado. Se establecen las especificaciones de masa, volumen y potencia. Además se establecen las interfaces con otros subsistemas y las estimaciones de posibles costos. La fase de salida del diseño B1 incluye una profunda revisión de los requisitos y la especificación técnica para una próxima etapa del diseño. La primera integración supervisada por CNES, sería en noviembre de 2012, con la presentación del primer prototipo.
- La etapa de diseño fase B2 se basa en los resultados generados de la fase B1, donde todos los implicados en el diseño han integrado por primera vez sus dispositivos de fase B1, y han acordado un plan de desarrollo afín, para lograr la meta de construcción del primer prototipo en conjunto del instrumento final, operando de forma autónoma. El resultado de la fase B2 servirá como base fundamental en el desarrollo de la misión JEM-EUSO. La segunda integración tomaría lugar en Junio 2013, con la entrega de la versión definitiva de vuelo para EUSO-BALLOON.

Cabe resaltar que el equipo encargado del desarrollo del *Housekeeping*, comenzó su trabajo al principio de la fase B1.

La plataforma Pixqui se desarrolla como consecuencia de los prototipos presentados en la fase de desarrollo B2, con una adaptación para funcionar en el vuelo de la NASA, pero validará los prototipos probados hasta ese momento. El desarrollo de Pixqui comienza en Junio 2013, con fecha de vuelo para el 15 de Agosto 2013.

La última puesta a prueba del instrumento EUSO-BALLOON está programada para septiembre de 2013, integrando todos los sistemas de la carga útil. En la tabla 1.1 se muestra el cronograma del desarrollo del *Housekeeping*.

Tabla 1.1. Cronograma del desarrollo del *Housekeeping*.

| Housekeeping cronograma de desarrollo | | | | | | | | | | | | | | |
|---|------|----|----|----|------|----|----|----|----|----|----|----|----|----|
| Desarrollo | 2012 | | | | 2013 | | | | | | | | | |
| | 09 | 10 | 11 | 12 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 |
| Requerimientos prototipo B1 | █ | | | | | | | | | | | | | |
| Pruebas en protoboard | █ | | | | | | | | | | | | | |
| Diseño de PCB's prototipo B1 | █ | | | | | | | | | | | | | |
| Fabricación prototipo B1 | | █ | | | | | | | | | | | | |
| Ensamble prototipo B1 | | █ | | | | | | | | | | | | |
| Pruebas de vacío y ciclado térmico | | █ | | | | | | | | | | | | |
| Ensamble mecánico B1 | | █ | | | | | | | | | | | | |
| Firmware B1 | | █ | | | | | | | | | | | | |
| Integración B1 <i>Housekeeping-DP</i> | | | █ | | | | | | | | | | | |
| Mejoras de diseño en prototipo B2 | | | | █ | █ | | | | | | | | | |
| Diseño de bancos de prueba B2 | | | | | | █ | | | | | | | | |
| Fabricación prototipos B2 | | | | | | | █ | | | | | | | |
| Fabricación bancos de prueba B2 | | | | | | | | █ | | | | | | |
| Ensamble prototipos B2 | | | | | | | | | █ | | | | | |
| Ensamble bancos de prueba B2 | | | | | | | | | | █ | | | | |
| Ensamble mecánico B2 | | | | | | | | | | | █ | | | |
| Pruebas de vacío y ciclado térmico | | | | | | | | | | | | █ | | |
| Firmware B2 | | | | | █ | █ | █ | █ | █ | █ | █ | █ | █ | █ |
| Integración B2 <i>Housekeeping-LVPS</i> | | | | | | | | | | | █ | | | |
| Integración B2 <i>Housekeeping-DP</i> | | | | | | | | | | | | █ | | |
| Firmware B2 <i>Integración Housekeeping-PDM</i> | | | | | | | | | | | | | █ | █ |
| Integración B2 <i>Housekeeping-PDM</i> | | | | | | | | | | | | | | █ |
| Diseño conceptual y requerimientos Pixqui | | | | | | | | | | █ | | | | |
| Adaptaciones prototipo B2 a Pixqui | | | | | | | | | | | █ | | | |
| Adaptaciones bancos de prueba B2 a Pixqui | | | | | | | | | | | | █ | | |
| Firmware B2 Pixqui | | | | | | | | | | | | █ | █ | |
| Integración B2 <i>Housekeeping-Pixqui</i> | | | | | | | | | | | | | █ | |
| Ensamble mecánico Pixqui | | | | | | | | | | | | | █ | |
| Prueba EMC Pixqui | | | | | | | | | | | | | █ | |
| Pruebas de vacío y ciclado térmico Pixqui | | | | | | | | | | | | | █ | |
| Vuelo Pixqui-NASA | | | | | | | | | | | | | | █ |

2. Antecedentes del proyecto

Este capítulo es un compendio de los antecedentes teóricos requeridos para el desarrollo del proyecto. Se dará un breve repaso de conceptos de circuitos electrónicos y dispositivos de almacenamiento, además se hará énfasis en los temas de microcontroladores, adquisición de datos y en los protocolos e interfaces de comunicación serie.

2.1. Circuitos electrónicos

La electrónica es la rama de la física, y especialización de la ingeniería, que estudia y emplea sistemas cuyo funcionamiento se basa en la conducción y el control del flujo de los electrones u otras partículas cargadas eléctricamente en diferentes materiales y elementos, como los semiconductores.

La ingeniería electrónica estudia el uso de la energía eléctrica para transmitir, recibir y procesar información, siendo esta la base de la ingeniería en telecomunicaciones, la ingeniería informática y la ingeniería de control.

El campo de la electrónica juega un papel clave en muchas ramas de la ingeniería, ya que todas las disciplinas de ingeniería ocupan la electrónica como herramienta para desarrollar nuevos avances en su campo. Como resultados de estos desarrollos se tienen avances significativos tales como comunicaciones telefónicas y de datos, cómputo de alta velocidad, radio, visión artificial, sistemas de control, investigación espacial, procesamiento de señales, entretenimiento.

Un circuito es una red eléctrica de interconexión de dos o más componentes, tales como resistencias, inductores, condensadores, fuentes, interruptores y semiconductores, en el que existe al menos una trayectoria cerrada. Un circuito que tiene componentes electrónicos es denominado un circuito electrónico. Estos últimos pueden ser de carácter analógico, digital o ambos.

2.1.1. Analógicos

La electrónica analógica es una parte de la electrónica que estudia los sistemas en los cuales sus variables, voltaje y corriente varían de una forma continua en el tiempo, pudiendo tomar infinitos valores.

Amplificadores operacionales

Uno de los dispositivos más versátiles y de mayor uso es el amplificador operacional. El término operacional originalmente se refería al hecho de que se usaba en circuitos para sumar, restar, multiplicar e incluso resolver ecuaciones diferenciales.

En la actualidad el amplificador operacional está inmerso en aplicaciones diversas tales como el acondicionamiento de señales, el control de procesos, las comunicaciones, las fuentes de potencia y señales. En la figura 2.1 se muestra la representación de un amplificador operacional.

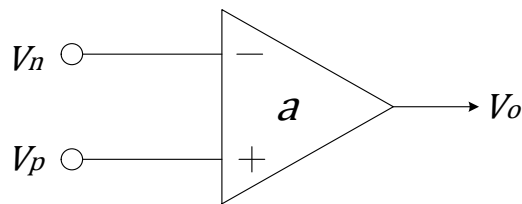


Figura 2.1. Representación de un amplificador operacional.

La función principal del amplificador operacional es producir una salida que es proporcional a la diferencia de voltajes entre sus dos entradas, de manera ideal se tiene lo siguiente:

$$V_o = a(V_p - V_n) = a\Delta V \quad \text{Ecuación 1}$$

Donde V_p y V_n son las entradas de voltaje no inversora e inversora respectivamente, y a es la ganancia.

Las principales características del amplificador operacional son:

- Impedancia de entrada muy elevada.
- Impedancia de salida muy baja.
- Ganancia en lazo abierto muy elevada, del orden de 100,000 o mayor.
- Voltajes y corrientes de polarización muy pequeños.
- Relación de rechazo en modo común CMRR (*Common Mode Rejection Ratio*) elevada.
- *Slew rate*. Es la relación entre la variación de la tensión de salida máxima respecto de la variación del tiempo, se mide en V/ μ s.
- Producto Ganancia - Ancho de Banda (GBP). Se define como el producto de la ganancia por el ancho de banda de una frecuencia específica, en lazo abierto. A ganancia unitaria, GBP es igual al ancho de banda disponible del amplificador.

Configuraciones básicas

En el amplificador operacional, con comportamiento controlado, se usa con realimentación negativa. Con esta conexión se obtienen muchas ventajas, ya que la ganancia del circuito ya no depende de la ganancia de lazo abierto, porque ahora tiene una ganancia de lazo cerrado. Esta ganancia de lazo cerrado depende sólo de los componentes externos que se conecten. Sus tres configuraciones básicas son:

- Amplificador operacional inversor
- Amplificador operacional no inversor
- Amplificador operacional seguidor de voltaje

Amplificador operacional inversor. Este circuito se caracteriza porque la señal de salida se encuentra desfasada 180° con respecto a la señal de entrada. Como se muestra en la figura 2.2, se conectan dos resistencias externas al amplificador, la señal de entrada se introduce por la terminal inversora y la terminal no inversora se conecta a tierra.

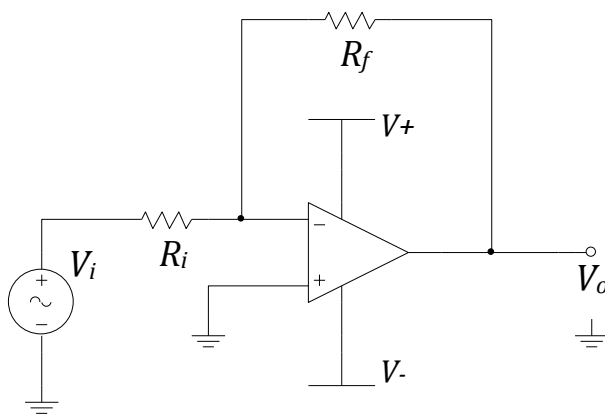


Figura 2.2. Circuito de un amplificador operacional inversor.

La amplitud del voltaje de salida de este circuito está dada por una ganancia de lazo cerrado que depende de la relación entre las dos resistencias.

El voltaje de salida está dado por la siguiente expresión:

$$V_o = -\frac{R_f}{R_i} V_i \quad \text{Ecuación 2}$$

Dónde:

V_o , es el voltaje en la salida

V_i , es el voltaje en la entrada

R_f , es la resistencia de realimentación

R_i , es la resistencia en la entrada

La ganancia en lazo cerrado de este circuito es:

$$A_{CL} = -\frac{R_f}{R_i} \quad \text{Ecuación 3}$$

Donde A_{CL} es la ganancia en lazo cerrado.

Amplificador operacional no inversor. En este circuito la señal de salida se encuentra en fase con respecto a la señal de entrada. Como se muestra en la figura 2.3, la señal de entrada es introducida por la terminal no inversora, la terminal inversora se conecta a tierra.

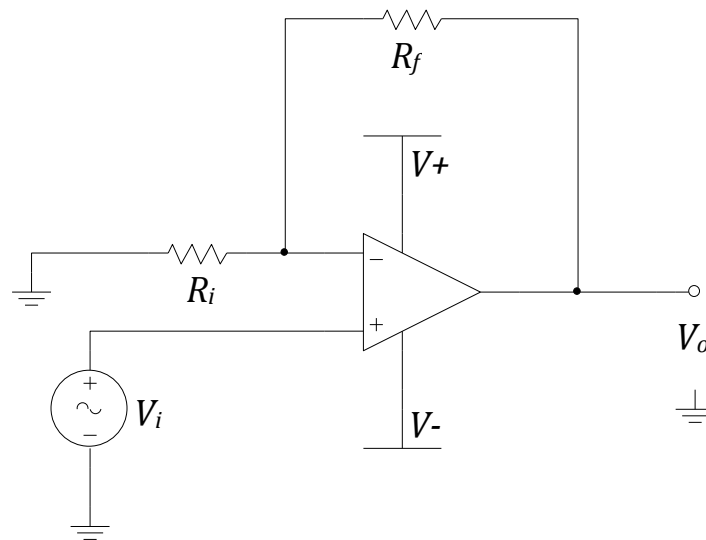


Figura 2.3. Configuración de un amplificador no inversor.

El voltaje en la salida en esta configuración está dado por la siguiente ecuación:

$$V_o = \left(\frac{R_f}{R_i} + 1\right) V_i \quad \text{Ecuación 4}$$

Dónde:

V_o , es el voltaje en la salida

V_i , es el voltaje en la entrada

R_f , es la resistencia de realimentación

R_i , es la resistencia en la entrada

La ganancia en lazo cerrado de este circuito es:

$$A_{CL} = \frac{R_f}{R_i} + 1 \quad \text{Ecuación 5}$$

Seguidor de Voltaje. Este circuito se caracteriza por tener una ganancia unitaria. El voltaje en la entrada es igual al voltaje en la salida, tanto en magnitud como en signo. En la figura 2.4 se muestra el diagrama de este circuito:

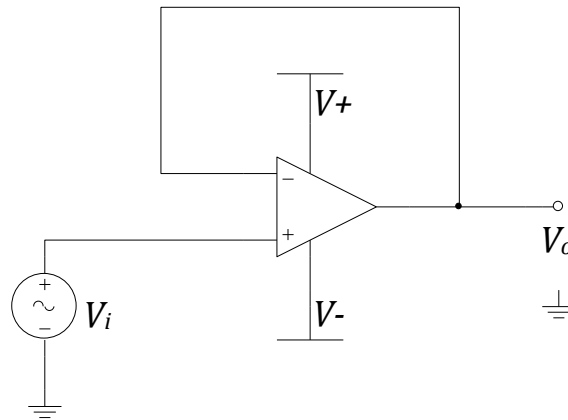


Figura 2.4. Configuración del seguidor de voltaje.

2.1.2. Digitales

En la electrónica digital las variables sólo pueden tomar valores discretos, teniendo siempre un estado perfectamente definido para una representación binaria, con valores de “1” y “0” lógicos.

Compuertas lógicas

Los dispositivos digitales más elementales se conocen con el nombre de compuertas lógicas. Éstas deben originalmente su nombre a su capacidad de permitir o retardar (como lo haría una compuerta) el flujo de la información digital. En general, una compuerta tiene una o más entradas y produce una salida que es función de los valores de entrada.

Las compuertas lógicas más básicas son: AND, OR y NOT, es fácil implementar cualquier clase de función lógica con ese tipo de compuertas. Los símbolos gráficos y las tablas de verdad de estas compuertas básicas se muestran en la figura 2.5.

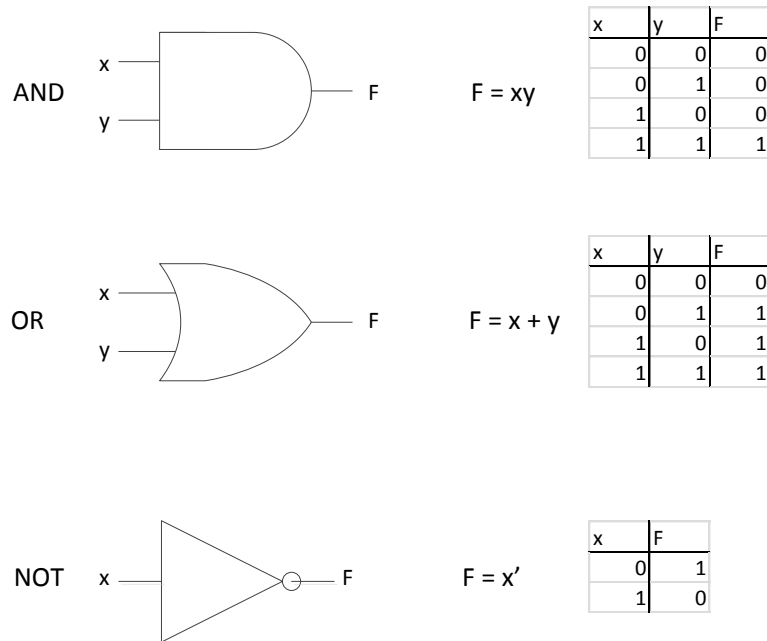


Figura 2.5. Simbología y tablas de verdad para las compuertas lógicas.

Decodificadores

Un decodificador es un circuito lógico con varias entradas y salidas que convierte las entradas codificadas en salidas codificadas, donde los códigos de entrada y salida son diferentes. La estructura general de un circuito decodificador se ilustra en la figura 2.6.

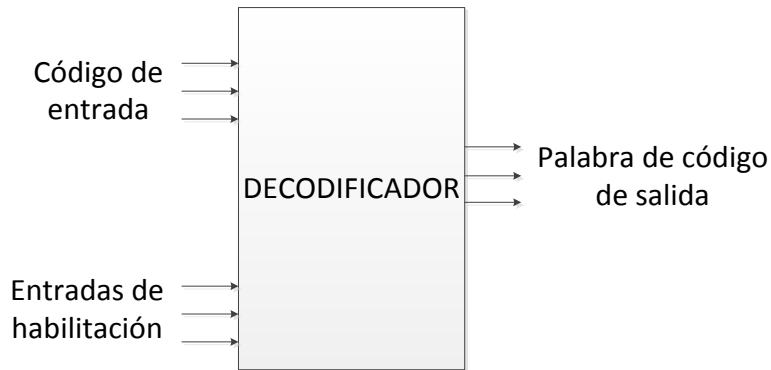


Figura 2.6. Estructura de un circuito decodificador.

Las entradas de habilitación, si se hallan presentes, deben ser afirmadas para que el decodificador efectúe su función de mapeo normal. El código de entrada que se utiliza con mayor frecuencia es un código binario de n bits. Por otro lado, el código de salida que se utiliza con mayor frecuencia es uno que contiene m bits, donde un bit está activo en todo momento. De este modo, por ejemplo, en un código de 4 dígitos con salidas activas altas, las palabras del código son 0001, 0010, 0100, 1000.

Multiplexores analógicos y digitales

Un multiplexor es un interruptor que conecta datos de n fuentes disponibles de entrada hasta su(s) salida(s). La figura 2.7 muestra las entradas y salidas de un multiplexor. Cuenta con entradas que seleccionan las señales de las fuentes disponibles, además de contar con una entrada de habilitación.

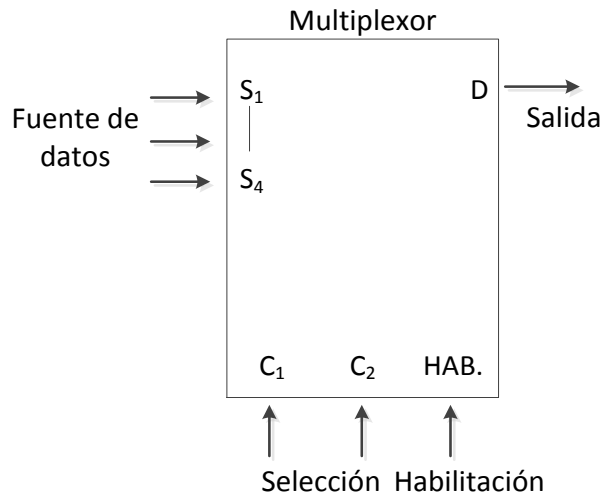


Figura 2.7. Estructura de un multiplexor.

Estos dispositivos son especialmente útiles cuando en alguna aplicación deben ser conmutadas múltiples fuentes de señales hacia un destino. Una aplicación común en computadoras es el multiplexor entre los registros de un microprocesador y su unidad aritmética lógica (ALU).

Los multiplexores digitales son dispositivos unidireccionales, la información fluye solamente desde las entradas hacia la salida. Por otro lado, también existen multiplexores analógicos, los cuales poseen la ventaja de ser bidireccionales, lo que permite que cada terminal del multiplexor pueda ser usada como entrada o como salida.

2.2. Sistemas de monitoreo y adquisición de datos

Un sistema es la combinación de dos o más elementos, subconjuntos y partes necesarias para realizar una o varias funciones. Tales funciones, dentro de un sistema de monitoreo y adquisición de datos, son la asignación de un número o valor a una propiedad o cualidad de algún objeto o evento que se requiere describir.

Un sistema de monitoreo y adquisición de datos está definido como una combinación de *hardware* y *software* que permite adquirir, procesar y representar de manera simple la información. Esta información posteriormente puede ser analizada y almacenada para los

finés específicos que se requieran. En la figura 2.8 se muestra una estructura general de un sistema de adquisición de datos.

Las características de los bloques que componen el sistema de adquisición de datos son:

- **Entrada de señales (analógicas, digitales).** Estas señales generalmente provienen de sensores analógicos o digitales, dependiendo de su origen se establecerá el tratamiento que se le dé a la señal.
- **Acondicionamiento.** En esta etapa se adapta la señal para ser procesada posteriormente.
- **Multiplexor.** Dependiendo de la cantidad de señales con las que se esté trabajando, es posible que sea necesario hacer uso de un multiplexor para poder trabajar con un número elevado de señales.
- **Conversión A/D.** Para procesar señales analógicas será necesario el uso de un convertidor analógico-digital (ADC), que entregue una señal adecuada en formato digital.
- **Procesamiento, presentación o almacenamiento.** En esta etapa se pueden almacenar los datos obtenidos, representar de alguna manera o hacer un procesamiento posterior.

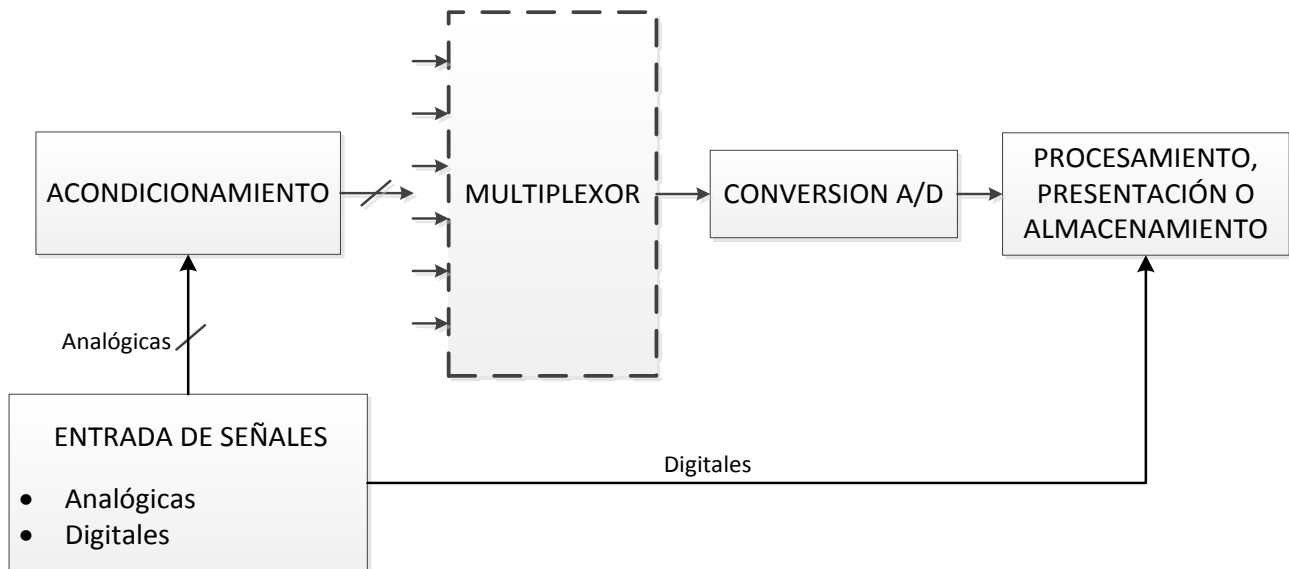


Figura 2.8. Sistema de adquisición de datos.

2.2.1. Acondicionamiento de señales

Los acondicionadores de señal son elementos que ofrecen a partir de la señal de salida de un sensor, una señal apta para ser procesada o registrada posteriormente. Consisten generalmente en circuitos electrónicos que ofrecen entre otras funciones:

- Filtrado
- Amplificación

Podemos considerar, como ejemplo, el tratamiento de una señal analógica; para su procesamiento habrá que hacer uso de un convertidor A/D. Este último tiene una impedancia de entrada definida, además de exigir que la señal aplicada sea continua, de una frecuencia limitada y que su amplitud esté dentro de ciertos límites determinados. Estas exigencias obligan a interponer un acondicionador de señal entre el sensor y el convertidor A/D.

Filtrado

Un filtro es el elemento de un sistema que discrimina una o varias señales determinadas que pasan a través de él, por medio de una selección de los componentes de frecuencias que componen estas señales. Los filtros son útiles a la entrada y salida de un sistema, debido a que ofrecen:

- Acondicionamiento de la señal de entrada al sistema
 - Eliminación de interferencias y ruido de las señales procedentes de sensores.
 - Limitación del ancho de banda al rango útil del sistema.
 - Eliminación de frecuencias superiores antes de la conversión A/D (*Anti-aliasing*).
 - Sintonización de señales útiles (P. ej. demodulación).
- Acondicionamiento de la señal de salida
 - Eliminación de armónicos innecesarios (P. ej. alisado de la salida de los conversores D/A).
 - Supresión de ruido e interferencias que halla introducido el sistema.

Los filtros se pueden clasificar atendiendo a dos conceptos distintos:

- a) El tipo de tecnología o componentes con que se fabrica el filtro.
- b) Su respuesta en frecuencia.

Considerando el primer concepto de clasificación se tiene tres tipos distintos:

1. Pasivos.
2. Activos.
3. Digitales.

Los filtros pasivos usan solamente como componentes los de tipo pasivo, es decir resistencias, capacitores e inductores.

Los filtros activos utilizan elementos que necesitan alimentación y no son considerados lineales, como los transistores, los amplificadores operacionales, los diodos, etc. Además usan componentes pasivos.

Los filtros digitales realizan la función del filtro a través de algoritmos numéricos.

En un filtro se puede hablar de 3 principales bandas de frecuencia, las cuales son:

- Banda de paso. Esta banda está definida por el intervalo de frecuencias que se desea dejar pasar.
- Banda de transición. Esta banda está definida por el intervalo de frecuencias donde la ganancia recibe una atenuación de -3 dB hasta llegar a una frecuencia donde la atenuación respecto a la ganancia es de -20 dB o mayor.
- Banda de supresión. Esta banda está definida por el intervalo de frecuencias donde se da la máxima atenuación posible.

En la figura 2.9 se muestran las bandas correspondientes a un filtro paso bajas.

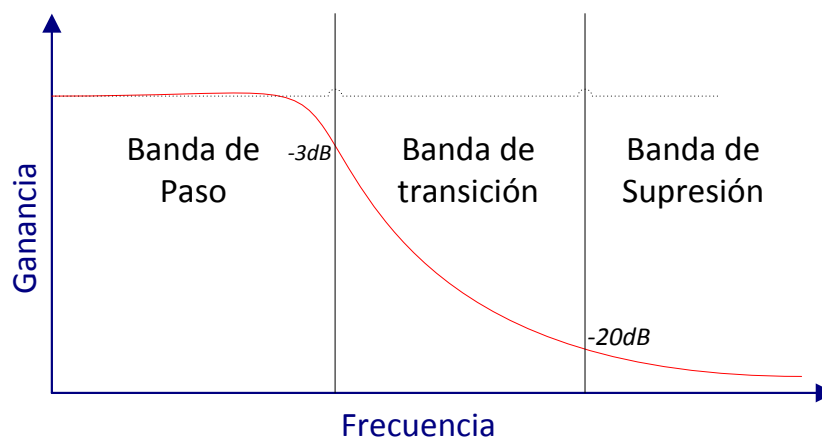


Figura 2.9. Bandas de trabajo de un filtro paso bajas.

Atendiendo al segundo concepto de clasificación los filtros también se dividen en 4 tipos distintos:

- **Paso bajas.** Este filtro se caracteriza por que su banda de paso se encuentra desde una frecuencia cero hasta una de corte, indica el fin de la banda de paso, seguida por las de transición y supresión. Suprimiendo frecuencias altas.
- **Paso altas.** A diferencia del filtro paso bajas, este comienza con la banda de supresión en bajas frecuencias, luego sigue la banda de transición hasta encontrar una frecuencia de corte que indica el inicio de la banda de paso, por lo que permite el paso de frecuencias altas.
- **Paso banda.** Este tipo de filtro tendrá 2 bandas de transición, 2 de rechazo y una de paso, permitiendo solamente pasar un intervalo seleccionado de frecuencias que van desde bajas a altas.
- **Rechazo de banda.** Al contrario del paso banda, éste cuenta con 2 bandas de paso, 2 de transición y entre éstas se encuentra una de rechazo que elimina un intervalo no deseado de frecuencias.

En la figura 2.10 se muestran los gráficos correspondientes a cada tipo de filtro expuesto anteriormente.

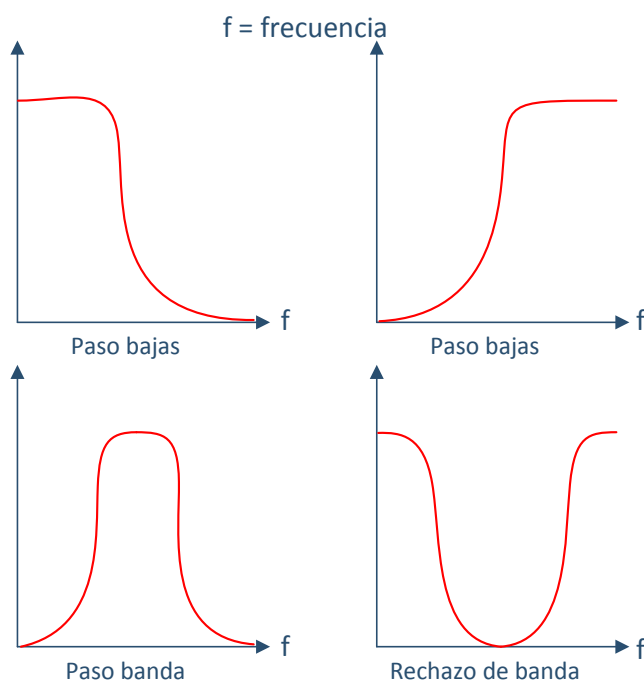


Figura 2.10. Filtros según su respuesta en frecuencia.

Amplificación

Los amplificadores son circuitos que se utilizan para aumentar el valor de la señal eléctrica de entrada, generalmente muy pequeña, y así obtener una señal a la salida con una amplitud mayor a la señal original. En estas aplicaciones los arreglos de amplificadores tipo diferencial y de instrumentación son muy utilizados, debido a sus características de ganancia y de rechazo a modo común.

Amplificador diferencial básico

La característica de un amplificador diferencial es que su salida es proporcional a la diferencia de las señales de sus dos entradas. En la figura 2.11 se muestra un amplificador diferencial básico, construido por medio de un amplificador operacional y cuatro resistencias.

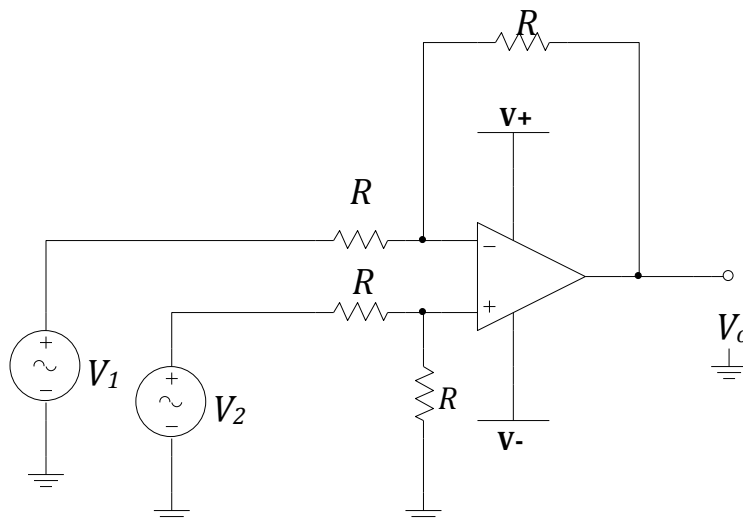


Figura 2.11. Amplificador diferencial.

El voltaje de salida de este circuito está dado por:

$$V_o = V_2 - V_1 \quad \text{Ecuación 6}$$

Dónde:

- V_o , es el voltaje de salida
- V_1 , es el voltaje que se aplica en la terminal inversora
- V_2 , es el voltaje que se aplica en la terminal no inversora

La ecuación 6 es válida sólo si el valor de R es el mismo en el arreglo de la figura 2.11.

Amplificadores de instrumentación

El amplificador de instrumentación es uno de los amplificadores más útiles, precisos y versátiles que se dispone en la actualidad. Su utilización es común en aparatos que trabajan con señales muy débiles, tales como equipos médicos para minimizar el error de medida.

El amplificador de instrumentación está diseñado para tener las siguientes características:

- Muy alta impedancia de entrada
- Muy alto rechazo en modo común CMRR
- Una ganancia estable
- Voltajes y corrientes de offset muy bajos
- Impedancia de salida baja

El amplificador de instrumentación básico se compone de tres amplificadores operacionales, los amplificadores A_1 y A_2 corresponden a la primer etapa, la salida de ésta es acoplada aun amplificador diferencial básico con ganancia unitaria. En la figura 2.12 se muestra un esquema básico del amplificador de instrumentación.

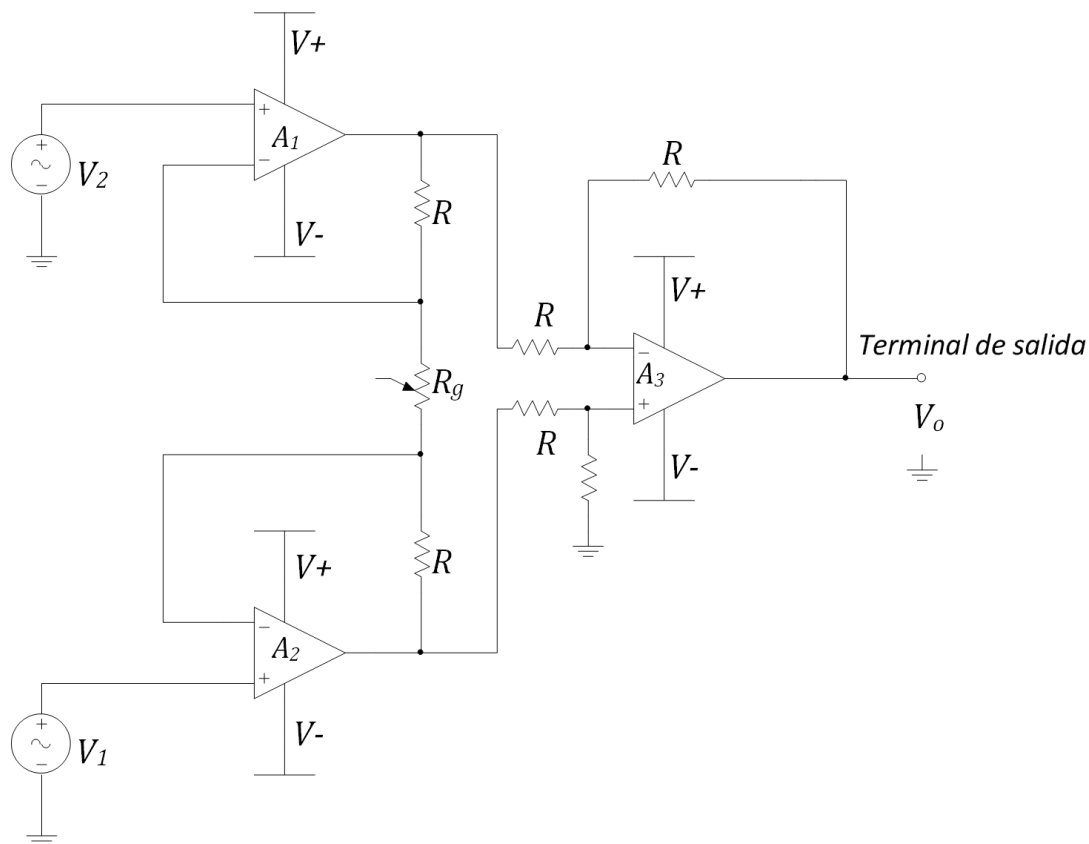


Figura 2.12. Esquemático de un amplificador de instrumentación.

Dónde:

- R_g es la resistencia que ajusta la ganancia del amplificador de instrumentación.
- V_1 es el voltaje en la entrada positiva del amplificador A_1 .
- V_2 es el voltaje en la entrada positiva del amplificador de A_2 .

La salida de este amplificador se determina con la siguiente ecuación:

$$V_0 = \left(1 + \frac{2R}{R_g}\right) (V_2 - V_1) \quad \text{Ecuación 7}$$

Para darle más versatilidad y desempeño al amplificador de instrumentación se puede abrir el lazo de realimentación negativa del amplificador operacional A_3 y agregar tres terminales, esto se muestra en la figura 2.13, como una mejora a su configuración básica. Estas tres terminales se conocen como terminal de salida (O), de detección (S) y de referencia (R).

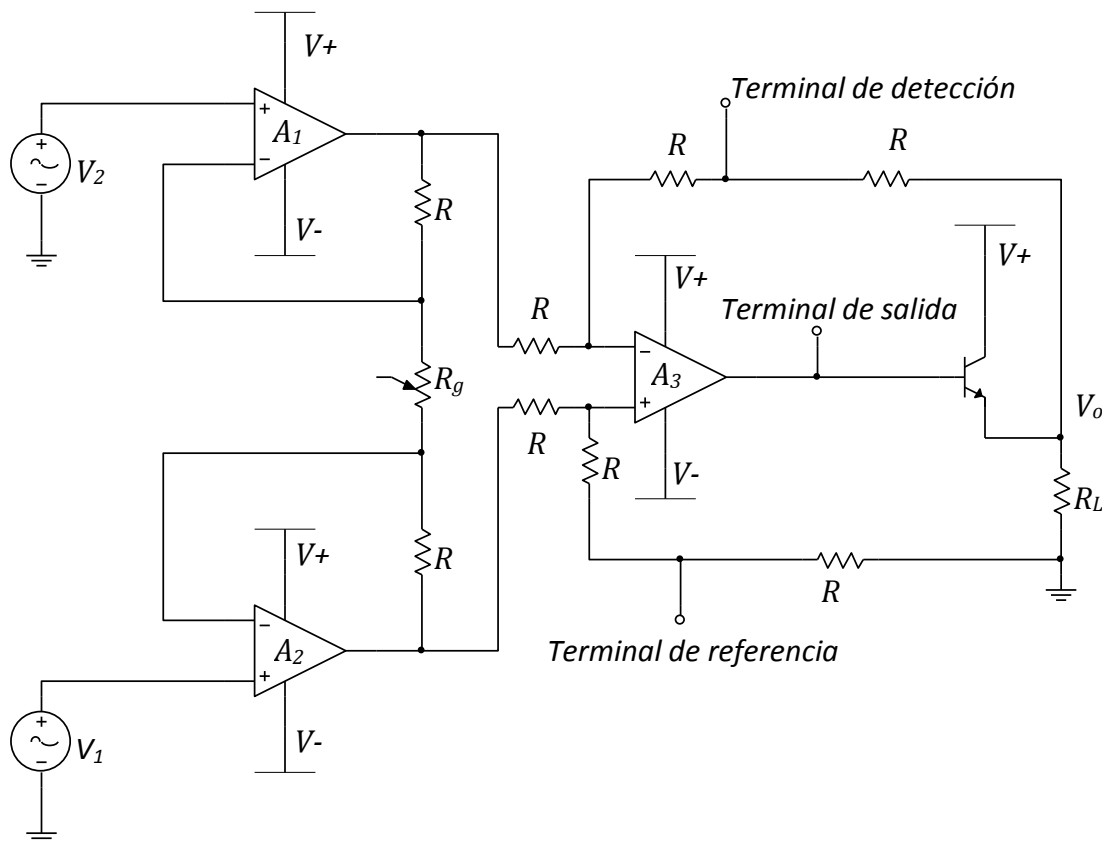


Figura 2.13. Mejora al amplificador de instrumentación.

De esta manera, si fuese necesario utilizar conexiones en las terminales muy largas o un transistor amplificador de corriente entre el amplificador de instrumentación y la carga, se producirían caídas de voltaje a través de los alambres de conexión. Para eliminar esta caída, la terminal de detección y la de referencia se conectan directamente a la carga. De esta manera la resistencia del alambre se suma por igual a las resistencias en serie con las terminales de

detección y de referencia para minimizar cualquier desigualdad. Además, al detectar el voltaje en las terminales de carga en vez de la terminal de salida del amplificador, la retroalimentación actúa para mantener constante el voltaje en la carga.

Su capacidad de lograr altas ganancias con valores de resistencias bajas, impedancia de entrada extremadamente alta y un muy alto rechazo a señales de modo común, hace que el amplificador de instrumentación sea muy utilizado en amplificación de señales muy pequeñas.

2.2.2. Sensores de temperatura

El monitoreo de temperatura tiene una importancia fundamental en numerosos procesos, así que es imprescindible disponer de mediciones de ésta lo más precisas y confiables posibles.

Se dispone de una gran cantidad de sensores para realizar la medición de temperatura, la elección del sensor correcto depende principalmente de factores tales como:

- El rango de temperaturas a medir.
- La velocidad de respuesta del sensor.
- El costo.
- La precisión que entrega el sensor.
- Requerimientos de mantenimiento.

Los sensores de temperatura que se pueden encontrar en el mercado son los siguientes: termopares, dispositivos de temperatura resistivos como los RTD (*Resistance Temperature Detector*) y los termistores, termómetros infrarrojos, dispositivos bimetálicos, dispositivos de expansión de líquidos, dispositivos de cambio de estado y sensores digitales de temperatura

Los sensores que más comúnmente se usan son los termopares, los RTD y los termistores, aunque actualmente las soluciones integradas comienzan a tener más presencia.

Termopares. Se componen esencialmente de dos tiras o hilos de metales diferentes y unidos en un extremo. Los cambios en la temperatura de esa junta provocan un cambio en la fuerza electromotriz (FEM) entre los dos extremos. Cuando la temperatura sube, por ejemplo, esta salida FEM del termopar aumenta, aunque no necesariamente de manera lineal. En la figura 2.14 se muestra un termopar típico.



Figura 2.14. Termopar típico.

RTD. El fundamento de los sensores RTD es la variación de la resistencia de un conductor con la temperatura. Los metales empleados normalmente para estos sensores son platino, cobre, níquel y molibdeno.

Los RTDs se caracterizan por un tiempo lento de respuesta y baja sensibilidad, y como requieren excitación de corriente pueden ser propensos a auto calentamiento. En la figura 2.15 se muestra un RTD.



Figura 2.15. Sensor de temperatura RTD.

LM35. Es un sensor de temperatura en circuito integrado de precisión. La salida de voltaje es linealmente proporcional a la temperatura en grados Celsius.

El LM35 tiene ventajas sobre sensores calibrados en grados Kelvin [K], esto se debe a que ya no se le tiene que restar algún voltaje a la salida para calibrarlo a grados Celsius (°C). Este no requiere calibración externa para proveer buena precisión sobre un rango de -55°C a 155 °C. En la figura 2.16 se muestra un LM35.

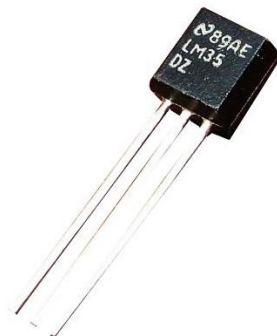


Figura 2.16. Sensor de temperatura LM35.

Termistores

Los termistores son resistores variables con la temperatura. Si su coeficiente de temperatura es negativo se denominan NTC (*Negative Temperature Coefficient*), mientras que si es positivo se denominan PTC (*Positive Temperature Coefficient*). En la figura 2.17 se muestra un termistor típico.



Figura 2.17. Termistor típico.

El fundamento de los termistores está en la dependencia de la resistencia de los semiconductores con la temperatura, debida a la variación con ésta del número de portadores. Al aumentar la temperatura lo hace también el número de portadores de carga reduciéndose la resistencia, en el caso de los que presentan coeficiente de temperatura negativo.

Las aplicaciones de los termistores se pueden dividir entre las que están basadas en un calentamiento externo del termistor, que son relativas a la medida de temperatura, y las que se basan en calentarlo mediante el propio circuito de medida para aplicaciones tales como la limitación de corriente de arranque dentro de circuitos de suministro de energía.

Para los termistores NTC, en un margen de temperaturas reducido (50°C), la dependencia se puede considerar de tipo exponencial, de la forma:

$$R_T = R_0 e^{B\left(\frac{1}{T} - \frac{1}{T_0}\right)} \quad \text{Ecuación 8}$$

Donde R_T es la resistencia a una temperatura T , R_0 es la resistencia a 25°C, u otra temperatura de referencia, T_0 es dicha temperatura expresada en Kelvin (K). El parámetro B se denomina temperatura característica del material y tiene valores de 2000 K a 5000 K. En la figura 2.18 se muestran unas curvas características que indican la variación de resistencia de un termistor NTC con respecto de la temperatura.

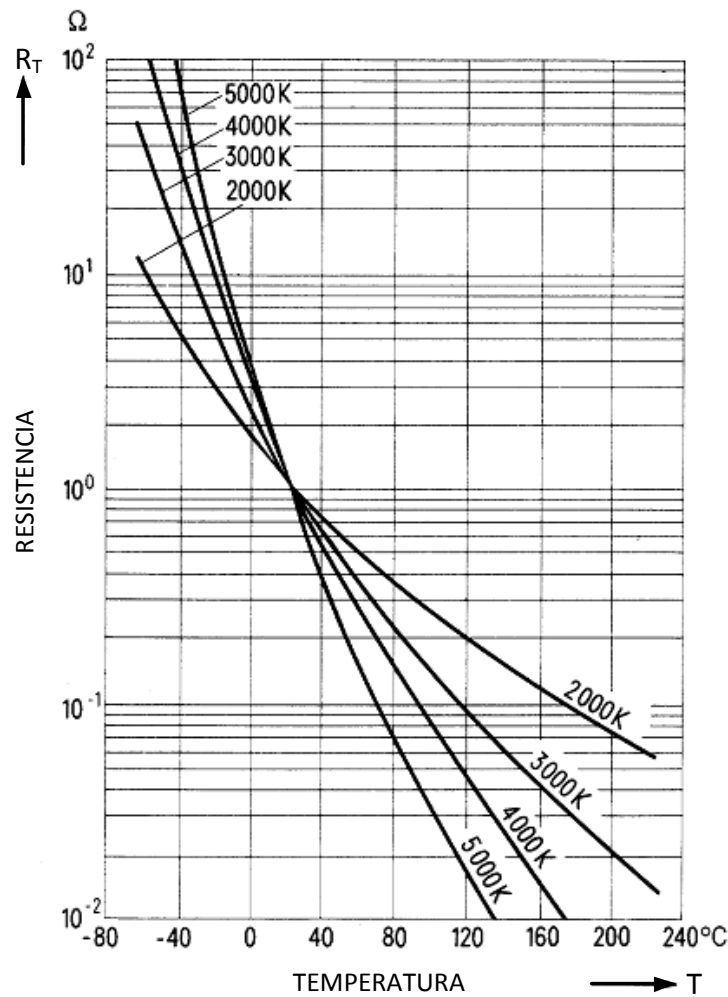


Figura 2.18. Curva característica de un termistor.

Modelo Steinhart-Hart. Para mediciones de temperaturas exactas, la curva de resistencia/temperatura de un termistor debe ser descrita con más detalle. La ecuación de *Steinhart-Hart*, ecuación 9, es una aproximación de tercer orden que es ampliamente aceptada y usada para definir la curva de resistencia con respecto a la temperatura.

$$\frac{1}{T} = a + b \ln R_T + c (\ln R_T)^2 + d (\ln R_T)^3 \quad \text{Ecuación 9}$$

En la ecuación 9, los términos a , b , c , d se obtienen midiendo R_T a cuatro temperaturas distintas y resolviendo el sistema de ecuaciones resultante; a partir de los cuatro coeficientes obtenidos se puede calcular el valor de cualquier temperatura con un error cercano a sólo 0.0015°C, en un rango de 0 a 100°C.

Sensores digitales de temperatura

Los sensores digitales de temperatura son aquellos que combinan la función primaria de medición de temperatura y funciones tales como el procesamiento de la señal que generan y proveer una interfaz de comunicación hacia el exterior. Un sensor digital generalmente es más caro que un sensor convencional, pero si además del costo que implica se consideran cuestiones de mantenimiento, fiabilidad, etc., el costo total de un sensor convencional puede ser mucho mayor.

El nivel de complejidad de un sensor digital puede ser muy variado. Además de la detección de señales, puede incluir:

- Acondicionamiento de la señal
- Correcciones de ganancia y linealidad
- Compensación ambiental
- Escalado y conversión de unidades
- Comunicación digital
- Autodiagnóstico

Los sensores de temperatura digitales ofrecen una alta precisión, alta linealidad en un rango de -55°C a 150°C ; ofrecen también salidas tanto digitales como analógicas, entre otras prestaciones.

Excepto por los sensores de temperatura digitales, todos los demás sensores de temperatura presentan un comportamiento no lineal, esto requiere una compleja circuitería para compensar esa no linealidad o en su caso realizar una compensación por *software*. Además, los sensores de temperatura convencionales generalmente requieren ser calibrados manualmente.

2.2.3. Conversión analógica-digital

Los procesos que se dan en el mundo real producen señales analógicas, cuya variación es continua. La velocidad de esta variación a veces es muy lenta, como en el caso de las variaciones de temperatura, pero también pueden ser muy rápidas, como en el caso de un sistema de audio. Una manera de describir los procesos analógicos es mediante una representación digital que permita un adecuado procesamiento.

Cuando se requiere de un procesamiento digital de la señal que se está midiendo, es necesaria una vinculación entre los mundos analógico y digital, para esto se requiere de dispositivos capaces de efectuar este tipo de tareas.

ADC

Un convertidor analógico-digital (CAD) o bien por sus siglas en inglés ADC (*Analog to Digital Converter*) es un dispositivo que ofrece una salida digital a partir de una señal analógica de entrada. El ADC funciona tomando muestras de la amplitud de una señal en intervalos uniformes de tiempo, asignando un valor digital a cada muestra tomada.

Los siguientes procesos se realizan en una conversión analógico-digital:

- **Muestreo:** Consiste en tomar valores de amplitud de la señal en intervalos discretos de tiempo. La velocidad con que se tomen las muestras determinará la frecuencia de muestreo.
- **Retención:** Las muestras que se toman se deben retener cierto tiempo para que el ADC lo pueda evaluar y asignar un valor correcto. Para esto se usan circuitos llamados circuitos de muestreo y retención (*simple & hold*) que mantienen un nivel constante de voltaje por un intervalo de tiempo definido.
- **Cuantificación:** En este proceso el ADC mide cada valor de amplitud leído del nivel discreto más cercano, utiliza un código digital con 2^N valores diferentes, donde N es la resolución en bits del ADC.
- **Codificación:** La codificación permite asignar valores numéricos binarios equivalentes a los valores de voltajes obtenidos en la cuantificación.

Dos conceptos importantes relacionados con la conversión A/D son:

Frecuencia de muestreo. Para que una señal pueda ser reconstruida fielmente, es necesario prestar atención a la frecuencia con que se tomarán las muestras con el ADC. Una frecuencia de muestreo pequeña provocará que no pueda ser reconstruida la señal, mientras que una frecuencia de muestreo excesiva genera un uso adicional de energía y recursos del sistema.

Teorema de muestreo de Nyquist. Una señal continua en el tiempo $f(x)$ puede ser completamente reconstruida si se muestrea a una tasa tal que $f_{muestreo} > 2f_{max}$, donde f_{max} es la frecuencia máxima de $f(x)$.

Los criterios más importantes a tomar en cuenta al momento de seleccionar un ADC son:

- Número de canales de entrada
- Resolución

- Frecuencia de muestreo
- Rango de entrada de voltaje
- Error de offset y de ganancia
- Linealidad
- Monotonicidad
- Relación señal/ruido

Los convertidores ADC más comunes son:

- ADC Comparador de Tipo Rampa
- ADC de aproximaciones sucesivas
- ADC *flash*
- ADC Sigma-Delta
- ADC paralelo

La mayoría de los microcontroladores integran en su interior convertidores analógicos digitales de tipo aproximaciones sucesivas.

2.3. Sistemas embebidos

Se entiende por sistema embebido a una combinación de *hardware* y *software*, sumado tal vez a algunas piezas mecánicas o actuadores, diseñado para tener una función específica. Es común que en el uso de estos dispositivos se encuentre un procesador y un programa ejecutándose. Muchas veces un sistema embebido es un componente de un sistema mucho más grande, como por ejemplo los sistemas de frenos o el sistema de inyección de combustible en automóviles actuales.

Una de las ventajas de los sistemas embebidos es su flexibilidad, ya que a la hora de realizar alguna modificación resulta mucho más sencillo modificar unas líneas de código al *software* del sistema embebido que reemplazar todo el circuito integrado.

Un uso muy común de los sistemas embebidos es en los sistemas de tiempo real, entendiéndose por sistemas en tiempo real a aquellos sistemas en los que el control del tiempo es vital para el correcto funcionamiento. Los sistemas en tiempo real necesitan realizar ciertas operaciones o cálculos en un límite de tiempo. Donde ese límite de tiempo resulta crucial. Un ejemplo claro de un sistema de tiempo real es el control de tráfico aéreo.

Componentes de un sistema embebido

Un sistema embebido está formado principalmente por un microprocesador y un *software* que se ejecute sobre éste. Todo sistema embebido necesitará cierta cantidad de memoria (para guardar el *software*, almacenar los datos que se obtengan en el proceso, etc.), la cual puede incluso encontrarse dentro del mismo circuito integrado del procesador. Además, normalmente un sistema embebido contará con una serie de salidas y entradas necesarias para comunicarse con el mundo exterior.

Debido a que las tareas realizadas por sistemas embebidos son de relativa sencillez, los procesadores comúnmente usados cuentan con registros de 8 o 16 bits. En su memoria sólo reside el programa destinado a gobernar una aplicación determinada. Sus líneas de entrada/salida se utilizan para la conexión a sensores y actuadores del dispositivo a controlar.

2.4. Microcontroladores

Dentro del mundo de la electrónica existen básicamente dos maneras de diseñar o construir un circuito o un sistema específico para alguna aplicación: mediante componentes discretos y circuitos integrados, o bien utilizando microcontroladores.

Un microcontrolador es un circuito integrado que internamente contiene una unidad de procesamiento central CPU (*Central Processing Unit*), también llamado microprocesador, memoria para almacenar programas y datos, puertos de entrada y salida, además de periféricos adicionales, determinados por el tipo de microcontrolador. A diferencia de los microprocesadores, los microcontroladores son unidades autosuficientes, ya que contienen todo lo necesario para operar por sí solos. En particular, los microcontroladores están diseñados para tareas de monitoreo y control de procesos.

Un microcontrolador es una herramienta muy poderosa que permite crear sistemas sofisticados de adquisición y procesamiento de datos, bajo un programa que lo gobierna todo. Se pueden clasificar con base al número de bits que procesan. Los de 8 bits han sido los más populares y se pueden encontrar en la mayoría de las aplicaciones, los de 16 y 32 bits ofrecen mejores prestaciones en cuanto al nivel de procesamiento de datos que pueden dar. Con respecto a estos últimos, en los años más recientes su uso se ha incrementado a medida que se requieren implementar sistemas más complejos y también por el hecho de que su precio ha disminuido drásticamente.

Las aplicaciones más frecuentes donde se pueden encontrar microcontroladores son las siguientes:

- Industria automovilística
- Industria aeroespacial
- Control y automatización
- Electrodomésticos
- Telecomunicaciones

En la figura 2.19 se muestra el esquema básico de un microcontrolador:

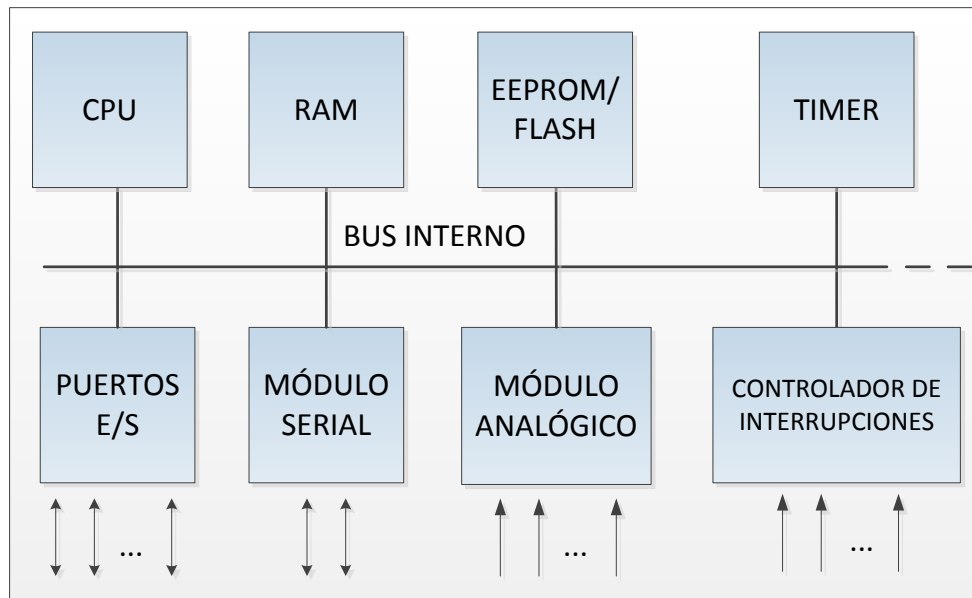


Figura 2.19. Diagrama de bloques básico de un microcontrolador.

Los componentes principales de un microcontrolador son:

- **Unidad de Procesamiento Central (CPU):** Contiene la unidad aritmética lógica, la unidad de control, el contador de programa, el registro acumulador, la pila o *stack* y los registros generales.
- **Memoria:** Existe la memoria de datos y la memoria de programa. La memoria de datos consiste en una memoria RAM. La memoria de programa consiste en una memoria ROM, EEPROM o *flash* que almacena el código del programa que se ejecutará.
- **Puertos E/S:** Son una de las principales características de los microcontroladores, permiten la interacción del microcontrolador con el exterior.
- **Módulo analógico:** Contiene periféricos tales como comparadores analógicos, convertidores analógico-digitales (ADC) y en algunos casos convertidores digital-analógico (DAC).

- **Módulo serial:** Contiene interfaces seriales de comunicación tales los como módulos UART (*Universal Asynchronous Receiver-Transmitter*), SPI (*Serial Peripheral Interface*), I2C (*Inter-Integrated Circuit*), entre otros.
- **Timer:** Se puede contar con uno o más *timers* /contadores de diferentes resoluciones, que pueden ser usados para monitorear eventos de tiempo. Además se pueden configurar para generar salidas PWM (modulación por ancho de pulso).

2.4.1. Arquitectura de microcontroladores

Dependiendo de la disposición de los bancos de memoria dentro del microcontrolador, se pueden distinguir dos configuraciones: Von Neumann y Harvard. Estas arquitecturas son modelos generales del *hardware* de los microcontroladores, que representan dos soluciones diferentes al problema de establecer la conexión entre la CPU y la memoria.

Arquitectura Von Neumann

En la arquitectura Von Neumann, el programa y los datos son almacenados en el mismo banco de memoria y se acceden a través del mismo bus. El uso de esta arquitectura fue ampliamente extendido en los microprocesadores para PC, debido a que en ellos siempre existirá la necesidad de tener el menor número de terminales y conexiones internas. En la figura 2.20 se representa un esquema de la arquitectura Von Neumann.

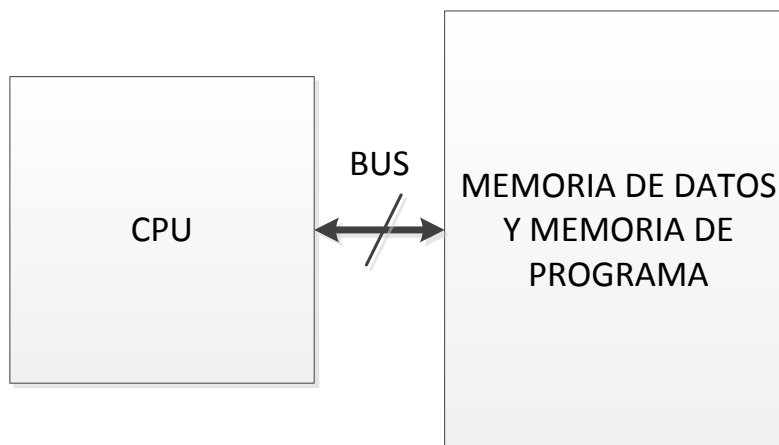


Figura 2.20. Arquitectura Von Neumann.

Arquitectura Harvard

En la arquitectura Harvard la memoria de datos y la de programa están separadas. El acceso a la memoria se hace en canales separados, razón por la cual el acceso a uno de los bancos de memoria no podrá causar conflicto con el otro, como puede suceder en la arquitectura Von

Neumann. Debido a que en un microcontrolador todos los componentes del sistema están dentro de un circuito integrado, no se tiene la necesidad de reducir el número de terminales de la CPU, provocando que el uso de esta arquitectura se extendiera en microcontroladores. En la figura 2.21 se representa un esquema de la arquitectura Harvard.

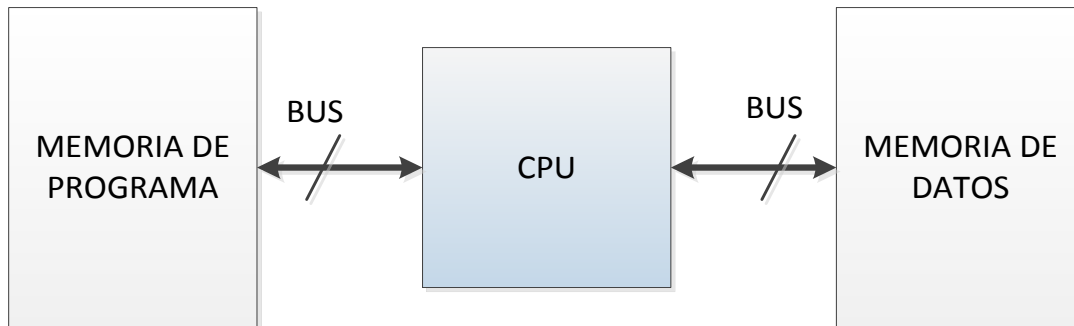


Figura 2.21. Arquitectura Harvard.

2.4.2. Familias de microcontroladores

Dentro de la gran variedad de microcontroladores existente en el mercado, se puede hacer una distinción, con base en los fabricantes y las familias que derivan de cada fabricante. Entre los más importantes se encuentran:

- **ATMEL.** Ofrece familias tales como:
 - **AVR.** En esta familia se ofrecen opciones con arquitecturas de 8 y de 32 bits, con dispositivos tales como los tinyAVR, megaAVR, AVR XMEGA y AVR UC3 de 32 bits.
Esta familia cuenta con dispositivos con encapsulados de 6 hasta 144 terminales, con velocidades de reloj máximas de 66 MHz, soporte para operaciones nativas de DSP (*Digital Signal Processing*), controladores DMA (*Direct Memory Access*), periféricos USB (*Universal Serial Bus*), CAN (*Controller Area Network*), LIN (*Local Interconnect Network*), entre otros.
 - **ARM.** Esta familia se basa en el uso de microprocesadores ARM de 32 bits, con dispositivos que operan a un máximo de 120 MHz, con encapsulados de 32 a 144 terminales, amplia variedad de opciones de comunicación serie entre otros.
- **FREESCALE.** Cuenta con opciones de 8, 16 y 32 bits y soluciones basadas en ARM.
 - La familia de 8 y 16 bits cuenta con dispositivos tales como los S08, HC08, HC05, HC11, S12, HC12, HC16 con encapsulados de 8 a 160 terminales, velocidades de operación de hasta 50 MHz, opciones de comunicación CAN, USB, SPI, Ethernet, entre otros.

- La familia de 32 bits consta de ColdFire y Kinetis (basado en ARM), con frecuencias de operación máximas de 50 MHz, encapsulados que van desde los 32 a los 144 terminales y múltiples opciones de interfaces seriales.
- **TEXAS INSTRUMENTS.** Cuenta con microcontroladores de 16 y 32 bits, con familias tales como MSP430, C2000, *Tiva* y *Hercules* (las últimas dos basadas en ARM). Con encapsulados de 8 a 144 terminales, frecuencias de operación de 16 MHz a 300 MHz, periféricos para comunicación CAN, USB, ETHERNET, LIN, módulos de protección de memoria, entre otros
- **MICROCHIP.** Cuenta con dispositivos de 8, 16 y 32 bits, como PIC10, PIC12, PIC16, PIC18, PIC24, dsPIC30, dsPIC33 y PIC32. Con opciones de encapsulados de 6 a 144 terminales, frecuencias de operación de hasta 80 MHz, periféricos para comunicación CAN, ETHERNET, LIN, USB, controladores DMA, soporte para operaciones de DSP, entre otros.

2.4.3. Microcontroladores AVR

Los AVR son una familia de microcontroladores RISC (*Reduced Instruction Set Computer*) de Atmel. Esta familia fue concebida por dos estudiantes del *Norwegian Institute of Technology* y posteriormente refinada y desarrollada por Atmel. El acrónimo fue reportado como *Advanced Virtual RISC*.

El AVR fue diseñado desde un comienzo para la ejecución eficiente de código en C.

La familia de AVR es muy numerosa, con variantes tales como tinyAVR, megaAVR, AVR XMEGA y AVR UC3.

tinyAVR. Esta familia está orientada a aplicaciones que requieren encapsulados pequeños con un número de terminales reducido, partiendo desde 6 terminales, con opciones de memoria *flash* de hasta 8 kB, frecuencias de operación de hasta 20 MHz y cuenta con periféricos tales como ADC, memoria EEPROM (*Electrically Erasable Programmable Read-Only Memory*), módulo SPI, UART (*Universal Asynchronous Receiver Transmitter*), PWM (*Pulse Width Modulation*). Algunos miembros de esta familia aceptan voltajes de polarización de 0.7 V.

megaAVR. Esta familia se caracteriza por contar con dispositivos con una mayor cantidad de terminales con respecto a los tinyAVR, además de contar con una cantidad mayor de memoria *flash* disponible (un máximo de 256kB). Cuenta con periféricos tales como controladores CAN, LIN, USB, para LCD (Liquid Crystal Display), módulos UART, ADC, DAC, sensores de temperatura integrados y frecuencia de operación de hasta 20 MHz.

AVR XMEGA. Orientados para aplicaciones de tiempo real, contienen mejores periféricos con respecto a megaAVR, como ADCs y DACs de 12 bits, controladores DMA, módulos de encriptación de datos AES (*Advanced Encryption Standard*), mayor número de módulos UART, PWM, I2C, SPI, etc. Los encapsulados varían de entre 44 a 100 terminales, con frecuencias de operación de hasta 32 MHz.

AVR UC3. Basado en una arquitectura de 32 bits, es capaz de realizar de forma nativa operaciones de DSP, cuenta con controladores DMA, conectividad CAN, LIN, Ethernet, salidas PWM en todas las terminales de entrada/salida, variantes de encapsulados de 48 a 144 terminales, hasta 512 kB de *flash*, entre otras características.

2.4.4. *Software de desarrollo*

Durante el proceso de resolución de un problema que precise tareas de cómputo o procesamiento de datos, una de las etapas fundamentales es la del desarrollo del *software*. Se pueden considerar una serie de fases o pasos comunes durante el desarrollo de éste:

- Diseño de algoritmo
- Codificación
- Compilación y ejecución
- Verificación
- Depuración
- Mantenimiento
- Documentación

La primera fase conduce a un diseño detallado del algoritmo requerido. Durante la codificación se implementa el algoritmo en un código escrito en algún lenguaje de programación, reflejando de la mejor manera posible lo que se busca. Las fases de compilación y ejecución traducen y ejecutan el programa. En las fases de verificación y depuración se buscarán errores y tratarán de eliminarse. Durante el mantenimiento se tratará de corregir errores inesperados y mejorar el rendimiento del programa. Por último, la documentación es de suma importancia, ya que registrará las fases del ciclo de vida del *software* y durante este proceso se recogen los pasos seguidos en cada fase para su ejecución.

Compiladores de alto nivel

Los compiladores de alto nivel son las herramientas de programación más potentes que existen. El compilador traducirá el programa escrito en lenguaje de alto nivel (C, Basic, Pascal u otro) en código ensamblador y luego generará los archivos de depuración (*.dbg, *.cof, *.d31, etc.) necesarios y de ejecución (*.hex). Al trabajar con un lenguaje de alto nivel el programador no debe preocuparse por las características de *hardware* ni por el ensamblador nativo de cada microcontrolador. Inclusive es posible adaptar un código para un microcontrolador de otra marca, por ejemplo, de Freescale o Atmel. A la característica anterior se le conoce como portabilidad.

Las características del lenguaje C fueron ideadas para el trabajo con sofisticados procesos, propios de los ordenadores. Muchas de esas características resultan ventajosas y son aprovechadas por el hardware de los microcontroladores. Además, dada la robustez y la aceptación del lenguaje C, se le ha tomado como referencia para lenguajes de otros propósitos como Java, JavaScript, php o de Matlab, entre otros.

Lenguaje C

C es un lenguaje de programación de propósito general que ha sido estrechamente asociado con el sistema UNIX, lenguaje con el cual fue desarrollado. Este lenguaje no está ligado a ningún sistema operativo ni a ninguna máquina en específico.

C proporciona las construcciones fundamentales que requieren los programas bien estructurados: agrupación de proposiciones, toma de decisiones (*if-else*), selección de un caso entre un conjunto de ellos (*switch*), iteración de una condición de paro (*while, for, do*), terminación prematura de ciclos (*break*) y un método definido para llamada a funciones y para uso de parámetros.

Gran parte de la potencia del lenguaje C reside en su habilidad de combinar comandos simples de bajo nivel y complicadas funciones de alto nivel, esto permite el acceso a los *bytes* y palabras reservadas de un procesador. Esto es especialmente útil cuando se trabaja con microcontroladores, ya que para tener un control sobre ellos es necesario acceder a sus registros para poder modificar su funcionamiento.

Otras consideraciones a destacar para el lenguaje C usado en microcontroladores son:

- Control de periféricos internos y externos del circuito integrado
- Servicio de interrupciones
- Soporte de diferentes configuraciones de RAM/ROM
- Alto nivel de optimización

- Uso eficiente de conjuntos limitados de instrucciones

Entorno de desarrollo Atmel Studio

Atmel Studio es el entorno de desarrollo integrado (IDE, *Integrated Development Environment*) de Atmel para la elaboración de proyectos con sus productos, relacionados con microcontroladores. Anteriormente Atmel mantuvo dos IDEs separadas para sus microcontroladores, una que soportaba microcontroladores de 8 bits mientras que la otra daba soporte a microcontroladores con arquitectura de 32 *bits*.

En 2011 Atmel unificó ambas IDEs creando el AVR Studio 5 y en 2012 comienza a dar soporte a sus microcontroladores con procesadores ARM, de esta manera el nuevo entorno de desarrollo pasó a llamarse Atmel Studio 6. Entre las herramientas que incluye Atmel Studio se tienen las siguientes:

- Un editor de códigos, que permite mostrar el código en diferentes colores, según la instrucción de la que se trate, mostrar números de línea, etc.
- Un administrador de proyectos, que da soporte para trabajar en ensamblador y a varios compiladores de C para AVR.
- Un simulador, el AVR Simulator, para programas escritos tanto en C como en ensamblador.
- Un paquete de librerías y APIs (*Application Programming Interface*) para facilitar la interacción con los periféricos de los microcontroladores, desde los puertos hasta los módulos USB o CAN.

En la figura 2.22 se muestra una captura de pantalla del entorno de desarrollo Atmel Studio 6.

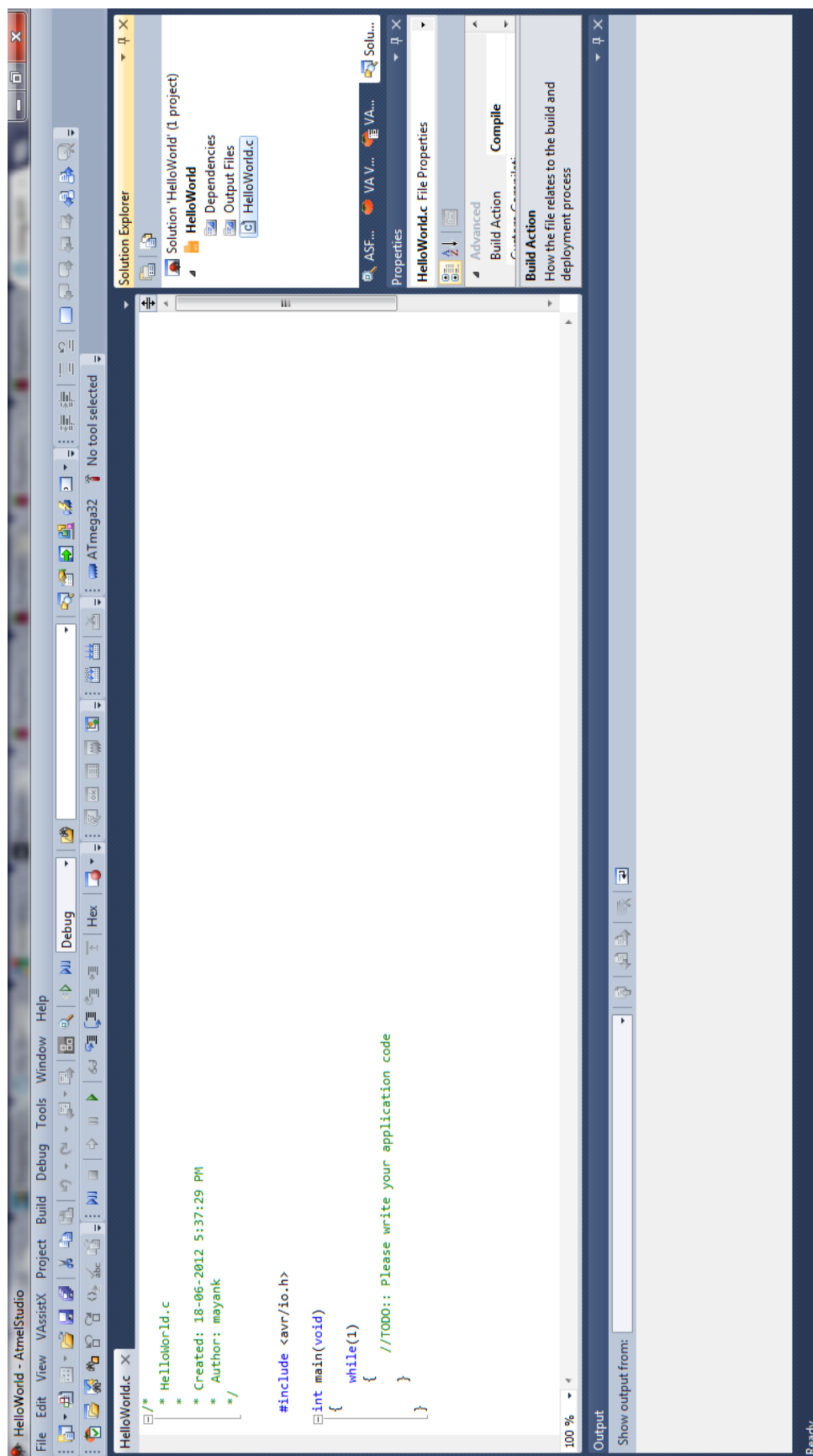


Figura 2.22. Entorno de desarrollo Atmel Studio.

2.5. Comunicación serie

La comunicación serie es un procedimiento que utilizan dos o más dispositivos para el intercambio de información de forma secuencial. Para establecer una comunicación serie es necesario definir uno o varios emisores de información (maestros) y uno o varios receptores (esclavos), que utilizan una conexión para el intercambio de la información. Los tipos de conexión [9] empleados pueden ser el *simplex* mostrado en la figura 2.23, en donde sólo se presenta flujo de información de maestro a esclavo. El *full-duplex* se observa en la figura 2.24, en este se tiene intercambio de información de maestro a esclavo y viceversa, al mismo tiempo. En la figura 2.25 se presenta el *half-duplex*, en el cual sólo se tiene intercambio de información en un solo sentido a la vez, es decir, si el maestro se encuentra transmitiendo datos el esclavo no lo hace, de igual manera para el sentido contrario.

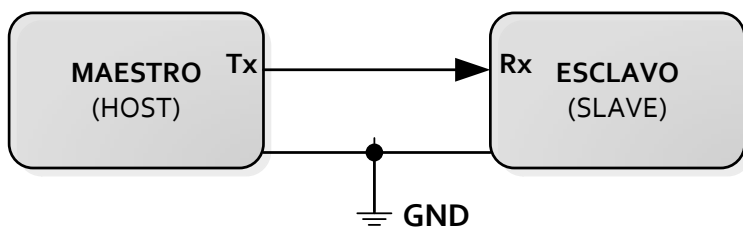


Figura 2.23. Conexión tipo *Simplex*.

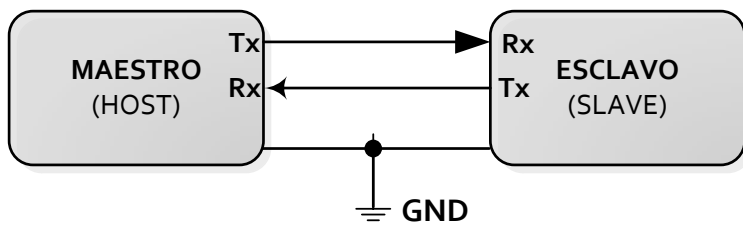


Figura 2.24. Conexión tipo *Full-Duplex*.

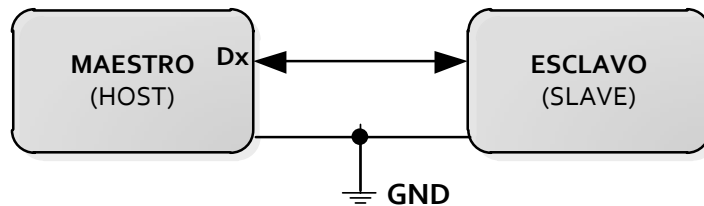


Figura 2.25. Conexión tipo *Half-Duplex*.

Así mismo existen configuraciones o topologías para realizar la conexión entre los dispositivos, dichas topologías son llamadas Punto a Punto, *Multi-Drop* y *Multi-Punto*. La primera describe a solo un maestro y a solo un esclavo, como se observa en la figura 2.26. La segunda a varios esclavos conectados al bus de un solo maestro, como se observa en la figura

2.27 y la última se refiere a varios esclavos conectados a un bus con más de dos maestros, mostrado en la figura 2.28. Se debe mencionar que en esta configuración sólo un maestro y un esclavo pueden enviar información a la vez.

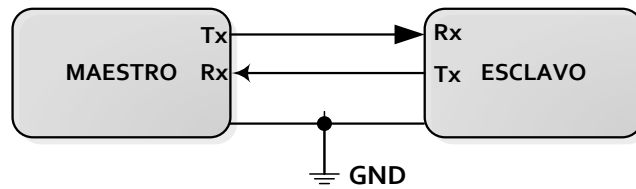


Figura 2.26. Topología Punto a Punto.

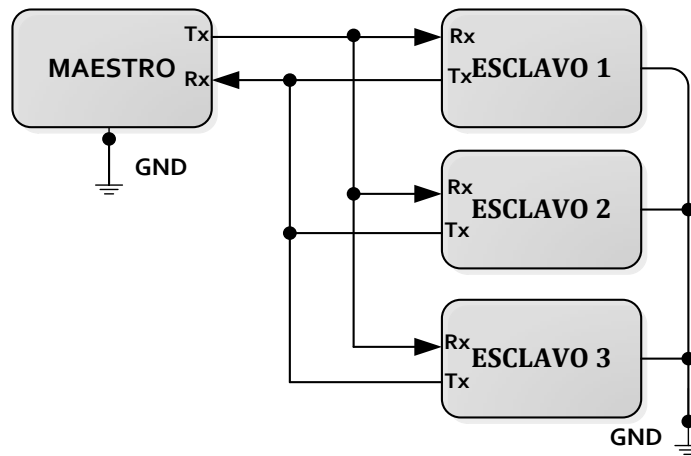


Figura 2.27. Topología Multi-Drop.

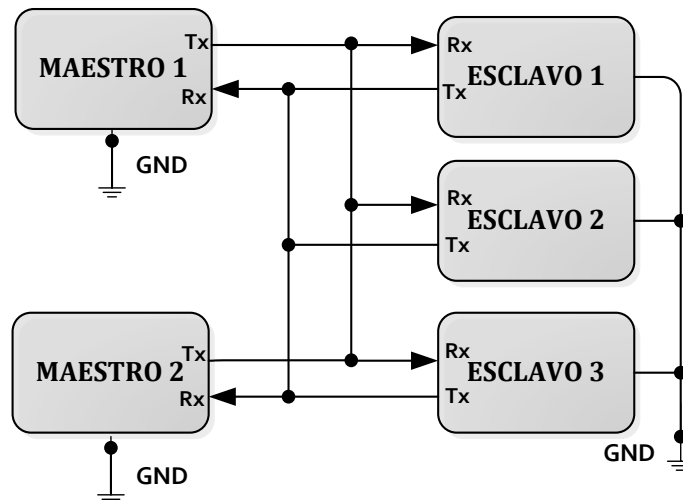


Figura 2.28. Topología Multi-Punto.

En la comunicación serie se puede encontrar de dos tipos: asíncrona y síncrona.

2.5.1. Comunicación asíncrona

La comunicación asíncrona se da cuando el proceso de sincronización entre emisor y receptor se realiza en cada palabra de código transmitido. Esta sincronización se lleva a cabo a través de unos *bits* especiales que definen el entorno de cada código. También se dice que se establece una relación asíncrona cuando no hay ninguna relación temporal entre la estación que transmite y la que recibe.

El intercambio de información o protocolo entre dos dispositivos de forma asíncrona puede ser definido dependiendo los requisitos del sistema, pero básicamente se deben definir las siguientes características:

- Velocidad de transmisión: Indica el número de bits por segundo que se transfieren, su unidad son los baudios.
- Número de *bits* de datos: Se refiere a la cantidad de bits de información. A esta característica se le conoce como paquete, un paquete generalmente contiene 8 *bits*.
- *Bit* de inicio: Indicará el inicio de transmisión, al hacer un cambio de estado en el *bus*.
- *Bits* de parada: Se utilizan para indicar el fin de la comunicación de un paquete y proporcionar un margen para sincronizar el *bus*. Los valores típicos son 1, 1.5 o 2 *bits*.
- Paridad: Es utilizada para verificar si se presentaron errores en la transmisión serial. Existen cuatro tipos de paridad, que son par, impar, marcada y espaciada, aunque también se puede no usar paridad.

En la figura 2.29 se puede observar la transmisión de un paquete (0x49) con algunas de las características que se han mencionado.

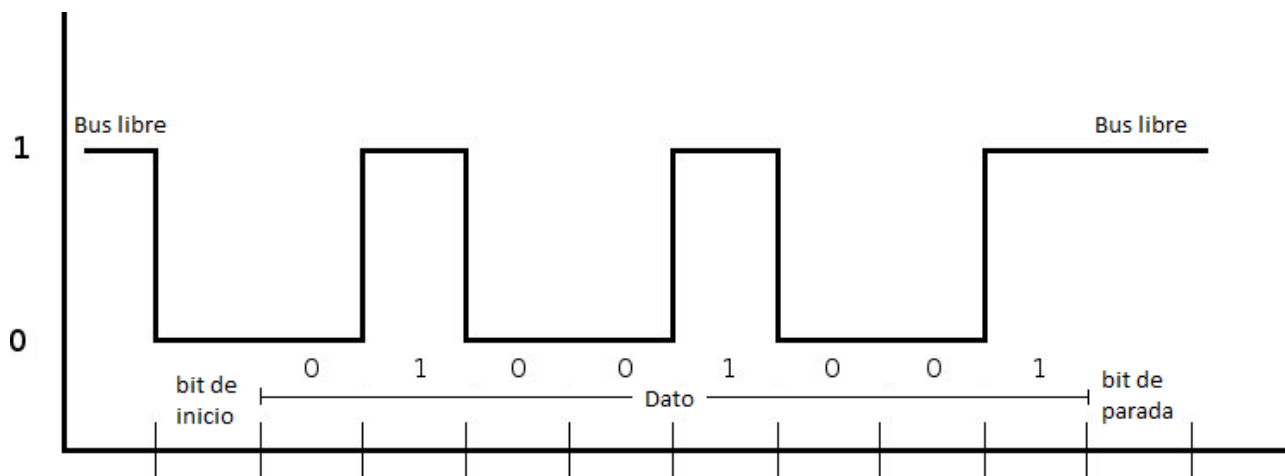


Figura 2.29. Protocolo de comunicación serie.

Al inicio de la transmisión se debe verificar que el *bus* se encuentre libre, posteriormente enviar el *bit* de inicio seguido del dato o paquete, por último se envía el *bit* de parada para dejar el *bus* libre. Algunos ejemplos de estándares eléctricos que manejan comunicación asíncrona son: RS-232, RS-422, entre otros.

Estándar RS-232

El estándar TIA/EIA-232, también conocido como RS-232, establece las características eléctricas de las señales que se utilizan para la transferencia de datos [10], así como los conectores apropiados y protocolo de comunicación.

Las señales que se utilizan en RS-232 son del tipo *single-ended*, es decir que tienen una referencia de tierra, su esquema de conexión se presenta en la figura 2.30.

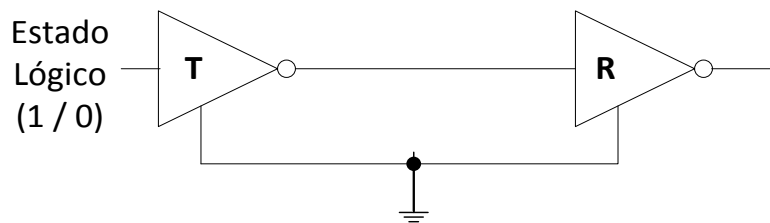


Figura 2.30. Interfaz RS-232.

Las características eléctricas que establece el estándar son:

- En el lado de transmisión un estado lógico 0 se representa por un voltaje positivo de 5V a 15V y un estado lógico 1 se detecta entre los niveles negativos -5V a -15V.
- En el lado de recepción un estado lógico 0 se representa por un voltaje positivo de 3V a 15V y un estado lógico 1 se detecta entre los niveles negativos -3V a -15V.
- El voltaje en la terminal del conector no deberá exceder ± 25 V.
- Cada línea deberá tener una impedancia mínima de 3 k Ω y una impedancia máxima de 7 k Ω , que generalmente es parte del circuito receptor.
- La capacitancia máxima de carga en la línea es de 2,500 pF, que es equivalente a una longitud máxima en el conductor de 15 a 20 metros.
- *SlewRate* (SR) máximo de 30V/ μ s en el transmisor, esto con el fin de evitar el problema de ruido inducido (*Crosstalk*) entre las señales al cambiar de estado.
- La velocidad máxima de transmisión se establece en 20 kbit/s con la máxima longitud de conductor. Si se reduce la longitud del conductor en práctica se pueden establecer velocidades de hasta 120 kbit/s.

De acuerdo al estándar también encontramos un protocolo definido. Como parte de este protocolo se definen características mecánicas y eléctricas especiales del mismo, ya que utiliza

señales de control para la comunicación. Estas señales deben usar un conector DB de 25 terminales o de 9 terminales.

En la tabla 2.1 se puede observar la distribución de las señales por conector, el nombre indica la función que cumple en el protocolo.

Tabla 2.1. Asignación de terminales en los conectores para RS-232.

| DB25 | DB9 | Nombre | Nombre formal |
|------|-----|--------|---------------------------------------|
| 1 | | FG | <i>Protective Ground</i> |
| 2 | 3 | TD | <i>Transmitted Data</i> |
| 3 | 2 | RD | <i>Received Data</i> |
| 4 | 7 | RTS | <i>Request To Send</i> |
| 5 | 8 | CTS | <i>Clear To Send</i> |
| 6 | 6 | DSR | <i>Data Set Ready</i> |
| 7 | 5 | SG | <i>Signal Ground</i> |
| 8 | 1 | DCD | <i>Data Carrier Detect</i> |
| 9 | | | <i>+ P</i> |
| 10 | | | <i>- P</i> |
| 11 | | | <i>No assigned</i> |
| 12 | | SDCD | <i>Secondary Data Carrier</i> |
| 13 | | SCTS | <i>Secondary Clear To Send</i> |
| 14 | | STD | <i>Secondary Transmitted</i> |
| 15 | | TC | <i>Transmission Signal</i> |
| 16 | | SRD | <i>Secondary Received Data</i> |
| 17 | | RC | <i>Receiver Signal Element Timing</i> |
| 18 | | | <i>No assigned</i> |
| 19 | | SRTS | <i>Secondary Request To Send</i> |
| 20 | 4 | DTR | <i>Data Terminal Ready</i> |
| 21 | | SQ | <i>Signal Quality Detector</i> |
| 22 | 9 | RI | <i>Ring Indicator</i> |
| 23 | | | <i>Data Signal Rate Selector</i> |
| 24 | | | <i>Transmitter Signal</i> |
| 25 | | | <i>No assigned</i> |

Estándar RS-422

El estándar TIA/EIA-422 [11][12], también conocido como RS-422 fue diseñado para implementar mejoras el estándar RS-232. Utiliza señales balanceadas del tipo diferencial para resolver los problemas de *crossstalk* que se presentan en interfaces *single-ended*, esto permite aumentar la velocidad de transmisión. Además de eliminar problemas de ruido debido a las entradas comunes. La limitación de la longitud de los cables disminuye, permitiendo una conexión *multi-drop* con hasta 10 receptores a la vez. Estas ventajas han hecho que se vea favorecido el estándar RS-422 sobre el RS-232 en la industria.

El diagrama de conexión entre transmisor (D) y receptor (R) de esta interfaz se presenta en la figura 2.31. En la misma se puede notar que se coloca una impedancia del lado del receptor, esto con el fin de evitar reflexión de las señales. A esta conexión con terminación del lado receptor se le llama circuito balanceado.

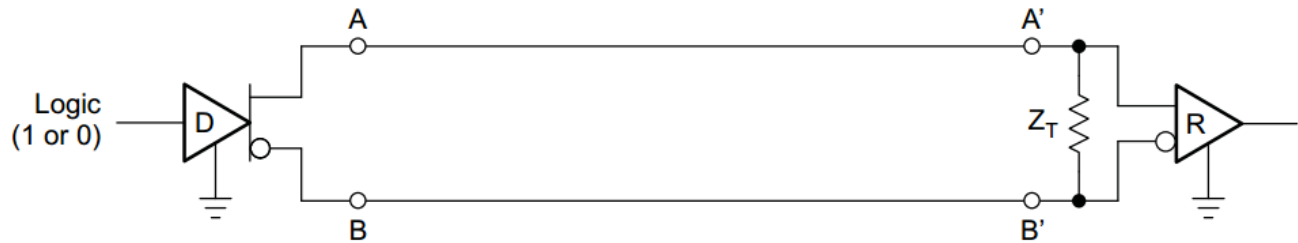


Figura 2.31. Interfaz RS-422 y LVDS.

Las características eléctricas que establece el estándar son:

- En el lado de transmisión un estado lógico 0 se representa por un voltaje positivo de 2V a 6V y un estado lógico 1 se detecta entre los niveles negativos -2V a -6V.
- En el lado de recepción un estado lógico 0 se representa por un voltaje positivo de 200 mV a 6V y un estado lógico 1 se detecta entre los niveles negativos -200 mV a -6V.
- El voltaje en la terminal del conector no deberá exceder ± 10 V.
- La línea de transmisión soporta hasta 10 cargas máximo, considerando una carga equivalente a una impedancia 4 k Ω .
- La terminación de una línea balanceada debe ser de al menos 100 Ω , para asegurar un voltaje mínimo ± 200 mV.
- El conductor puede tener una longitud máxima de 1,200 metros.
- Puede alcanzar velocidades de transmisión hasta de 10 Mbit/s. Sin embargo esta velocidad no se logra a la distancia máxima especificada.
- El estándar no define protocolo ni conector.

La gráfica que se muestra en la figura 2.32 fue obtenida de forma empírica por el fabricante *Texas Instrument*, usando un cable de cobre con calibre 24 AWG, con una capacitancia de 16 pF/ft. El cable es par trenzado con terminación de 100 Ω en la carga. Se puede observar un comportamiento de la longitud del cable contra la velocidad de transmisión. Además se incluye en modo de comparación las velocidades alcanzados con otros estándares, entre ellos el RS-232.

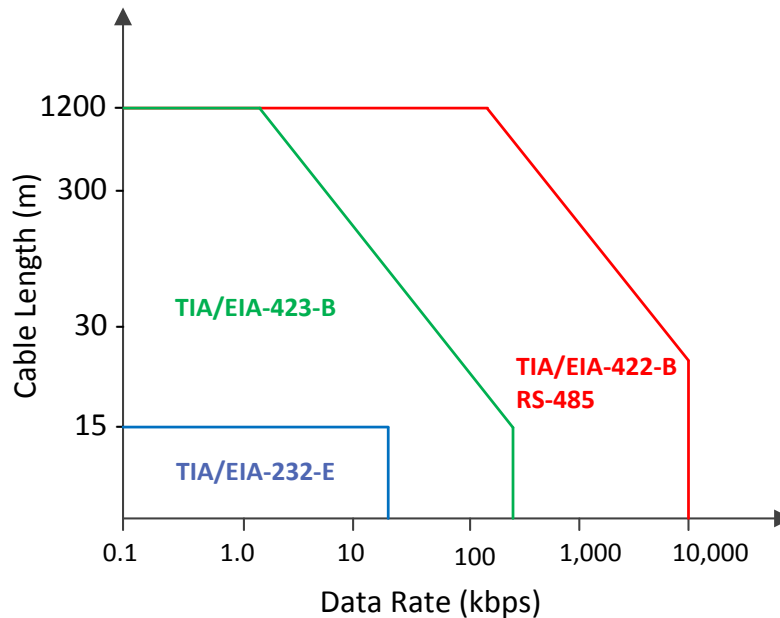


Figura 2.32. Velocidad de transmisión vs longitud del conductor [13][14].

Estándar TIA/EIA-644

El estándar TIA/EIA-644, mejor conocido como LVDS (*Low-Voltage Differential Signaling*) [15][16], es un método utilizado para transmitir datos a altas velocidades y baja potencia. Algunas de las características de la interfaz se presentan a continuación:

- En la transmisión las señales tienen niveles de voltaje ± 250 mV a ± 450 mV.
- Los receptores pueden detectar hasta niveles comprendidos entre ± 100 mV.
- Las líneas deben ser balanceadas y con una terminación de 100Ω .
- El estándar recomienda una velocidad máxima de 655 Mbps, aunque también menciona velocidad teórica máxima de 1.923 Gbps.
- El cambio de dirección en la corriente del resistor de terminación, indicará en el receptor una salida válida para un estado lógico 0 o 1, esto se define desde en el voltaje de transmisión, en la figura 2.31 estaría definido por la diferencia de potencial en las terminales a y b.
- La longitud del cable está definida por una impedancia máxima de 32Ω , que es el equivalente a una pérdida de 80 mV de la señal, a estas limitantes no se asegura una buena transmisión de datos.

Al tratarse de una conexión diferencial, el diagrama de la interfaz LVDS es igual al de la interfaz RS-422, representado en la figura 2.31.

2.5.2. Comunicación síncrona

En la comunicación síncrona, el transmisor y el receptor se sincroniza a la frecuencia de un reloj. La comunicación síncrona es más eficaz que la asíncrona, debido a que son innecesarios los *bits* de inicio y de parada. En vez de eso, los datos se envían en bloques que son mucho más grandes que un solo carácter. Los bloques comienzan con una secuencia de identificación de *bits* que permite un entramado apropiado y a menudo identifica también el contenido del bloque.

El costo de la mayor eficiencia de la comunicación síncrona, es una mayor complejidad tanto en *software* como en *hardware*. En la práctica, los sistemas asíncronos se utilizan para comunicación entre microcomputadoras. Los sistemas síncronos se utilizan para la comunicación de mayor velocidad común en sensores y señales analógicas digitalizadas, como las que se encuentran en sistemas complejos de adquisición de datos. Ejemplos de estándares eléctricos que manejan comunicación síncrona son: SPI, I2C, entre otros.

Estándar SPI

SPI es un estándar de comunicación serie, es de tipo *Full-Duplex* y utiliza la topología *Multi-Drop* [17]. Fue implementado por MOTOROLA para sus dispositivos, pero actualmente un gran número de dispositivos electrónicos utilizan este estándar, aunque éste no es una especificación oficial.

La interacción entre los dispositivos de este estándar es de tipo maestro–esclavo. En esta interfaz se utilizan 4 líneas unidireccionales, las cuales son:

Línea MOSI (*Master Output Slave Input*): Esta línea se encarga de llevar el flujo de datos del maestro al esclavo.

Línea MISO (*Master Input Slave Output*): Esta línea se encarga de llevar el flujo de datos del esclavo al maestro.

Línea de reloj (*SCK, Serial Clock*): Esta línea es generada y manipulada solamente por el maestro, ésta se encarga de sincronizar la transmisión de datos, ya que indica cuando los datos son válidos.

Línea de selección de dispositivo (*CS, Chip Select*): Esta línea es controlada por el maestro, se encarga de seleccionar el esclavo con el cual se desea tener comunicación, cambiando de nivel lógico. En una red donde haya más de un esclavo se utiliza una línea de *Chip Select* para cada uno.

También debe tener una línea de referencia (GND). La figura 2.33 muestra la conexión del *bus* SPI; como se puede observar entre más esclavos se conecten al *bus*, será necesaria la adición de líneas de selección o *Chip Select* (CS1, CS2, CS3,...,CSn).

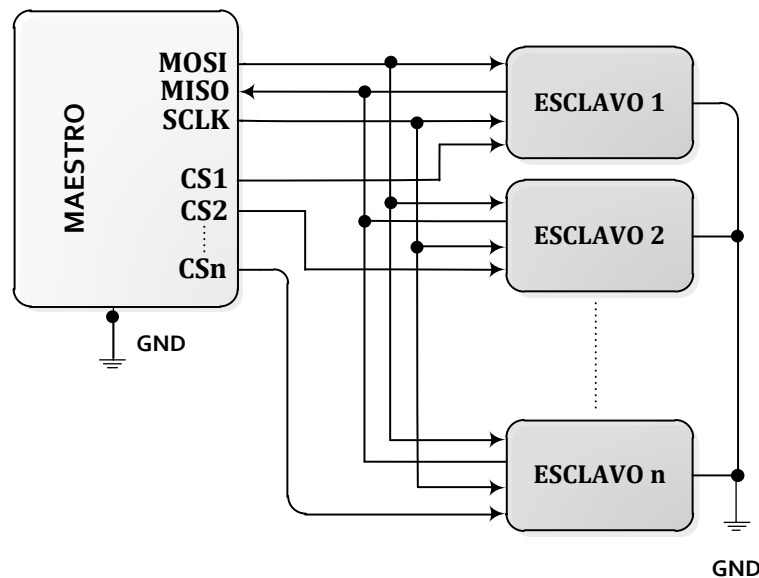


Figura 2.33. Bus SPI.

La forma de las señales para enviar y recibir información a través del bus SPI se observa en la figura 2.34.

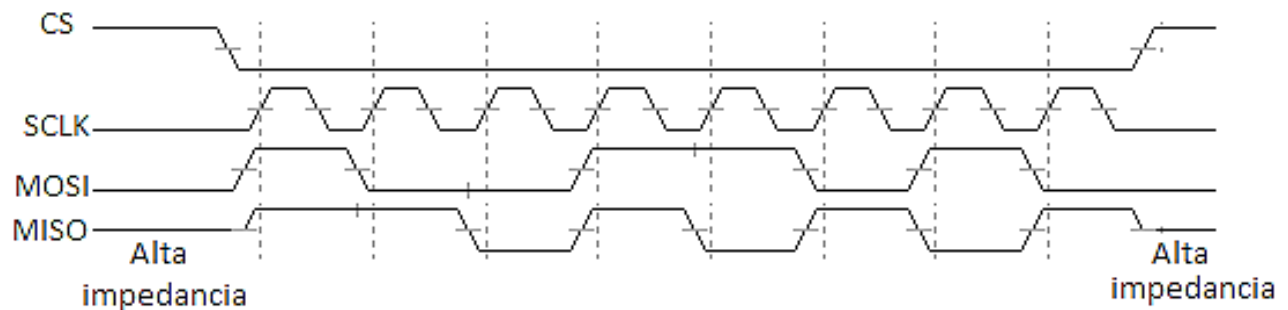


Figura 2.34. Protocolo SPI.

El procedimiento para establecer intercambio de información entre el maestro y el esclavo es:

- Establecer el modo en que el reloj operará. Existen cuatro modos de operación del estándar SPI, los cuales dependen de dos características del reloj. La primera es conocida como polaridad del reloj o *Clock Polarity* (CPOL), esto es el nivel de voltaje que tiene el reloj en estado libre (*IDLE STATE*); a la segunda se le conoce como fase del reloj o *Clock Phase* (CPHA), esto se refiere al flanco en el cual se toma la muestra de la información que se intercambia.

En la figura 2.35 se muestra que la combinación de estas dos características da como resultado 4 modos de operación, los cuales se definen como 0, 1, 2 y 3.

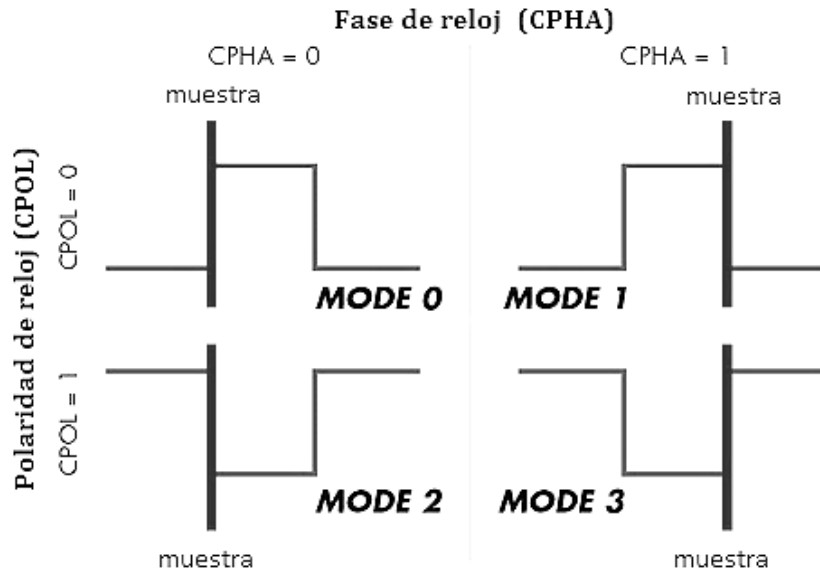


Figura 2.35. Modos de SPI.

- **MODE 0.** En este modo el reloj se encuentra en estado libre cuando su línea está en un nivel lógico 0 y la muestra es tomada en el flanco de subida del reloj.
- **MODE 1.** En este modo el reloj se encuentra en estado libre cuando su línea está en un nivel lógico 0 y la muestra es tomada en el flanco de bajada del reloj.
- **MODE 2.** En este modo el reloj se encuentra en estado libre cuando su línea está en un nivel lógico 1 y la muestra es tomada en el flanco de bajada del reloj.
- **MODE 3.** En este modo el reloj se encuentra en estado libre cuando su línea está en un nivel lógico 1 y la muestra es tomada en el flanco de subida del reloj.

Los pasos para iniciar una comunicación con un esclavo SPI son:

1. Habilitar la línea de selección de dispositivo mediante la línea de *Chip Select*, cambiando de estado ésta. La mayoría de los dispositivos son seleccionados cuando detectan esta línea en estado bajo (*ActiveLow*).
2. El maestro envía el dato a transmitir por la línea MOSI. Éste y el esclavo seleccionado se sincronizarán con la línea de reloj SCLK. El esclavo recibirá instrucciones o datos del maestro por la línea MOSI. Si el esclavo responde transmitirá información al maestro por la línea de MISO.
3. Deshabilitar el dispositivo con la línea de *Chip Select*, al terminar el intercambio de información. Esto se logra cambiando la línea de CS a su estado inicial, es decir, nivel alto en este caso.

Un aspecto importante a considerar es que en un *bus* con varios esclavos, sólo se puede seleccionar un esclavo a la vez. Los esclavos que no son seleccionados deben de mantener un estado de alta impedancia en su terminal de MOSI, si no es así se pueden presentar errores en los datos que el esclavo seleccionado esté transmitiendo al maestro.

En las comunicaciones seriales asíncronas o síncronas, es común la implementación de códigos para la detección de errores, dentro de algún protocolo de comunicación específico.

2.5.3. Detección de errores en la transmisión de datos

Un protocolo de comunicación es un conjunto de reglas y normas que permiten que dos o más entidades de un sistema de comunicación se comuniquen entre sí, para transmitir información por medio de cualquier tipo de variación de una magnitud física. Se trata de las reglas o el estándar que define la sintaxis, semántica y sincronización de la comunicación, así como posibles métodos de recuperación de errores. Los protocolos pueden ser implementados por *hardware*, *software*, o una combinación de ambos.

La transmisión de información está sujeta principalmente a tres tipos de perturbaciones: ruido, interferencia y diafonía.

- Ruido: Es cualquier señal indeseada y variable al azar que disminuye el contenido de información. Causan atenuación o distorsión a la señal que se transmite.
- Interferencia: Término empleado generalmente en radiocomunicación, se refiere a señales de la misma frecuencia o muy cercanas que perturban a la de interés.
- Diafonía: Describe la perturbación electromagnética que otros cables cercanos u otros canales próximos ejercen sobre el nuestro.

Estas perturbaciones son la causa importante por la que la señal eléctrica se deteriora al viajar por el canal de comunicación. Esta degradación de la señal puede hacer que se reciba en el receptor un carácter distinto al que fue emitido por el extremo transmisor, cuando esto ocurre se ha producido un error.

Disminuir los efectos de estos tres elementos perturbadores es de primordial importancia, los filtros, el uso de blindajes, la selección correcta de trayectos y técnicas de detección y corrección de errores se usan para esto.

La comprobación de redundancia cíclica (CRC) es un código de detección de errores, es usado frecuentemente en redes digitales y en dispositivos de almacenamiento para detectar

cambios accidentales en los datos. Los bloques de datos ingresados en estos sistemas contiene un valor de verificación adjunto, basado en el residuo de una división de polinomios, el cálculo es recursivo. Este código es un tipo de función que recibe un flujo de datos de cualquier longitud como entrada y devuelve un valor de longitud fija como salida. El término suele ser usado para designar tanto a la función como a su resultado. Pueden ser usadas como suma de verificación para detectar la alteración de datos durante su transmisión o almacenamiento. Los CRC son particularmente efectivos para detectar errores ocasionados por ruido en los canales de transmisión.

La detección y corrección de errores es una práctica importante para el mantenimiento e integridad de los datos a través de diferentes procedimientos y dispositivos, tales como medios de almacenamiento confiables.

2.6. Almacenamiento de datos

Los dispositivos de almacenamiento de datos son periféricos de un sistema que actúan como medio de soporte para guardar datos. Existen diversos dispositivos de almacenamiento de datos, entre los que se encuentran el DVD (*Digital Versatile Disc*), CD-ROM (*Compact Disc - Read Only Memory*), discos duros, memoria SD (*Secure Digital*), memoria EEPROM, la memoria *flash*, entre otros.

2.6.1. Memorias ROM

La memoria ROM (Read Only Memory, o Memoria de Sólo Lectura), almacena información y puede conservarla sin necesidad de polarización; a veces denominada memoria no volátil, dado que no se borra cuando se apaga el sistema. Existen diferentes memorias de tipo ROM, entre ellas encontramos: EEPROM, *flash* o SD.

- La memoria EEPROM es un tipo de memoria de sólo lectura programable y que se borra eléctricamente. En este tipo de memoria se puede escribir y leer entre 100,000 y 1,000,000 de veces. Para lograr la programación de este tipo de memorias se debe acceder a cada localidad de memoria a la vez para poder escribir o leer.
- La memoria *flash* se trata de una memoria que esencialmente representa a una EEPROM pero con la ventaja de poder escribir y leer múltiples localidades de memoria a la vez. Son utilizada para almacenar grandes cantidades de información.
- Las memorias SD es un tipo de memoria *flash*, con la particularidad que poseen un microcontrolador para direccionar a las localidades de memoria que se quiere escribir.

Otras de las características de este tipo de memorias son que tienen implementado un algoritmo de corrección de errores (ECC) y de seguridad, administración de energía y control de reloj. Para tener acceso a la escritura o lectura a este tipo de memorias se utilizan dos tipos de protocolos, por un lado tiene un protocolo propio llamado *SD card* y uno estándar que es el SPI. Actualmente se tienen memorias SD con capacidades mayores a 32 GB de diferentes clases (*Class 2, 4, 6, 10*), cada clase representa la velocidad de transferencia de datos que soporta, por ejemplo, la Clase 2 soporta hasta 2 Mb/s.

En el desarrollo de este capítulo se han dado las bases necesarias para el desarrollo del proyecto en cuestión. En el capítulo siguiente se proporcionarán los requerimientos solicitados por la colaboración, además se desarrollarán las propuestas de diseño del *Housekeeping*, bancos de prueba y *software*.

3. Diseño y desarrollo

En este capítulo se abordará el diseño del *Housekeeping*, considerando para ello los requerimientos parcialmente definidos para la primera versión. Posteriormente, a partir de pruebas con prototipos desarrollados para simular las interfaces físicas y su comportamiento, pruebas en condiciones de presión y temperatura similares a las de operación, e integraciones con los subsistemas reales desarrollados por los colaboradores europeos, se presentará una segunda versión, como modelo final de vuelo del *Housekeeping*.

La misión EUSO-BALLOON es una versión reducida del JEM-EUSO, que consiste en un telescopio a bordo de un globo estratosférico del CNES de Francia. El instrumento será montado en una estructura llamada góndola [18], como la que se muestra en la figura 3.1.

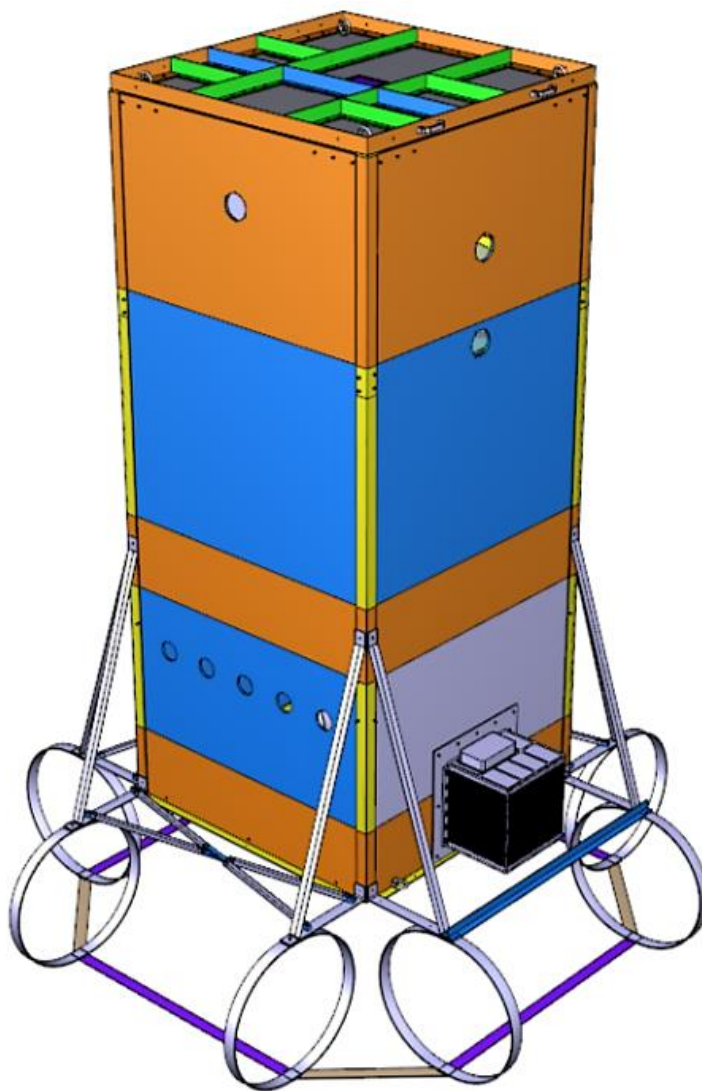


Figura 3.1. Góndola de la misión EUSO-BALLOON.

Su óptica estará formada por un sistema de 3 lentes de Fresnel de $1\text{m} \times 1\text{m}$ y una superficie focal hecha de 36 fotomultiplicadores multi-ánodos (PDM), con 64 píxeles cada uno. La matriz de fotomultiplicadores junto con su electrónica de lectura, activación y alimentación se conocen como un PDM [19] (*Photo Detector Module*). La parte de discriminación de eventos y almacenamiento de datos será realizada por un sistema que se denomina DP [10] (*Data Processor*), éste se encargará de procesar toda la información colectada por el sistema PDM. En conjunto servirá para estudiar las partículas de altas energías que inciden a la atmosfera y emiten luz ultravioleta a su paso.

3.1. Filosofía de operación

El *Housekeeping* tendrá las siguientes tareas: encender y apagar cada uno de los subsistemas con los que interactúa en el instrumento, verificar el estado de encendido/apagado de cada subsistema, supervisar los voltajes y las corrientes en las LVPS (*Low Voltage Power Supply*). El *Housekeeping* contará con una interfaz para la adquisición de señales de sensores analógicos, distribuidos en todo el instrumento, tendrá un *bus* serie para transmitir comandos e información a través de la interfaz de la CPU (*Central Processing Unit*), y otro bajo el mismo protocolo para comunicarse directamente con la estación de tierra NOSYCA (*New Operational System for the Control of Aerostat*) a través del módulo en el instrumento llamado SIREN (*Network Interface System NOSYCA*) y finalmente un *bus* dedicado para la recolección de datos de los otros subsistemas que conforman el DP. El *Housekeeping* funcionará en un modo de control lento, es decir, dentro de las escalas de tiempo del orden de 1 a unos pocos segundos.

3.2. Requerimientos y especificaciones

El *Housekeeping* estará alojado en el DP. El sistema DP se encargará de realizar las siguientes tareas:

- Control y configuración de toda la electrónica principal del sistema
- Registrar eventos de tiempos de arribo y posición de la carga
- Proveer las señales de sincronización de todo el instrumento
- Gestionar el envío y recepción de telecomandos y el sistema de telemetría
- Supervisar la integridad de los diferentes subsistemas mediante el *Housekeeping*
- Controlar el sistema de almacenamiento de datos del instrumento

EL DP está conformado por los siguientes subsistemas: CCB [21] (*Control Cluster Board*), CLKB [22] (*Clock Board*), GPS (*GPS receiver*), *Housekeeping* y LVPS-DP. Todos los subsistemas estarán colocados dentro de un gabinete *subrack* 6U con divisiones de 3U para las medidas estándar *eurocard*. Cada subsistema deberá utilizar un gabinete que cumpla las especificaciones para una integración de forma modular al *subrack* y cada gabinete debe contar con protección de EMC (*Electromagnetic Compatibility*) en la parte del panel frontal. En las figuras 3.2 y 3.3 se muestran el gabinete y los contenedores de cada subsistema respectivamente.



Figura 3.2. Estructura donde se integran los subsistemas del DP.



Figura 3.3. Gabinetes para cada subsistema del DP.

Además de la interacción con el DP, el *Housekeeping* tendrá que interactuar con los siguientes sistemas: PDMB (*PDM Board*), HVPS [23] (*High Voltage Power Supply*), SIREN y LENSES [24] (Sistema óptico del instrumento).

Otro punto importante a tomar en cuenta son las condiciones ambientales a las que estará sometido el sistema. Se tiene que asegurar que el sistema opere sin ningún problema en las condiciones de temperatura, altitud y presión propias del vuelo.

3.2.1. Condiciones de operación

Se espera que durante el vuelo se presenten las siguientes condiciones:

- Altitud: 40 km
- Presión: 3 mbar
- Rango de temperaturas: -30°C a +50°C
- Tiempo de ascenso: ~ 3 horas
- Tiempo de operación a 40 km: ~ 12 horas
- Tiempo de descenso: ~ 30 minutos

Para asegurar que el *Housekeeping* opere de manera correcta en dichas condiciones, se seleccionarán componentes que cumplan con el rango de temperaturas establecido. Además, posteriormente se deberán realizar pruebas de validación en las condiciones de presión y temperatura esperadas.

3.2.2. Interfaces de comunicación externas

Dentro del sistema DP, el *Housekeeping* tendrá que interactuar con diversos subsistemas para realizar monitoreo de sus parámetros, su diseño deberá de cumplir con limitaciones de potencia y tamaño dentro del espacio asignado para ser montado.

Arquitectura del DP e interacción con otros subsistemas

En el diagrama de la figura 3.4 se muestra la interacción del *Housekeeping* con los diversos subsistemas del instrumento. A partir de este diagrama se establecen las interfaces necesarias para lograr dicha interacción.

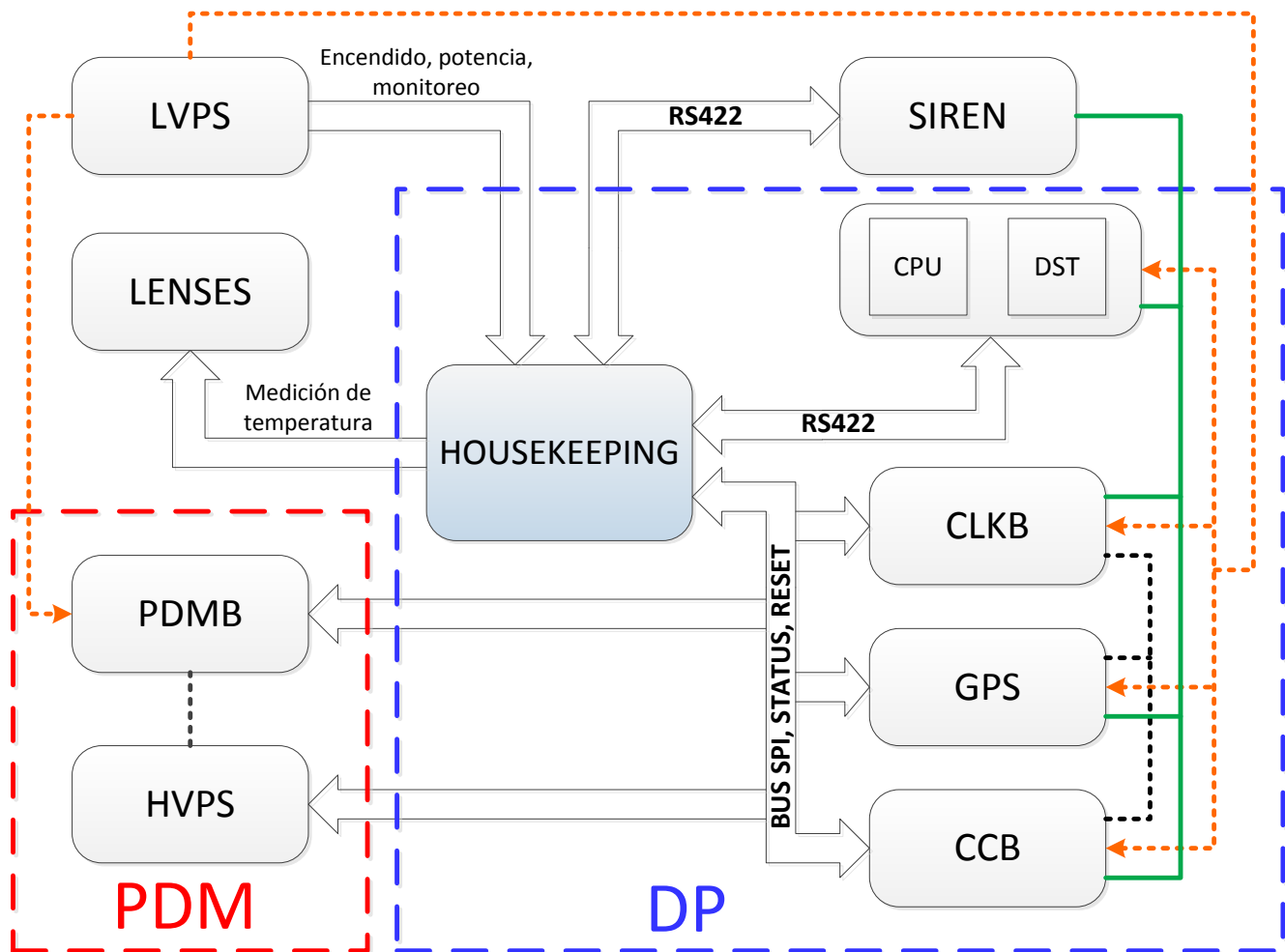


Figura 3.4. Diagrama general del DP.

CPU. La unidad central de procesamiento será una computadora, cuya placa madre contará con un procesador Intel Atom. Estará encargada de llevar a cabo las siguientes tareas:

- Decidir cuándo encender o apagar cualquier subsistema a través de comandos enviados al *Housekeeping*, quien ejecutará la acción.
- Realizar tareas de calibración periódicas del instrumento.
- Comenzar la adquisición y almacenamiento de datos. Este último a través del subsistema DST (*Data Storage*).
- Definir los modos de disparo para la adquisición de datos con CCB.
- Solicitar información al *Housekeeping* para monitorear los subsistemas alimentados por LVPS y leer alarmas provenientes de otros subsistemas.
- Tomar en cuenta el planeamiento de contingencia en tiempo real.
- Realizar descargas de información de modo periódico a través de la interfaz SIREN.

Las tareas específicas en la interacción con el *Housekeeping* son:

- Controlar el encendido y apagado de los subsistemas por medio de comandos a través de un protocolo serial.
- Recibir la lectura de todos los parámetros que el *Housekeeping* puede leer en forma periódica.
- Recibir información de alarmas de forma periódica.
- Tomar decisiones en caso de que se presenten alarmas.
- Enviar comandos de *reset* a los subsistemas o de reprogramación en caso de ser requerido.

DST. El sistema de almacenamiento de información está compuesto por un arreglo de dos unidades de estado sólido SSD (*Solid State Disk*), operando en el modo redundante a prueba de fallas RAID-1 (*Redundant Array Inexpensive Disk*). Junto con CPU, estará encargado de almacenar todos los datos importantes y recolectados a lo largo de su funcionamiento.

La tarea del *Housekeeping* con el subsistema DST será la de registrar las condiciones de temperatura para ser enviadas al sistema SIREN. Esto con el fin de que SIREN pueda decidir cuándo enviar el comando para encender el sistema DST. Esta decisión es de suma importancia, ya que se debe verificar que DST pueda operar dentro de su rango de temperatura que va de 0 a 55 °C.

SIREN. Este sistema se encarga de comunicar al instrumento con la estación de control en tierra NOSYCA. Donde todos los datos enviados a SIREN se envían por telemetría a tierra. El sistema SIREN incluye un *router* IP y varios puertos seriales configurables. El sistema en tierra y SIREN proveen las siguientes interfaces de comunicación o servicios:

- Ethernet / IP
- NTP (*Network Time Protocol*)
- Salida 1PPS (*Pulse Per Second*)
- Hasta 8 Canales asíncronos RS232 *full-duplex*
- Un Canal RS-422 síncrono para envío de telemetría a alta velocidad

Las tareas específicas en la interacción con el *Housekeeping* son:

- Verificar las condiciones de temperatura antes de enviar un comando para encender la CPU y DST. Esto lo realiza mediante el intercambio de información con el *Housekeeping* de sus sensores de temperatura.
- Funciona como un sistema redundante de CPU, para el envío de comandos al *Housekeeping*.

CLKB. Este subsistema está encargado de las siguientes tareas:

- Generar y distribuir un par de bases de tiempo de 40 MHz y 400 KHz para la sincronización de eventos en todos los dispositivos del instrumento.
- Contiene interfaces de comunicación con los subsistemas CCB, GPS y *Housekeeping*.

En cuanto a la comunicación con el *Housekeeping*, se contemplan las siguientes tareas:

- **Recolección de datos por un bus de comunicación serie**
 - CLKB envía sus valores de temperatura interna al *Housekeeping*. Éste debe ser capaz de generar alarmas en caso de que se alcancen valores críticos, además de que debe almacenar dichos valores.
 - Envío de valores de voltaje leídos internamente al *Housekeeping*. Éste debe tener la capacidad de generar alarmas al recibir valores que rebasen los límites establecidos y posteriormente almacenarlos.
 - Como requerimiento de la interfaz de comunicación serie, ésta deberá establecerse en SPI usando el estándar LVDS, a una tasa de 2 Mb/s.
- **Líneas dedicadas para generación de alarmas y reprogramación del dispositivo FPGA (*Field Programmable Gate Array*)**
 - Líneas *Done* y *Alarm*. Ambas son líneas de alarma generadas por el subsistema CLKB, *Housekeeping* debe ser capaz de tomar acciones al leer alguna de las dos alarmas.
 - Líneas *Reset* y *Program_B*. El *Housekeeping* utiliza estas líneas para enviar señales de reinicio y reprogramación en el sistema.
 - Se requiere que las líneas cumplan con el estándar LVDS.

GPS. El sistema receptor de GPS tiene las tareas de:

- Proveer una estampa de tiempo UTC (*Coordinated Universal Time*) con una precisión de 1 μ S, además de proporcionar la posición del instrumento para cada evento registrado en vuelo.
- Proveer datos correctos y señales de posicionamiento con una precisión de 10 m desde tierra hasta los 42 km sobre el nivel del mar.
- Contiene interfaces con CLKB, CPU y *Housekeeping*.

En cuanto a la comunicación con el *Housekeeping*, se contemplan las siguientes tareas:

- Leer un sensor de temperatura digital colocado en GPS a través del bus de comunicación serie.
- Leer una señal de voltaje con un ADC colocado en GPS a través del bus de comunicación serie.

CCB. Este subsistema contiene una conexión con el subsistema PDMB, a través de la cual procesa y clasifica los datos obtenidos por el sistema de fotodetectores del mismo. Además contiene interfaces de comunicación con los subsistemas CPU, CLKB y *Housekeeping*.

En cuanto a la comunicación con el *Housekeeping*, se contemplan las siguientes tareas:

- **Recolección de datos por un bus de comunicación serie**
 - Envío de valores de temperatura interna de CCB al *Housekeeping*. Éste debe ser capaz de generar alarmas en caso de que se alcancen valores críticos, además de que debe almacenar dichos valores.
 - Envío de valores de voltaje y corriente leídos internamente al *Housekeeping*. Igualmente el *Housekeeping* debe tener la capacidad para generar alarmas al recibir valores que rebasen los límites establecidos y posteriormente almacenarlos.
 - Envío de valores de registros al *Housekeeping*, que indican el estado del funcionamiento interno del FPGA.
 - Como requerimiento de la interfaz de comunicación serie, ésta deberá establecerse en SPI usando el estándar LVDS, a una tasa de 2 Mb/s.
- **Líneas dedicadas para generación de alarmas y reprogramación del dispositivo FPGA**
 - Líneas *Done* y *Alarm*. Ambas son líneas de alarma generadas por el módulo CCB, *Housekeeping* debe ser capaz de tomar acciones al leer alguna de las dos alarmas.
 - Líneas *Reset* y *Program_B*. El *Housekeeping* utiliza estas líneas para enviar señales de reinicio y reprogramación en el sistema.
 - Se requiere que las líneas cumplan con el estándar LVDS.

PDMB. Este subsistema está encargado de la detección de partículas de alta energía. Se comunica con el *Housekeeping* a través del bus de comunicación serie, intercambiando valores de temperaturas y voltajes.

HVPS. Este sistema está encargado de generar los voltajes necesarios para la operación de los fotodetectores incluidos en PDM. Contiene un esquema de circuitos multiplicadores de voltaje en el orden de 1000 V que alimentan a los fotomultiplicadores.

En cuanto a la interacción con el *Housekeeping*, se contemplan las siguientes tareas:

- **Control de voltajes de referencia para los fotomultiplicadores.** El *Housekeeping* se encarga de generar los voltajes de referencia que varían de 0 V a 2.44 V con una serie de circuitos DAC (*Digital to Analog Converter*). Estos voltajes de referencia alimentan a los multiplicadores de voltaje.

- **Control de interruptores para los fotomultiplicadores.** El *Housekeeping* debe controlar los interruptores de encendido/apagado de los fotomultiplicadores a través de una serie de expansores de puertos digitales.
- **Control por medio de comunicación serie.** El control de los dispositivos DAC y expansores debe realizarse a través del bus de comunicación serie. Los dispositivos se encuentran en el sistema HVPS.
- **Línea dedicada de alarma.** Se comparte una línea dedicada que indica el estado general del funcionamiento del sistema HVPS. El *Housekeeping* monitorea esta línea y genera una alarma en caso de un cambio de estado de dicha línea.

LVPS. Proporciona la alimentación para el *Housekeeping* y los demás subsistemas. Contiene monitores de voltaje y corriente, cuyos valores son enviados al *Housekeeping*, también contiene los relevadores que controlan el encendido de cada subsistema del instrumento. Este control se lleva a cabo por medio del *Housekeeping*.

Las tareas en las que interactúa con el *Housekeeping* son:

- Envía al *Housekeeping* señales de monitoreo de voltaje y corriente de todos los subsistemas que alimenta. Todas las señales monitoreadas se envían en forma de señales de voltaje que han sido acondicionadas para variar en un rango de 0 V a 5 V.
- Contiene un sistema de encendido/apagado de todos los subsistemas que es controlado por el *Housekeeping*, a través de una interfaz llamada *High Level Command*. Además está encargado de enviar señales que indican el correcto estado de encendido/apagado (*Contact Closure*) de los subsistemas que el *Housekeeping* usará para la generación de alarmas.

LENSES. Este módulo es el sistema óptico del instrumento, el *Housekeeping* está encargado de monitorear su temperatura a través de una serie de sensores de temperatura colocados en varias zonas del instrumento.

3.2.3. Requerimientos de potencia y montaje

La potencia máxima a consumir por el *Housekeeping* no deberá ser mayor a 6 W. Por lo anterior, debemos considerar como requerimiento de diseño 5 W.

El *Housekeeping* deberá ser colocado dentro de un gabinete de medidas europeas estándar, *eurorack*. En este gabinete se integrarán también los demás subsistemas, deberá ser

integrado en un módulo independiente, debido a su gran número de interfaces se hará una propuesta que cumpla con las especificaciones requeridas para integrarse en el *eurorack*.

Las dimensiones del módulo independiente donde estará el *Housekeeping* son:

- Panel Frontal :
 - Alto: 128.4 mm (3U)
 - Ancho: 141.9 mm (28 HP)
- Profundidad: 227 mm
- Volumen efectivo (sin contar las paredes del gabinete):
 - Alto: 120 mm
 - Ancho: 135 mm
 - Profundidad: 220 mm

3.2.4. Metodología empleada

La metodología a seguir en el desarrollo del *Housekeeping* es:

1. Análisis de los requerimientos con los diferentes subsistemas con los que se tendrá interacción y definición de diferentes interfaces que se necesitan para lograrlo.
2. Selección de los dispositivos correctos para llevar a cabo la tarea requerida, esto con base en una comparación entre las diferentes opciones disponibles en el mercado, buscando utilizar componentes COTS (*Commercial Off The Shelf*) [25] para su rápida adquisición y la minimización de costos.
3. Desarrollo de las interfaces necesarias para la comunicación con los diferentes subsistemas, identificando los dispositivos necesarios para lograr lo anterior.
4. Validación de las interfaces propuestas a través de simulaciones y pruebas de laboratorio.
5. Diseño, fabricación y verificación de los circuitos impresos necesarios para el prototipo final.
6. Caracterización y validación del prototipo a nivel de *hardware*.
7. Fabricación de bancos de prueba de los subsistemas para la implementación de un *firmware* de funcionamiento en condiciones más reales.
8. Implementación de un protocolo interno de comunicación para CPU/SIREN y *Housekeeping*.
9. Entregar un *firmware* autónomo para el dispositivo de vuelo.

3.3. Propuesta y diseño del primer prototipo del *Housekeeping*

En esta sección se aborda el diseño de la primera versión del *Housekeeping*, considerando la selección de los dispositivos necesarios a partir de los requerimientos previamente establecidos, el diseño de las interfaces requeridas para establecer la comunicación con los subsistemas, el desarrollo del *software* implicado en dicha comunicación, la simulación de interfaces y la fabricación de los prototipos en línea de producción de circuitos impresos dentro del laboratorio.

3.3.1. Selección del dispositivo principal de control

Para la selección del dispositivo principal de control se consideraron FPGAs, DSPs (*Digital Signal Processor*) y microcontroladores, sin embargo, conociendo el tipo interfaces requeridas y tomando en cuenta que para esta misión el uso de componentes de categoría militar o espacial podría comprometer los cronogramas acordados en la colaboración debido a que los tiempos de entrega para nuestro país se extendían demasiado, se decidió utilizar un microcontrolador de categoría industrial. En la tabla 3.1 se muestra una tabla comparativa con los distintos microcontroladores que se evaluaron.

Tabla 3.1. Comparación de microcontroladores.

| Microcontrolador | ATmega2560 | LM3S9B92 | RM48L940-ZWT | PIC32MX795F512 | MSP430F5438 |
|----------------------------|-------------------------|----------------------------|----------------------------|------------------------|---------------------------|
| Temperatura | -40°C a 85°C | -40°C a 85°C | -40 a 105 | -40°C a 85°C | -40 °C a 85 °C |
| Arquitectura | 8 bits | 32 bits | 32 bits | 32 bits | 16 bits |
| Frecuencia máxima | 16 MHz | 80 MHz | 200 MHz | 80 MHz | 18 MHz |
| Voltaje de operación | 5V | 5V | 3.3 V | 3.6 V | 3.6 V |
| Memoria <i>flash</i> | 256 KB | 256 KB | 3072 KB | 512 KB | 256 KB |
| RAM | 8 KB | 96 KB | 256 KB | 128 KB | 16 KB |
| EEPROM | 4 KB | No disponible | 64 KB | No disponible | No disponible |
| TIMER/ RESOLUCIÓN | 6 timers | 4 timers | 32 timers | 5 timers | 3 timers |
| ADC/RESOLUCIÓN | 16 canales / 10 bits | 2x16 canales/10 bits | 2x12 canales/12 bits | 16 canales/ 10 bits | 5 canales/12 bits |
| USART/UART | 4 | 3 | 2 | 6 | 4 |
| SSI | 1 | 2 | 2 | 3 | 4 (compartido) |
| I2C | 1 | 2 | 1 | 2 | 4 |
| GPIO | 50 | 65 | 120 | 83 | 87 |
| USB | 1 | No disponible | No disponible | 1 | 1 |
| Interrupciones internas | 57 | Número no especificado | 8 | 96 | Número no especificado |
| Interrupciones externas | 6 | | 46 | 5 | |
| WatchDog | Disponible | Disponible | Disponible | Disponible | Disponible |
| Precio aprox. | 70 dls | 130 dls | 203 dls | MXN \$1,060.00 | 149 dls |

Considerando que en el ICN se contaba con la infraestructura, materiales y experiencia en el manejo de la familia ATmega de Atmel, se seleccionó esta familia para implementar el control del *Housekeeping*.

Microcontrolador AVR como núcleo de *Housekeeping*

El microcontrolador seleccionado para ser el núcleo del *Housekeeping* cuenta con una arquitectura de 8 bits, que posee un gran número de periféricos para comunicación externa, además de ofrecer una buena relación de costo con respecto a las prestaciones que ofrece. Está basado en la arquitectura Harvard y tiene la capacidad de ejecutar instrucciones en un

solo ciclo de reloj. En la figura 3.5 se muestra un diagrama de bloques de la arquitectura Harvard del microcontrolador.

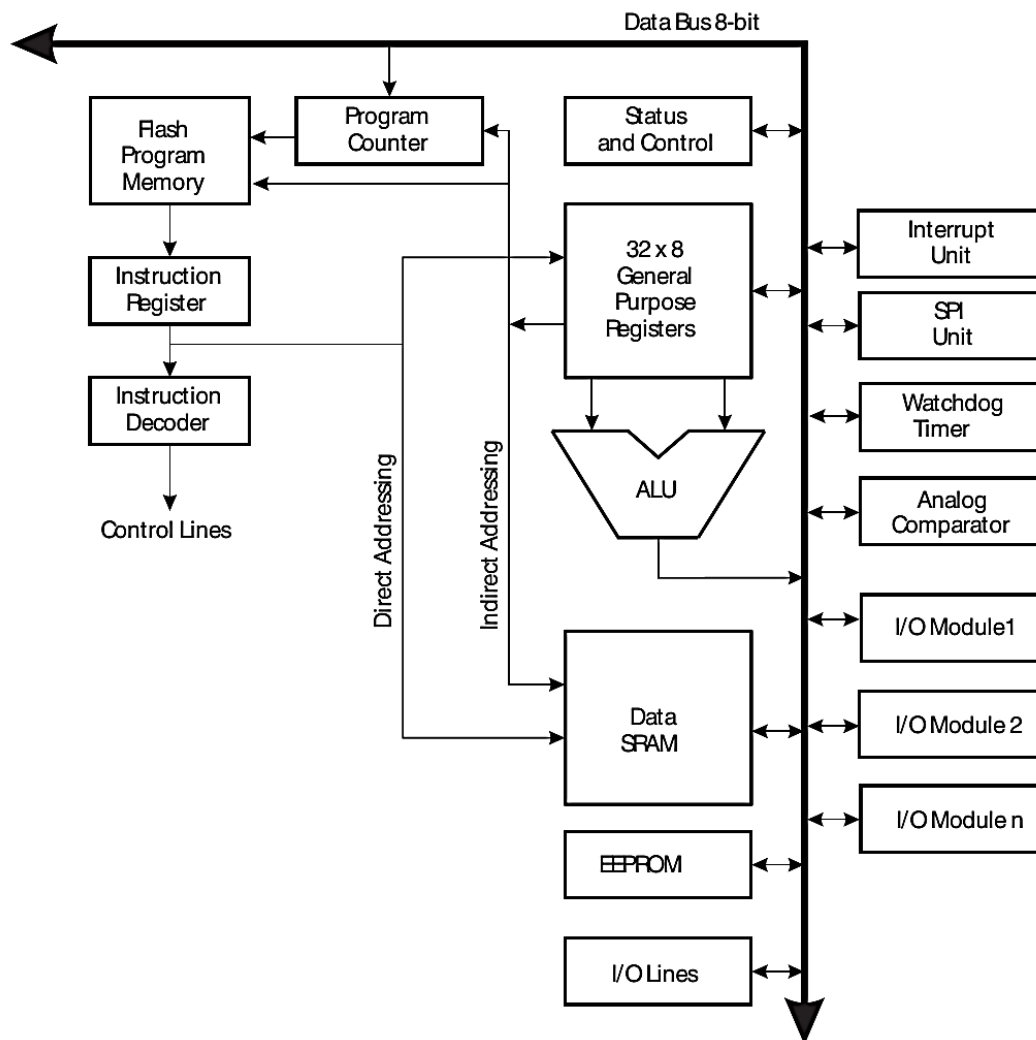


Figura 3.5. Arquitectura del microcontrolador.

Características principales:

- Frecuencia de operación máxima de 16 MHz
- Memoria *flash* de 256 kB
- Memoria SRAM (*Static Random Access Memory*) de 8 kB
- Memoria EEPROM de 4 kB
- ADC de 16 canales, con 10 bits de resolución
- 4 módulos USART (*Universal Synchronous Asynchronous Receiver/Transmitter*)
- Módulos SPI e I2C
- Dos módulos *timer* de 8 bits y cuatro de 16 bits
- 8 canales PWM de 8 bits y 6 canales PWM de 16 bits

- En total, cuenta con 86 terminales de entrada/salida
- Cuenta con los puertos PORTA, PORTB, PORTC, PORTD, PORTE, PORTF, PORTG, PORTH, PORTJ, PORTK y PORTL
- Posibilidad de programación vía ISP (*In-System Programmer*)
- 8 fuentes de interrupción externa
- Temperatura de operación de -40°C a 85°C
- Voltaje de operación de 2.7 V a 5.5 V

En la figura 3.6 un diagrama de bloques interno del microcontrolador.

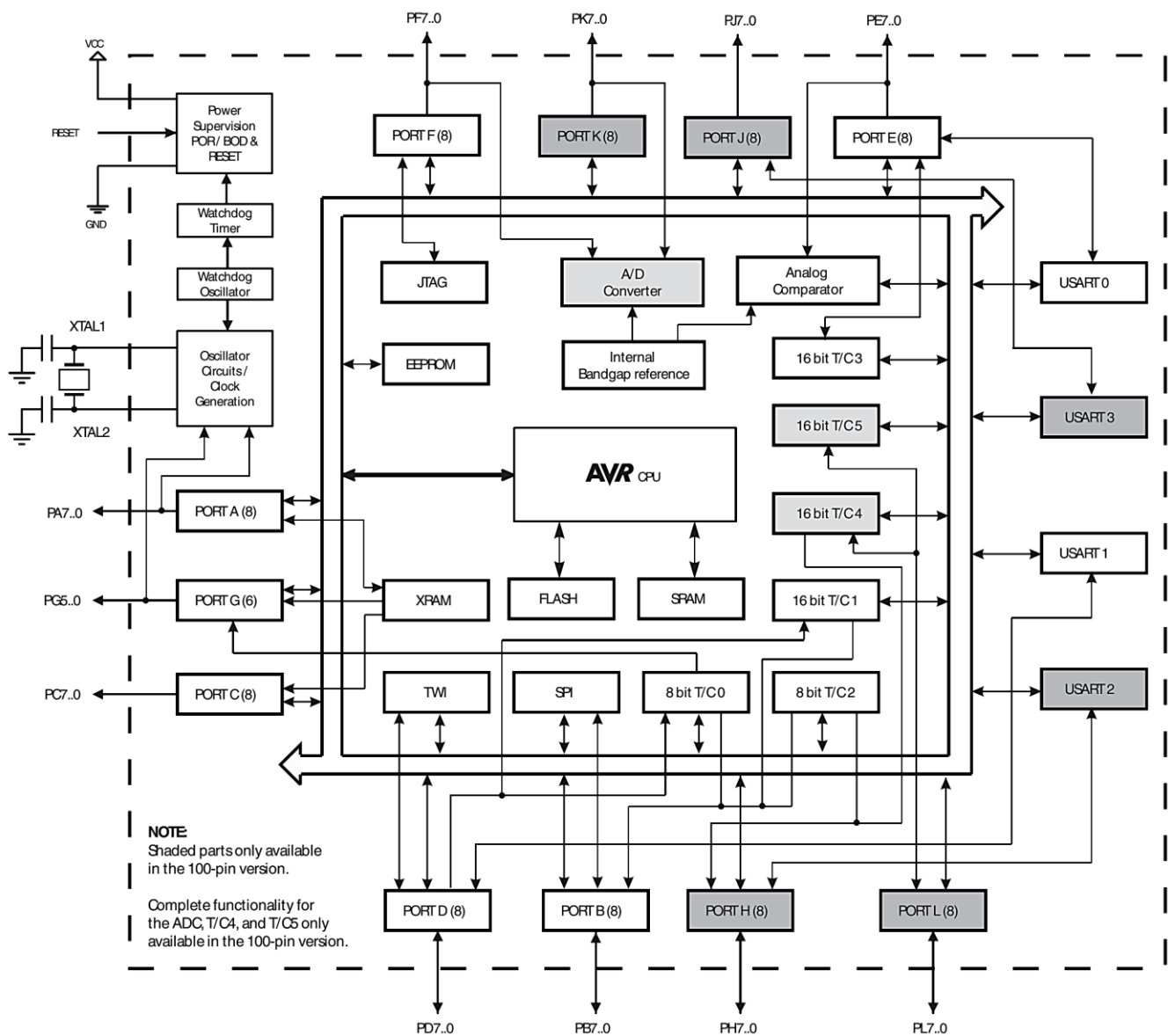


Figura 3.6. Diagrama interno del microcontrolador.

3.3.2. Diseño de las interfaces de *hardware*

En la comunicación del *Housekeeping* con los demás subsistemas se requiere contar con las interfaces: RS-422, SPI LVDS, lectura de señales provenientes de sensores de temperatura, lectura de voltajes y control de encendido y apagado de los subsistemas. En la figura 3.7 se muestra un diagrama con las especificaciones de las interfaces requeridas por subsistema.

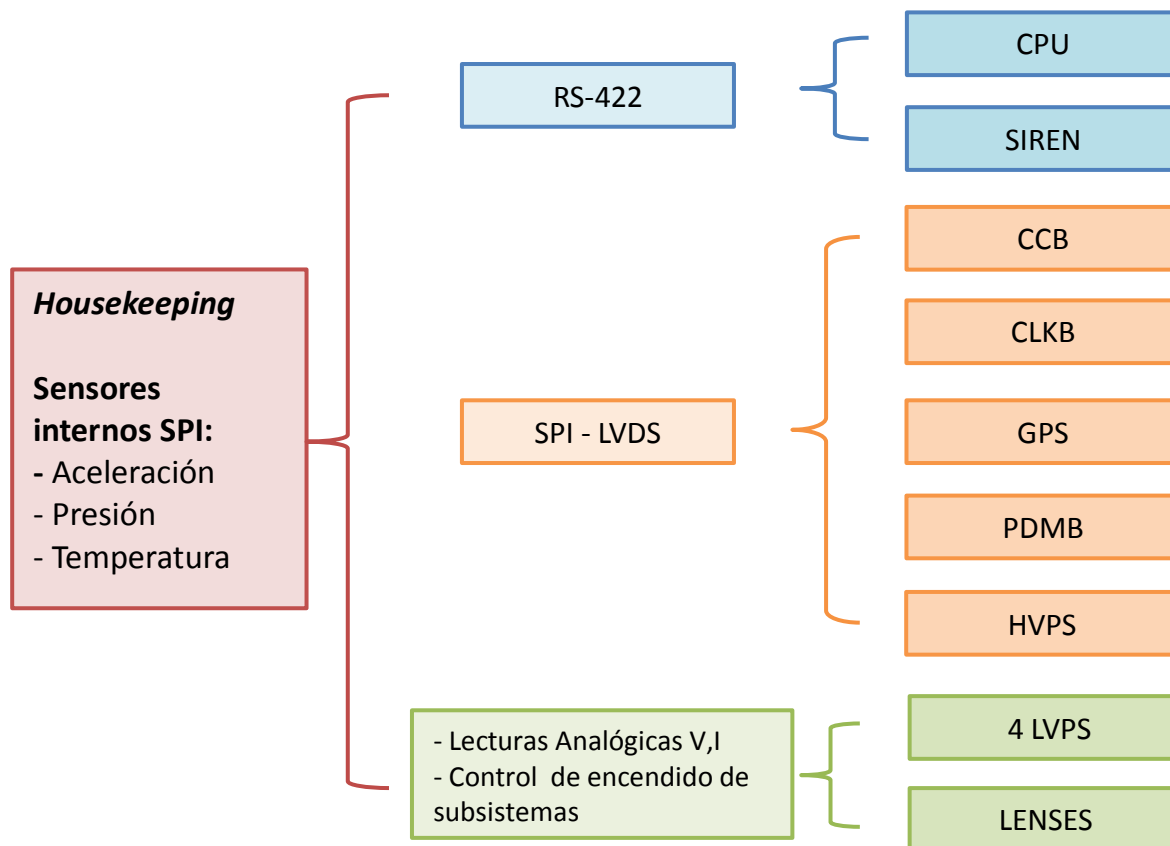


Figura 3.7. *Housekeeping* y sus interfaces externas.

Comunicación serie RS-422

Para establecer comunicación entre CPU y el *Housekeeping*, así como entre el *Housekeeping* y SIREN, se usará el estándar RS-422.

Para establecer la comunicación vía RS-422 se usará un transceptor, es decir, un dispositivo que contiene un transmisor y un receptor en un mismo encapsulado, por lo que solamente se usará uno para cada canal de comunicación usado en SIREN y otro para CPU. En la figura 3.8 se muestra la interfaz propuesta.

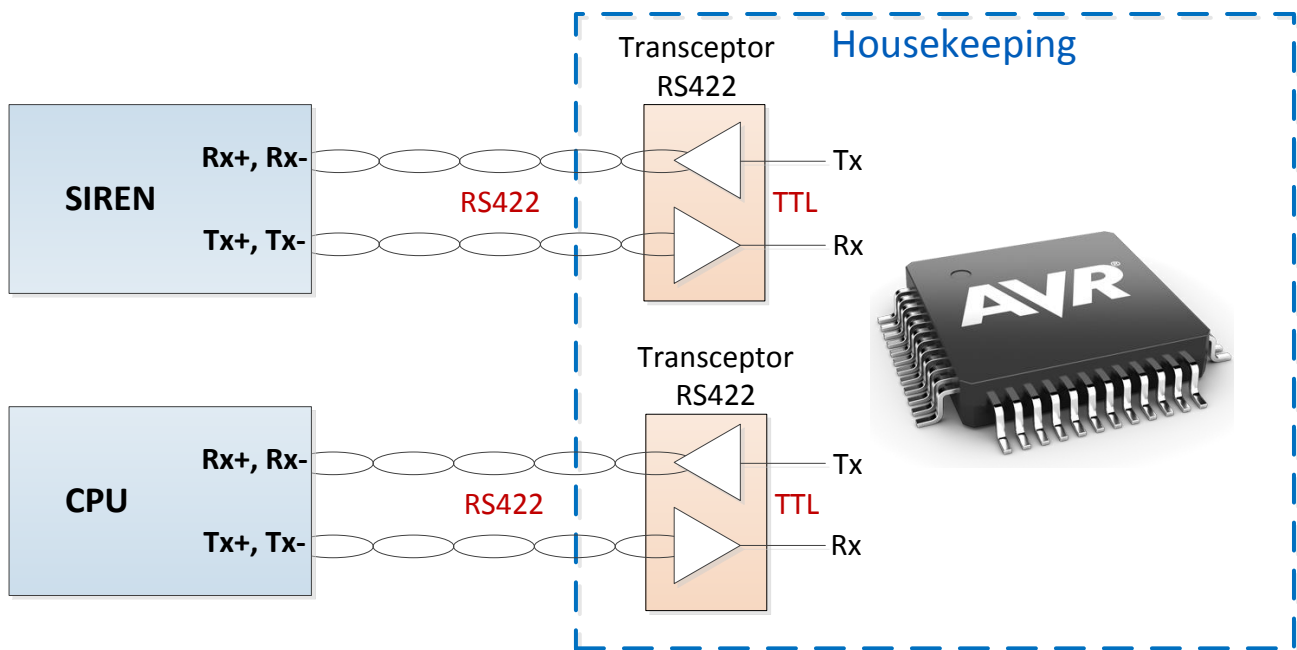


Figura 3.8. Propuesta de la comunicación con el estándar RS-422.

Comunicación diferencial SPI

Para establecer comunicación con los subsistemas CLKB, CCB, GPS, HVPS y PDMB, se hará uso del estándar SPI con una frecuencia de reloj de 2 MHz, por requerimientos del proyecto. Se usarán circuitos integrados capaces de manejar las características de la interfaz LVDS, receptores y transmisores LVDS.

Las terminales SPI del microcontrolador serán conectadas de la siguiente manera:

Las señales *MOSI*, *CLK* y *Chip Select* (*CS*) provenientes del microcontrolador se conectan a un transmisor LVDS que adecúa dichas señales para ser enviadas a los subsistemas. En la figura 3.9 se muestra la conexión de las diferentes señales para los subsistemas con interfaz SPI.

Para controlar las diferentes señales de *Chip Select* que requieren los esclavos, se hace uso de un circuito decodificador. Con el microcontrolador se manipulan las entradas de este decodificador y cada señal de salida es conectada a su respectivo transmisor LVDS. En la figura 3.10 se muestra el esquema de esta propuesta.

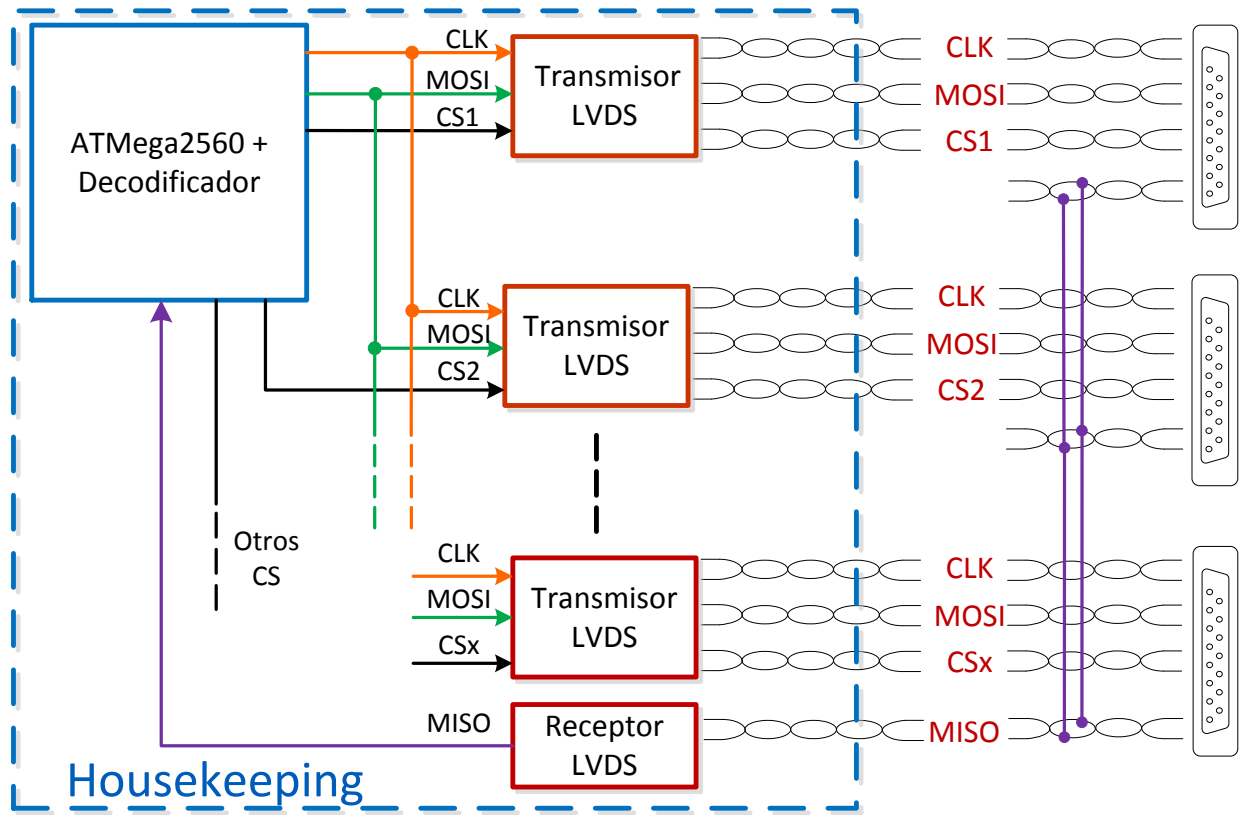


Figura 3.9. Propuesta de comunicación SPI.

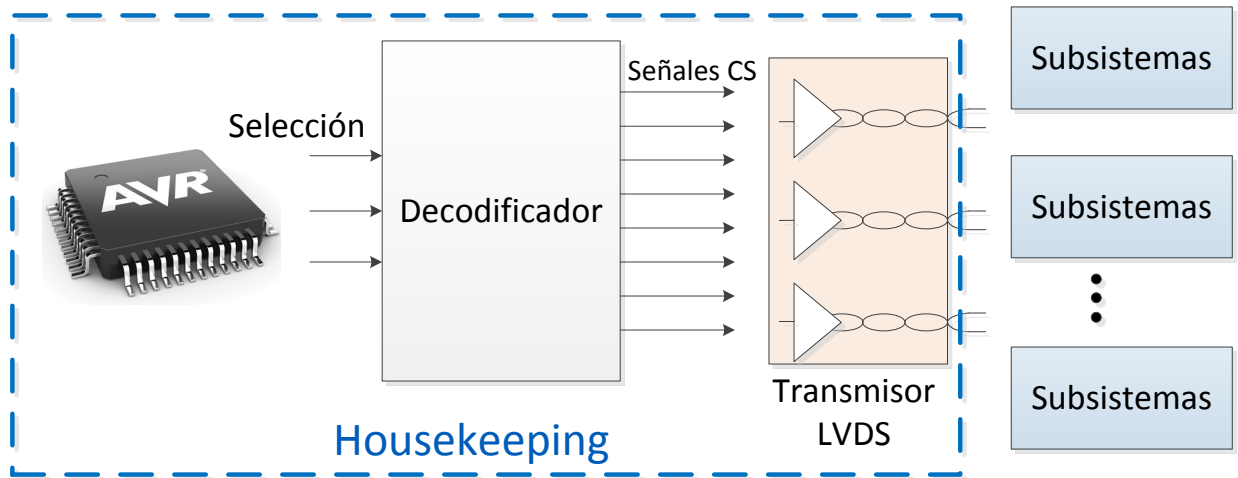


Figura 3.10. Propuesta del control de las señales *Chip Select*.

Control del encendido y apagado de los subsistemas

El circuito usado es llamado *High Level Command (HL_CMD)* y se muestra en la figura 3.11. La duración del pulso de salida depende del tiempo que se mantiene el pulso en la entrada.

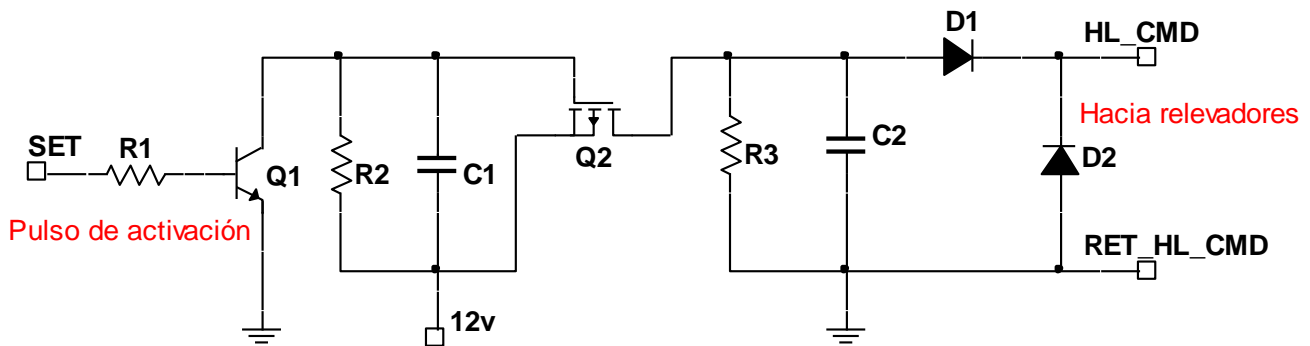


Figura 3.11. Circuito *High Level Command*.

Para encender o apagar un subsistema, se usarán dos circuitos *High Level Command*, uno es para encender y el otro es para apagar. El funcionamiento de éstos será el mismo, la diferencia es que cada uno se conecta a diferentes terminales de un relevador, uno va conectado a la terminal *set* (*ON*) y el otro a la terminal *reset* (*OFF*) del relevador. Los relevadores usados son de tipo DPDT (doble polo doble tiro) con *latch*.

Al tratarse de 7 subsistemas los que se tienen que encender y apagar, se tienen que controlar 7 relevadores, lo que da una totalidad de 14 circuitos *High Level Command*.

Debido a la cantidad de circuitos *High Level Command* que se tienen que controlar, se hace uso de un circuito decodificador, de esta manera se destinan menos terminales del microcontrolador para el encendido y apagado de subsistemas. En la figura 3.12 se muestra el esquema del control de los 14 circuitos de *High Level Command*.

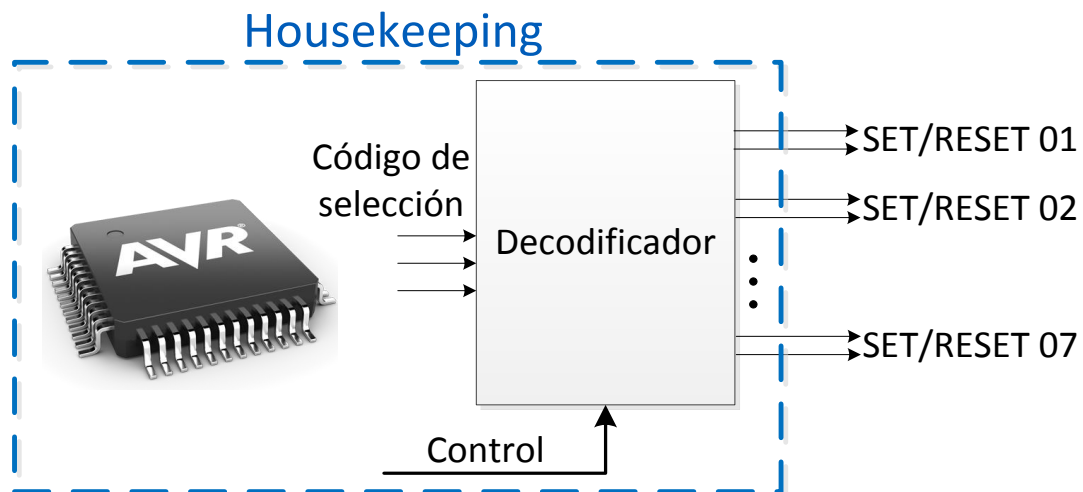


Figura 3.12. Control de los *High Level Command*.

Verificación del estado ON/OFF de subsistemas

El *Housekeeping* requiere identificar el estado de un subsistema después de que ha enviado un comando de encendido o apagado. Cuando se manda un comando de encendido, el *Housekeeping* tiene que leer como respuesta un estado “alto” en una terminal específica de los relevadores (Señal *Contact Closure*). Con esto verifica que el sistema se encendió.

Para lograr esto, se realiza la conexión que se muestra en la figura 3.13. Se tomarán 5 V de una terminal del *Housekeeping*, esta será conectada a una terminal de uno de los interruptores del relevador, la otra terminal del interruptor regresa a *Housekeeping*, para ser leída por éste. Cuando se manda a encender un subsistema, este interruptor se cierra, lo que provoca que el *Housekeeping* lea 5V. Cuando se manda un pulso de *reset*, este interruptor se abre, con este el *Housekeeping* leerá 0V, esto gracias a una resistencia de *pull down*.

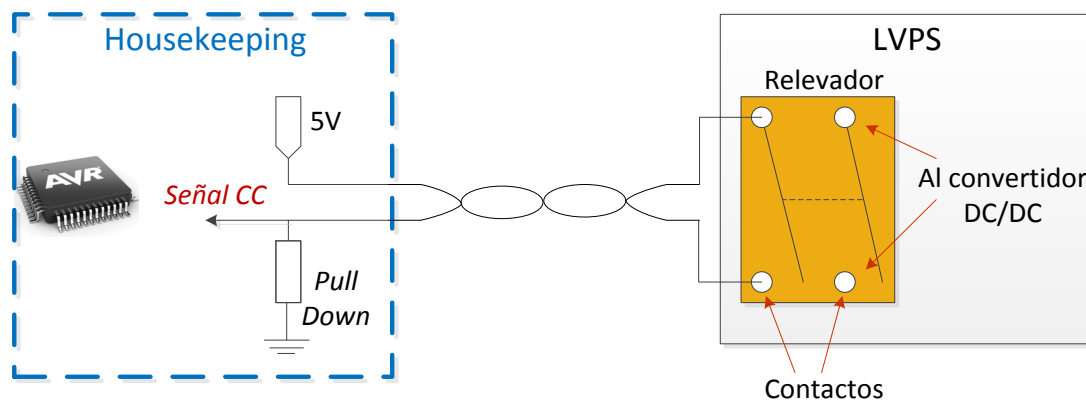


Figura 3.13. Conexión con relevadores de las LVPS.

Adquisición analógica

Para verificar el funcionamiento correcto de las LVPS, el *Housekeeping* tiene que ser capaz de monitorear el voltaje y la corriente que está consumiendo cada subsistema. Se tienen que monitorear 18 señales, la lectura de las corrientes se hará de forma indirecta, ya que la LVPS entregará un voltaje, el cual es proporcional al valor de corriente.

Debido al número de señales que se tienen que monitorear, los canales ADC del microcontrolador no son suficientes, por esta razón se utilizan multiplexores analógicos. El microcontrolador se encargará de manipular los selectores del multiplexor. Cada señal de voltaje que se recibe viene acompañada de su retorno, lo cual evita problemas de ruido al ser enviadas dichas señales desde las LVPS. La salida del multiplexor se conectará a la entrada de un amplificador de instrumentación y finalmente la salida del amplificador se conectará a un ADC del microcontrolador. En la figura 3.14 se muestra la forma en que la señal analógica es adquirida.

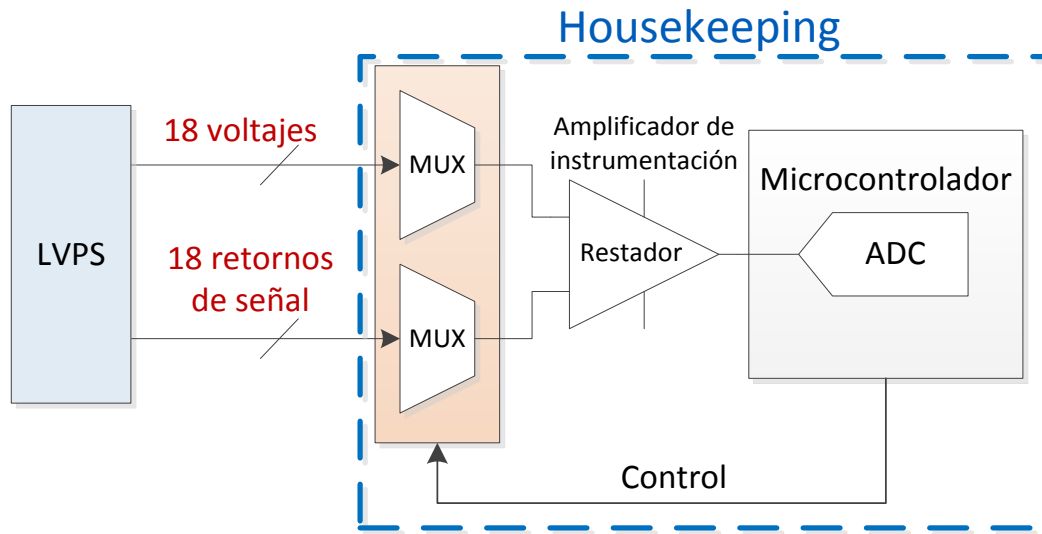


Figura 3.14. Propuesta del monitoreo analógico.

Otras variables analógicas son las correspondientes a las temperaturas en LENSES, para ello se utilizarán 9 termistores colocados en diferentes posiciones dentro del sistema de lentes. En la figura 3.15 se presenta el diagrama de conexiones.

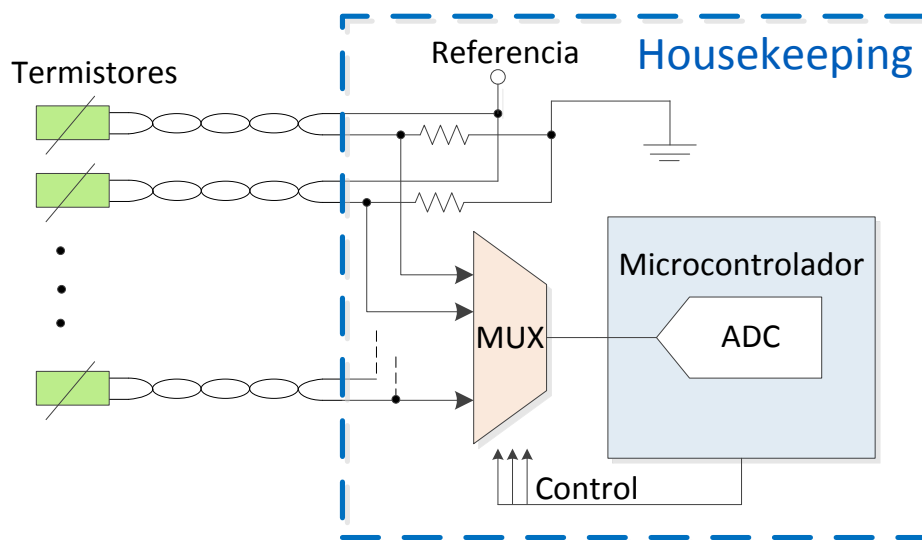


Figura 3.15. Propuesta para la medición de la temperatura.

Líneas dedicadas de subsistemas

Dedicadas a alarmas. Las líneas dedicadas a generación de alarmas son gestionadas por los subsistemas con los que hay interacción directa. Cuando ocurra alguna anomalía dentro de cada subsistema, esta línea cambiará de estado para indicar el problema. Estas líneas llegan al *Housekeeping* de forma diferencial y se conectan a un receptor LVDS que adecúa la señal para ser leída correctamente.

Se manejan dos tipos de líneas de alarma:

- **Alarm.** Esta línea de alarma cambia de estado cuando se detecte un error dentro del FPGA.
- **Done.** Es la línea de alarma más crítica que usan los sistemas con FPGAs, cambiará de estado cuando ocurra un error crítico que implique una reprogramación forzada del FPGA.

Dedicadas al reinicio y reconfiguración de subsistemas. Estas líneas son generadas por el *Housekeeping*. Sirven para el reinicio y la reprogramación de los FPGAs después de que se ha detectado algún error. Se envían a través de un transmisor LVDS.

Se distinguen dos tipos de alarmas:

- **Reset.** Esta línea se activa (cambio de estado) cuando se recibe un comando de reinicio por parte de CPU, responde a la señal de *Alarm*.
- **Program_B.** Esta línea responde a la recepción de la alarma de tipo *Done*, cuando se detecta el cambio de estado en la señal *Done*, el *Housekeeping* tiene que iniciar la reprogramación del FPGA cambiando de estado la línea de *Program_B*.

En la figura 3.16 se muestran las líneas dedicadas para los subsistemas correspondientes.

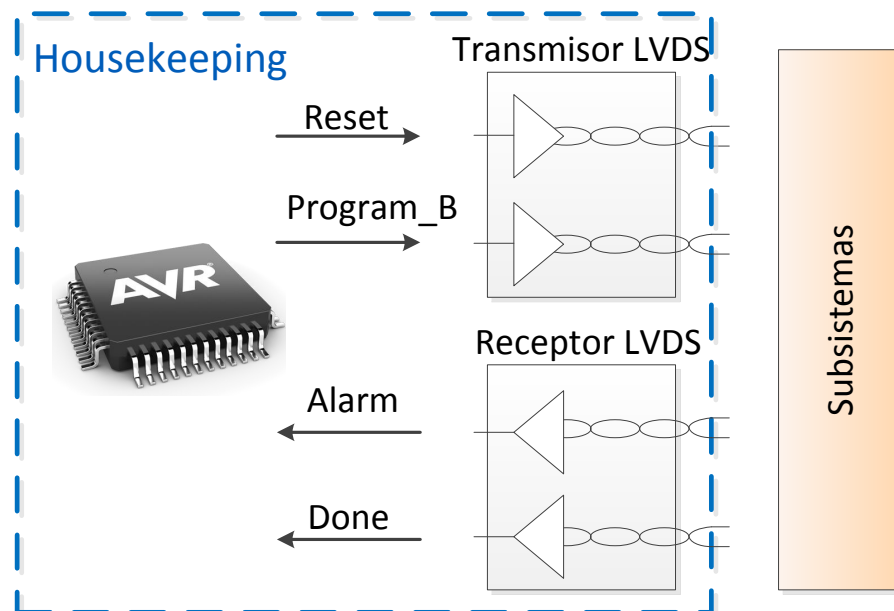


Figura 3.16. Líneas dedicadas para subsistemas.

Dispositivos internos SPI

Durante el desarrollo del *Housekeeping*, se consideró como parte fundamental de la misión tomar datos relevantes de las condiciones de operación durante el vuelo. De manera que se decidió colocar sensores dentro del *Housekeeping*, los cuales estarían tomando mediciones continuamente. Estas mediciones serán almacenadas en la tarjeta microSD del *Housekeeping* para su posterior análisis.

Los sensores colocados dentro del *Housekeeping* fueron los siguientes:

- Acelerómetro (SPI, 3 ejes, resolución 10 bits , ± 16 g escalable)
- Sensor digital de presión (SPI, 50 kPa a 115 kPa ± 1 kPa exactitud)
- Sensor digital de temperatura (SPI, rango: -55 °C a 115 °C)

En la figura 3.17 se muestran los sensores utilizados.



Figura 3.17. Sensores internos del *Housekeeping*.

Se eligieron sensores con interfaz SPI, con capacidad para operar dentro de las condiciones esperadas en vuelo. Éstos serían conectados directamente a la interfaz SPI del microcontrolador, sin necesidad de usar alguna interfaz LVDS.

Una vez que se eligieron los sensores internos que se utilizarán, se elige el tipo de cableado exterior entre el *Housekeeping* y los subsistemas.

Consideraciones acerca del cableado exterior

Un punto de suma importancia es el cableado exterior, ya que se tiene que asegurar la integridad de las señales, analógicas y digitales, que se intercambian entre los diferentes subsistemas y el *Housekeeping*. Para asegurar un intercambio de señales confiable y menos sensible a errores se hacen las siguientes consideraciones:

- Para el intercambio de señales digitales se han elegido interfaces diferenciales para conseguir mayor inmunidad al ruido, mayor tolerancia relacionada con la longitudes de cable y cierto aislamiento al no estar referidas directamente a tierra.

- Con respecto a las señales analógicas, no se intercambian señales diferenciales, pero todas las señales de monitoreo y control *ON/OFF* provenientes de LVPS vienen acompañadas de una señal de retorno. Esto es necesario, ya que para monitorear los voltajes y corrientes que entrega cada LVPS, se requiere de esta señal de retorno para no leer un voltaje no referido.

Se asegurará el uso de cableado adecuado para la conexión con los diferentes subsistemas, para ello se consideraron el cable trenzado con y sin blindaje.

Cable trenzado blindado

El cable trenzado blindado ofrece cierta inmunidad al ruido exterior, debido a que el recubrimiento que tiene actúa como una jaula de Faraday, beneficiando la integridad de la señal que se transmite. El rango de temperatura de operación de este tipo de cable es de -20 °C a 80 °C, por lo que éste cumple con los requerimientos térmicos de la misión. En la figura 3.18 se presenta una muestra del cable.



Figura 3.18. Cable trenzado blindado.

Cable trenzado sin blindaje

El cable trenzado sin blindaje, es más susceptible a las interferencias externas, con respecto al que tiene blindaje, sin embargo, aún ofrece buena integridad en las señales diferenciales, gracias a su construcción en pares trenzados. Éste también cumple con las condiciones térmicas establecidas por la misión, ya que su rango de temperatura de operación es de -20 °C a 80 °C. En la figura 3.19 se presenta una muestra del cable.



Figura 3.19. Cable trenzado sin blindaje.

Se realizó una comparación con respecto a los precios, ésta se muestra en la tabla 3.2; y con base en esta comparación se decidió usar cable trenzado sin blindaje, ya que a pesar de que el cable trenzado con blindaje parece ser una mejor opción, éste es más difícil de conseguir y el precio aumenta considerablemente.

Tabla 3.2. Comparación de precios entre los diferentes tipos de cable.

| Tipo de cable | Temperatura de operación | Precio en dólares (50 m) |
|----------------------------|--------------------------|--------------------------|
| Cable trenzado blindado | -20°C a 80°C | 472.50 |
| Cable trenzado sin blindar | -20°C a 80°C | 141.50 |

Como se optó por utilizar cable sin blindar, se compensará agregando carcasas (*backshells*) de metal en las terminaciones de los cables, para proporcionar mayor robustez e inmunidad al ruido en los cables. En la figura 3.20 se muestran los *backshells*.



Figura 3.20. *Backshells* para cables.

3.3.3. Simulación de interfaces externas

Un aspecto importante a considerar en el diseño de sistemas electrónicos es la simulación del comportamiento de los mismos. Para ello y en relación a nuestro proyecto realizamos la simulación de las interfaces de comunicación RS-422, SPI y el control de las líneas de *set* y *reset* utilizadas en las LVPS.

Simulaciones RS-422

Se verificó el comportamiento de la interfaz RS-422 por medio de una simulación realizada en el *software* PROTEUS.

La simulación se llevó a cabo con los siguientes dispositivos:

- Una versión reducida del microcontrolador utilizado en este proyecto
- Transmisor para RS-422
- Receptor para RS-422

La simulación consiste en mandar un *byte* de dato desde el microcontrolador. Por medio de los transmisores, los niveles de voltajes de la señal de salida del microcontrolador se adecúan al estándar RS-422. Posteriormente al llegar al receptor, el nivel de voltaje de la señal se convierte a TTL para ser leído por una terminal virtual de PROTEUS, el dato enviado también es leído para comparar que el dato se transmite de forma correcta. La simulación se muestra en la figura 3.21.

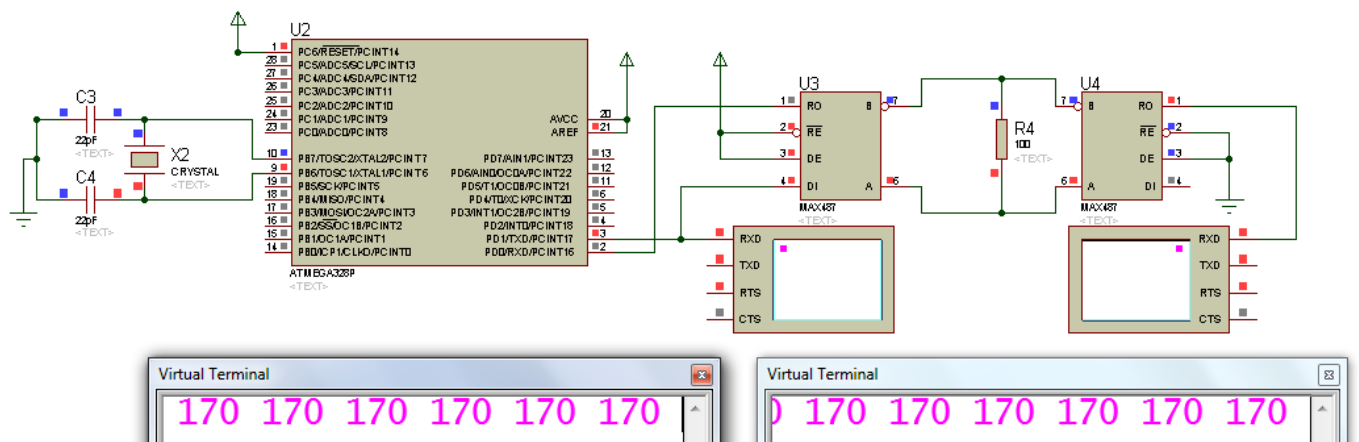


Figura 3.21. Simulación de la interfaz RS-422.

En la figura 3.22 se muestra el diagrama de flujo del programa que se implementó en el microcontrolador:

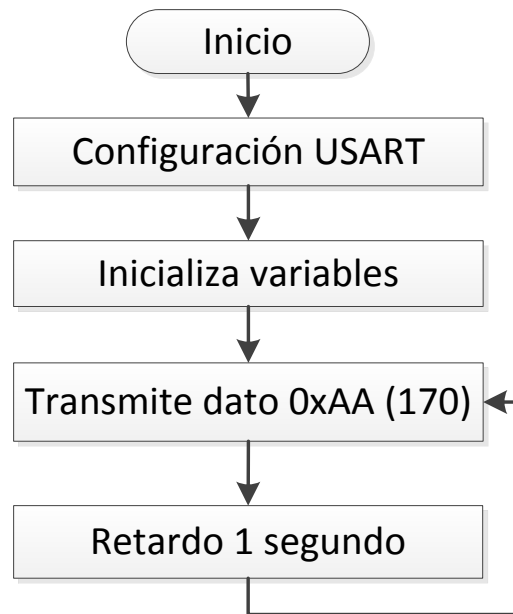


Figura 3.22. Diagrama de flujo de la transmisión serie.

Se puede observar en la figura 3.20 que en la simulación de la interfaz RS-422, el *byte* de prueba fue transmitido de forma exitosa. En la terminal de la izquierda de la simulación se puede observar el valor del *byte* que se transmite, posteriormente en la ventana de la derecha se puede observar que el valor del *byte* en el receptor es 170. La simulación de esta interfaz tuvo el comportamiento esperado.

Simulaciones SPI

Para llevar a cabo una simulación de la comunicación SPI se utilizó un microcontrolador, que actuará como maestro, un sensor de temperatura y un reloj en tiempo real, éstos últimos trabajando como esclavos.

Primero se simuló la comunicación entre el sensor de temperatura y el microcontrolador, como se muestra en la figura 3.23. El valor de temperatura que entrega el sensor en la simulación, es mostrada en un LCD. En la parte derecha de la figura se puede observar las tramas que son intercambiadas entre el microcontrolador y el sensor, así como un registro en dicho sensor que muestra el dato que será enviado al microcontrolador.

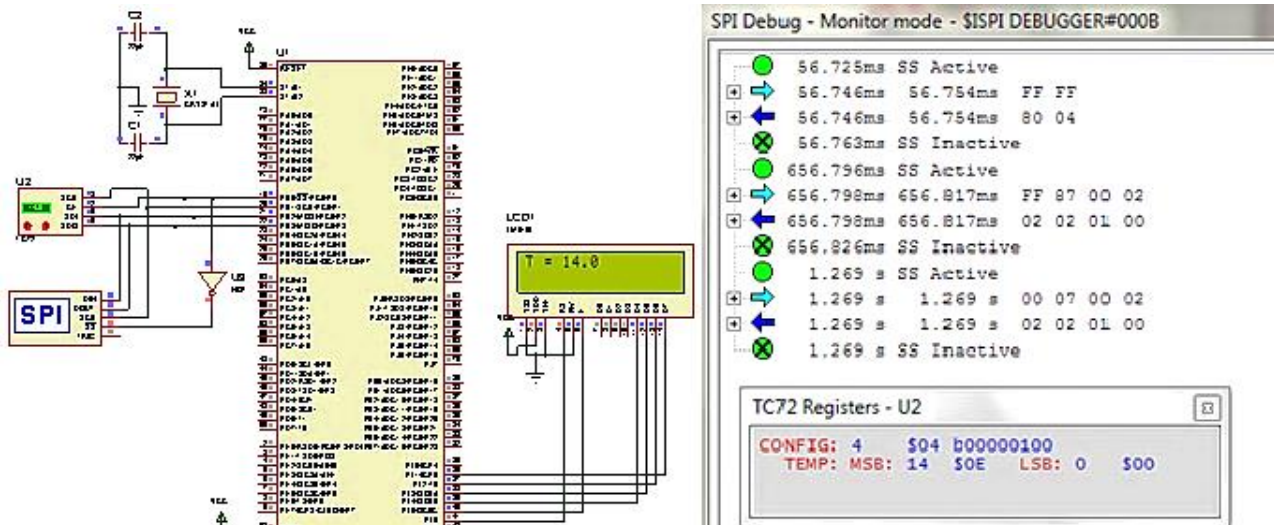


Figura 3.23. Comunicación microcontrolador - sensor de temperatura.

En esta simulación se puede observar que el intercambio de información se llevó a cabo de forma correcta. En el LCD controlado por el microcontrolador se muestra el valor de 14.0. El sensor de temperatura usa dos registros para mandar el valor de temperatura medida, uno de estos es para mandar la parte entera y el otro para la parte decimal; en la parte derecha de la imagen se muestra que se encuentran almacenados los valores 14 y 0 que componen la temperatura medida, en sus respectivos registros.

Otra de las simulaciones que se realizaron, fue la comunicación entre el microcontrolador y un reloj de tiempo real, los resultados se muestran en la figura 3.24.

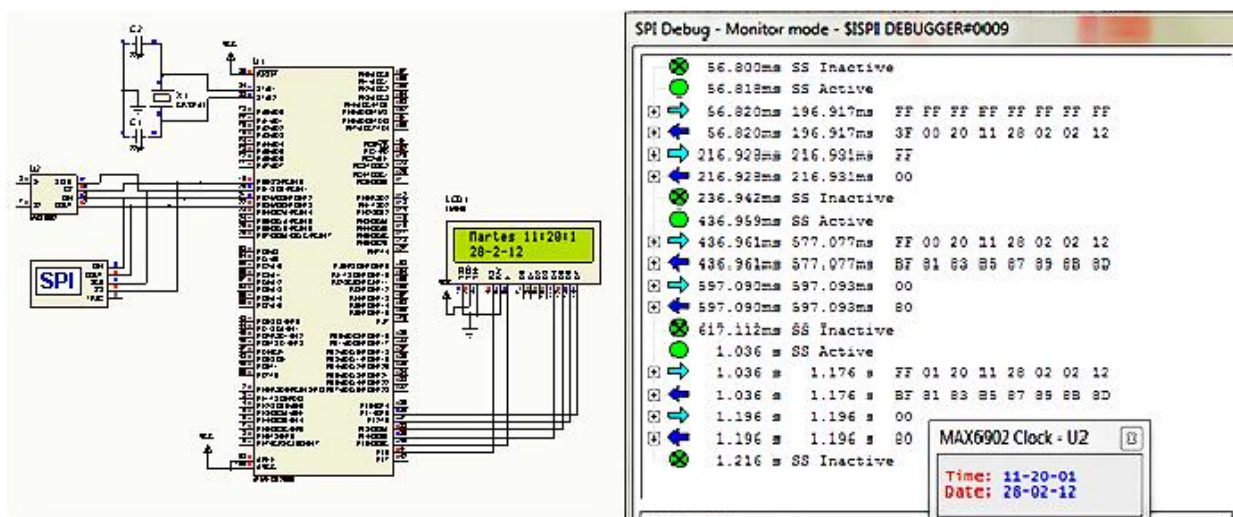


Figura 3.24. Comunicación microcontrolador – reloj de tiempo real.

En esta simulación con el reloj de tiempo real, se puede verificar que la transmisión se realizó de la forma esperada. Los datos mostrados en el LCD, son los que fueron recibidos por el

microcontrolador y éstos son los mismos que el sensor está transmitiendo, como se observa en la parte derecha de la figura 3.24.

Finalmente se simuló la comunicación con dos esclavos a la vez, el resultado es mostrado en un LCD, en la figura 3.25.

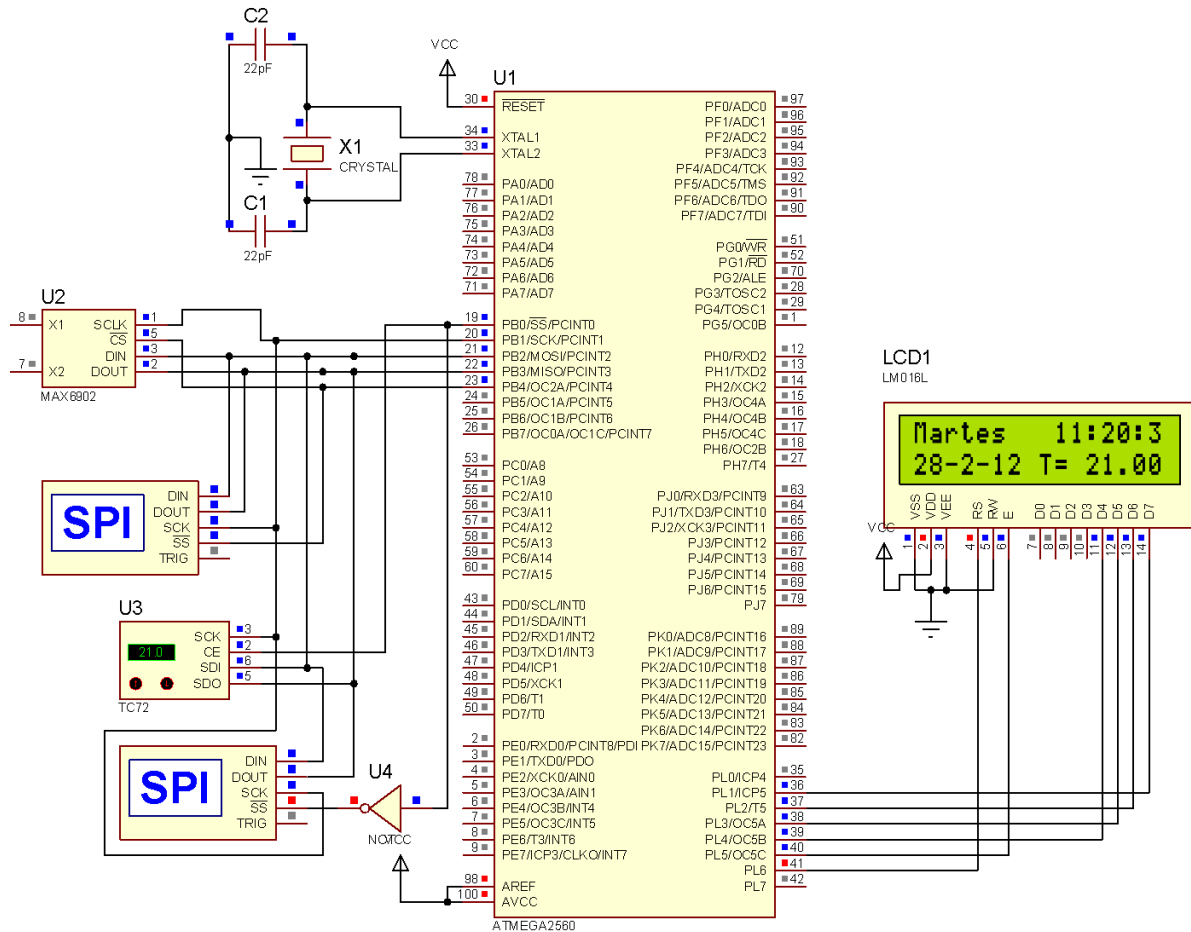


Figura 3.25. Comunicación SPI con dos esclavos.

En esta simulación el microcontrolador selecciona uno de los esclavos para establecer comunicación utilizando la línea de *Chip Select* del sensor, posteriormente se establece comunicación con el otro dispositivo. Como resultado de esta simulación se verificó que la comunicación fue posible de forma correcta con cada uno de los dos esclavos. En el LCD de la figura 3.25 se muestran los datos que los dos sensores transmiten al microcontrolador.

En la figura 3.26 se muestra el diagrama de flujo básico para comunicarse con un sensor por medio de SPI.

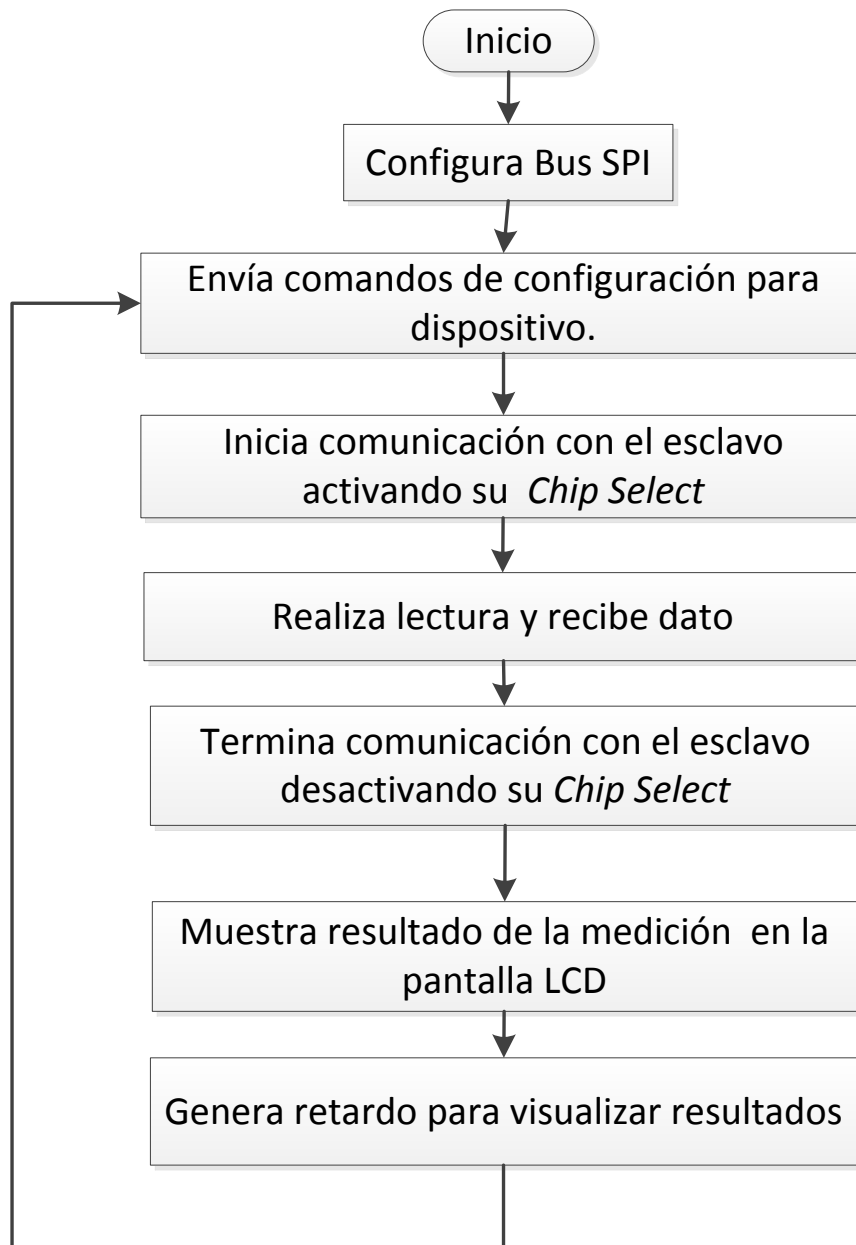


Figura 3.26. Configuración y lectura de sensor en SPI.

Simulación del control de encendido de subsistemas

El control de las señales de *set* y *reset* de cada relevador de las LVPS se simuló utilizando un circuito decodificador. Se trata de un decodificador 4:16 donde 7 líneas corresponden a un *set* y otras 7 a un *reset*. En la figura 3.27 se muestra el circuito simulado.

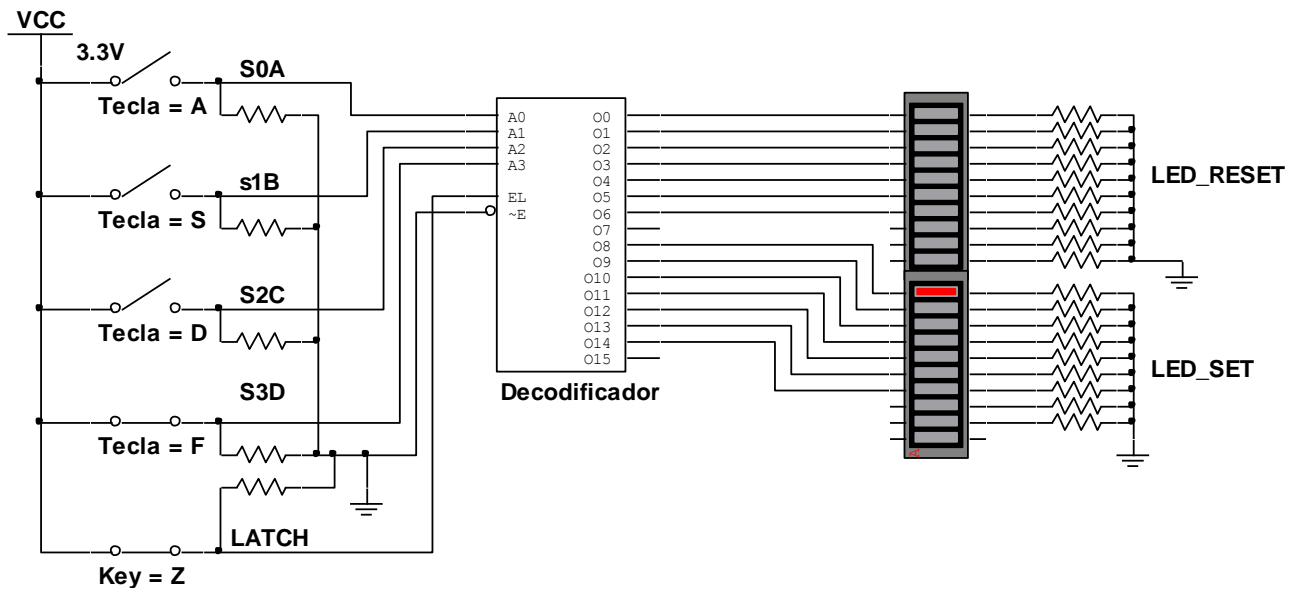


Figura 3.27. Simulación del control del decodificador.

La salida del decodificador es controlada por un habilitador y una terminal de *latch*, en la simulación se observa que permanece en un estado fijo. Así mismo, la salida será conectada a la entrada de los circuitos de *High Level Command*.

Circuito *High Level Command*

En la figura 3.28 se muestra la simulación en el *software* MULTISIM del circuito *High Level Command*.

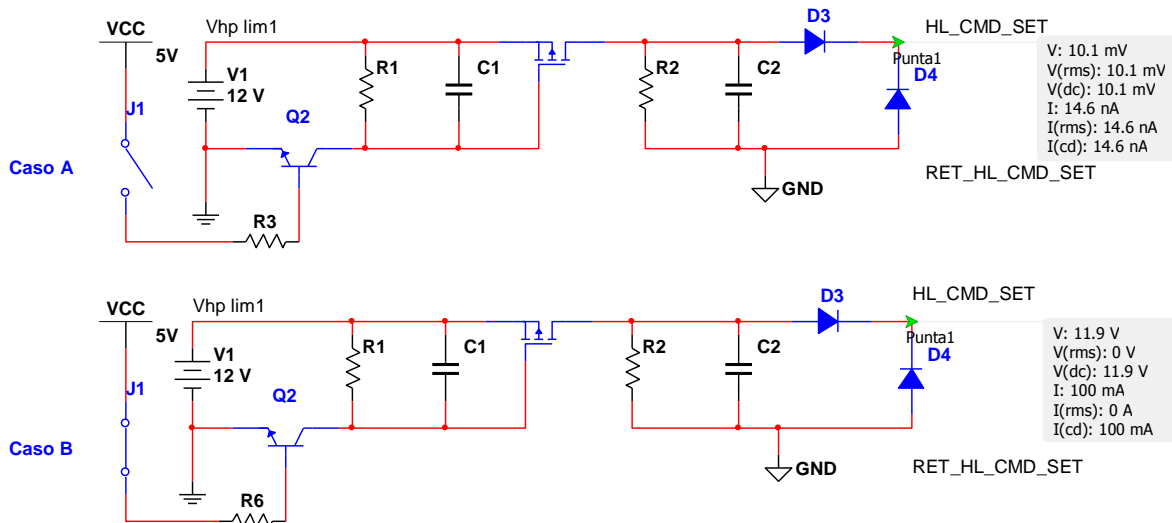


Figura 3.28. Simulación de *High Level Command*.

En esta simulación, un pulso del microcontrolador es simulado por medio del interruptor J1.

Cuando el interruptor J1 se encuentra abierto (caso A), se está simulando un cero lógico a la entrada del circuito de *High Level Command*, por lo que a la salida del circuito tendremos 0 V. Por otro lado, cuando el interruptor se cierra (caso B), se está simulando un uno lógico a la entrada del circuito de *High Level Command*, en este caso en la salida del circuito se tiene un pulso de 12 V que se encargará de cambiar de estado un relevador que controla el encendido y apagado de cada subsistema por medio de las LVPS. Cabe mencionar que la duración del pulso de salida depende del tiempo en que se presente un uno lógico en la entrada del circuito de *High Level Command*. Para el control del *reset* se sigue la misma filosofía.

Una vez que se ha comprobado el comportamiento de las interfaces en las simulaciones, el siguiente paso en el diseño electrónico es la implementación de estos circuitos.

3.3.4. Pruebas preliminares

Como pruebas preliminares se implementaron los circuitos simulados. De esta forma se hicieron pruebas a los módulos del microcontrolador de forma independiente. Con el fin de poder caracterizar cada uno, se hicieron las siguientes pruebas:

- Prueba al módulo USART.
- Comunicación serie RS-422 utilizando los transceptores propuestos y una PC.
- Prueba al módulo SPI.
- Comunicación SPI para comunicarse con sensores.
- Medición de temperatura con termistores.
- Control y encendido de relevadores utilizando decodificador.
- Activación de *Set* y *Reset* de relevadores mediante el circuito *High Level Command*.

Prueba al módulo USART

Una prueba muy importante para la selección del microcontrolador para el *Housekeeping*, fue verificar si éste es capaz de establecer una comunicación de tipo *full-duplex* con cualquier dispositivo capaz de establecerla. La prueba consistió en establecer comunicación entre dos microcontroladores, los datos que intercambiaban estos dos se midieron con ayuda de un osciloscopio digital con módulo de comunicaciones seriales. En la figura 3.29 se pueden apreciar las conexiones realizadas para la prueba.

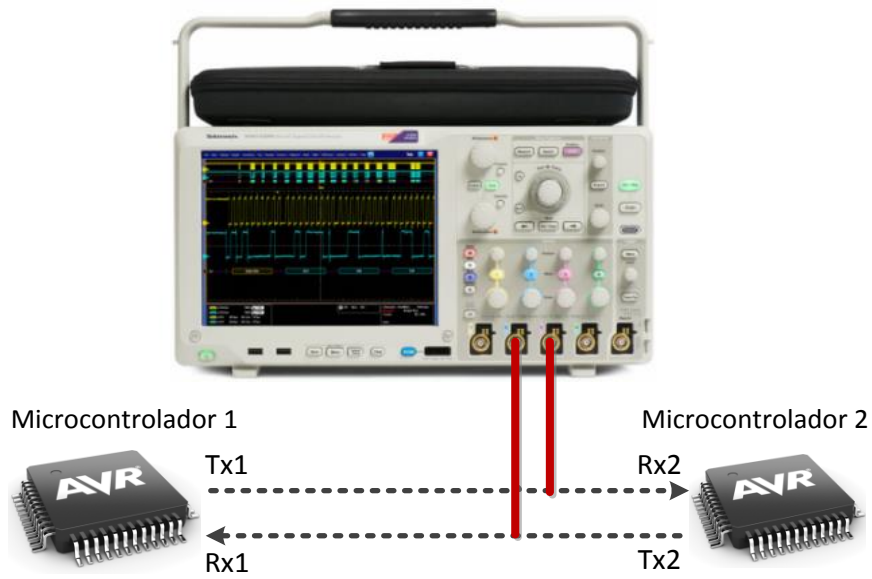


Figura 3.29. Esquema de la prueba de comunicación *full-duplex*.

El microcontrolador 1 envió una trama de 10 *bytes* con valor de 0x31 al microcontrolador 2. Simultáneamente, el microcontrolador 2 envió una trama de 10 *bytes* con valor 0x32 al microcontrolador 1. Durante este proceso se obtuvo una captura de la medición de la comunicación con el osciloscopio, mientras ésta se efectuaba a una tasa de 9600 *bauds*, el resultado se muestra en la figura 3.30. En esta figura se puede ver que los dos microcontroladores transmiten datos al mismo tiempo, con esto se comprueba que la transmisión *full-duplex* es posible con este microcontrolador.



Figura 3.30. Prueba de comunicación *full-duplex*.

Una vez que se realizó esta prueba, se procedió a implementar un circuito que es capaz de establecer los niveles eléctricos de RS-422.

Implementación de circuito de comunicación con RS-422

En esta prueba un microcontrolador se encarga de transmitir el valor que está leyendo de uno de sus canales ADC. Este valor se envía a otro microcontrolador por medio del estándar RS-422. El dato recibido es mostrado en la pantalla de una PC. La prueba se muestra en la figura 3.31.

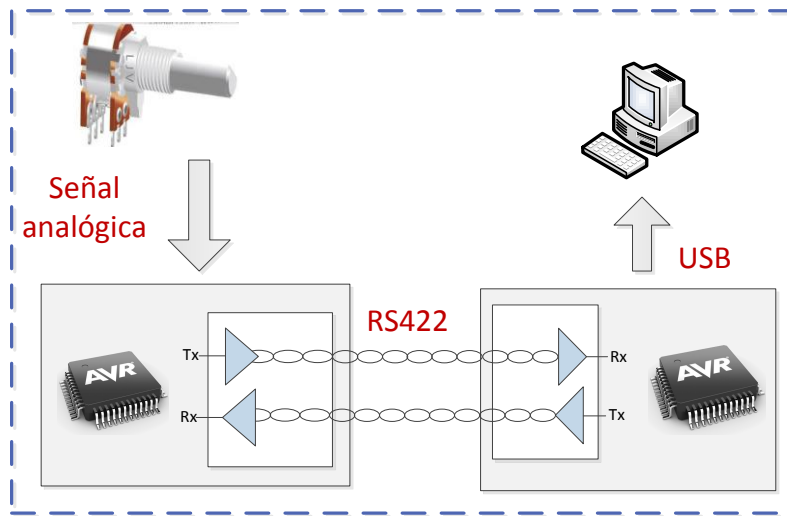


Figura 3.31. Prueba de comunicación RS-422.

En las figuras 3.32 y 3.33 se muestran los diagramas de flujo para los programas de recepción y transmisión, respectivamente.

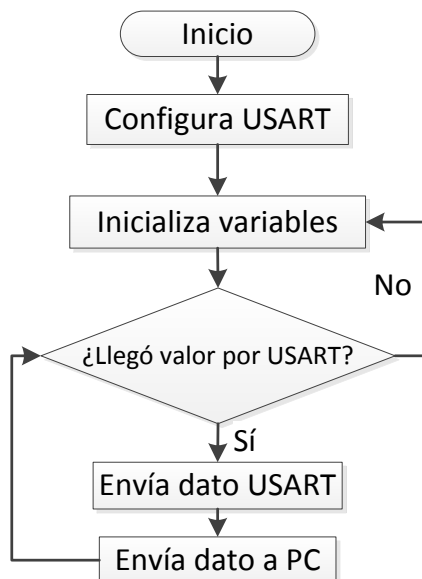


Figura 3.32. Recepción por RS-422.

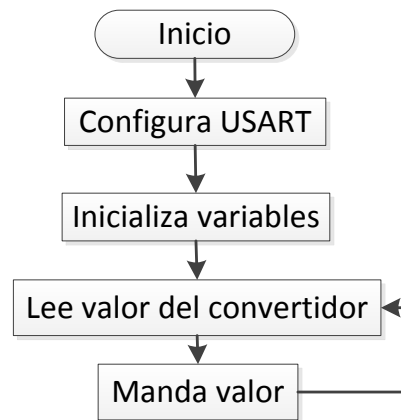


Figura 3.33. Transmisión por RS-422.

Implementación del circuito de comunicación con SPI

Una vez efectuadas las simulaciones con el estándar SPI, se implementaron las pruebas de comunicación entre el microcontrolador y algunos dispositivos que utilizan el estándar SPI.

En la figura 3.34 se muestra una tarjeta de prueba con un acelerómetro SPI. Se construyó un circuito con el microcontrolador para establecer comunicación con el sensor. Los datos obtenidos son mostrados en una pantalla de computadora, figura 3.35.

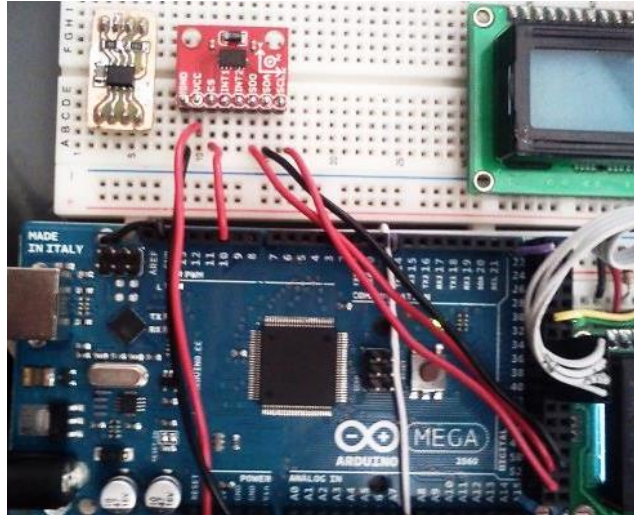


Figura 3.34. Acelerómetro SPI.

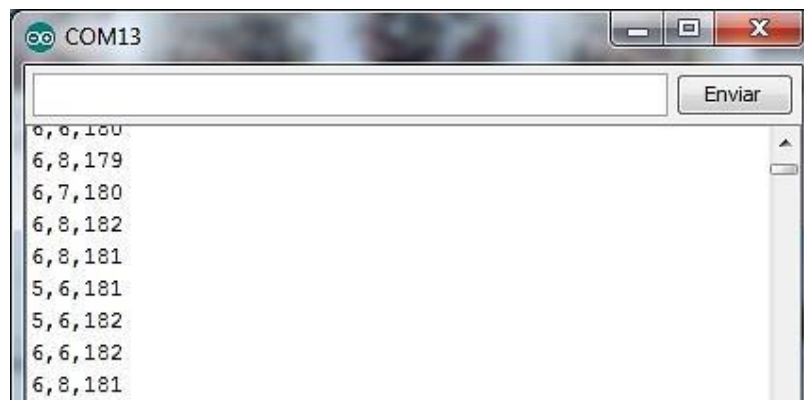


Figura 3.35. Datos del acelerómetro.

Con la intención de integrar las pruebas ya realizadas, se implementaron los circuitos SPI y RS-422 para realizar pruebas de comunicación con el microcontrolador, figura 3.36. Esta prueba consistió en comunicar un sensor de presión SPI con un microcontrolador. La información recolectada por el microcontrolador fue enviada a un puerto serie de una PC1 vía USB, a su vez esta misma información fue transmitida vía RS-422 a otra PC2, el *baud rate* utilizado fue de 115,200.



Figura 3.36. Esquema de comunicación SPI y RS-422.

En la figura 3.37 se muestra el diagrama de flujo general del funcionamiento de esta prueba.

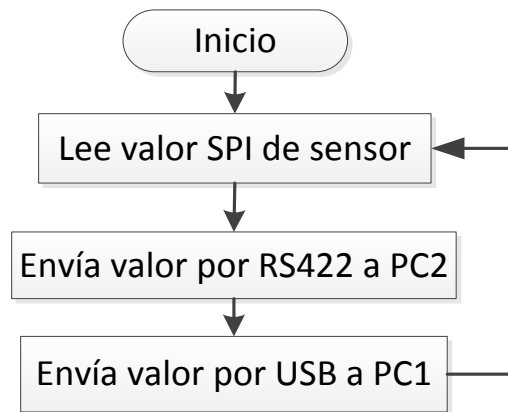


Figura 3.37. Diagrama de flujo de la prueba SPI-RS-422.

Con esta prueba se verifica el funcionamiento de las dos interfaces SPI y RS-422 en un microcontrolador.

Adquisición de señales analógicas

Para caracterizar el módulo ADC del microcontrolador, se realizó una prueba leyendo temperatura con termistores. En la figura 3.38 se muestra el diagrama de la configuración de la prueba.

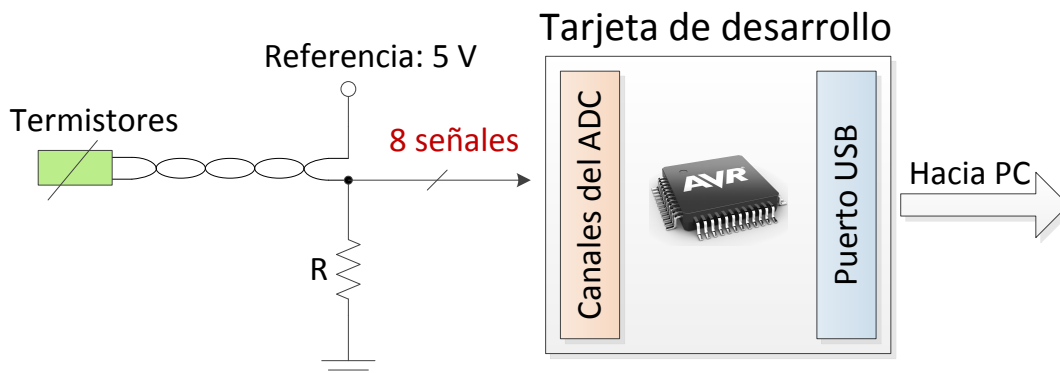


Figura 3.38. Diagrama de la configuración de pruebas del ADC.

En esta prueba se midió el valor de la temperatura ambiente con termistores, el microcontrolador se encargó de procesar la señal proporcionada por los termistores para mandarla vía USB al puerto serie de una computadora, figura 3.39.

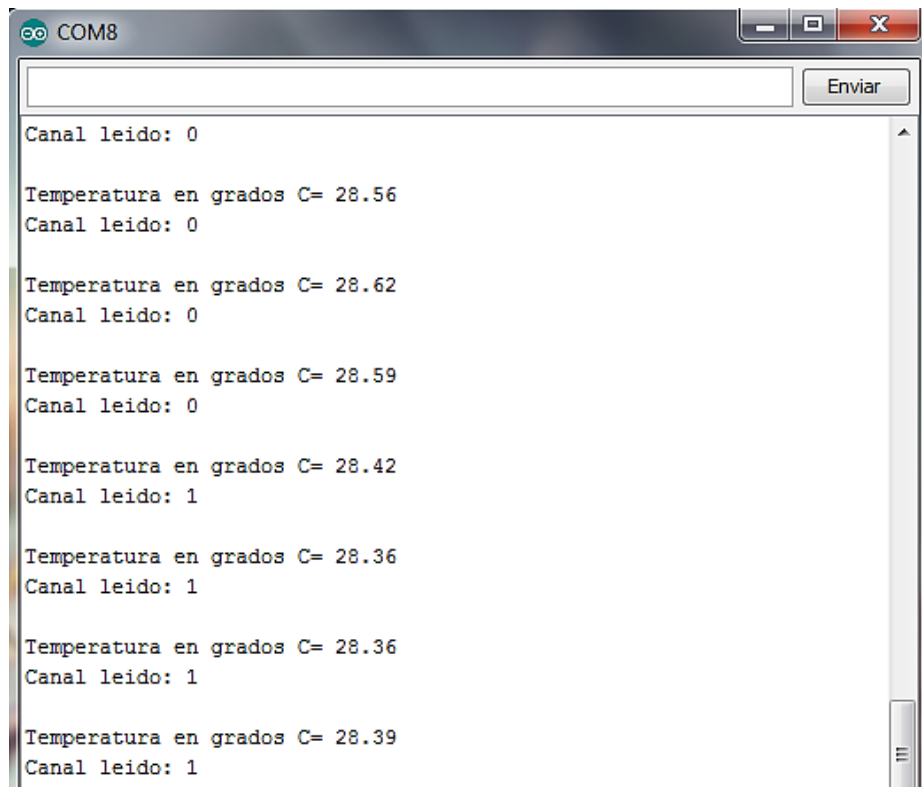


Figura 3.39. Temperatura medida.

En la figura 3.40 se muestra el diagrama de flujo del programa que realiza la lectura de la temperatura.

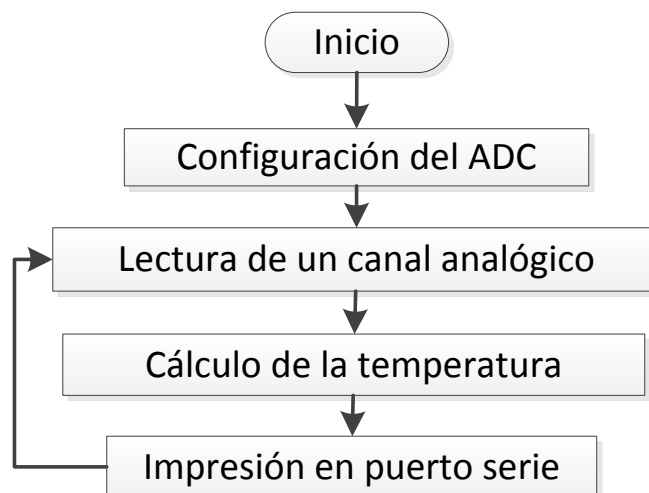


Figura 3.40. Medición de temperatura.

Al finalizar la prueba pudimos observar que los resultados fueron los esperados, al arrojar un valor cercano a 28 °C, temperatura ambiente.

High Level Command

En esta parte se implementaron dos circuitos de *High Level Command*, uno para enviar la señal de *set* y el otro para enviar la señal de *reset*, al relevador. El diagrama de bloques de la prueba se muestra en figura 3.41.

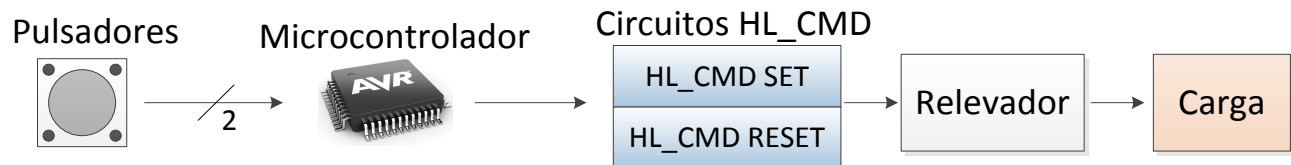


Figura 3.41. Prueba de *High Level Command*.

La prueba en general consistió en controlar los dos circuitos de *High Level Command* con un microcontrolador y con dos pulsadores conectados a éste último. El microcontrolador era capaz de detectar qué pulsador fue presionado para mandar el pulso correspondiente. La función de los pulsadores fue establecer qué pulso se iba a mandar al relevador. Cada *High Level Command* se conectó a las terminales *set* y *reset* de un relevador de tipo DPDT con *latch*.

Como primera prueba se conectaron dos LEDs indicadores como carga al relevador, para verificar el correcto envío de las señales *set* y *reset*. La prueba se desarrolló de la siguiente manera: Al presionar el pulsador asignado para mandar un *set*, el microcontrolador activó el circuito de *High Level Command* encargado de mandar un *set* al relevador, de esta manera el LED indicador de la señal de *set* se enciende, mientras que el LED indicador de la señal de *reset* se apaga. Por otra parte, cuando se presionó el pulsador asignado para enviar un *reset*, el LED indicador de la señal de *reset* se encendió, apagándose el LED correspondiente a la señal de *set*. En la figura 3.42 se muestra los circuitos usados.

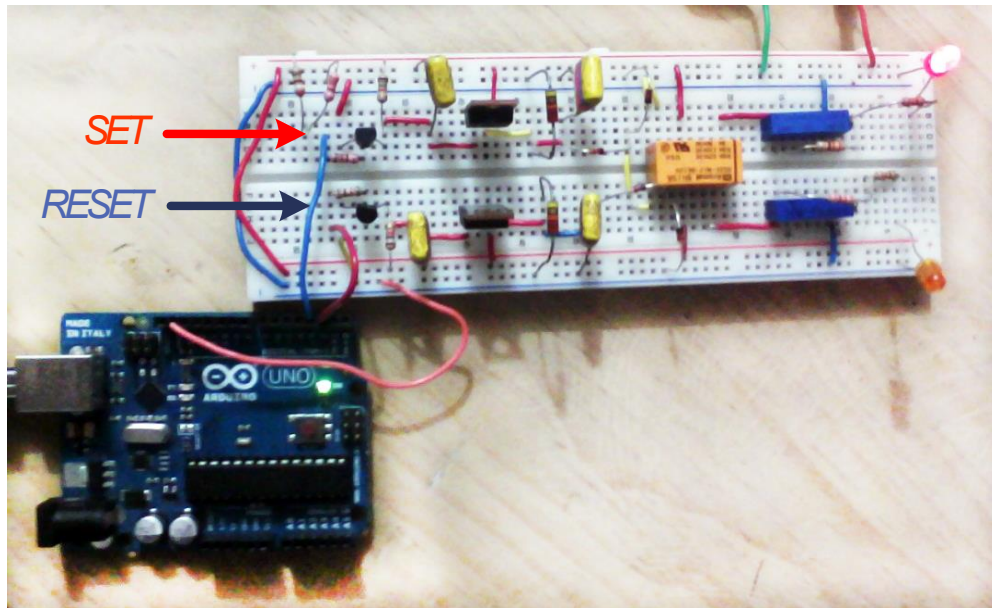


Figura 3.42. Circuito *High Level Command*.

En la segunda prueba que se realizó, se cambió la carga que tenía el relevador y se colocó un circuito de prueba para verificar que se energizara correctamente.

Al presionar el pulsador asignado para mandar un *set*, el microcontrolador activó el circuito de *High Level Command* encargado de mandar un *set* al relevador, de esta manera el microcontrolador y el circuito de prueba fueron encendidos. Por otra parte, cuando se presionó el pulsador asignado para mandar un *reset*, el microcontrolador y el circuito de prueba fueron apagados. En la figura 3.43 se muestra la prueba en el orden siguiente: circuito de *High Level Command*; circuito de prueba.

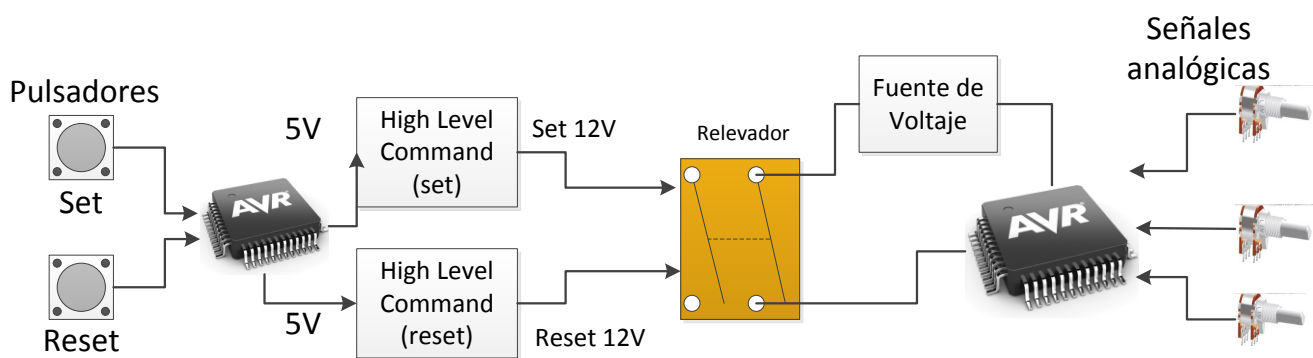


Figura 3.43. Circuito de prueba para *High Level Command*.

En la figura 3.44 se muestra el diagrama de flujo del programa con el que se probó la interfaz *High Level Command*.

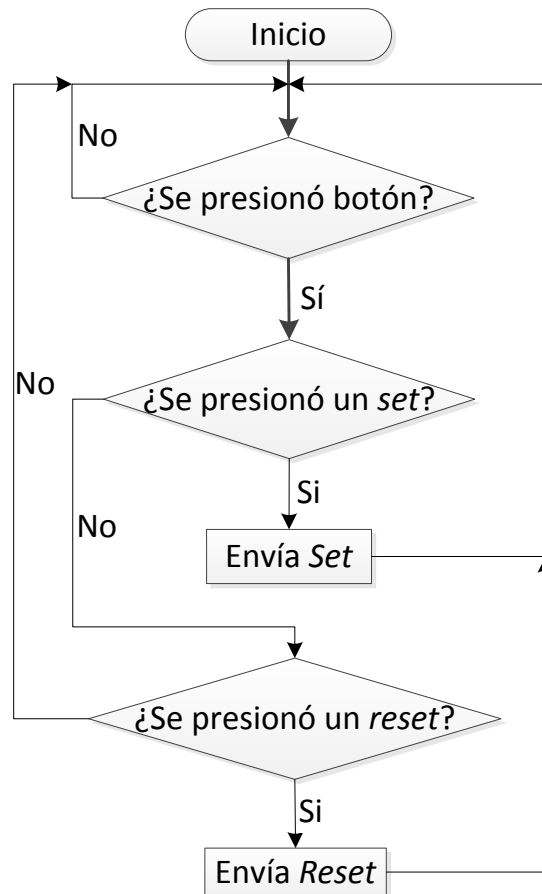


Figura 3.44. Prueba de *High Level Command*.

Al realizar esta prueba, el circuito tuvo el comportamiento esperado y un resultado parecido al de la simulación realizada previamente, con esto el encendido y apagado de subsistemas fue implementado de forma correcta.

3.3.5. Diseño de *software*

El desarrollo de las pruebas preliminares, así como la simulación de las interfaces externas, implicó el desarrollo de *software* para llevarlas a cabo, por lo que el *software* desarrollado fue para:

- Comunicación RS-422
- Comunicación SPI
- Control de encendido y apagado de subsistemas
- Adquisición analógica

Los bloques de *software* anteriores se integraron con el objetivo de dar una estructura general capaz de cumplir con la filosofía de operación. El alcance de esta primera etapa se presenta en el diagrama de flujo contenido en la figura 3.45.

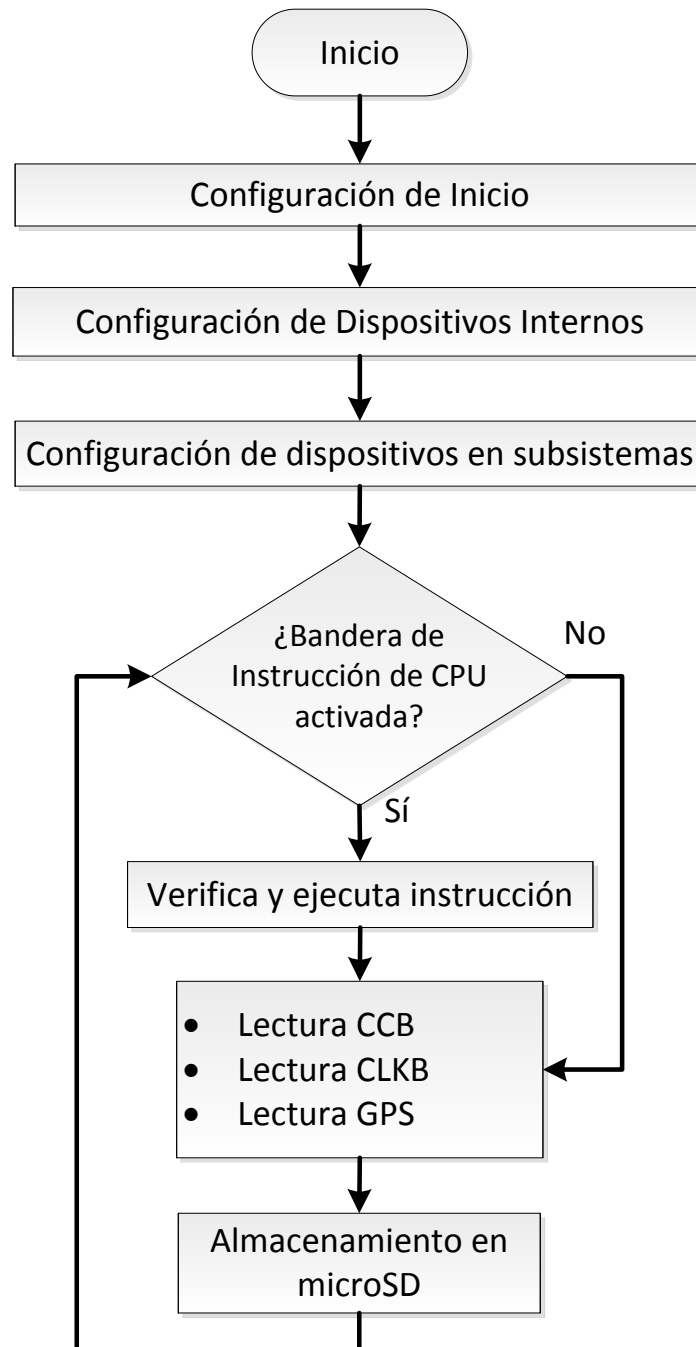


Figura 3.45. Proceso principal del *Housekeeping*.

La primera acción después de encender el *Housekeeping* es hacer la configuración de inicio, las acciones realizadas en esta etapa se definen en la figura 3.46.

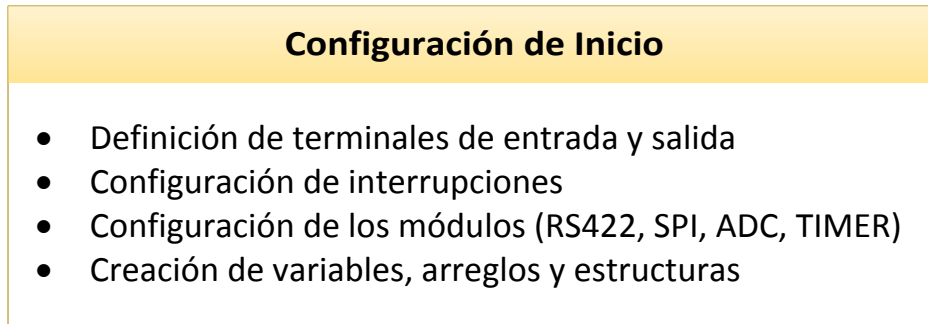


Figura 3.46. Configuración de inicio del *Housekeeping*.

Posteriormente se hace una configuración de los sensores internos y externos, las tareas realizadas en estas etapas se enlistan en las figuras 3.47 y 3.48 respectivamente. Se debe resaltar que en esta versión los sensores fueron configurados para adquirir sus valores de forma constante. Además se configura la microSD para poder almacenar los datos mediante el bus SPI.

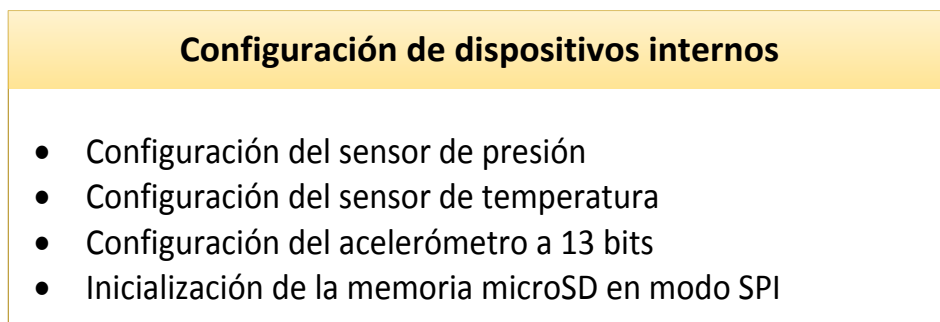


Figura 3.47. Configuración de dispositivos internos del *Housekeeping*.

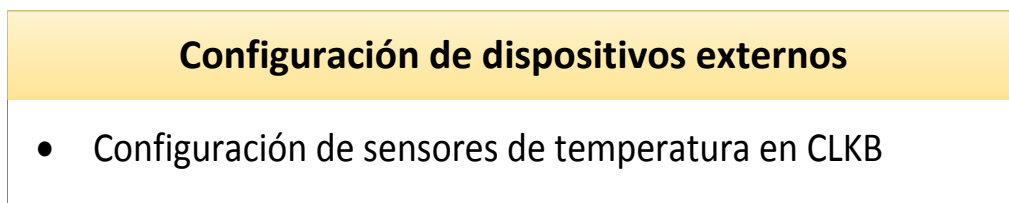


Figura 3.48. Configuración de dispositivos externos del *Housekeeping*.

Después de ejecutar la configuración de los dispositivos se verifica el estado de la bandera de instrucción de CPU, la cual es activada si se presenta una interrupción de recepción de datos en la USART, ésta tiene su propia rutina y se presenta en la figura 3.49.

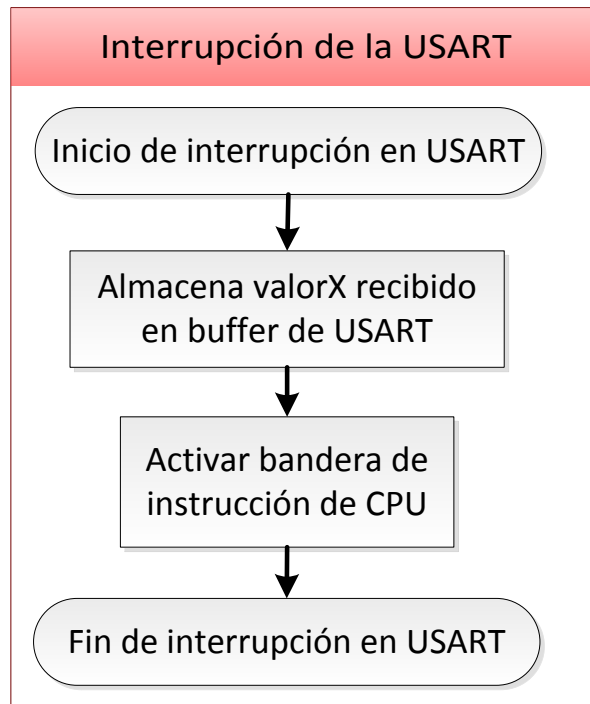


Figura 3.49. Interrupción de USART.

Al llegar un dato a una de las USART del microcontrolador, inicia su rutina de interrupción, la cual consiste en almacenar los datos que llegan de forma consecutiva en un arreglo, al mismo tiempo se pone en alto una bandera que indica que ha ocurrido una interrupción. Si la bandera de interrupción está activada se procede a leer los valores almacenados en el arreglo durante la rutina de interrupción y se ejecuta alguno de los casos mostrados en la figura 3.50, en caso contrario se informa que hay un error, posteriormente se desactiva la bandera.

Posteriormente se realizan las lecturas de CCB, CLKB y GPS. El procedimiento para leer estos subsistemas se presenta en las figuras 3.51 y 3.52. Consiste en enviar comandos y recibir respuesta por parte de los subsistemas (en comunicación *full-duplex*), almacenar la respuesta y operarla matemáticamente. Al final se muestran valores para verificar su valor

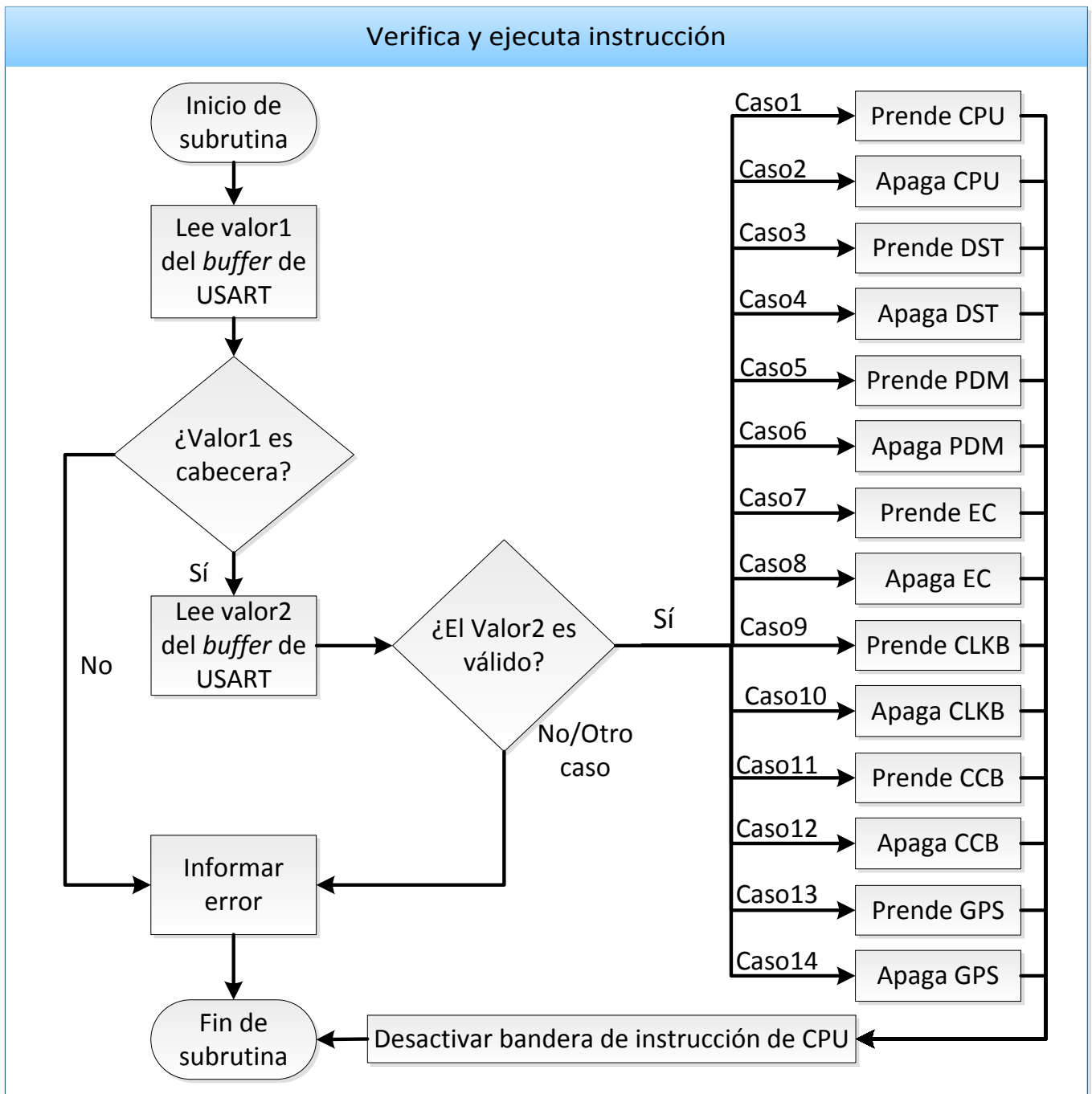


Figura 3.50. Instrucción de CPU a *Housekeeping*.

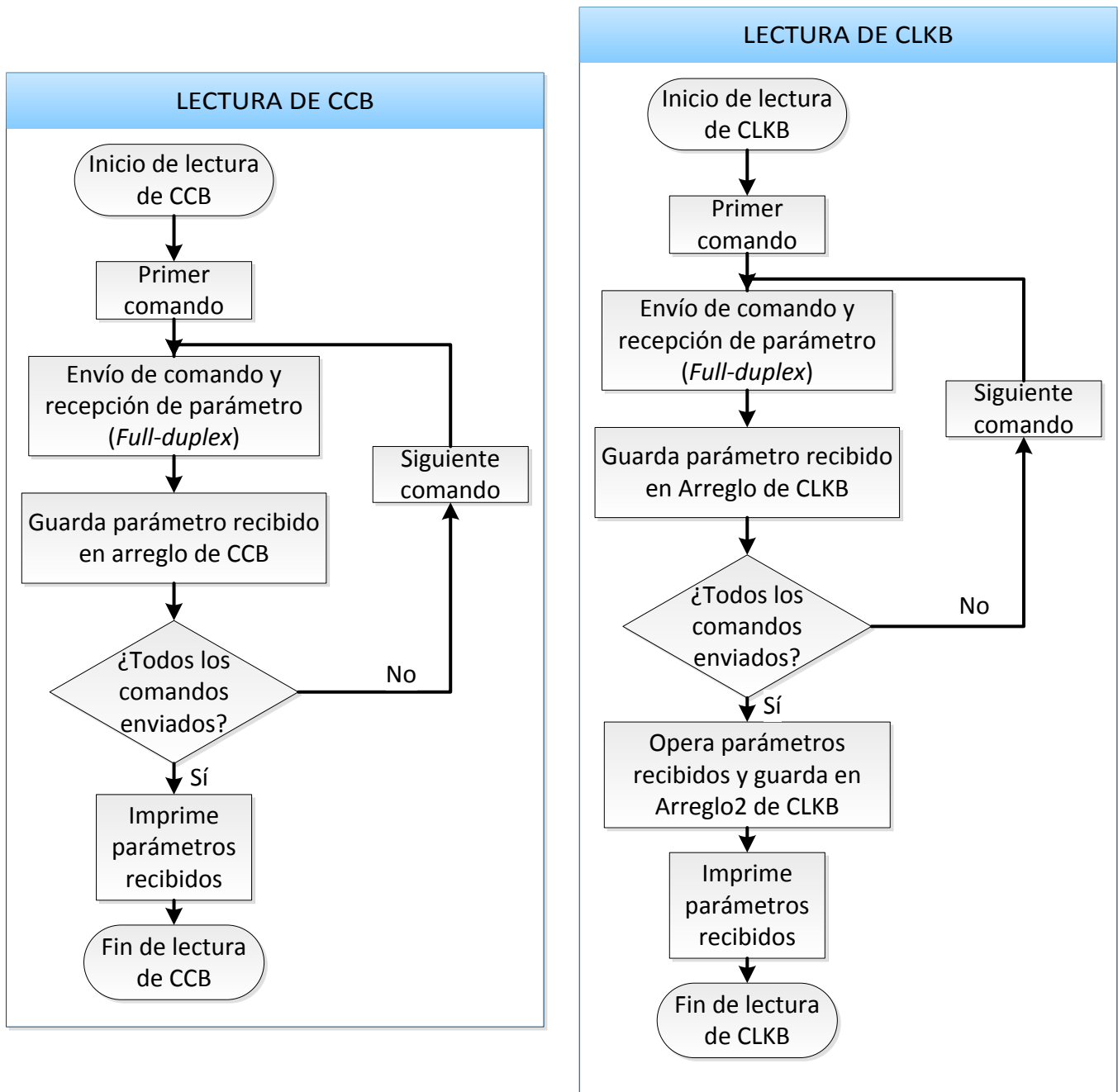


Figura 3.51. Lectura de CCB y CLKB en *Housekeeping*.

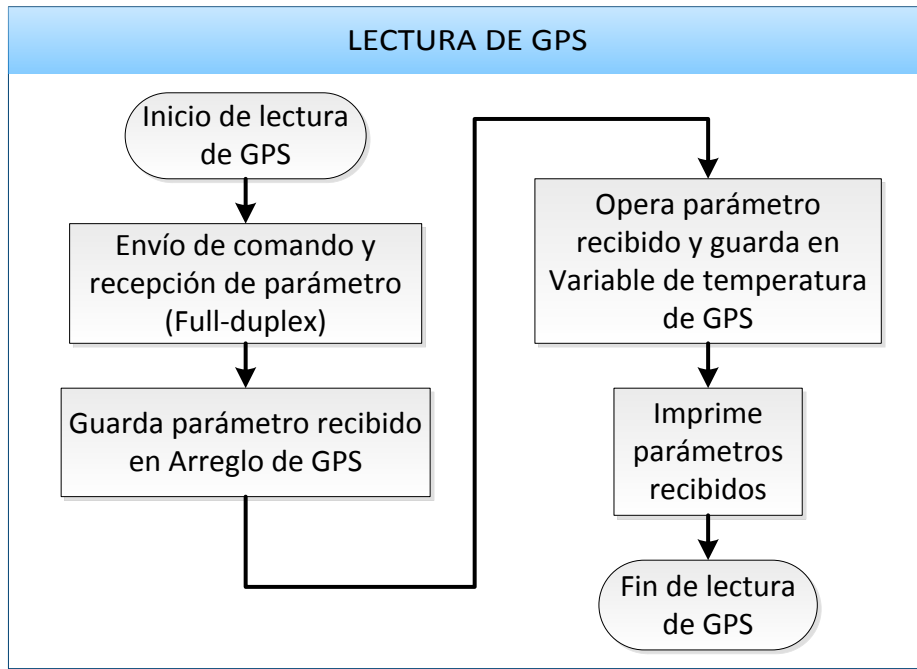


Figura 3.52. Descripción de lectura de GPS en *Housekeeping*.

El procedimiento para almacenar información obtenida de cada subsistema en la microSD se presenta en la figura 3.53. La información almacenada debe estar operada matemáticamente para su fácil interpretación. Posteriormente se repite el ciclo descrito anteriormente a partir de la verificación de la bandera de instrucción de CPU.

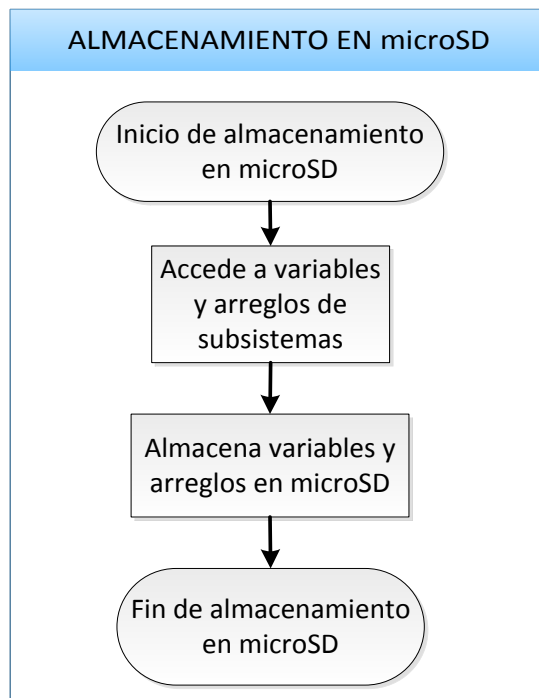


Figura 3.53. Descripción del almacenamiento en microSD.

3.3.6. Desarrollo de prototipos

La necesidad de desarrollar complejos sistemas de *software* y de *hardware* en periodos cortos de tiempo, que aseguren una alta confiabilidad y reducidos costos, está produciendo importantes cambios en la forma de construirlos.

La utilización de *hardware* comercial listo para usarse (COTS) en el sistema a construir, simplifica en gran medida el desarrollo debido a que ofrece bajos costos y reducidos tiempos de adquisición, comparados con los componentes de calificación militar o espacial. Esto permite construir una aplicación buscando y ensamblando componentes, desarrollados por terceros, que combinados adecuadamente satisfacen los requisitos del sistema.

De acuerdo a la filosofía de diseño basada en componentes COTS, aplicada al diseño del *Housekeeping*, la selección de componentes debe realizarse de tal manera que éstos cumplan con las condiciones de operación esperadas, y que puedan ser adquiridas de manera rápida y a un precio razonable.

El rango de precios de los componentes de calificación militar o espacial, puede ser incluso hasta 100 veces mayor en comparación con un componente comercial o industrial. En la tabla 3.3 se puede observar que el costo de un componente de calificación militar se incrementa 10 veces al costo de un comercial, mientras que el espacial lo hace 100 veces más.

Tabla 3.3. Comparativa de precios para componentes.

| Tipo/Aplicación | Conector | Precio unitario (USD)** | Cantidad | Total | Entrega |
|--|--------------|-------------------------|----------|----------|-------------|
| Comercial | DE-9 macho | \$0.83 | 1 | \$0.83 | Inmediata |
| | DA-15 macho | \$2.86 | 1 | \$2.86 | Inmediata |
| | DB-25 macho | \$1.40 | 1 | \$1.40 | Inmediata |
| Total | | | | \$5.09 | |
| Militar | DE-9 macho | \$11.21 | 1 | \$11.21 | 15-30 días* |
| | DA-15 macho | \$9.61 | 1 | \$9.61 | 15-30 días* |
| | DB-25 macho | \$24.65 | 1 | \$24.65 | 15-30 días* |
| Total | | | | \$45.47 | |
| Espacial | MDM-9 macho | \$157.06 | 1 | \$157.06 | 30-60 días* |
| | MDM-15 macho | \$174.11 | 1 | \$174.11 | 30-60 días* |
| | MDM-25 macho | \$215.69 | 1 | \$215.69 | 30-60 días* |
| Total | | | | \$546.86 | |
| *Requiere permisos de importación; **Todos los precios en dólares americanos | | | | | |

Dadas las limitaciones de tiempo y del presupuesto disponible durante el desarrollo del *Housekeeping*, el uso de componentes espaciales y militares no era viable en tiempos de entrega, ya que requieren permiso de importación aprobado por el gobierno. La única solución viable es usar componentes de tipo industrial para lograr entregar un prototipo en el tiempo establecido y que además cumpliera con las especificaciones de temperatura, ya que éstas caen dentro del rango de los componentes industriales.

Asignación de señales en conectores

Después del intercambio de información con los equipos encargados en el desarrollo de los subsistemas con los que se comunica el *Housekeeping*, en donde se discutió sobre las líneas de comunicación que eran manejadas con cada subsistema, se procedió a congelar el diseño de interfaces de cada uno, de tal forma que se pudiera continuar con el diseño de cada subsistema de forma independiente.

En el caso del *Housekeeping* se optó por usar conexiones basadas en conectores D-sub. Un D-sub son dos o más filas paralelas de contactos (terminales), que por lo general están rodeados por un escudo metálico en forma de "D", que proporciona el apoyo mecánico y protección contra las interferencias electromagnéticas. La forma de "D" garantiza la orientación correcta en la conexión.

Los conectores D-sub fueron inventados por ITT Cannon, en 1952. El número que incluye la nomenclatura de los D-sub hace referencia a la cantidad de terminales de cada conector, hace uso del prefijo D para todas las series, seguido por una letra que indica el tamaño de la cubierta o carcasa "D" (A=15 terminales, B=25 terminales, C=37 terminales, D=50 terminales, E=9 terminales), seguido por el número de contactos que lleva el conector, seguido por el género del conector (M=macho, F=hembra). En la figura 3.54 se muestran los conectores D-Sub.

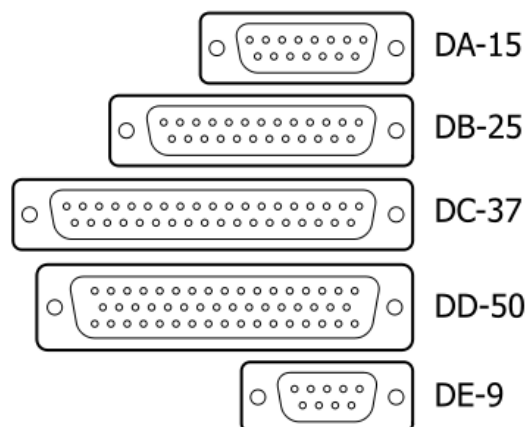


Figura 3.54. Conectores D-sub.

Los escudos metálicos (*backshell*) se conectan a los blindajes de los cables cuando se utilizan conectores de este tipo, creando una barrera eléctrica continua que cubre el cable entero y el sistema de conexión, evitando que los ruidos electromagnéticos interfieran en la comunicación.

En la figura 3.55, se muestra la distribución de señales para los conectores con interfaz RS-422.

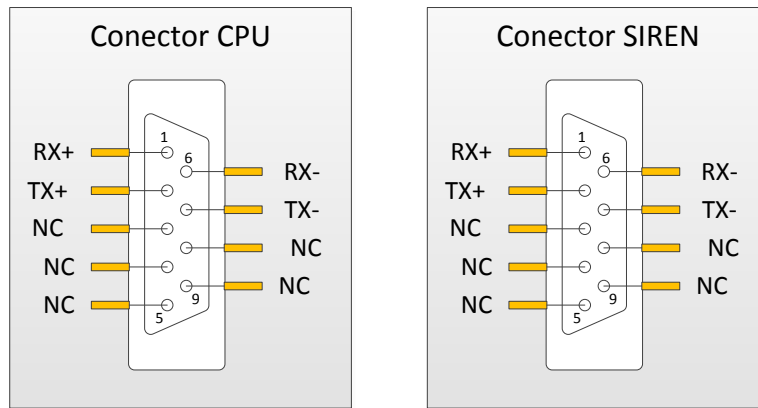


Figura 3.55. Señales en conectores con interfaz RS-422.

En las figuras 3.56 y 3.57 se muestran la distribución de señales para los conectores con interfaz SPI.

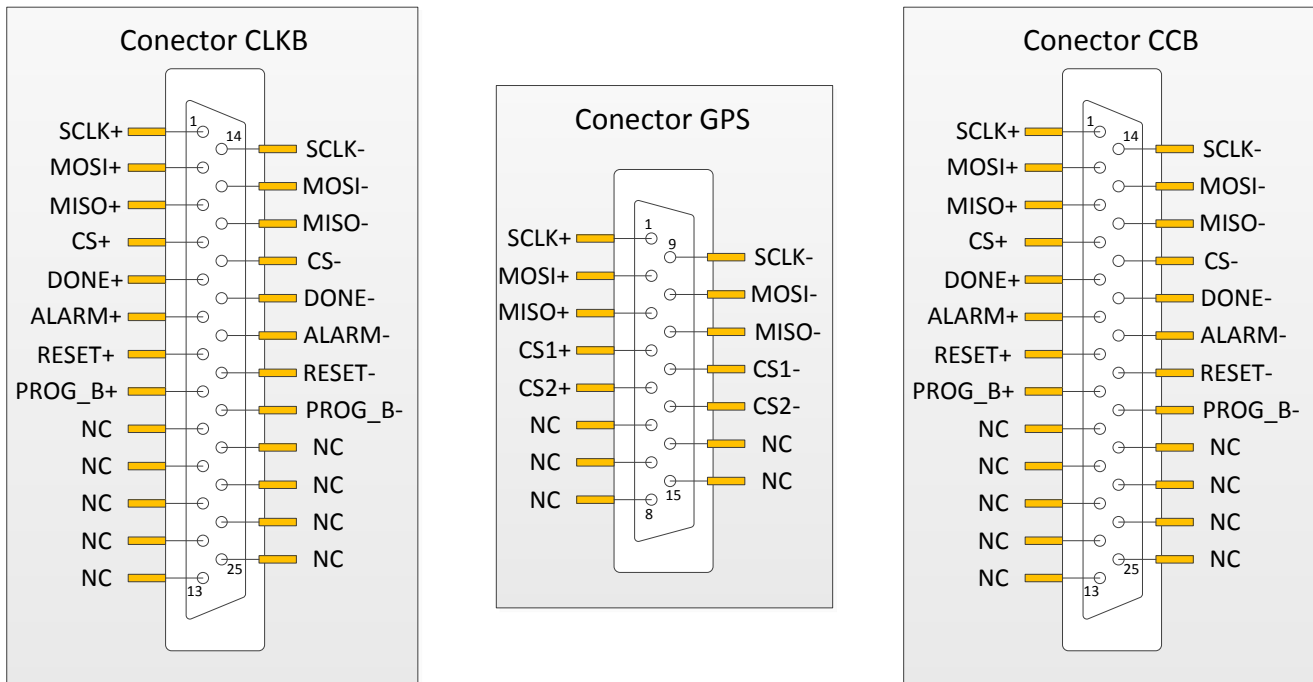


Figura 3.56. Señales en conectores con interfaz SPI.

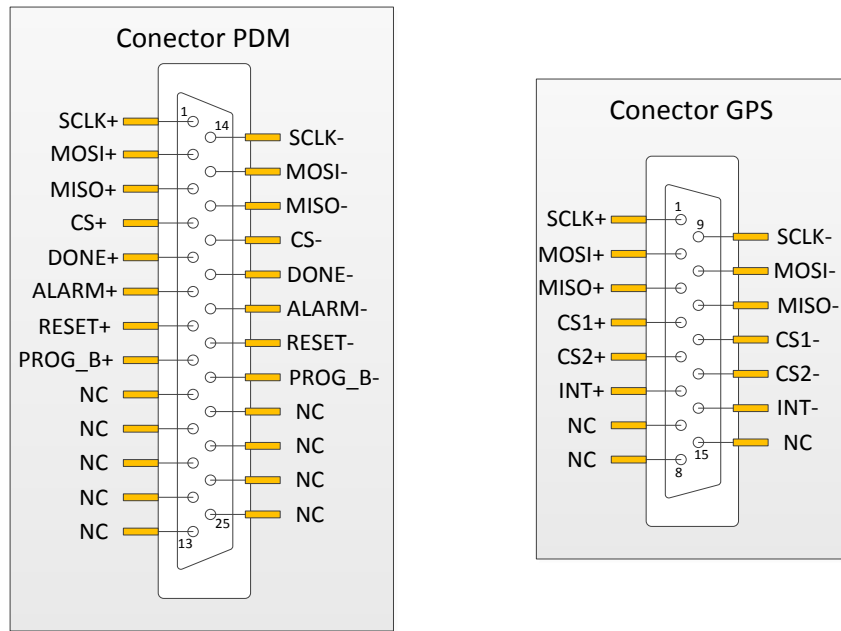


Figura 3.57. Señales en conectores con interfaz SPI.

En las figuras 3.58 y 3.59, se muestra la distribución de señales para los conectores con interfaz con las LVPS.

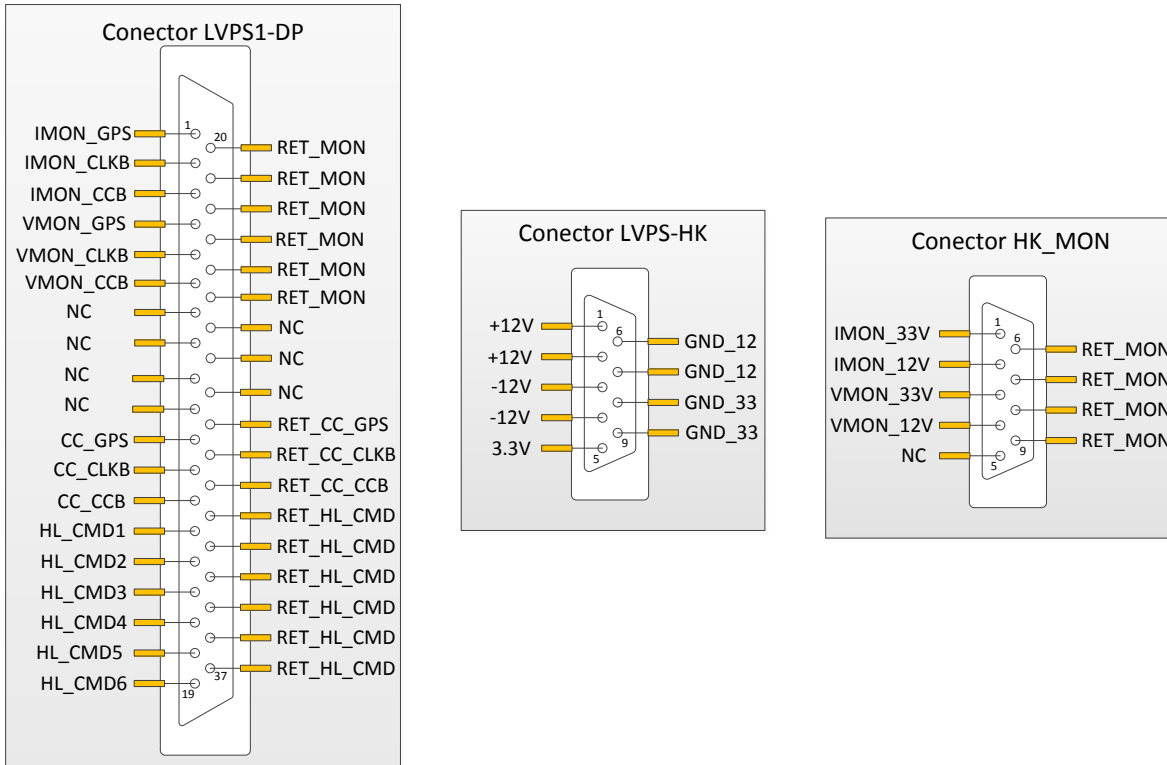


Figura 3.58. Distribución de señales en conectores con las LVPS.

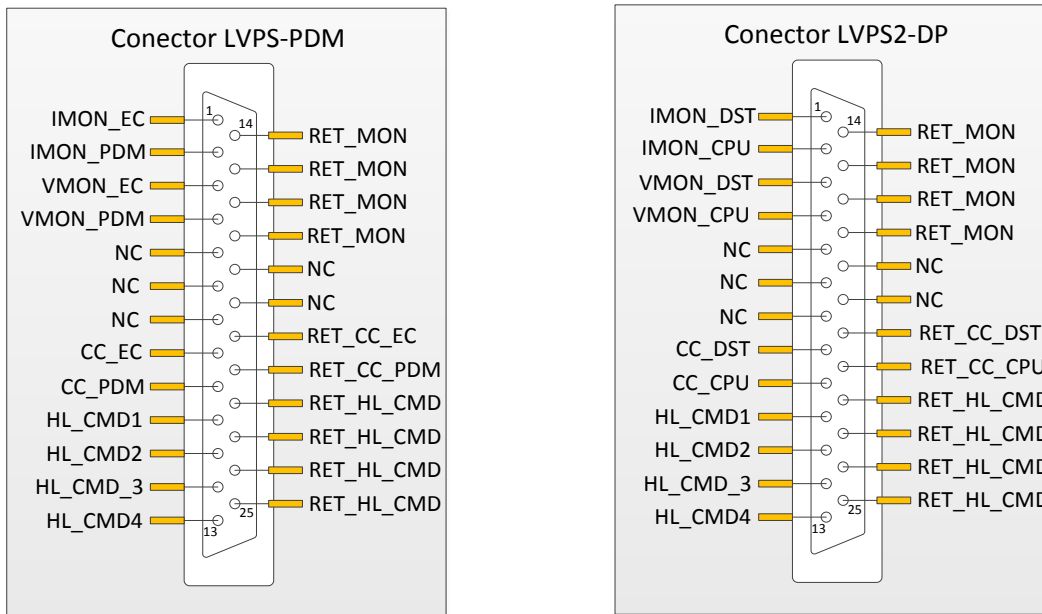


Figura 3.59. Distribución de señales en conectores con las LVPS.

Uso de recursos del microcontrolador

La tarjeta de desarrollo usada en el microcontrolador cuenta con 64 terminales disponibles, se utilizaron 63. En la tabla 3.4 se muestra el uso de las terminales de los puertos de entrada/salida del microcontrolador. La leyenda ND indica que la terminal no se encuentra disponible en la tarjeta de desarrollo.

Tabla 3.4. Uso de los puertos de E/S del microcontrolador.

| | Terminales usadas | | | | | | | |
|-------|-------------------|----|----|----|----|----|----|----|
| | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PORTA | - | x | x | x | x | x | x | x |
| PORTB | x | x | x | x | - | x | x | x |
| PORTC | - | - | - | x | x | x | - | - |
| PORTD | - | ND | ND | ND | - | - | - | - |
| PORTE | ND | ND | x | x | x | ND | x | - |
| PORTF | - | - | - | x | - | - | - | - |
| PORTG | ND | x | x | ND | ND | - | x | x |
| PORTH | ND | x | x | x | - | ND | x | - |
| PORTJ | ND | ND | ND | ND | ND | ND | x | - |
| PORTK | - | x | - | x | - | - | - | - |
| PORTL | x | x | x | x | x | X | x | x |

| Código de colores | |
|-------------------|---|
| Entrada | - |
| Salida | x |

En la tabla 3.5 se muestra el uso de las terminales analógicas del microcontrolador. La mayoría de las terminales de este tipo se usaron como entradas analógicas, mientras que sólo una se usó como salida digital.

Tabla 3.5. Uso de canales analógicos.

| Canales analógicos | Uso |
|--------------------|----------------|
| Canal 0 | Sí |
| Canal 1 | Sí |
| Canal 2 | Sí |
| Canal 3 | No usado |
| Canal 4 | Sí |
| Canal 5 | Sí |
| Canal 6 | Sí |
| Canal 7 | Sí |
| Canal 8 | Sí |
| Canal 9 | Sí |
| Canal 10 | Sí |
| Canal 11 | Sí |
| Canal 12 | Sí |
| Canal 13 | Sí |
| Canal 14 | Salida digital |
| Canal 15 | Sí |

En las tablas 3.6 y 3.7, respectivamente, se muestra el uso de los módulos *timer* y seriales del microcontrolador. Mientras que en la tabla 3.8 se muestra el uso aproximado de la memoria *flash* y EEPROM.

Tabla 3.6. Uso de los módulos *timer*

| Módulos <i>timer</i> | Uso |
|----------------------|----------|
| Timer0 | No usado |
| Timer1 | No usado |
| Timer2 | No usado |
| Timer3 | Sí |
| Timer4 | Sí |
| Timer5 | No usado |

Tabla 3.7. Uso de los módulos de comunicación

| Módulos seriales | Uso |
|------------------|----------|
| USART0 | Sí |
| USART1 | No usado |
| USART2 | Sí |
| USART3 | Sí |
| SPI | Sí |
| I2C | No usado |

Tabla 3.8. Uso de memoria *flash* y EEPROM.

| Uso de memoria | |
|----------------|-------------|
| <i>flash</i> | ~ 59 kB |
| EEPROM | ~ 200 bytes |

Diseño de tarjetas en circuito impreso

El diseño y la fabricación asistidos por computadora es una disciplina que estudia el uso de sistemas informáticos como herramienta de soporte en todos los procesos involucrados en el diseño y la fabricación de cualquier tipo de producto. Esta disciplina se ha convertido en un requisito indispensable para la industria actual que se enfrenta a la necesidad de mejorar la calidad, disminuir los costos y acortar los tiempos de diseño y producción. La única alternativa para conseguir este triple objetivo es la de utilizar la potencia de las herramientas informáticas actuales e integrar todos los procesos, para reducir los costes (de tiempo y dinero) en el desarrollo de los productos y en su fabricación.

CAD es el acrónimo de *Computer Aided Design* o diseño asistido por computadora. Se trata de la tecnología implicada en el uso de ordenadores para realizar tareas de creación, modificación, análisis y optimización de un diseño. De esta forma, cualquier aplicación que incluya una interfaz gráfica y realice alguna tarea de ingeniería se considera *software* de CAD. Las herramientas de CAD abarcan desde herramientas de modelado geométrico hasta aplicaciones a medida para el análisis u optimización de un producto específico.

Distribución de señales de la tarjeta de desarrollo del microcontrolador

El diseño se construye tomando como referencia la distribución de señales de la tarjeta de desarrollo del microcontrolador. Esta huella de las señales, como se observa en la figura 3.60, se debe conservar con el fin de diseñar tarjetas que se apilen sobre esta última. Dado el gran número de subsistemas con los que interactúa el *Housekeeping*, se decidió implementar la electrónica en cinco tarjetas apiladas, con el fin de que si fuese necesario modificar alguna interfaz se contara con la flexibilidad de cambiar sólo una tarjeta y no todo el instrumento.

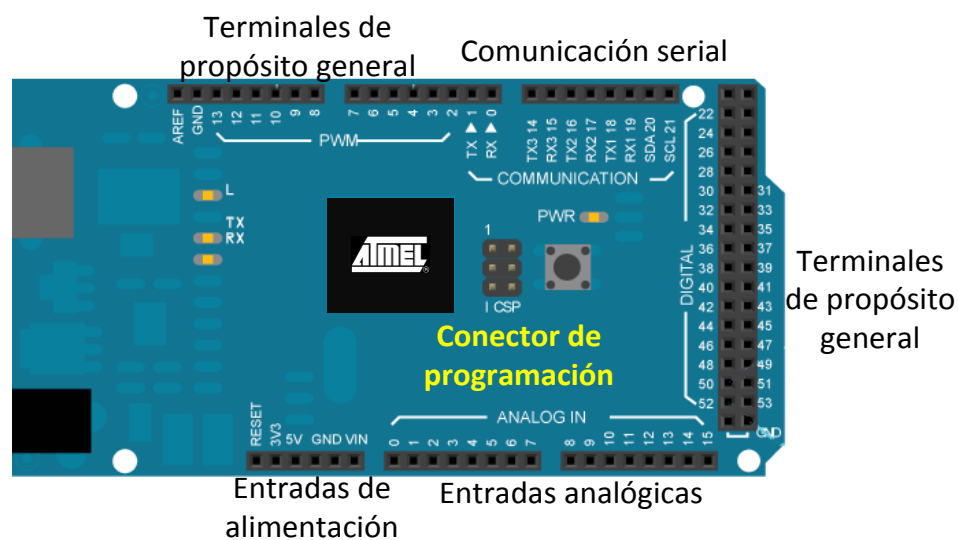


Figura 3.60. Distribución de señales en la tarjeta de desarrollo.

La distribución de las interfaces con los subsistemas, en las cinco tarjetas, se muestra en el diagrama de la figura 3.61. Los nombres de los subsistemas indican la conexión de una tira doble de terminales que conectarán al *Housekeeping* con cables de par trenzado a los conectores de ubicados en el panel frontal.

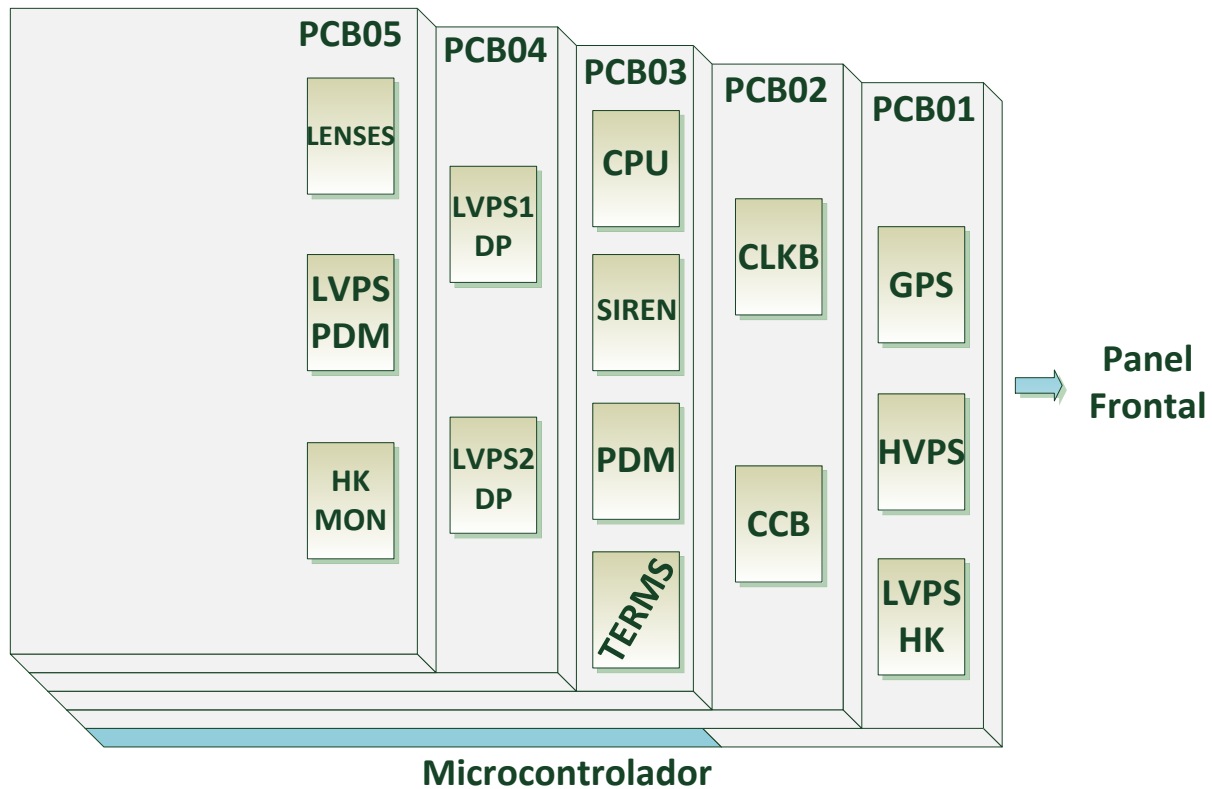


Figura 3.61. Distribución de interfaces en el *Housekeeping*.

Cada una de las PCBs de la estructura contiene conectores tipo *header*, a través de los cuales se colocará un cable que conectará hacia el panel frontal. El diseño modular facilita la sujeción de estos conectores al panel frontal del gabinete del *Housekeeping*.

Diseño de los circuitos impresos

En el diseño de los circuitos impresos, la consideración principal fue separar los circuitos digitales de los analógicos, con el fin de evitar problemas de ruido en los cuales, la electrónica digital se viera afectada por inducciones o mal funcionamiento de la parte analógica. De tal forma que las PCB04 y PCB05 están dedicadas únicamente al manejo de señales analógicas y el control de relevadores por medio de HL_CMD. Por otro lado las PCB01-03 contienen las señales para los buses de comunicación serie basados en los estándares RS-422 y SPI.

En las figuras 3.62 a 3.66 se presenta los cinco diseños de PCB de las tarjetas correspondientes al *Housekeeping*, el *top* a la izquierda y el *bottom* a la derecha.

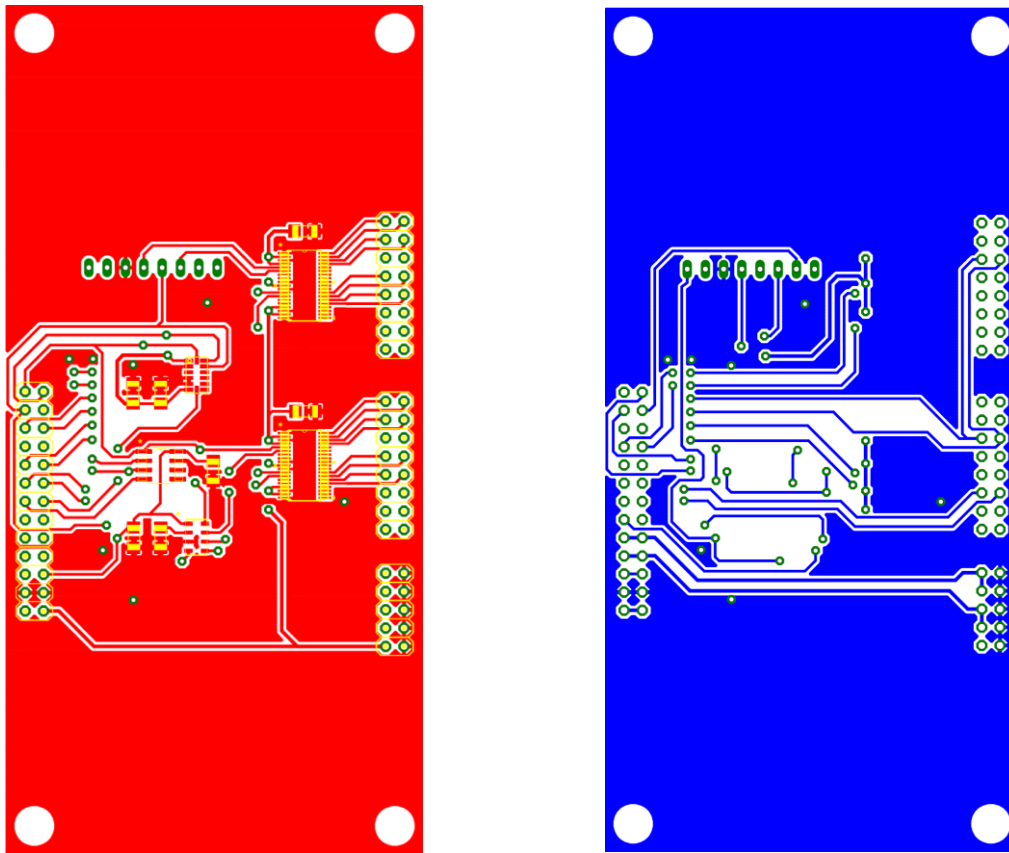


Figura 3.62. Layout de la PCB01.

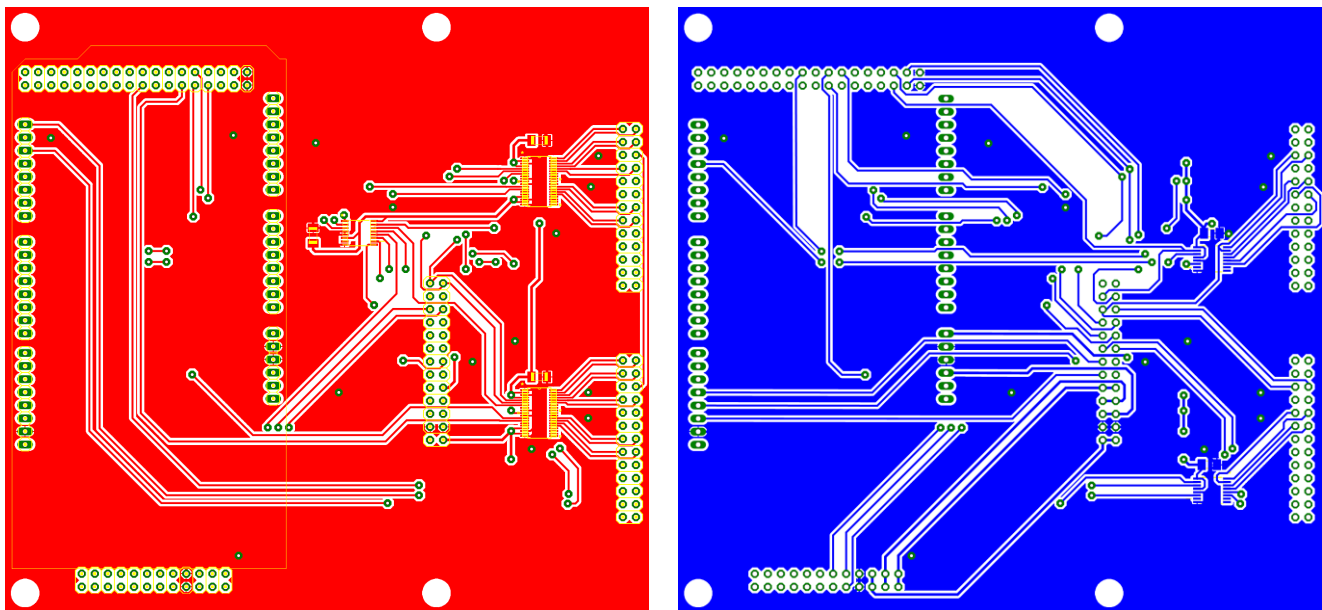


Figura 3.63. Layout de la PCB02.

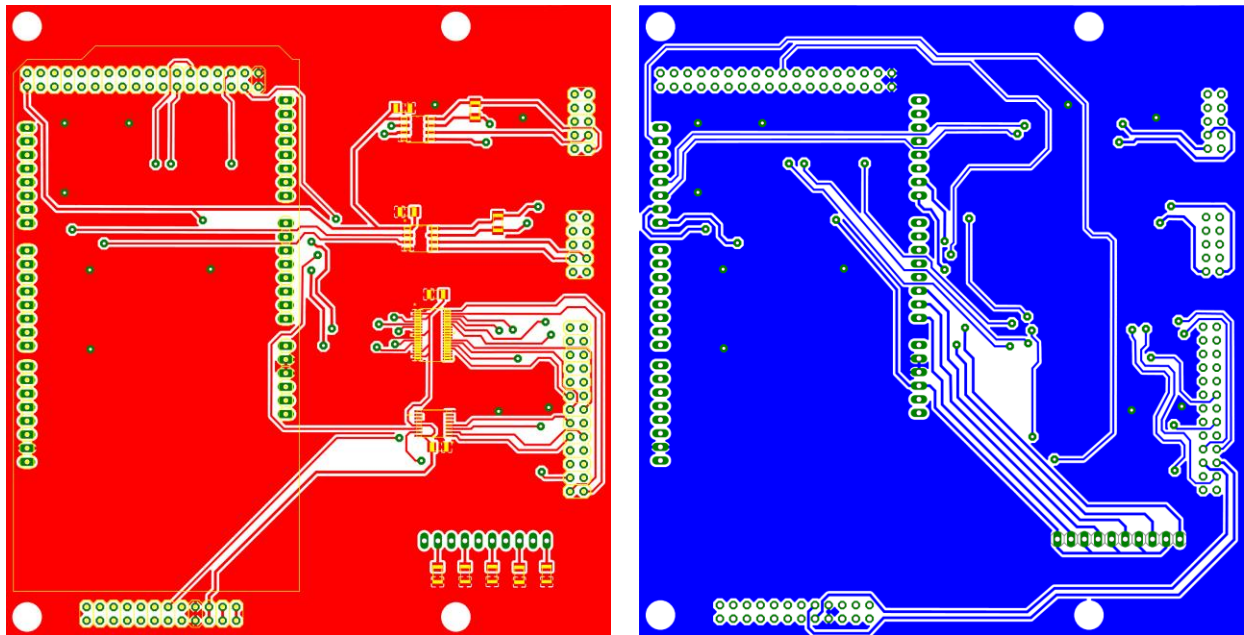


Figura 3.64. Layout de la PCB03.

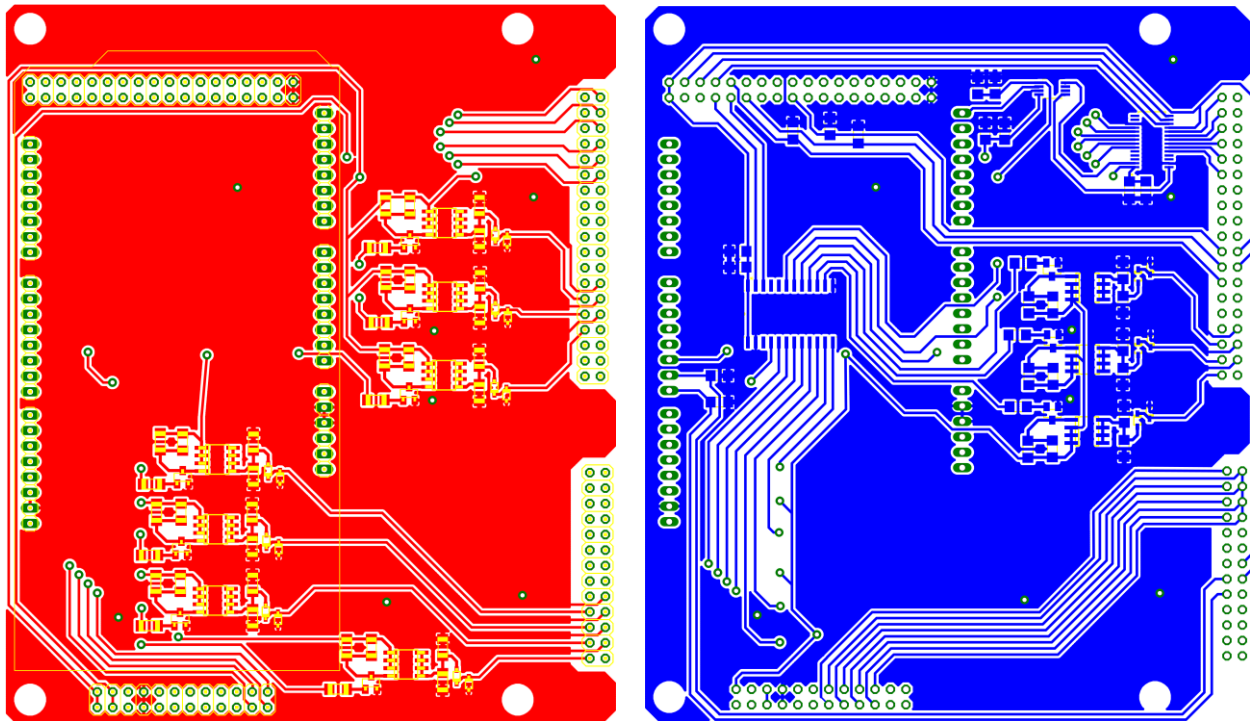


Figura 3.65. Layout de la PCB04.

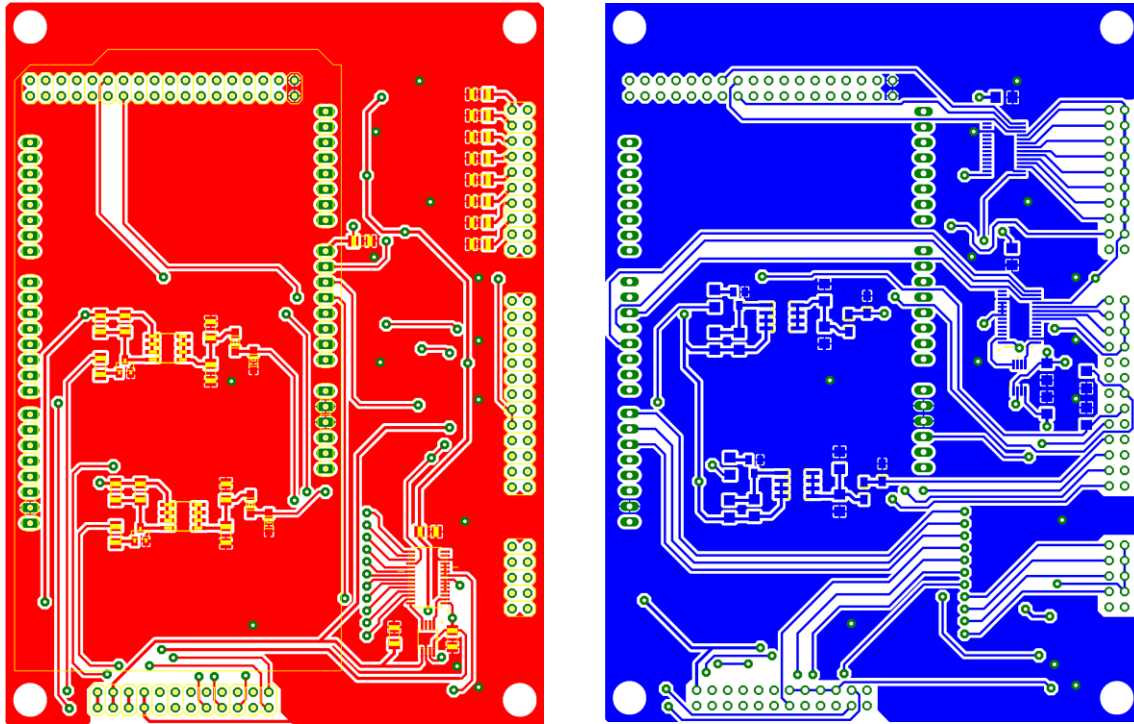


Figura 3.66. Layout de la PCB05.

Una de las ventajas del *software* CAD es el hecho de que puede generar simulaciones en tercera dimensión de los diseños, lo cual nos permite observar el diseño final de las tarjetas a fabricar. Esto puede ayudar a realizar un análisis de ensamble del prototipo incluso antes de haberlo montado físicamente. En las figuras 3.67 a 3.71 se muestra una de las caras de cada PCB diseñada, en vista 3D.

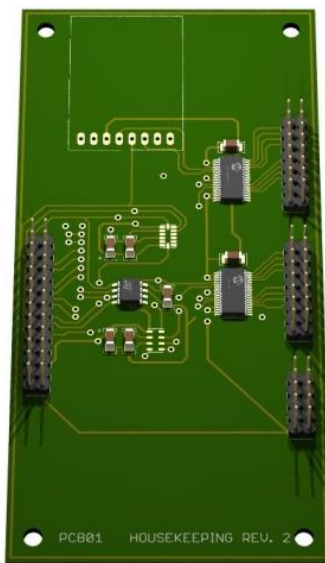


Figura 3.67. Modelo 3D de la PCB01.

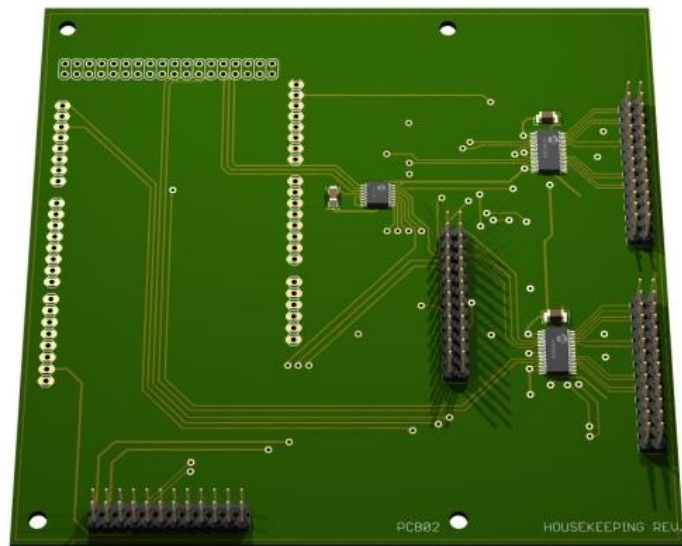


Figura 3.68. Modelo 3D de la PCB02.

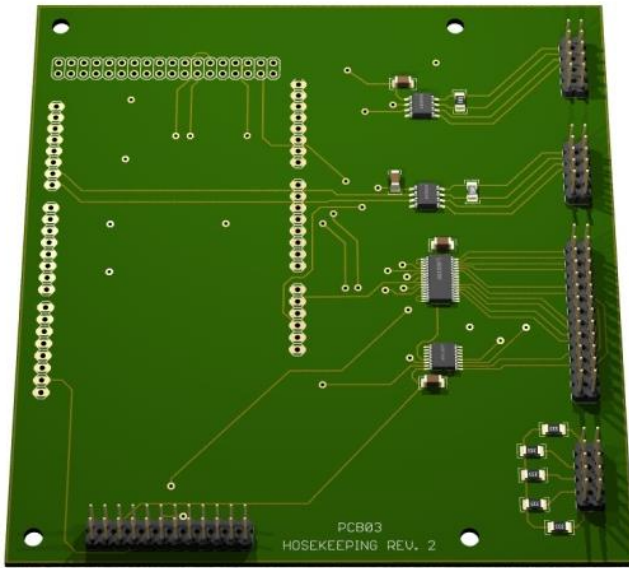


Figura 3.69. Modelo 3D de la PCB03.

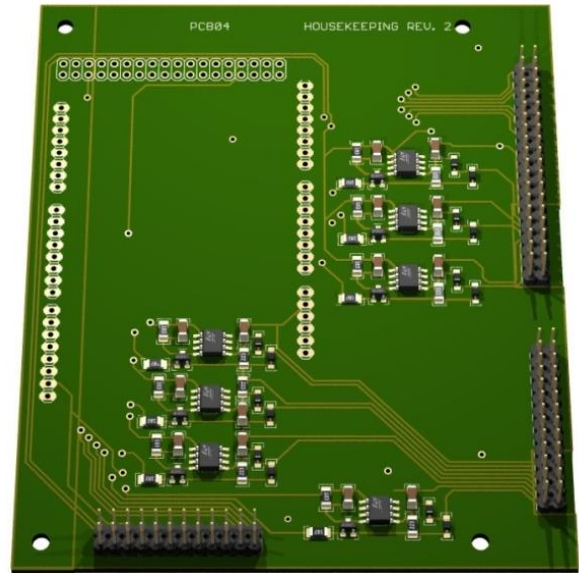


Figura 3.70. Modelo 3D de la PCB04.

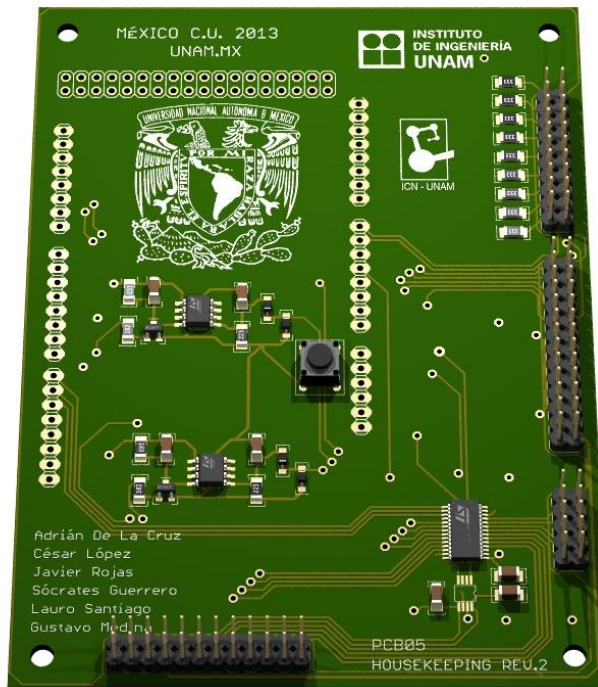


Figura 3.71. Modelo 3D de la PCB05.

Fabricación de prototipos

CAM es el acrónimo de *Computer Aided Manufacturing*, o manufactura asistida por computadora. Se refiere al uso de aplicaciones de *software* computacional de control numérico (NC), para crear instrucciones detalladas que conducen a las máquinas de

herramientas para manufactura de partes controladas numéricamente por computadora (CNC). Los fabricantes de diferentes industrias dependen de las capacidades de CAM para producir partes de alta calidad.

En esta etapa la optimización de diseños en CAD nos permitirá generar los archivos *gerber* bajo el estándar RS-274x, para llevar a cabo la fabricación de las tarjetas de circuito impreso bajo la línea de producción de la empresa LPKF *Laser & Electronics*. Las especificaciones para esta línea de producción se presentan en la tabla 3.9.

Tabla 3.9. Especificaciones de la línea producción LPKF.

| Características | Estándar | Capacidad máxima |
|--|------------------|------------------------|
| Capas | 1 o 2 | (4, 6) |
| Material | FR4 | |
| Espesor de cobre | 35 μm | 105 μm |
| | 70 μm | 140 μm |
| Espesor de tarjeta | 1.6 mm | 0.5 mm 0.6 mm |
| | | 0.8 mm 1.0 mm |
| | | 1.2 mm 2.0 mm |
| | | 2.4 mm 3.2 mm |
| Terminado superficial | | Platinado en estaño |
| Mascara antisoldante | Verde | Estándar |
| Dimensiones máximas de tarjetas | A4 | 210 mm \times 297 mm |
| Diámetro mínimo de perforación | 0.4 mm | 0.1 mm |
| Ancho mínimo de pista | 0.2 mm | 0.1 mm |
| Espacio mínimo de pista | 0.2 mm | 0.1 mm |
| Separación mínima SMT (<i>pitch</i>) | 0.2 mm | 0.1 mm |

Siguiendo las especificaciones del fabricante se utilizaron láminas de cobre de $\frac{1}{2}$ de onza, que equivale a 35 μm de espesor por cara, y se fabricaron las 5 tarjetas de circuito impreso por medio de CAM.

Línea de producción LPKF

Una vez que se finalizó el diseño de los circuitos impresos por computadora, se procedió a la fabricación de éstos. Para ello se utilizaron las máquinas de la línea de producción LPKF para tarjetas de circuito impreso, instalada dentro del Laboratorio de Detectores del ICN, ésta consta de las siguientes herramientas:

- Máquina fresadora. Esta máquina realiza el devaste de cobre de las placas fenólicas, es decir, realiza el trazado de pistas. También realiza las perforaciones necesarias que

serán la conexión entre las caras de las tarjetas, y las perforaciones para los componentes *through-hole*.

- Máquina de *through-hole*. En ésta se realiza un proceso químico de deposición de cobre sobre las placas perforadas por la máquina fresadora. Une dos o más capas de cobre por medio de las perforaciones que realizó la máquina fresadora.
- Laminadora. Es utilizada para la colocación de la máscara antisoldante que sirve para proteger el cobre de la tarjeta.
- Máquina de colocación de soldadura. Esta máquina coloca soldadura en pasta sobre la tarjeta, basándose en un molde con las perforaciones de los *pads*, también llamado estencil.
- Máquina para colocar los componentes. Mediante aire comprimido se sujetan los componentes con la finalidad de llevarlos a su posición en la tarjeta.
- Horno de reflujo. Se utiliza para realizar el proceso de soldado de los componentes electrónicos con las tarjetas.
- Adicionalmente se cuenta con una máquina para hacer circuitos de hasta seis capas.

El proceso de fabricación de las tarjetas de circuito impreso sigue el siguiente orden:

- Se hacen las perforaciones que serán la conexión entre las dos capas de la tarjeta.
- La tarjeta perforada se somete al proceso de deposición química de cobre.
- Terminada la deposición química de cobre se realiza el devastado de las pistas en la tarjeta. En la figura 3.72 se muestra la tarjeta maquinada por la fresadora y después de pasar por la deposición química.

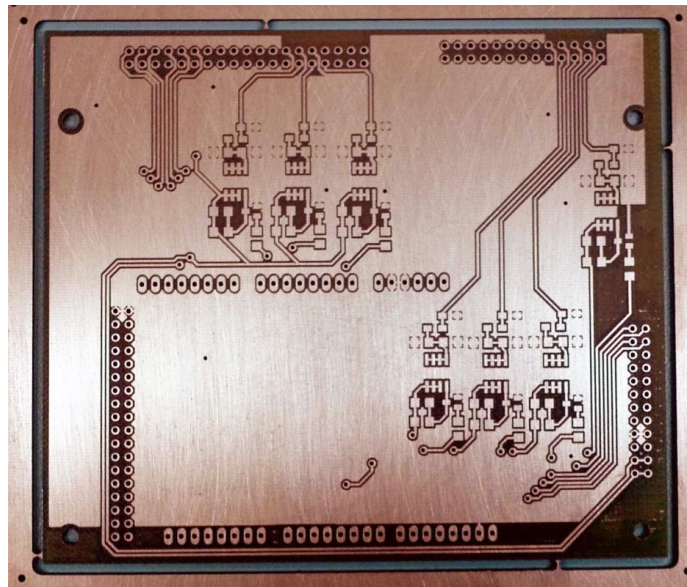


Figura 3.72. Devastado de la cara superior de la PCB04.

- Una vez que se finaliza el devastado de la tarjeta, esta se baña en estaño líquido. En la figura 3.73 se muestra cara *top* la tarjeta devastada y bañada en estaño líquido.

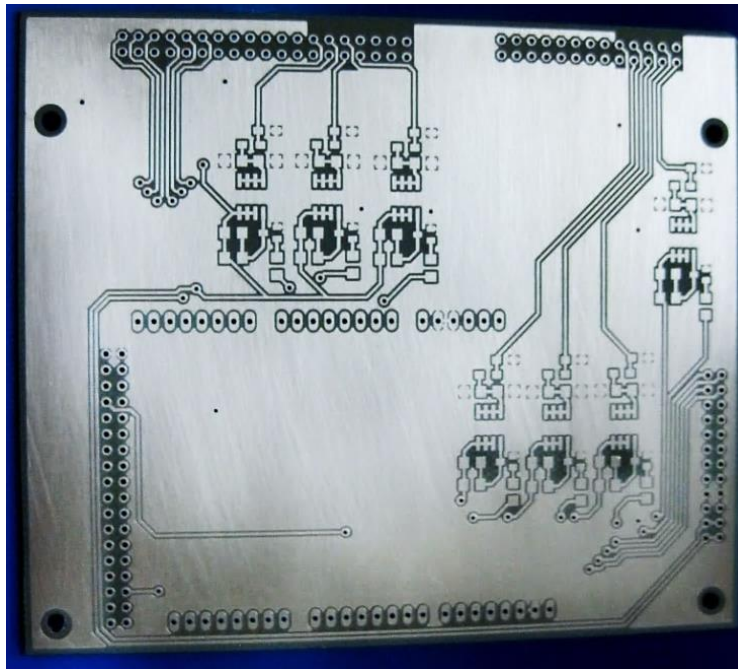


Figura 3.73. Platinado de la cara superior de la PCB04.

- Una vez que la tarjeta fue bañada en cobre, se le coloca la máscara antisoldante. En la figura 3.74, se muestra la cara *bottom* de la tarjeta, con la máscara antisoldante aplicada.

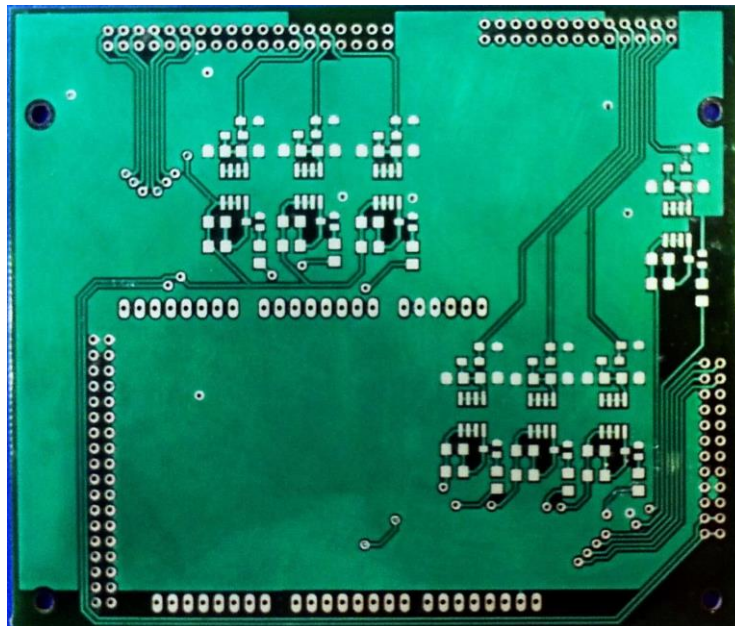


Figura 3.74. Cara superior de la PCB04 con máscara antisoldante.

- Ya que se colocó la máscara antisoldante, se procede a revisar las tarjetas con el microscopio, esto con el fin de verificar que no existan cortos entre pistas o entre los *pads* de los circuitos integrados que se van a colocar.

En la figura 3.75 se muestran los *pads* de uno de los dispositivos vistos con ayuda del microscopio.

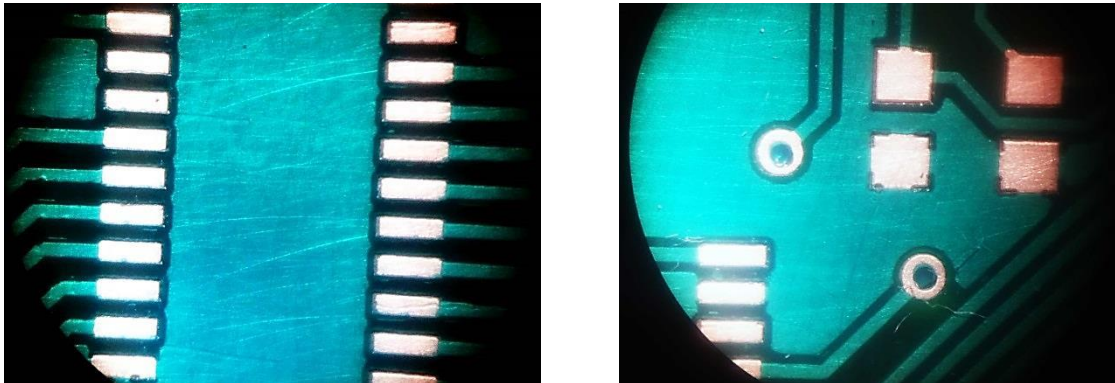


Figura 3.75. Vista de los *pads* con el microscopio.

- Con el estencil se coloca la soldadura en pasta y se colocan los componentes electrónicos, como se muestra en la figura 3.76.

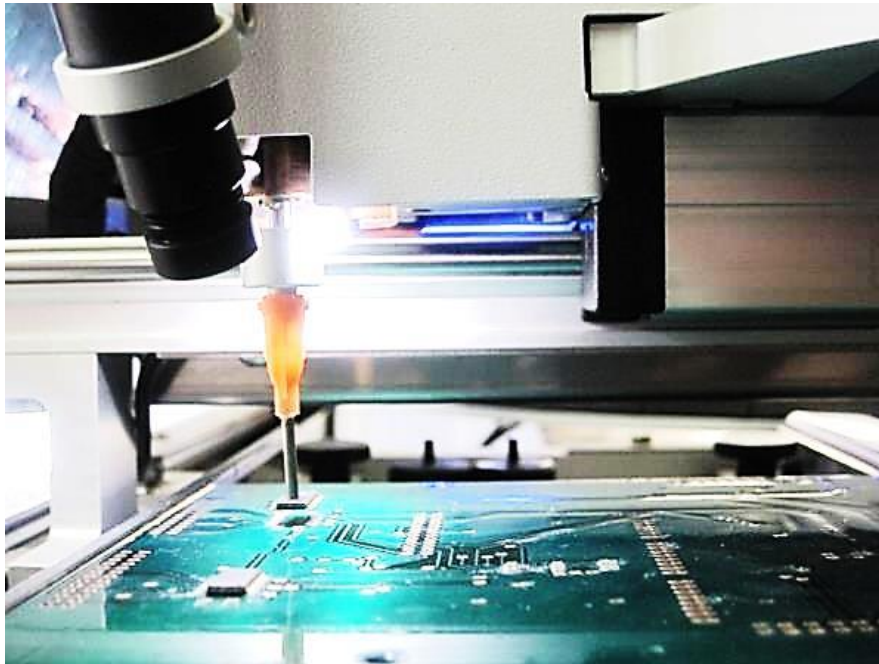


Figura 3.76. Colocación de componente.

- Posteriormente se procede al soldado de componentes en el horno de reflujo como se muestra en la figura 3.77.



Figura 3.77. Tarjeta en el horno de reflujo.

En la figura 3.78 se pueden observar las tarjetas terminadas, incluido el montaje de los componentes electrónicos de cada una de ellas. Posteriormente se procedió a apilarlas para tener el sistema completo como se muestra en la figura 3.79.

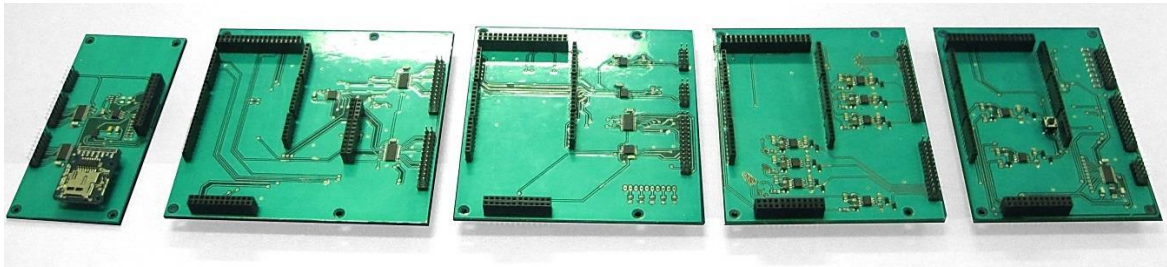


Figura 3.78. Tarjetas de *Housekeeping*.

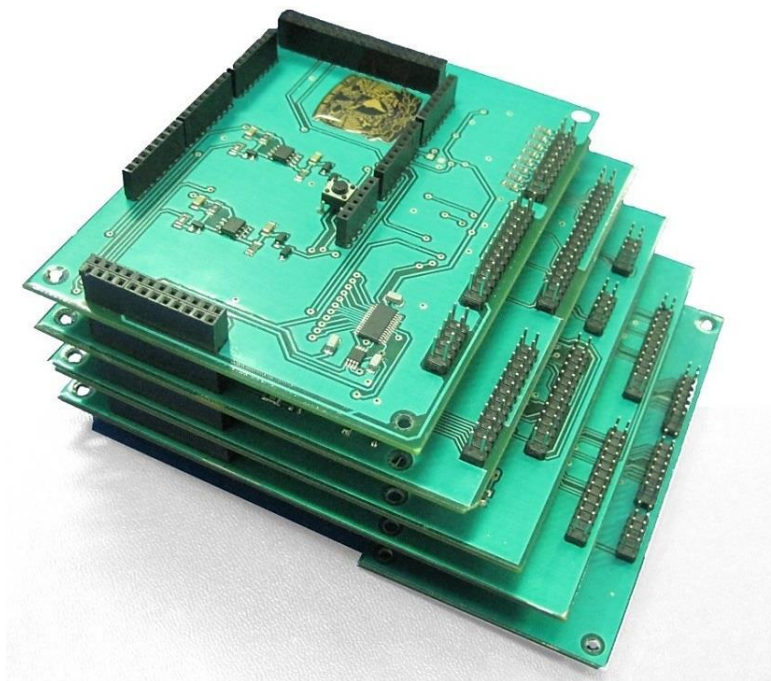


Figura 3.79. *Housekeeping* ensamblado.

Montaje en gabinete

Una restricción del diseño se dio por la limitación de espacio necesario para hacer la conexión de cables, para montar y desmontar conectores de forma rápida.

Recordando que el volumen efectivo es de 135 mm x 120 mm x 220 mm (ancho, alto y profundidad, respectivamente), se diseñó respetando las dimensiones máximas posibles. En las figuras 3.80 y 3.81 se muestran la distribución de las medidas y la disposición de tarjetas del *Housekeeping*, respetando las dimensiones.

Primer nivel: Placa de desarrollo del microcontrolador y PCB01.

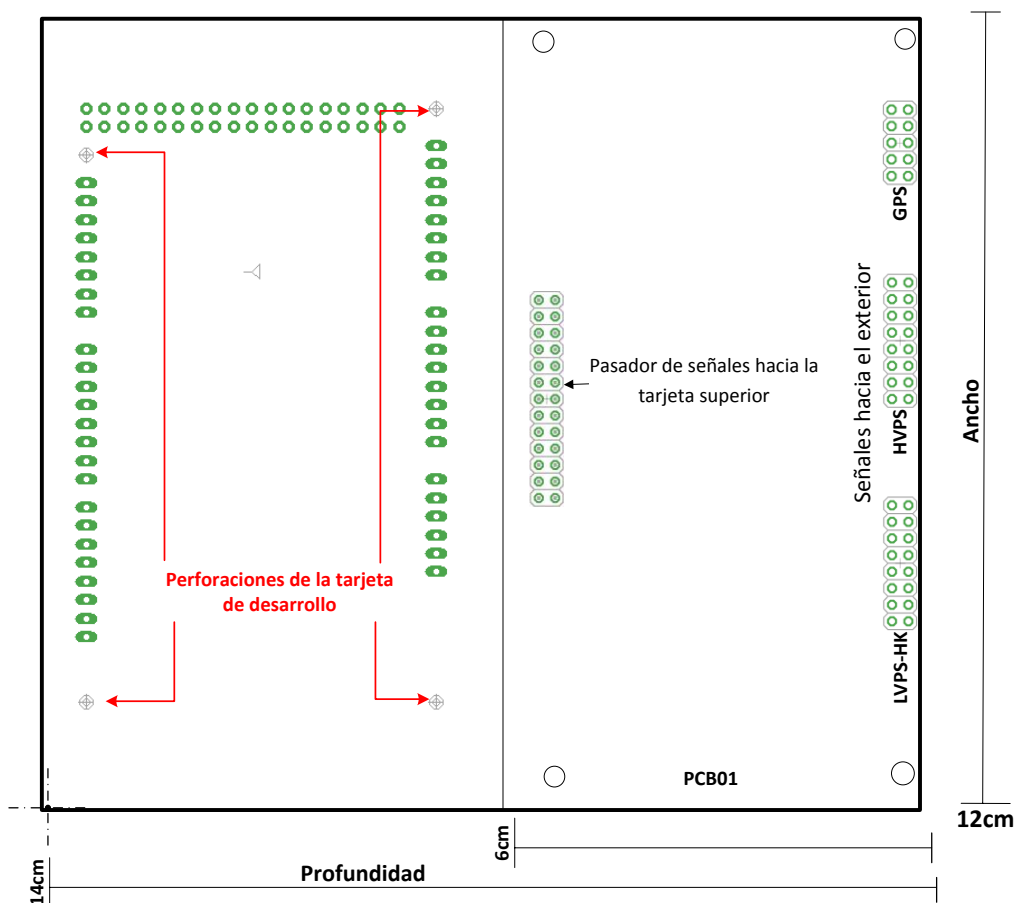


Figura 3.80. Medidas del primer nivel de la estructura del *Housekeeping*.

- Se montaría todo sobre una base de 14 cm x 12 cm, dejando libre aproximadamente 47% de base disponible.
- La placa de desarrollo del microcontrolador se sostendrá con tornillos, a través de los cuatro agujeros que tiene.
- La PCB01 se sostendrá con cuatro tornillos, situados en cada esquina de la placa cuyo diámetro es de 3.2mm.

Niveles superiores: PCB02 – PCB05

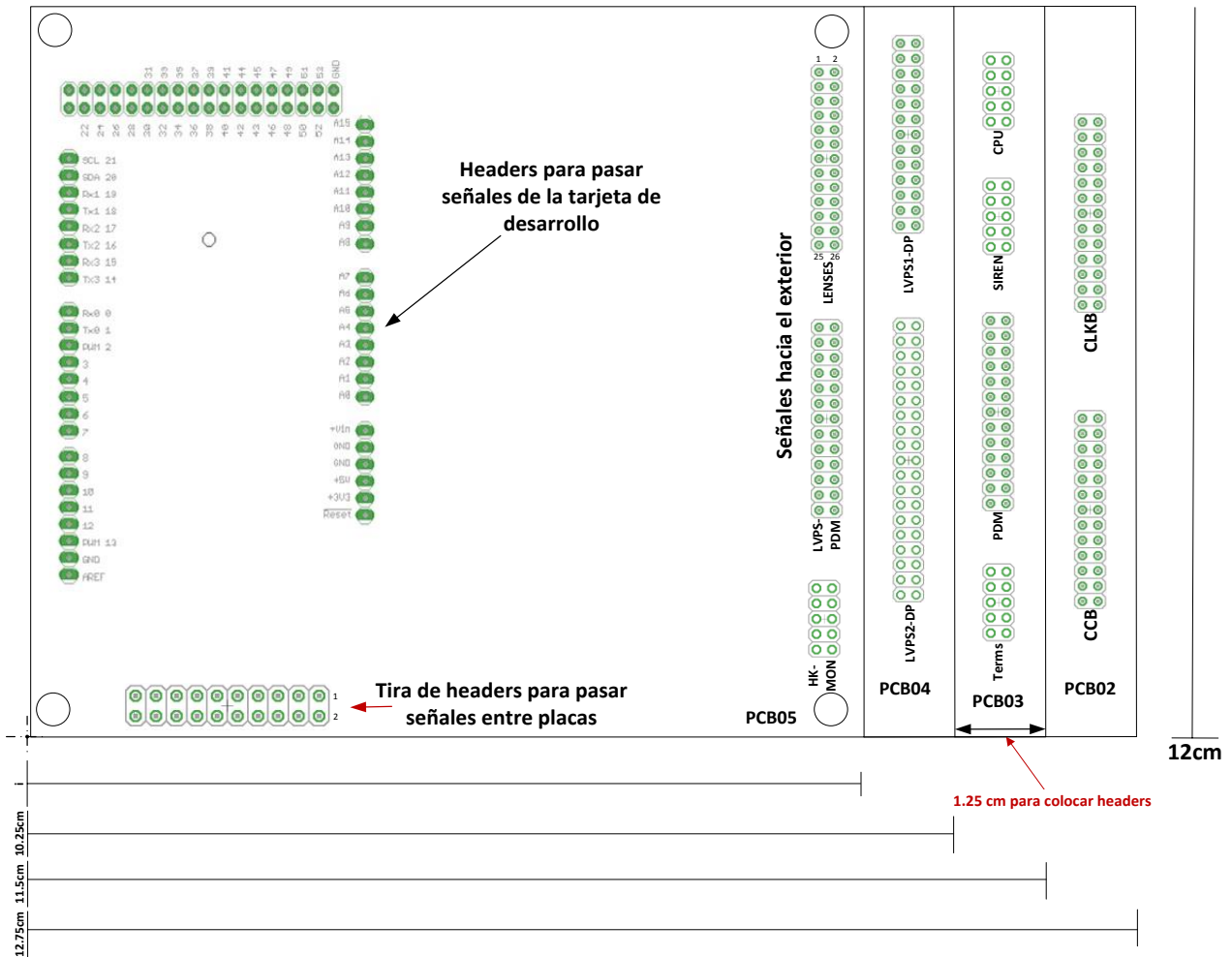


Figura 3.81. Vistas de las PCB02 a PCB05.

- Los agujeros de las placas de PCB02 a PCB04 tendrán la misma posición que los agujeros de PCB05.
- Los dos agujeros del lado derecho coincidirán hasta la PCB01.

En la figura 3.82 se muestran las medidas de la vista vertical de la estructura, en donde se observa que no se rebasa el límite para la medida de altura establecida.

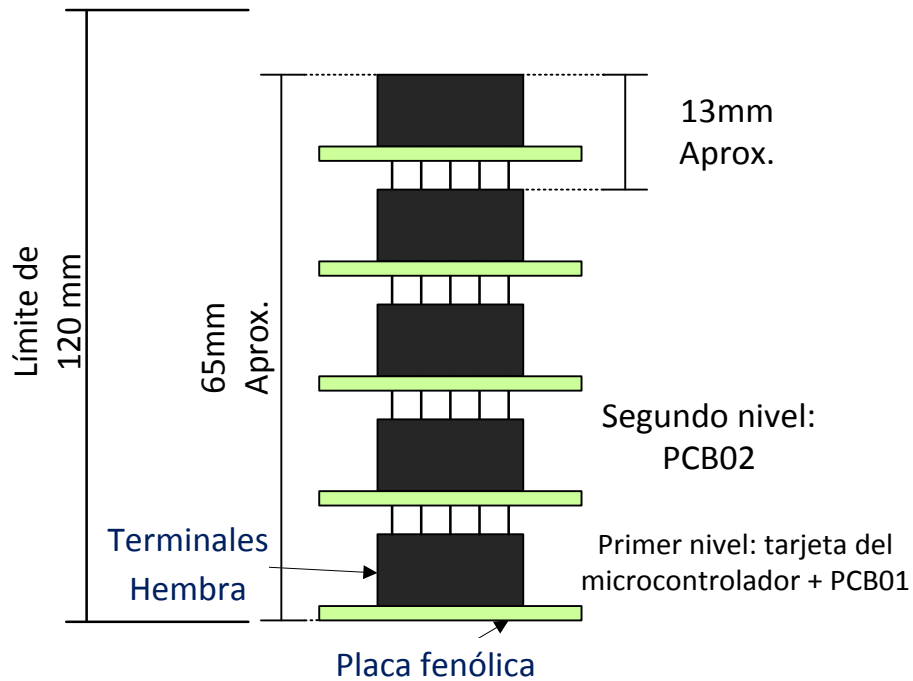


Figura 3.82. Vista de la estructura del *Housekeeping*.

El gabinete para montar esta versión del *Housekeeping* fue fabricado en México, se presenta en la figura 3.83. Este gabinete fue diseñado conforme a los requerimientos de espacio definidos anteriormente.



Figura 3.83. Gabinete para el *Housekeeping*.

Se generó un modelo CAD con las huellas de los conectores D-Sub para hacer las perforaciones en el panel frontal. En la figura 3.84 se muestra la distribución de los conectores para el panel frontal.

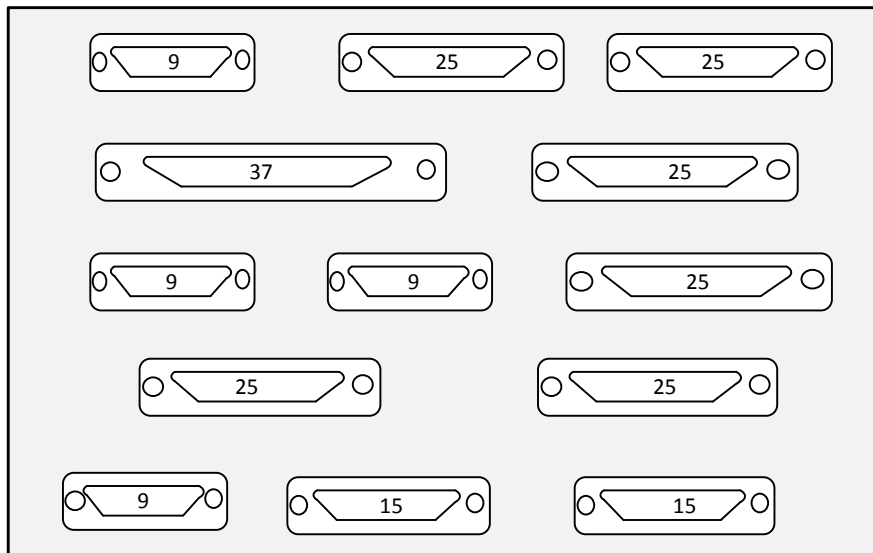


Figura 3.84. Panel frontal del gabinete del *Housekeeping*.

3.4. Bancos de pruebas de subsistemas

Con el objetivo de lograr un mayor avance en el desarrollo del *software* del *Housekeeping*, sin la necesidad de realizar pruebas directamente con los diferentes subsistemas, se desarrollaron los bancos de pruebas para las diferentes interfaces del *Housekeeping*.

3.4.1. Banco de pruebas para CPU y SIREN

Debido a la necesidad de contar con alguna interfaz para realizar las pruebas necesarias para validar la comunicación con CPU y SIREN, se diseñó un banco de pruebas que contiene ambas interfaces. Con este banco de pruebas se pueden intercambiar todos los comandos disponibles para el protocolo RS-422 utilizado con CPU y SIREN, además de que permite realizar la depuración del comportamiento del *Housekeeping*, enviando a un puerto serie toda la información importante acerca del funcionamiento de éste, como datos de monitoreo analógico/digital, valores de registros, etc.

Hardware

Este banco de pruebas está controlado por el mismo microcontrolador usado por el *Housekeeping*, además de contar también con los mismos transceptores para el estándar RS-422. Para cada interfaz que se simula, se tiene asignado un módulo USART independiente, y debido a que el funcionamiento para ambas interfaces es el mismo, se colocó un puente o *jumper*, para seleccionar la interfaz a simular. SIREN siempre tendrá la función de depuración, además de poder ser usado para el control del *Housekeeping*.

Este banco de pruebas es controlado por una conexión serial hacia una computadora, a través de la cual se tiene interacción con el usuario por medio de una interfaz de comandos. En la figura 3.85 se muestra un diagrama de bloques del banco de pruebas.

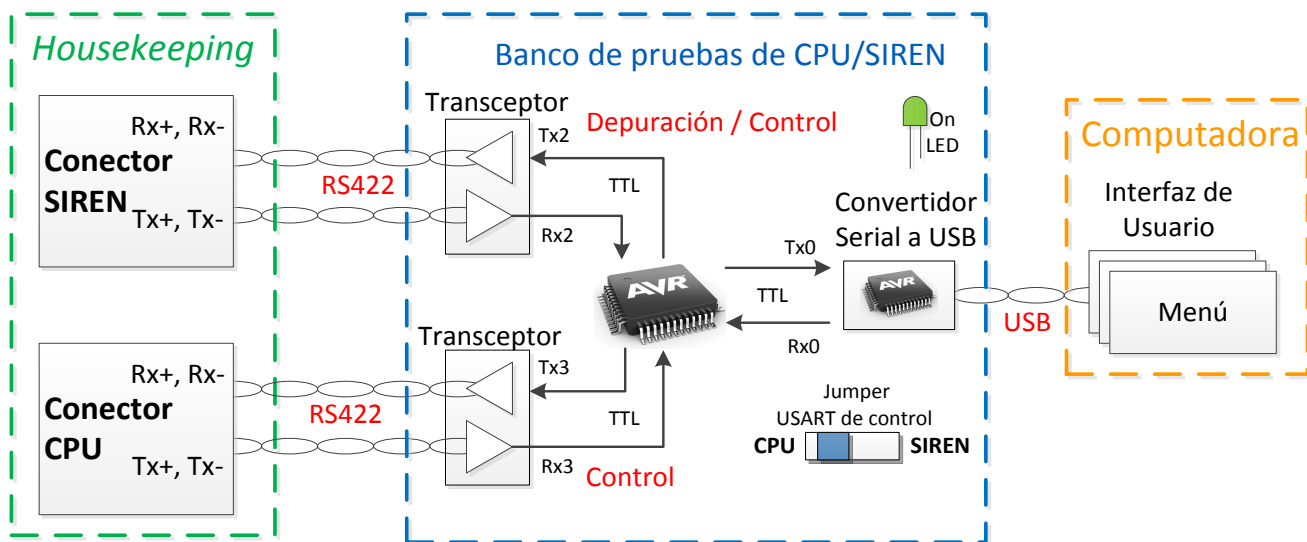


Figura 4.85. Banco de pruebas de CPU/SIREN.

La asignación de señales en los conectores de la tarjeta es la misma que se usa en las interfaces reales con CPU y SIREN. En la figura 3.86 muestra el circuito impreso en vista 3D del banco de pruebas, y la distribución de señales en el conector de SIREN y CPU.

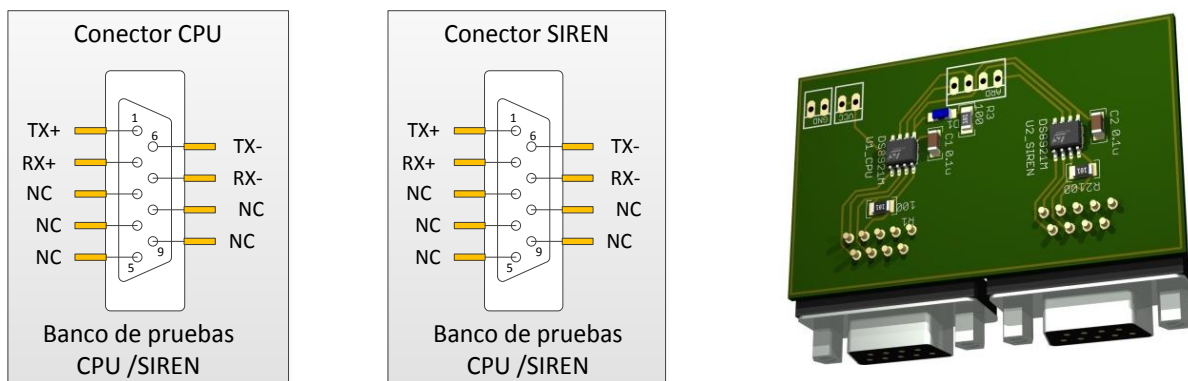


Figura 3.86. Señales en conectores y modelo 3D.

Las primeras pruebas de conexión eléctrica que se realizaron fueron sin implementar algún protocolo, sólo se realizó el envío y la recepción de un *byte*. Esto se hizo por medio de una terminal de computadora que enviaba el *byte* de la tecla presionada en el teclado, estos *bytes* eran recibidos y convertidos de USB a serial TTL a través del convertidor incluido en la tarjeta de desarrollo del microcontrolador. Este convertidor está implementado mediante un microcontrolador pequeño dedicado específicamente a esta función. Del lado de recepción de

Housekeeping detectaba el *byte* recibido y se le asignaba una función básica como mostrar el *byte* recibido o encender algún relevador.

El resultado de esta prueba llevó a la construcción del primer banco que simula los subsistemas de RS-422 del *Housekeeping*. Con el diseño del *hardware* y la conexión eléctrica probada con el *Housekeeping*, el siguiente paso es comenzar a trabajar en el *firmware*. En la siguiente figura 3.87 se muestra una imagen del banco colocado sobre la tarjeta de desarrollo.

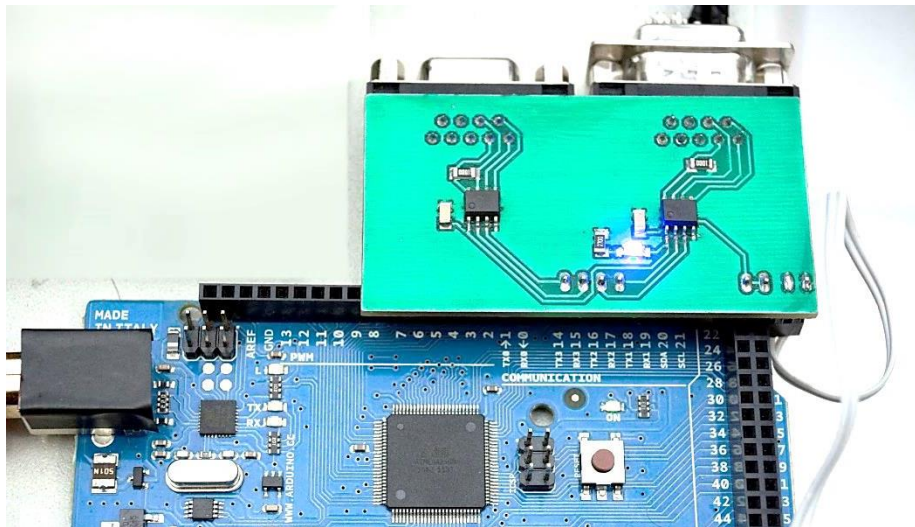


Figura 3.87. Banco de pruebas de CPU y SIREN.

Desarrollo del *firmware* y propuesta de comunicación con el *Housekeeping*

El *firmware* es un bloque de instrucciones de máquina para propósitos específicos, grabado en memoria, normalmente de lectura/escritura (ROM, EEPROM, *flash*, etc), que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo. Está fuertemente integrado con la electrónica del dispositivo, siendo el *software* que tiene directa interacción con el *hardware*, ya que es el encargado de controlarlo para ejecutar correctamente las instrucciones externas.

El *firmware* de este banco de pruebas se basa en un protocolo de comandos en estándar RS-422 con las siguientes características:

- Comunicación asíncrona
- Punto a punto
- Orientado a carácter (ASCII), de 8 *bits* por paquete
- Capacidad de establecer comunicación *full-duplex*

Para la comunicación con CPU se realizó una propuesta de comandos, de 1 *byte*, para realizar tareas con todos los subsistemas del *Housekeeping*. Estas tareas que se muestran en la tabla 3.10, agrupadas en tres secciones realizan lo siguiente:

- CPU/SIREN, contiene la mayoría de comandos para encender y apagar los relevadores encargados del suministro de potencia en LVPS, además de efectuar la reprogramación de FPGAs y el envío de alarmas.
- SIREN, permite la lectura de sensores de temperatura y el envío de comandos para encender y apagar la CPU.
- Subsistemas SPI, en este grupo de comandos se hace la lectura de parámetros para cada subsistema de SPI.

Tabla 3.10. Comandos entre CPU y *Housekeeping*.

| CPU / SIREN comandos para ejecutar acción | Descripción | Hex |
|--|----------------------------|------|
| | Enciende CLKB | 0X1A |
| | Enciende CCB | 0X1B |
| | Enciende GPS | 0X1C |
| | Enciende PDM | 0X1D |
| | Enciende EC | 0X1E |
| | Enciende DST | 0X18 |
| | Aparga CLKB | 0X8A |
| | Apaga CCB | 0X8B |
| | Apaga GPS | 0X8C |
| | Apaga PDM | 0X8D |
| | Apaga EC | 0X8E |
| | Aparga DST | 0X88 |
| | Mandar registro de alarma | 0X20 |
| | Manda datos leídos CPU | 0X40 |
| | Manda Reset CLKB | 0X7A |
| | Manda Reset CCB | 0X7B |
| | Manda Reset GPS | 0X7C |
| | No hay alarmas registradas | 0x2F |
| Manda Reset PDM | 0X7D | |
| CRC Error | 0x99 | |

| Subsistema s SPI (CCB, PDM, CLK, GPS) | Para todos subsistemas SPI | Hex |
|--|-------------------------------|------|
| | Leer temperatura 1 | 0x11 |
| | Leer temperatura 2 | 0x12 |
| | Leer temperatura 3 | 0x13 |
| | Leer voltaje 1 | 0x21 |
| | Leer voltaje 2 | 0x22 |
| | Leer voltaje 3 | 0x23 |
| | Leer posición 1 GPS | 0x31 |
| | Leer posición 2 GPS | 0x32 |
| | Leer posición 3 GPS | 0x33 |

| SIREN comandos para ejecutar acción | Descripción | Hex |
|---|------------------|------|
| | Lectura sensor 1 | 0x01 |
| | Lectura sensor 2 | 0x02 |
| | Lectura sensor 3 | 0x03 |
| | Lectura sensor 4 | 0x04 |
| | Lectura sensor 5 | 0x05 |
| | HK prueba OK | 0x21 |
| | HK prueba error | 0x2F |
| | Enciende CPU | 0X19 |
| | Apaga CPU | 0XF9 |
| | Alarm crítica | TBD |
| | CPU no responde | 0x4F |

Para cada subsistema se cuenta con un umbral de operación que indica los valores críticos de funcionamiento, así que para cada parámetro leído por el *Housekeeping*, se debe hacer una comparación con estos umbrales, con la finalidad de generar alarmas para cada subsistema.

Manejo de alarmas

El almacenamiento de alarmas se realiza como se muestra en la tabla 3.11 y consta de un arreglo de 3 *bytes* que recibe el nombre de registro de alarmas.

Tabla 3.11. Registro de alarmas.

| Registro de pila para alarmas | Alarma parte alta | Alarma parte baja |
|-------------------------------|-------------------------------|-------------------|
| Alarm1 | Subsistema que origina alarma | Tipo de Alarma |
| Alarm2 | | |
| Alarm3 | | |
| Alarm4 | | |

En el registro de alarmas, el primer *byte* corresponde al número que se le asigna al evento de alarma registrada, y consiste en una fila FIFO (*first input first output*). El siguiente *byte* corresponde al origen del subsistema que origina la alarma. Finalmente, el último *byte* corresponde al tipo de alarma que generó el subsistema.

En la tabla 3.12 se muestran los subsistemas que pueden originar alarmas, indicando con “x” el tipo de alarma que cada subsistema es capaz de generar. En la tabla 3.13 se muestran los tipos de alarma a generar, dependiendo del parámetro leído.

Tabla 3.12. Subsistemas que generan alarmas.

| Subsistemas | Hex | Tipos de Alarma | | | | | | | | |
|-------------|------|-----------------|------|------|------|------|------|------|------|------|
| | | 0xF3 | 0xFC | 0xA3 | 0xAC | 0xF5 | 0xC3 | 0xCC | 0x9F | 0xF9 |
| CLKB | 0xAA | X | x | - | x | - | x | x | - | - |
| CCB | 0xBB | X | x | - | x | - | x | x | - | - |
| GPS | 0xCC | X | x | x | - | - | x | x | - | - |
| PDM | 0xDD | X | x | - | x | - | x | x | - | - |
| EC | 0xEE | X | x | - | x | - | x | x | - | - |
| CPU | 0x99 | X | x | - | - | x | - | x | - | - |
| DST | 0x88 | X | x | - | - | x | - | x | - | - |
| HK | 0x77 | - | - | - | - | x | x | - | x | x |

Tabla 3.13. Alarmas válidas por subsistema.

| Tipos de alarma | Hex |
|--------------------------|------|
| Bajo voltaje | 0XF3 |
| Alta corriente | 0XFC |
| Baja temperatura por SPI | 0XA3 |
| Bajo voltaje por SPI | 0XAC |
| Temperatura analógica | 0XF5 |
| Falla de comunicación | 0XC3 |
| Mal cierre de relevador | 0XCC |
| Reinicio de HK | 0X9F |
| Bajo voltaje para HK | 0XF9 |

Las alarmas que los subsistemas pueden generar dependen de los parámetros leídos y son de distintos tipos. Algunos subsistemas comparten los mismos parámetros, como voltajes y corrientes, pero en el caso de temperaturas y condiciones de comunicación, algunos subsistemas no generan ese tipo de alarmas. El registro de alarmas será enviado únicamente si hay alarmas registradas, esto se controla por medio de una bandera del *Housekeeping*, que se activará al momento de registrarse un evento de alarma. Se implementó un comando para preguntar por el registro de alarmas, con este comando se revisará la bandera y de ser necesario, se responderá con las alarmas. En el caso de no haber alarmas, se implementó otro comando que indica esta situación.

La propuesta de control por comandos y el registro de alarmas funcionaron como se esperaba, esto derivó en la necesidad de crear un protocolo de comunicación entre CPU/SIREN más complejo, con la posibilidad de detectar errores en la comunicación.

Desarrollo del primer protocolo de comunicación

La primera propuesta de protocolo de comunicación entre CPU/SIREN y el *Housekeeping* cumple con las siguientes características:

- Comunicación maestro-esclavo: el *Housekeeping* sólo responde si CPU lo interrumpe.
- Comunicación asíncrona.
- Orientado a carácter: El intercambio de información se hace por medio de paquetes de *bytes* agrupados para formar una estructura completa de “n” *bytes*, cada uno con su significado correspondiente.
- Tasa de transferencia de 19,200 *bauds*.

Se agregó una comprobación de redundancia cíclica de 16 *bits*, para detección de errores. La detección de errores en 2 *bytes* fue necesaria, debido a que un parámetro está formado por 2

bytes, por lo que el valor del parámetro estará comprendido entre 0 a 65535. Los parámetros de monitoreo pueden ser enviados en unidades de centésimas o milésimas sin tener pérdida de información.

Comprobación de Redundancia Cíclica (CRC)

Para realizar la comprobación del CRC de un mensaje completo, se considera a éste como un polinomio y se divide entre otro polinomio fijo, llamado polinomio generador, a partir de esta operación se obtiene un residuo, el cual se anexa al mensaje y se transmite como un carácter de comprobación de bloque. En el receptor se repite el cálculo. Si no hay errores, el residuo debe ser el mismo que el que se transmitió; si éste es distinto ocurrió un error en la comunicación. En la ecuación 10 se muestra la forma en que se calcula el CRC de un dato.

$$CRC = R(x) = \frac{x^k P(x)}{G(x)} \quad \text{Ecuación 10}$$

Donde:

- $R(x)$, residuo de la división binaria
- k , indica el grado del polinomio
- $P(x)$, palabra o mensaje a ser transmitido
- $G(x)$, polinomio generador

La división en forma binaria está definida por la operación XOR. El polinomio generador elegido para la transmisión con CPU tiene el nombre **AC9A**, corresponde a un CRC de 16 bits que tiene la siguiente estructura:

$$G(x) = x^{16} + x^{14} + x^{12} + x^{11} + x^8 + x^5 + x^4 + x^2 + 1$$

Binario: 1 0101 1001 0011 0101

Como ejemplo se describen los pasos para calcular el CRC del dato 0x00A0h.

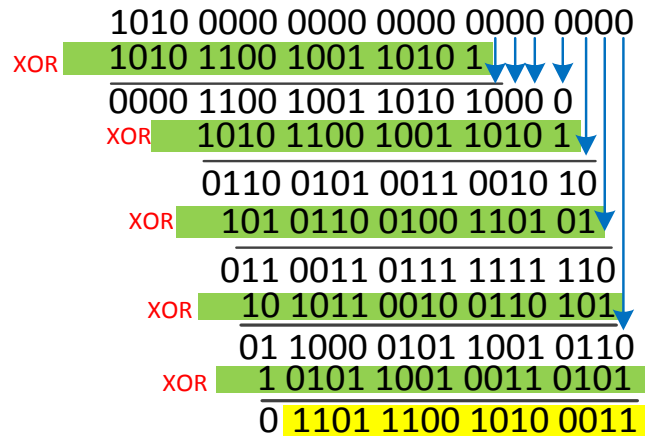
El grado del polinomio es, $k = 16$, de acuerdo a la expresión se calcula el numerador.

$$x^k P(x) = x^{16}(x^7 + x^5) = x^{23} + x^{21} + x^{19} + x^{18}$$

Binario: 1010 0000 0000 0000 0000 0000

Se procede a hacer la división binaria entre el numerador y el denominador, la cual se hace alineando los *bits* más significativos para comenzar a operar, posteriormente se agregarán los ceros ponderados que tiene el mensaje para seguir siendo operado por el polinomio

generador. Tal como se muestra en la figura 3.88, el residuo se toma como correcto hasta que no falte ningún cero por operar. El resultado serán los 16 bits menos significativos del residuo.



$$R(x) = 1101\ 1100\ 1010\ 0011\ (0xDCA3)$$

Figura 3.88. Ejemplo del calculo del CRC.

Dada a la complejidad de su implementación en código, se calculó solo para 2 bytes. A partir de esto se propuso el primer protocolo de comunicación, que consta de 8 bytes, su estructura se muestra en la figura 3.89.

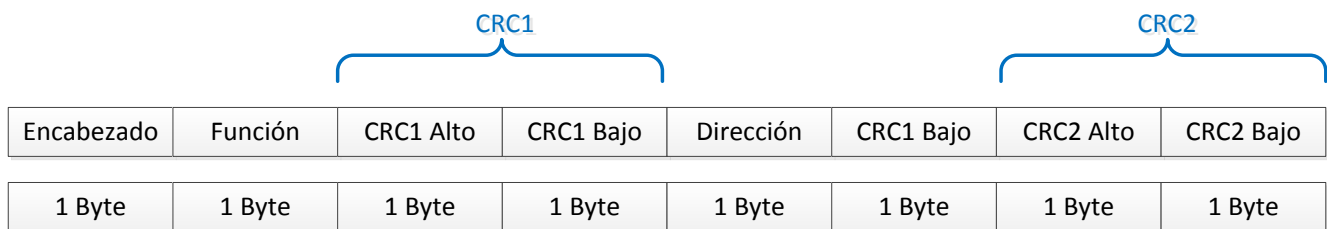


Figura 3.89. Estructura del protocolo propuesto.

Cada byte tiene el siguiente significado:

1. **Cabecera:** Indica que llega una petición de CPU.
2. **Comando:** Indica el tipo de comando; configurar un parámetro del *Housekeeping* o hacer una lectura de algún parámetro de éste.
3. **Parte alta del CRC1:** Parte Alta del CRC de los dos bytes anteriores.
4. **Parte baja del CRC1:** Parte baja del CRC de los dos bytes anteriores.
5. **Dirección:** Indica la dirección del subsistema.

6. **Dato:** Este *byte* indica el parámetro para configurar en el *Housekeeping*, si la trama fue enviada por CPU. Si el *Housekeeping* responde a una petición de CPU, este *byte* es un dato de lectura de algún parámetro.
7. **Parte alta del CRC2:** Parte Alta del CRC de los dos *bytes* anteriores.
8. **Parte baja del CRC2:** Parte baja del CRC de los dos *bytes* anteriores.

La interacción propuesta entre el *Housekeeping* y la CPU con este protocolo es la siguiente:

El *Housekeeping* se encuentra realizando alguna tarea y la CPU lo puede interrumpir con una petición. Esta petición puede consistir en configurar un parámetro o para hacer una lectura de algún parámetro que esté recibiendo de los otros subsistemas del DP.

Como resultado del desarrollo del banco de pruebas de CPU/SIREN, se inició la integración de *software* del *Housekeeping* con funciones básicas, que permitió integrar el control de los decodificadores para los circuitos de *High Level Command*, junto con la comunicación para SPI en modo diferencial.

3.4.2. Banco de pruebas SPI

El banco de pruebas SPI consiste en el diseño de un circuito que simula las condiciones de *hardware* y *software* de los subsistemas de DP, PDM y HVPS. Está basado en el mismo microcontrolador y el mismo tipo de receptores y transmisores LVDS que son utilizados en el *Housekeeping*. En el caso específico de GPS, se incluyeron sensores, ya que con este subsistema, se interactúa de esta forma y no con un FPGA.

En la figura 3.90 se presenta el diagrama de bloques de este banco de pruebas, el cual cumple con las siguientes funciones:

- Proporcionar la interfaz entre el *Housekeeping* y el microcontrolador que simula a los subsistemas.
- Enviar las señales TTL a un banco de pruebas de HVPS.
- El microcontrolador responde a los comandos enviados por el *Housekeeping* a través de la comunicación SPI. Los comandos de respuesta pueden ser modificados mediante una interfaz con una PC, por medio de su conector USB.
- Contiene los sensores de temperatura que contiene el subsistema GPS.
- Genera las líneas de alarma (*Alarm* y *Done*) y responde a las de *reset* (*Program_B* y *Reset*) mediante comandos enviados desde una CPU o por botones.

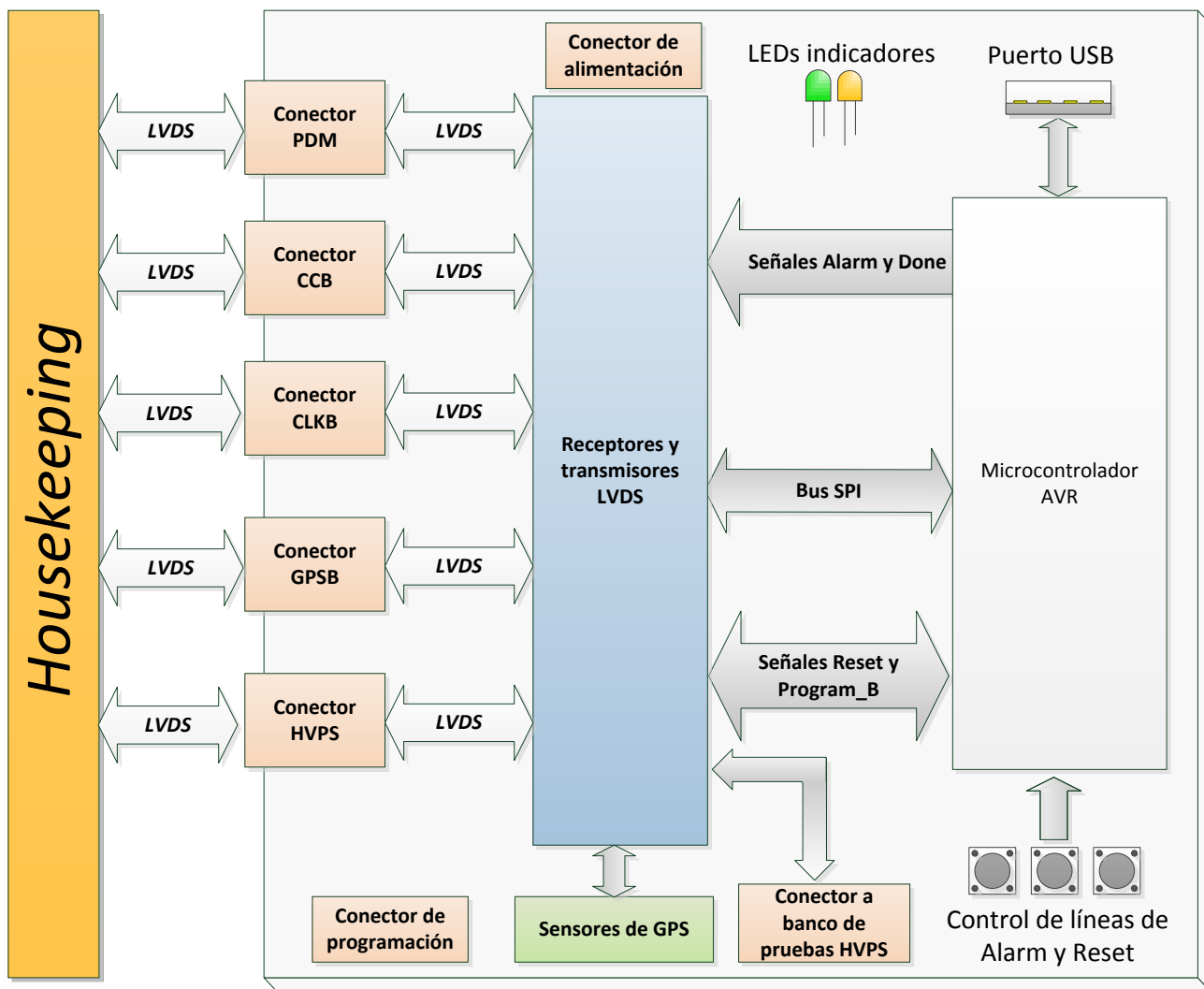


Figura 3.90. Diagrama de bloques del banco de pruebas SPI.

El diseño de este banco de pruebas se basó en el conocimiento adquirido durante la elaboración del *Housekeeping*. Además, se agregaron componentes para la interfaz con una computadora, una configuración de un FTDI que convierte los niveles de voltaje provenientes de la USART del microcontrolador a USB. También se utilizan compuertas OR y XNOR para controlar la actividad en los receptores y transmisores, es decir, controlar los estados en la salida.

El diseño del circuito impreso se realizó con la plataforma de diseño *Altium Designer*. En las figuras 3.91 y 3.92 se muestran la cara superior del diseño y la tarjeta finalizada.

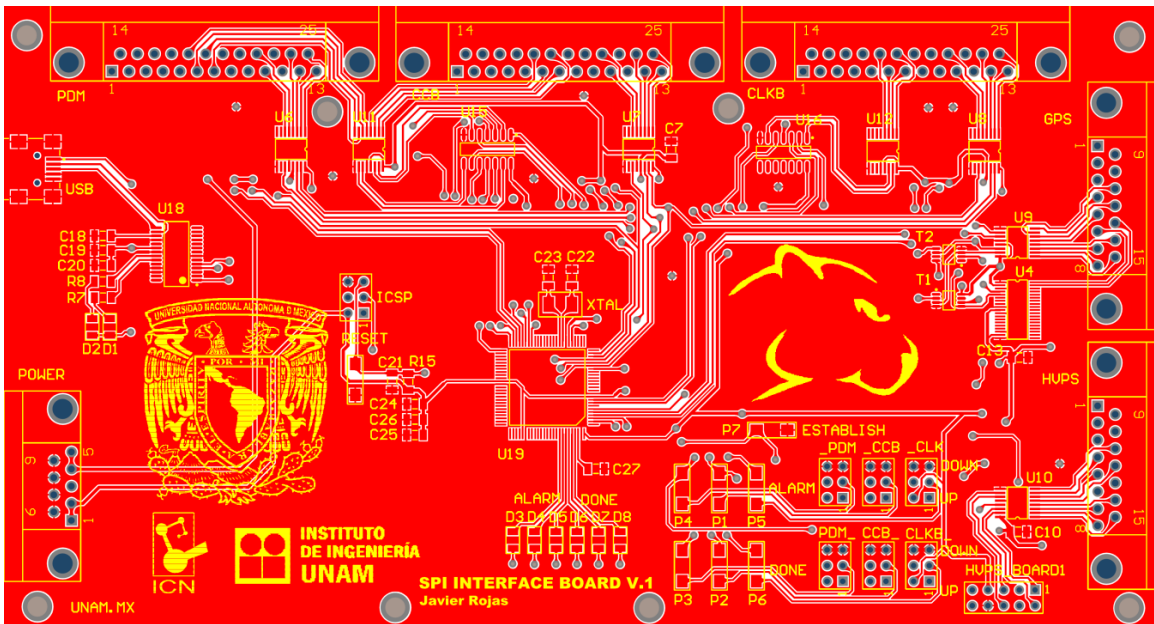


Figura 3.91. Cara superior del banco de pruebas de SPI.



Figura 3.92. Diseño terminado del banco de pruebas SPI.

Software

En cuanto al *firmware* desarrollado para este banco de pruebas, el proceso principal está indicado en la figura 3.93. Consiste en la creación de variables y configuración interna del microcontrolador (puertos, módulos e interrupciones). Posteriormente se queda en un ciclo

infinito, en donde está a la espera de interrupciones, las cuales son quienes rigen el comportamiento de las acciones a realizar por este banco de pruebas.

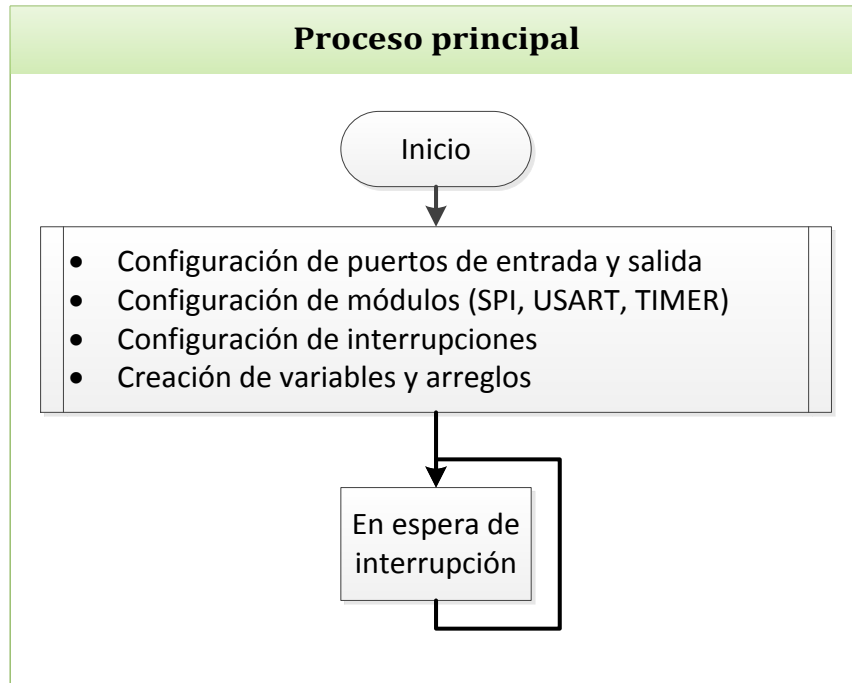


Figura 3.93. Diagrama de flujo del banco de pruebas SPI.

El banco de pruebas SPI funciona por petición, a través de interrupciones generadas en las interfaces de CCB, CLKB y PDM. Cuando el *Housekeeping* establece comunicación con este banco de pruebas, debe seleccionar un subsistema por medio de su *Chip Select* correspondiente, e indicar el tipo de parámetro que será leído.

La interrupción de comunicación por SPI consiste en detectar que se está seleccionando a algún dispositivo como esclavo, almacenar el comando enviado por el *Housekeeping*, verificar a qué subsistema se quiere simular y responder con un comando. Se repite el ciclo mientras esté seleccionado como esclavo. En la figura 3.94 se muestra el diagrama de flujo de la rutina de atención de esta interrupción.

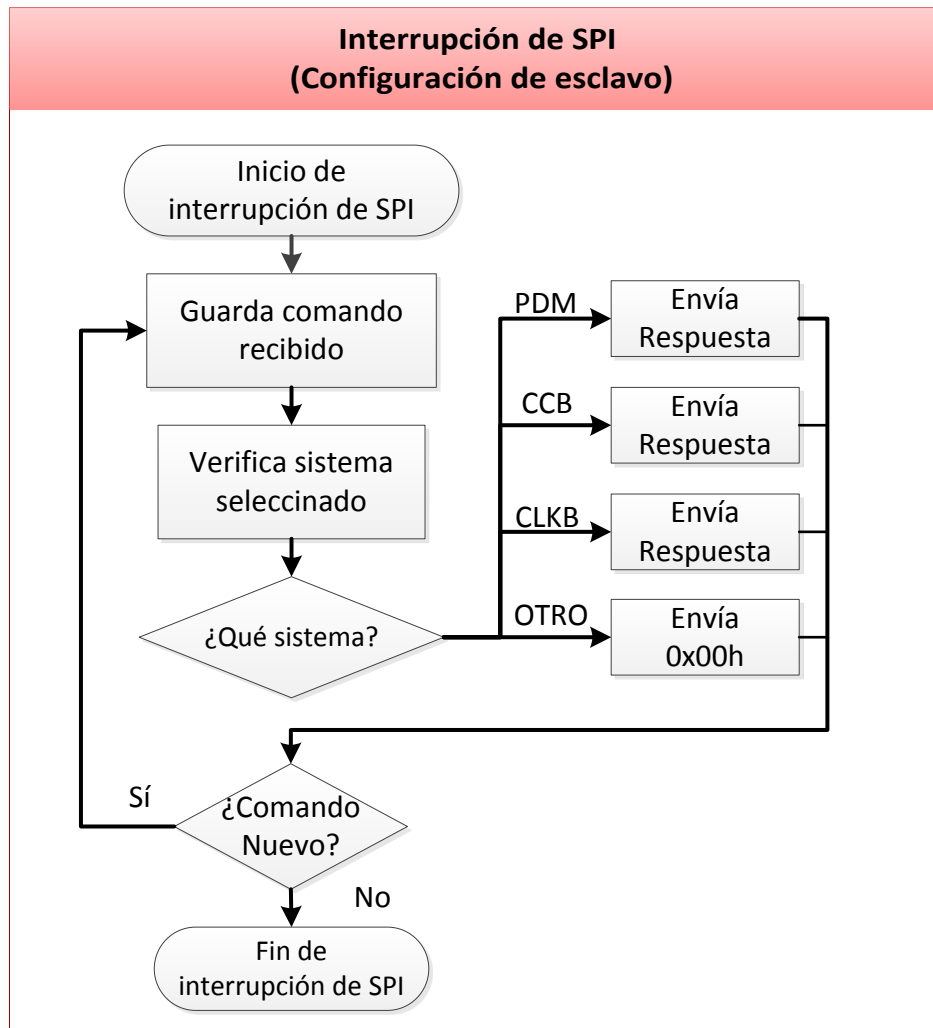


Figura 3.94. Diagrama de flujo de interrupción por SPI.

Para la generación de alarmas y recepción de *resets* se utilizan 6 de las interrupciones externas del microcontrolador. Al generar la interrupción se procede a verificar el estado en la línea deseada y se cambia su estado (si estaba en alto se cambia a bajo y viceversa). Para verificar dicho cambio se cuenta con unos LEDs indicadores.

La rutina de interrupción de la USART consiste en recibir una trama de tres *bytes*, para cambiar el valor de un parámetro de un subsistema. El primer *byte* indica el subsistema, el segundo el parámetro a cambiar y el tercero el valor que adquiere el parámetro.

En la figura 3.95 se muestran los diagramas de flujo de las rutinas de interrupción externas y de la USART. El diagrama de flujo de la interrupción externa es el mismo para las seis líneas (en diferente interrupción), mientras que en el diagrama de flujo de la interrupción de la USART sólo se muestra el cambio de un parámetro para un sistema; sin embargo, cada sistema tiene sus propios parámetros y cada uno de ellos puede adquirir tres valores diferentes.

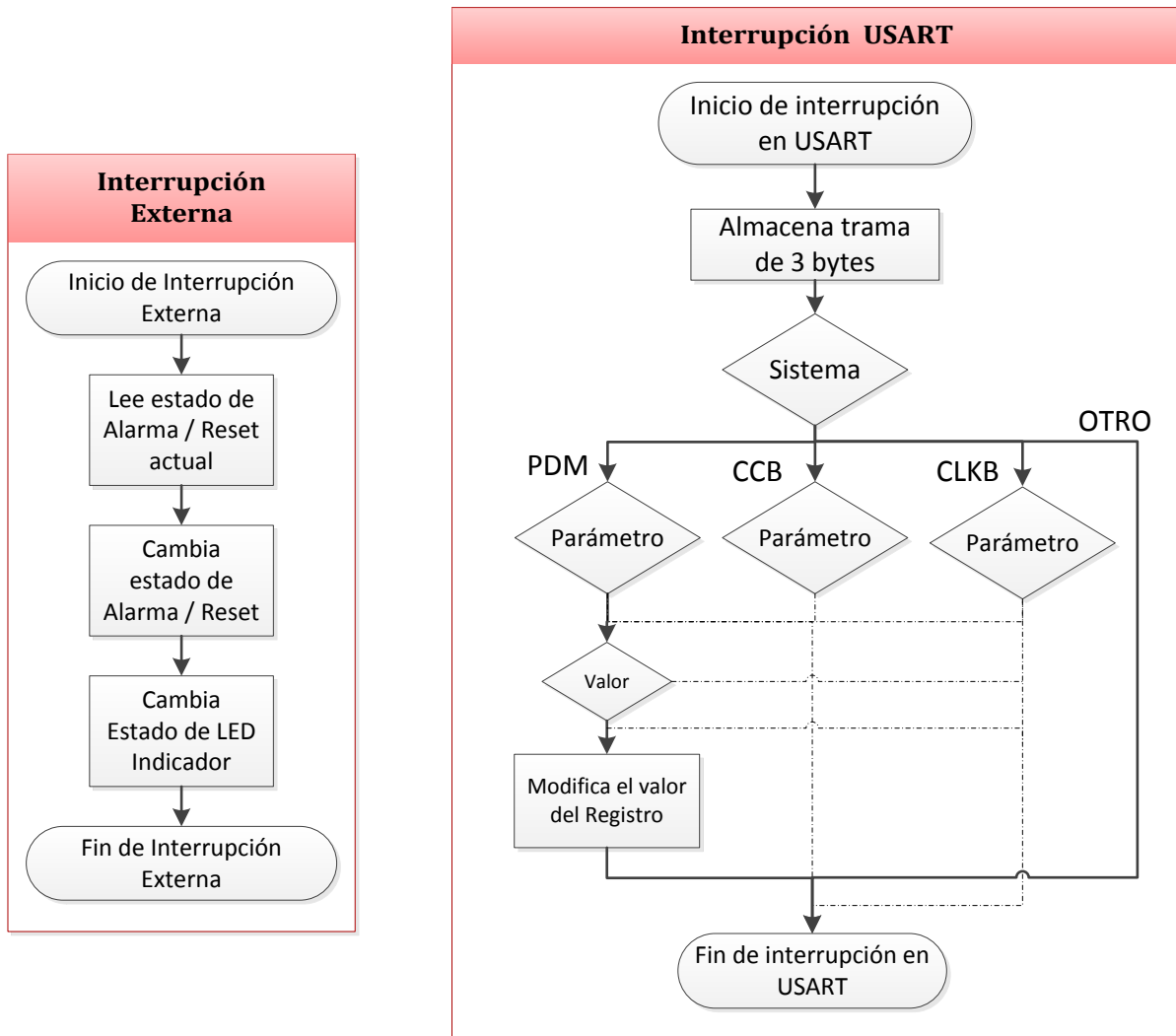


Figura 3.95. Diagrama de flujo de interrupción externa y de la USART.

Este banco de pruebas fue desarrollado con la finalidad de ser utilizado en cualquiera de las versiones que se implementen del *Housekeeping*, pero eventualmente se puede optimizar para cumplir con requerimientos más estrictos tanto en *hardware* como en *software*.

Banco de pruebas de HVPS

Para poder empezar con el diseño de este banco de pruebas, se describirá con detalle la interacción entre el *Housekeeping* y HVPS.

HVPS está encargado de generar los voltajes necesarios para la operación de los fotodetectores incluidos en PDM. Este último contiene un esquema de circuitos multiplicadores de voltaje en el orden de 1000 V que alimentan a los fotomultiplicadores. Necesita señales de voltaje de referencia para alimentar a los multiplicadores de voltaje,

señales de encendido/apagado (*ON/OFF*) y señales de estado (*STATUS*), además de una línea dedicada de alarma.

Un esquema que representa la interacción entre el *Housekeeping* y HVPS, en donde se especifican el número de señales a compartir y la interfaz usada se muestra en la figura 3.96:

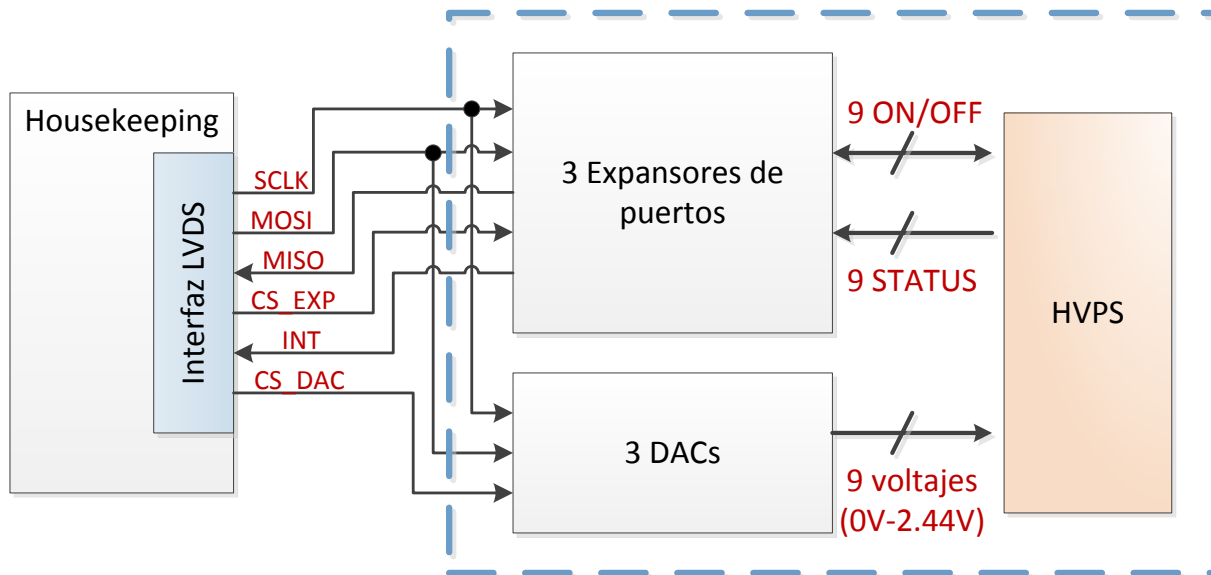


Figura 3.96. Interacción entre el *Housekeeping* y HVPS.

El control de los dispositivos asociados a HVPS se realiza a través de la interfaz LVDS, usando el protocolo SPI. Por propuesta del equipo encargado de HVPS se decidió una configuración con 6 líneas de comunicación, en donde sólo se usarán 2 señales *Chip Select* para la habilitación de los dispositivos. Estos dispositivos se encuentran en el módulo HVPS, por lo tanto el *Housekeeping* está a cargo de realizar un control a distancia de tales dispositivos.

Voltajes de referencia

Se requieren enviar 9 señales de voltaje que varíen dentro de un rango comprendido de 0 V a 2.44 V. Los circuitos integrados encargados de generar estos voltajes son los convertidores digitales/análogos. Este circuito integrado tiene las siguientes características:

- Interfaz SPI
- Frecuencia máxima de operación de 30 MHz
- Contiene 4 salidas diferentes con una resolución de 12 *bits*
- Cuenta con una referencia interna de precisión de 2.44 V
- Voltaje de operación de 2.7 V a 5.25 V

Señales **ON/OFF**, **STATUS** e interrupción

Para generar las señales necesarias para el control de los interruptores y la verificación del estado de los mismos, se requirió el uso de expansores de puertos.

Los circuitos integrados seleccionados tienen las siguientes características:

- Cuentan con 8 salidas bidireccionales
- Interfaz SPI con una frecuencia máxima de operación de 10 MHz
- Salida configurable de interrupción
- Cuenta con terminales de dirección para conectar varios expansores
- Voltaje de operación de 2.7 V a 5.5 V

Señales **ON/OFF.** Se requieren 9 señales bidireccionales para realizar el encendido y apagado de los interruptores que controlan los fotomultiplicadores de HVPS. Además estas líneas funcionan como un indicador del estado de los fusibles electrónicos que controlan cada salida de voltaje.

Señales **STATUS.** Se requieren 9 señales de entrada a través de las cuales HVPS indica el estado de carga de los fotomultiplicadores a través de un estado lógico predefinido.

Línea dedicada de alarma. Esta línea estará conectada directamente al *Housekeeping* para generar una interrupción. Cuando alguna de las señales *ON/OFF* o *STATUS* tenga un nivel lógico “bajo”, la línea de alarma se activará a través de la salida configurable de interrupción del expansor.

Diseño del banco de pruebas de HVPS

Con el fin de generar un *software* para el control de los dispositivos de HVPS, se diseñó un banco de pruebas que contuviera dichos dispositivos, de tal manera que al momento de requerirse la interacción entre el *Housekeeping* y HVPS se tuviera un avance.

Se diseñó una tarjeta que tuviera los dispositivos que deben ser controlados por el *Housekeeping*, además de proporcionar alguna interfaz para que el usuario sea capaz de interactuar con el banco de pruebas. Esta tarjeta se comunica con el *Housekeeping* a través del banco de pruebas SPI, que proporciona la alimentación, la interfaz LVDS y el conector entre ambos. En la figura 3.97 se muestra la conexión entre el *Housekeeping* y los bancos de prueba.

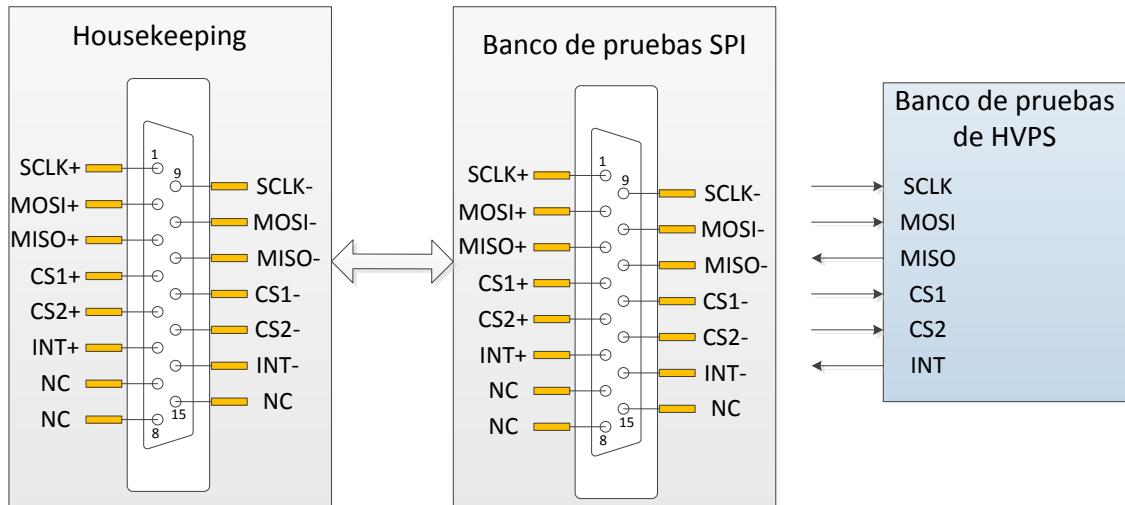


Figura 3.97. Conexión entre los bancos de prueba SPI y el *Housekeeping*.

Se requieren manejar todas las líneas implicadas en la operación HVPS: 9 *ON/OFF*, 9 *STATUS* y 9 voltajes de referencia. Para lograr esto se diseñó un circuito que contara con las siguientes características:

- 3 circuitos expansores de puertos y 3 convertidores digital-analógico
- Conector hacia un banco de pruebas de SPI que contiene la interfaz LVDS
- 9 líneas de entrada y salida para controlar las señales *ON/OFF*
- 9 líneas para generar las señales *STATUS*
- 9 líneas para la lectura de las señales de los DACs
- LEDs indicadores para las señales *ON/OFF*
- Botones para general manualmente cambios de nivel en las señales *STATUS*
- Interfaz para controlar con una computadora las señales *ON/OFF* y *STATUS*

Con las características anteriores, se requiere un dispositivo capaz de controlar 45 señales de entrada/salida, además de alguna interfaz para poder interactuar con el usuario a través de una computadora. El dispositivo seleccionado fue el mismo que se usó en el *Housekeeping*, el cual contiene un número suficiente de líneas de propósito general, además de contar con interfaces seriales para la comunicación.

El banco de pruebas de HVPS cuenta con una interfaz para ser controlado a través del puerto serie, usando un circuito integrado FTDI, que sirve para crear un puerto serie virtual a través del módulo USART del microcontrolador. A través de esta interfaz se pueden generar cambios de estado en las señales *ON/OFF* y *STATUS*, lo cual representaría una alarma para el *Housekeeping*, además de ofrecer la posibilidad de leer el voltaje generado por los DACs. El esquema general del banco de pruebas de HVPS se muestra en la figura 3.98.

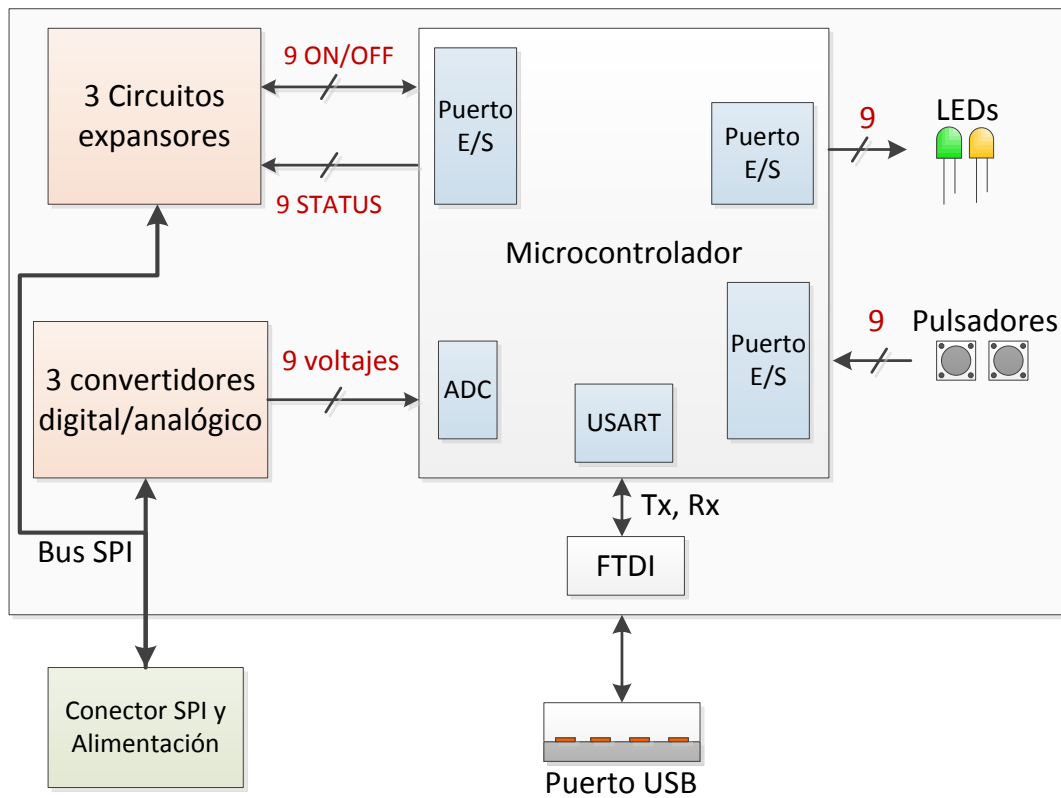


Figura 3.98. Diagrama del banco de pruebas de HVPS.

El diseño del circuito impreso se realizó con la plataforma Altium Designer. En las figuras 3.99 y 3.100 se muestran el diseño de la PCB y la tarjeta fabricada, respectivamente.

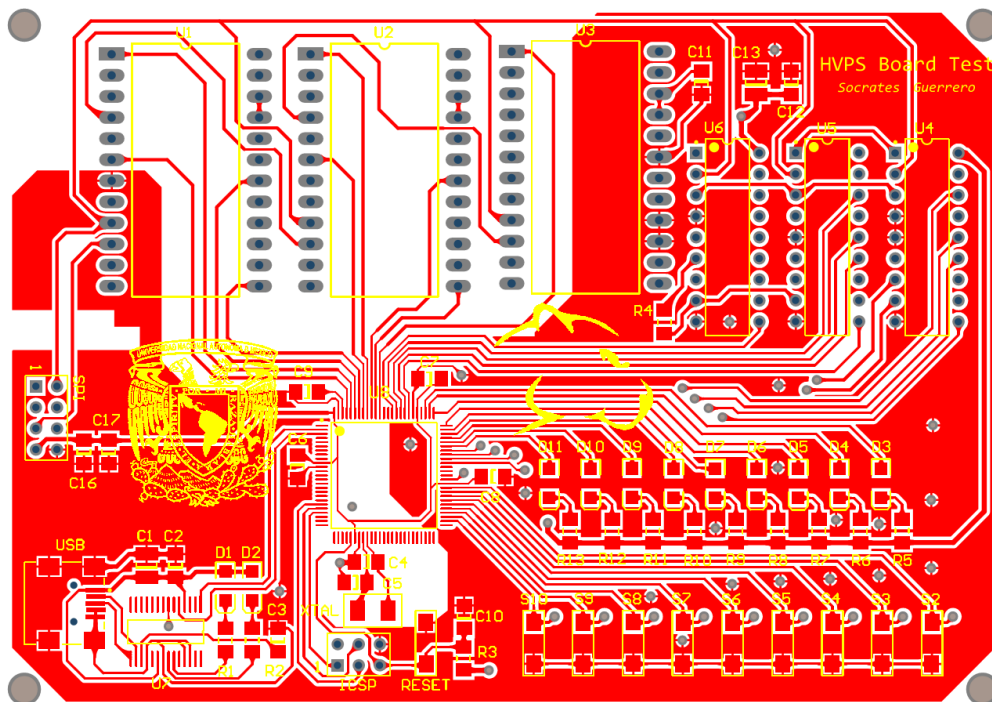


Figura 3.99. Diseño de PCB del banco de pruebas de HVPS.

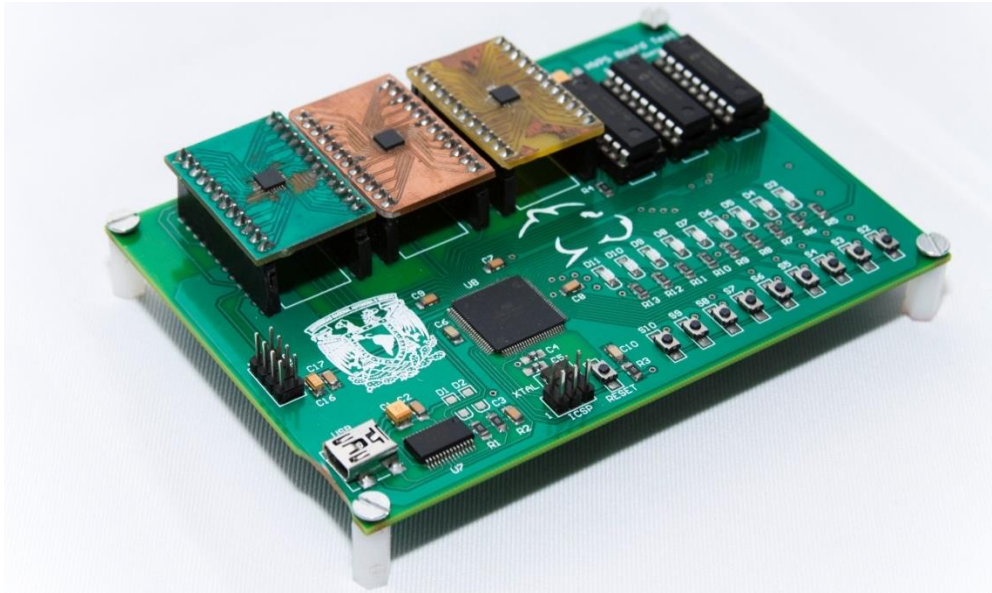


Figura 3.100. Banco de pruebas de HVPS ensamblado.

Software para el control del banco de pruebas de HVPS

Se diseñó un programa que a través del puerto serie es capaz de recibir comandos para realizar las siguientes tareas:

- **Cambio de nivel de señales *STATUS*.** Con este comando se genera un pulso en nivel “bajo” con una duración específica en alguna de las señales *STATUS*, para que el *Housekeeping* genere una alarma a través de esta condición. Pasado el tiempo definido, se levanta nuevamente la señal a su estado original.
- **Cambio de nivel de señales *ON/OFF*.** Con este comando se genera un pulso en nivel “bajo” para alguna de las señales *ON/OFF*, el microcontrolador esperará hasta que el *Housekeeping* reconfigure esta señal como salida y la establezca nuevamente en nivel alto. Un pulso en nivel “bajo” indica un mal funcionamiento en HVPS, por lo tanto el *Housekeeping* genera una alarma con esta condición.
- **Señales *ON/OFF* y *STATUS* en nivel bajo.** Este comando sirve para simular la situación en que alguna de las señales se quede en nivel “bajo” por un tiempo indefinido. El nivel de la señal se restablece introduciendo el mismo comando.
- **Lectura de los circuitos DAC.** Con este comando se realiza una lectura de los nueve voltajes generados por los DAC.

El *software* de control del banco de pruebas de HVPS comienza su operación cuando detecta los pulsos de encendido de los expansores, mismos que son generados a través del control que realiza el *Housekeeping*. A partir de este momento el usuario controlará el comportamiento de las señales *STATUS* y *ON/OFF*.

En la figura 3.101 se muestra un diagrama de flujo del programa que controla el banco de pruebas de HVPS.

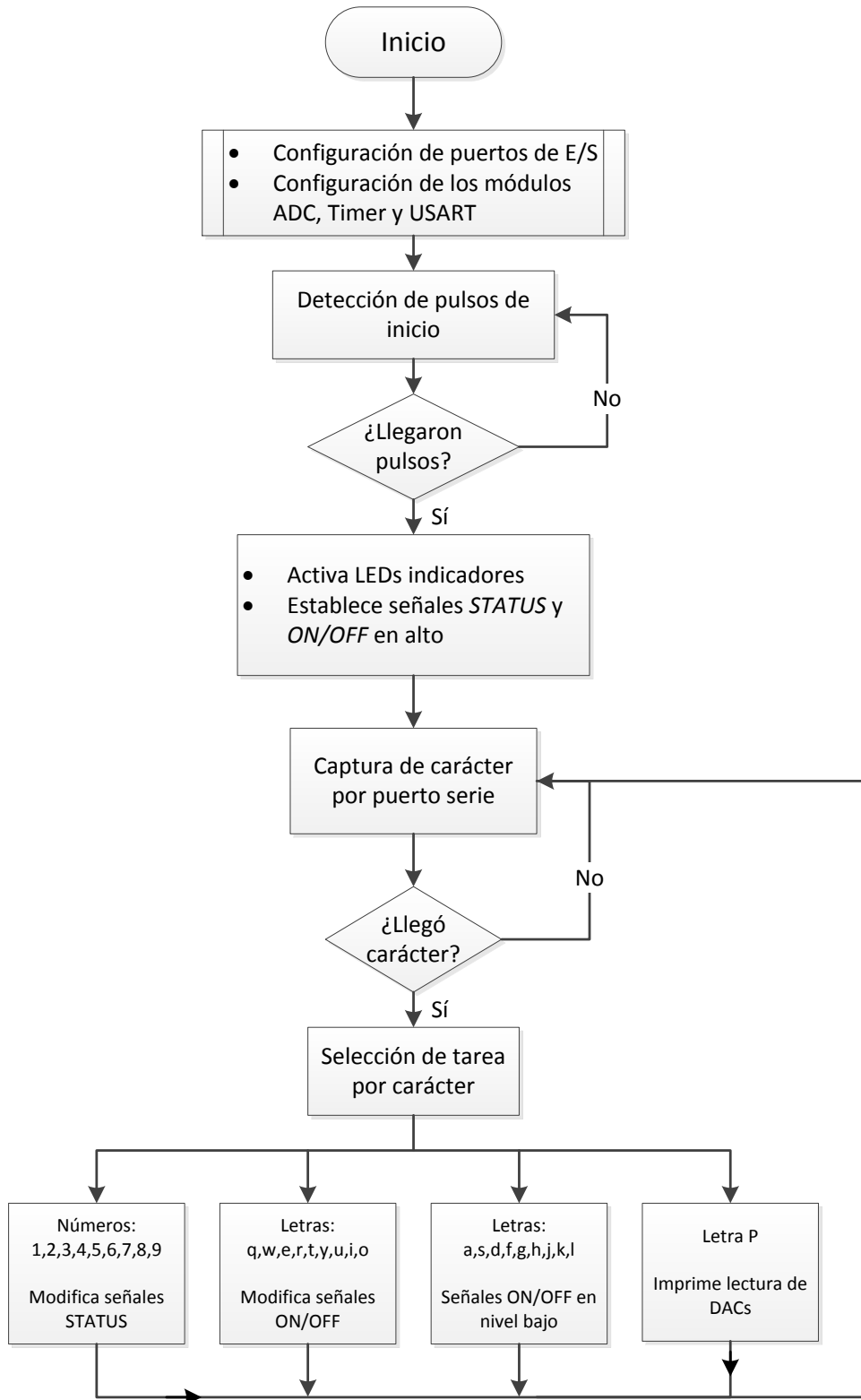


Figura 3.101. Software de control del banco de pruebas de HVPS.

3.4.3. Banco de pruebas de LVPS

Para validar el correcto funcionamiento de la adquisición de señales analógicas provenientes de las LVPS, por parte del *Housekeeping*, además del encendido y apagado de los subsistemas, se diseñó e implementó un tarjeta que fuera capaz de simular el comportamiento de las LVPS.

Subsistema LVPS

El subsistema LVPS consiste en una serie de módulos que proporcionan la alimentación para el *Housekeeping* y los demás subsistemas del DP. Cada LVPS contiene convertidores DC-DC, cuyo encendido/apagado es controlado a través de unos relevadores DPDT. Además, contiene circuitos monitores de voltaje y corriente, cuya salida de monitoreo está acondicionada en forma de voltajes que varían de 0 V a 5 V.

Por cada subsistema, LVPS proporciona un conector para la alimentación y otro destinado para el *Housekeeping*, donde se incluyen:

- 14 señales para activar y desactivar los relevadores (HL_CMD *set* y *reset*)
- 7 señales para la comprobación del estado ON/OFF de los relevadores (*Contact Closure*)
- 18 señales de monitoreo de voltajes y corrientes

Diseño del banco de pruebas

Se requería que este banco de pruebas entregara 18 voltajes para ser monitoreados por el *Housekeeping*, además de tener que contar con relevadores para ser controlados por medio de *High Level Command*. A su vez, con estos relevadores se leerían señales *Contact Closure*. De esta manera se simularía la activación y desactivación de subsistemas.

Debido a la versatilidad requerida por la tarjeta, se usaron circuitos convertidores digital-analógico, DAC, para generar los voltajes que monitoreara el *Housekeeping*. Cada voltaje representa el voltaje que entrega un monitor de las LVPS. Estos DACs son controlados por un microcontrolador, de esta manera el voltaje de salida de estos puede ser reprogramado.

Los relevadores que se usaron son los mismos con los que cuentan las LVPS. Éstos cuentan con dos estados, uno denominado *set* y otro denominado *reset*, que son manipulados por el *Housekeeping*. Además, para verificar de forma visual el estado de los relevadores se emplearon LEDs indicadores. Cuando el *Housekeeping* manda un pulso de activación (*set*), un interruptor se cierra permitiendo que se verifique la señal de *Contact Closure*, encendiendo un LED color verde. Cuando se manda un pulso de apagado (*reset*), se enciende un LED rojo. Un diagrama general del banco de pruebas de LVPS se muestra en la figura 3.102.

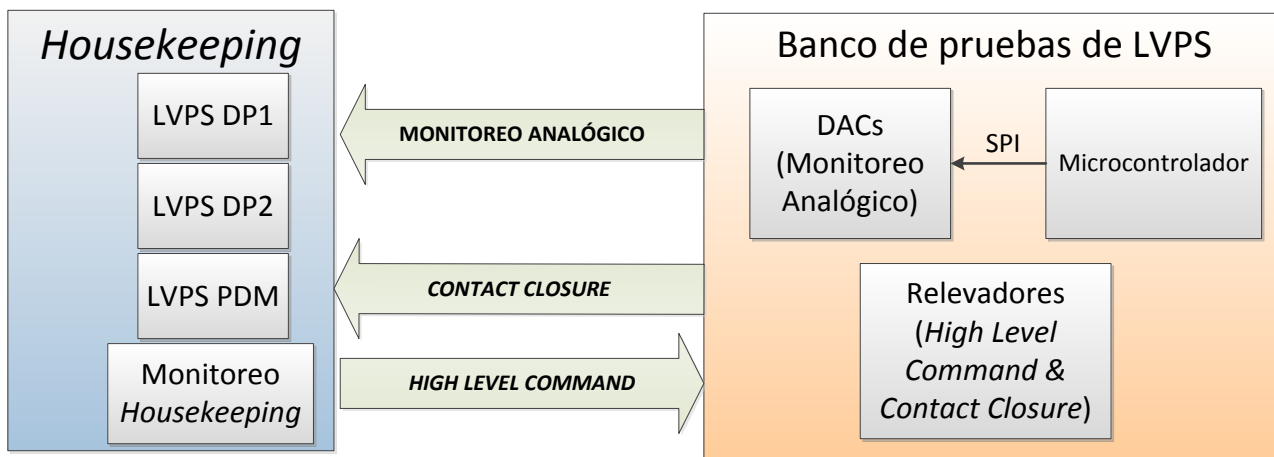


Figura 3.102. Diagrama del banco de pruebas de LVPS.

Cabe mencionar que para este banco de pruebas se colocó un *jumper*, con la finalidad de poder desactivar los LEDs por si es necesario. De esta manera la única forma de verificar el funcionamiento de los relevadores será de manera acústica, escuchándolos cambiar de estado.

Los DACs seleccionados tienen una resolución de 10 bits, cada empaquetado cuenta con 8 canales.

El número de voltajes que se requiere generar son 18, por lo que se usan 3 empaquetados. Éstos se controlan por medio del estándar SPI con un microcontrolador, el cual es programado para establecer los voltajes de salida de los DACs. Como se muestra en la figura 3.103, sólo se requieren 5 líneas para controlar los DACs.

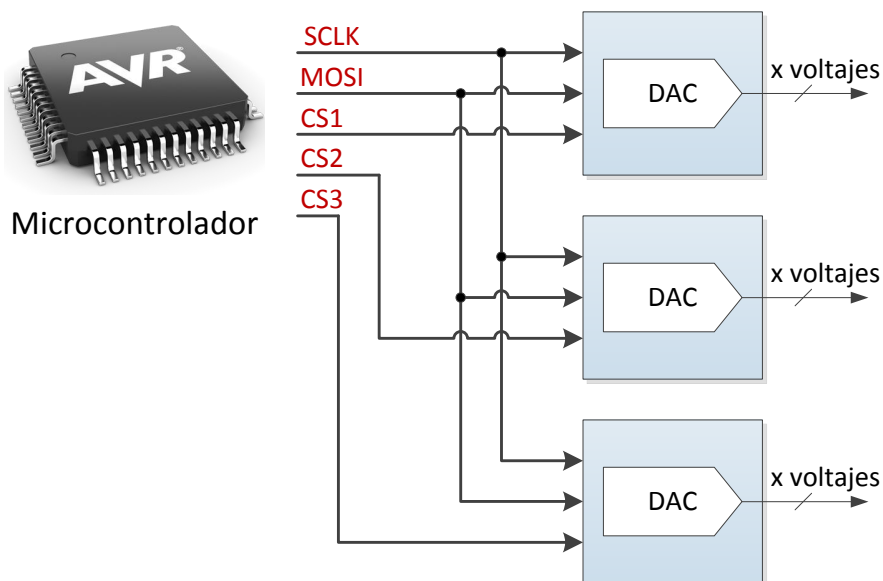


Figura 3.103. Control de los DACs del banco de pruebas de LVPS.

Software del banco de pruebas de LVPS

Para generar un voltaje de salida en alguno de los canales de los DACs, el microcontrolador se comunica con éste mandándole un comando de dos *bytes*, en este comando se especifica el canal que se desea activar, así como el valor del voltaje que se desea obtener en la salida del DAC.

El programa se desarrolló para que el microcontrolador fuera capaz de seleccionar el DAC correcto, así como el canal de éste. En la figura 3.104 se muestra el diagrama de flujo del programa del banco de pruebas de LVPS.

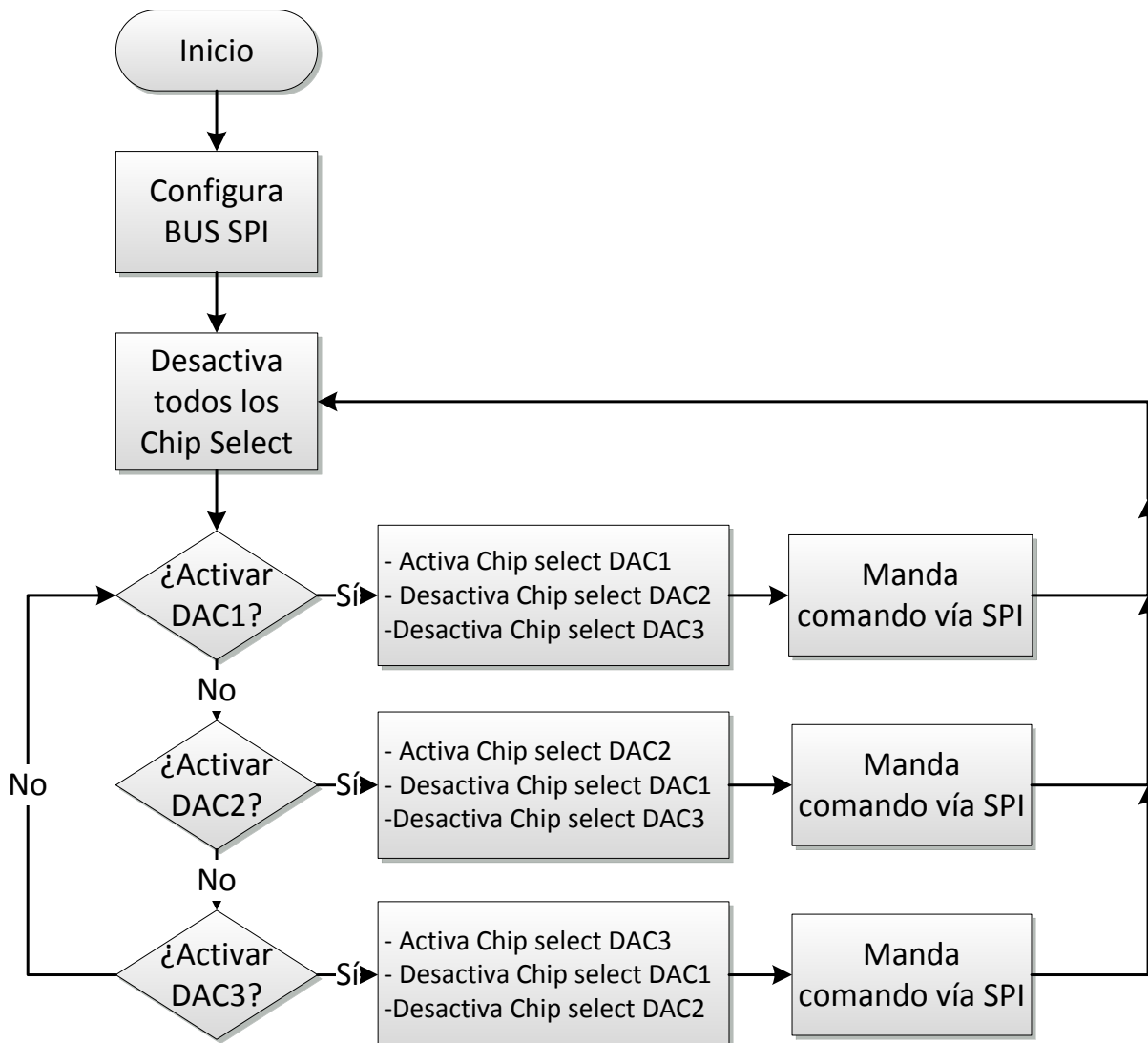


Figura 3.104. Control de los DACs.

Diseño del PCB

El diseño del PCB se llevó a cabo en el *software Eagle*, en la figura 3.105 se muestra la cara *top* del circuito impreso.

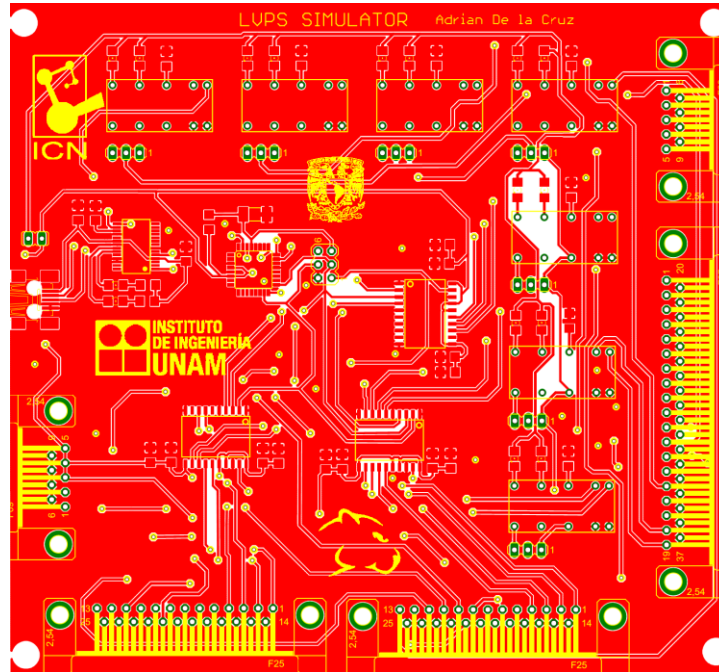


Figura 3.105. Vista *top* del banco de pruebas de LVPS.

Una vez que la PCB fue terminada de fabricar se procedió a colocar los dispositivos en ésta. En la figura 3.106 se muestra la tarjeta ya ensamblada en su totalidad.

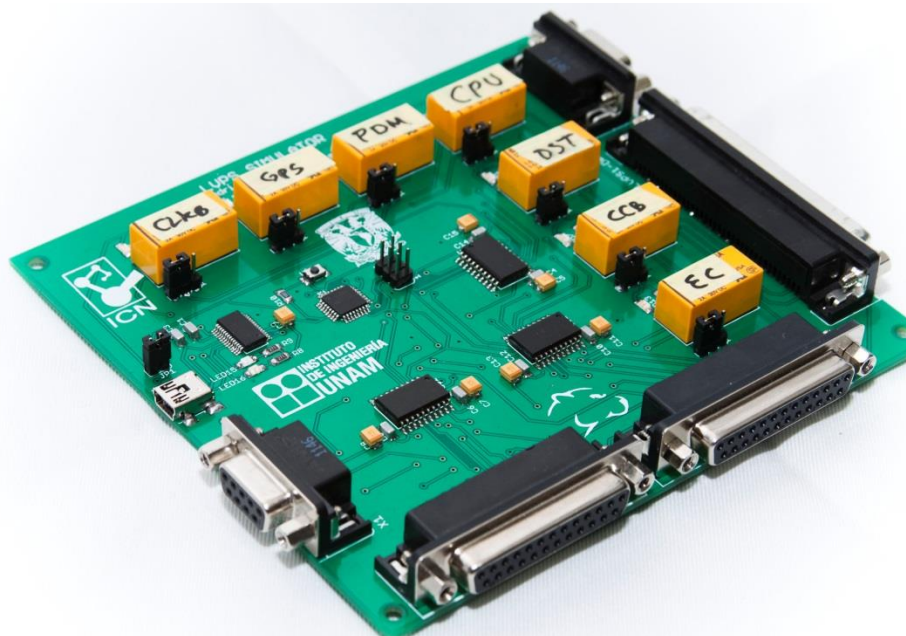


Figura 3.106. Banco de pruebas de LVPS ensamblado.

3.5. Versión de vuelo del *Housekeeping*

El primer prototipo fue probado en el Instituto Nacional de Física Nuclear (INFN), en Italia, junto con los primeros prototipos reales de sistemas que conforman el DP. Como resultado de esa integración se aprobó la congelación del *hardware*, pero se implementaron mejoras para prevenir fallos y convertir esta segunda versión en un prototipo de vuelo, desarrollando principalmente *firmware* para el *Housekeeping*, de acuerdo a los lineamientos de la filosofía de operación. En el capítulo 5 se abordarán los detalles de la primera integración del *Housekeeping* en el DP.

3.5.1. Medidas de protección contra fallas

Después de la primera integración, se estudiaron los casos en los que el *Housekeeping* puede quedar sin funcionamiento y el alcance de este tipo de sucesos. Esto es de vital importancia, debido a que se tiene que asegurar en la medida de lo posible que un error crítico en el *Housekeeping* afecte de manera mínima al funcionamiento del DP y de todo el instrumento.

De los casos estudiados, sólo se pudieron atender aquellos que no involucraran un cambio drástico en la configuración general del *hardware* del *Housekeeping*, debido al limitado tiempo para implementar dichos cambios y a que la colaboración congeló hasta esa etapa el diseño del *Housekeeping*.

Se trabajó en lo siguiente:

- **Control de habilitadores de dispositivos de LVDS.** Durante la primera integración, se comprobó que no todos los subsistemas con los que se tenía interacción pueden asegurar un estado de alta impedancia en sus líneas de comunicación, cuando éstos son deshabilitados durante la comunicación. Esto podría generar problemas debido a que la línea de comunicación MISO se comparte entre todos los esclavos, así que se modificó el manejo de dicha línea usando receptores diferentes y controlándolos con su terminal de habilitación. En la figura 3.107 se muestra la nueva configuración en la comunicación SPI.
- **Asegurar estado en líneas de reprogramación de FPGAs.** Como medida para evitar la reprogramación no deseada de los FPGAs, a través de las líneas *Program_B*, se colocaron resistencias de *pull-up* en éstas, para asegurar que la línea tenga un estado inicial alto. Anteriormente esto sólo se hacía mediante *software*, así que esta configuración se modificó para agregar confiabilidad.

- Asegurar estado de habilitación para el decodificador de líneas encendido y apagado de subsistemas.** Se colocaron resistores en la terminal *latch* que controla la salida del decodificador para el *High Level Command*, con el fin asegurar un estado inicial en esta terminal. Esto evitará cambios en la salida del decodificador que generen un apagado o encendido no deseado.

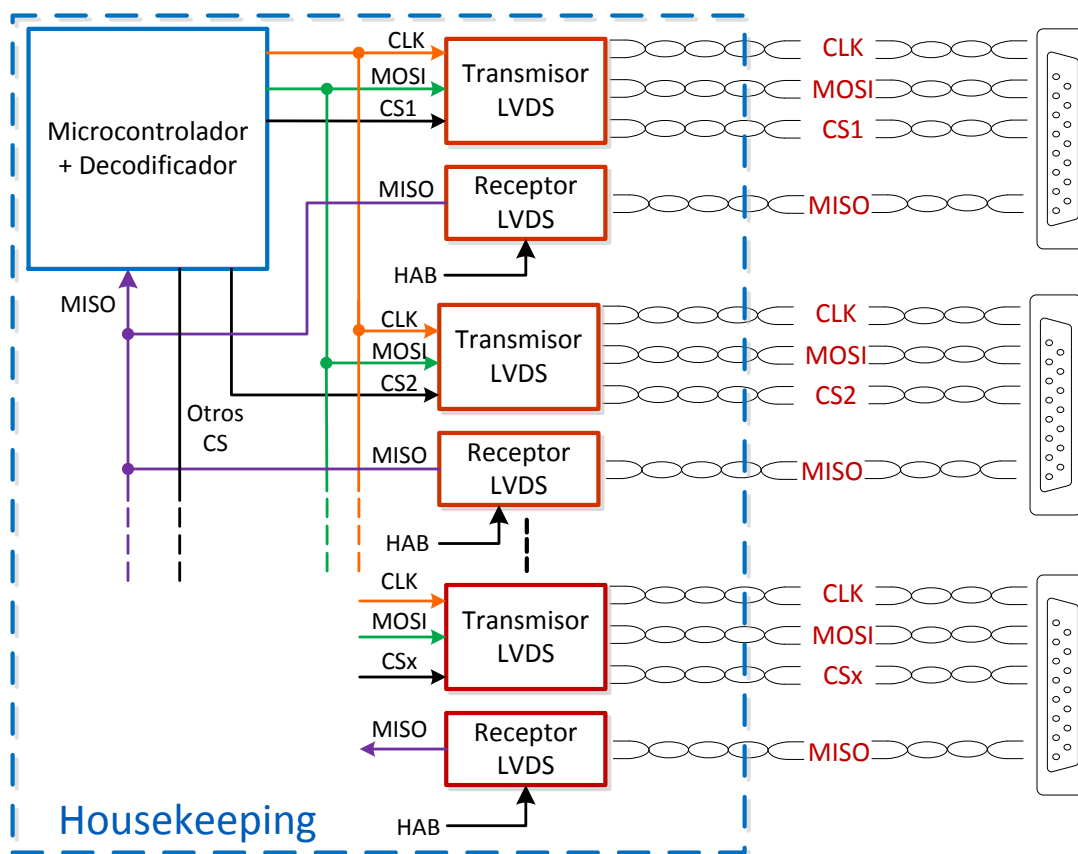


Figura 3.107. Diagrama general de la conexión SPI.

3.5.2. Diseño del *software* final

El *software* desarrollado en el primer prototipo del *Housekeeping* consistía en bloques de código, donde cada parte que controlaba algún módulo o interfaz funcionaba de forma independiente. Esto permitía probar partes de *hardware* de manera efectiva y al integrar se reutilizaba código uniéndose al preexistente, algo que era ineficiente porque las líneas de código aumentaban cuando se agregaba una funcionalidad para el *hardware* en la integración de *software*. Debido a lo anterior, se vio en la necesidad de crear librerías específicas para cada función, de tal forma que antes de cumplir con la filosofía de operación del *Housekeeping*, se buscó operar el microcontrolador con bloques embebidos de *software*, para ir subiendo el nivel de programación, de tal forma que implementando funciones específicas se pueda llegar a controlar el microcontrolador.

Desarrollo de librerías

Empezando por el primer nivel, se modificaron registros del microcontrolador para generar funciones que finalmente llegarían a controlar la interfaz de algún subsistema. En los siguientes cuadros se describe de forma simplificada las tareas que realiza cada una de las librerías hechas.

Llamaremos funciones de alto nivel a las que permiten al usuario interactuar con los módulos internos del microcontrolador. Mientras que las de bajo nivel están enfocadas a la configuración de registros dentro del microcontrolador. Esto permite que cualquiera de los desarrolladores pueda explotar las funcionalidades del microcontrolador sin tener que saber cómo configurarlo, permitiendo un desarrollo más rápido y sencillo de entender.

La librería para el control de los módulos USART del microcontrolador, es una de las librerías esenciales que se crearon, debido a que ésta permite depurar el *software* desarrollado, al comunicar el microcontrolador con una terminal de computadora para controlar y saber que sucede dentro del *Housekeeping*. En la figura 3.108 se muestran sus funciones.

Librería Serial

Funciones de alto nivel:

- Inicializar USART, indicando el *baudrate* y el número a usar. Se cuenta con 3 USART disponibles.
- Enviar un solo *byte* por USART seleccionada.
- Enviar un arreglo de caracteres por USART seleccionada.
- Enviar cadena de caracteres con salto de línea por USART seleccionada.
- Enviar un número en decimal sin signo por USART seleccionada.
- Enviar un número decimal con signo por USART seleccionada.
- Enviar un número en formato binario, octal o hexadecimal por USART.
- Enviar un número con punto decimal, indicando cuantas cifras después del punto y con signo por USART seleccionada.

Funciones de bajo nivel:

- Escribe los registros necesarios para establecer *baudrate*.
- Habilita la interrupción en recepción.
- Habilita el transmisor y receptor.
- Configura: transmisión, dato 8 *bits*, no paridad, 1 *bit* de parada.
- Transmisión de un *byte* por poleo de interrupción de bandera en el registro del *buffer*.

Figura 3.108. Librería serial del *Housekeeping*.

Junto con la implementación de la librería serial, se tuvo la necesidad de agregar la comprobación del código de redundancia cíclica, cada vez que se requería establecer comunicación con CPU. Ésta se utiliza para hacer el cálculo del CRC cuando se recibe una cadena de *bytes* en el receptor de la USART, y con ayuda de ésta se determina si los datos que llegaron son correctos. También se utiliza para la transmisión de datos del *Housekeeping* por medio de RS422, se calcula el valor del CRC para mandar el resultado de este en la cadena de *bytes* a enviar. En la figura 3.109 se muestran las funciones que realiza esta librería.

Librería CRC16

** Hace uso de la Librería Serial*

Funciones de alto nivel:

- Envía trama de ejecutar acción 5 *bytes*.
- Envía trama de lectura 6 *bytes*.
- Envía trama de respuesta a lectura 8 *bytes*.
- Envía trama de configuración 9 *bytes*.
- Envía trama de alarmas de baja prioridad 9 *bytes*.
- Envía trama de alarmas de alta prioridad 5 *bytes*.

Funciones de bajo nivel:

- Calcula CRC de una trama de “n” *bytes*.
- Une 2 *bytes* de CRC.
- Verifica CRC calculado y recibido.
- Calcula CRC de una trama de acción.
- Calcula CRC de una trama de lectura.
- Calcula CRC de una trama de respuesta de lectura.
- Calcula CRC de una trama de configuración.
- Calcula CRC de una trama de alarmas abaja prioridad.
- Calcula CRC de una trama de alarmas de alta prioridad.

Figura 3.109. Librería para calcular el CRC.

Otra librería que ayudó a depurar el *software* fue la de retardos, un ejemplo fue controlar el tiempo de las impresiones en pantalla en la terminal de computadora. Su implementación se vio utilizada para el control de señales de duración definida, como los pulsos de activación de relevadores, y para las conversiones analógicas digitales del microcontrolador. En la figura 3.110 se muestran sus funciones.

Librería Retardos

Funciones de alto nivel:

- Retardos en milisegundos.
- Retardo en microsegundos, con tiempo mínimo de 10 microsegundos.

Funciones de bajo nivel:

- Inicializa y configura los *Timers* 3 y 4.
- Crea una base de tiempo de 10 microsegundos.
- Crea una base de tiempo de 1 milisegundo.
- Crea una base de tiempo de 1 segundo.
- Habilita el modo captura y comparación con interrupción.
- Prescalamiento del reloj de 16 MHz para las bases de tiempo necesarias.

Figura 3.110. Librería de retardos del *Housekeeping*.

Una vez establecida la forma de depuración del *Housekeeping*, se procedió a incluir librerías más robustas para la interacción con sus subsistemas. En cuanto a la adquisición analógica se creó una librería que permitía controlar el módulo de conversión analógico-digital, modificando los registros correspondientes del microcontrolador. La figura 3.111 muestra la configuración que se hace con estos registros. Una ventaja de controlar la conversión analógica-digital del microcontrolador, es que se pueden hacer varias lecturas, invocando una sola vez la librería. Esto permite hacer promedios de la lectura para obtener valores más estables y evitar obtener lecturas erróneas en las señales.

Librería ADC

Funciones de alto nivel:

- Manda a hacer una lectura analógica del canal seleccionado.
- Lectura a un termistor de *Housekeeping*.
- Lectura de monitoreo de un voltajes en LVPS seleccionado.
- Lectura de monitoreo de una corriente en LVPS seleccionado.

Funciones de bajo nivel:

- Configura el registro de conversión analógica para canal 0 por defecto.
- Justificación a la derecha.
- Voltaje de referencia interno de 2.56V o 5V dependiendo de la lectura.
- Desactiva el *buffer* digital para las terminales analógicas del ADC.

Figura 3.111. Librería para la adquisición analógica del *Housekeeping*.

Una vez probada la librería de adquisición analógica, se implementó una librería que hacía uso de ésta para interactuar con los subsistemas que envían valores analógicos. Entre éstas se incluyen funciones de adquisición, monitoreo, almacenamiento y depuración para todas las lecturas analógicas. La librería se muestra en la figura 3.112.

Librería Analógica de Housekeeping

** Hace uso de la Librería ADC*

Funciones de alto nivel:

- Monitoreo de Lenses.
- Depuración del monitoreo de Lenses por el puerto serie.
- Comparación de umbrales altos de Lenses.
- Comparación de umbrales bajos de Lenses.

- Monitoreo de LVPS-HK.
- Depuración del monitoreo de LVPS-HK por el puerto serie.
- Comparación de umbrales altos de LVPS-HK.
- Comparación de umbrales bajos de LVPS-HK.

- Monitoreo de LVPS-PDM.
- Depuración del monitoreo de LVPS-PDM por el puerto serie.
- Comparación de umbrales altos de LVPS-PDM.
- Comparación de umbrales bajos de LVPS-PDM.

- Monitoreo de LVPS1-DP.
- Depuración del monitoreo de LVPS-DP1 por el puerto serie.
- Comparación de umbrales altos de LVPS-DP1.
- Comparación de umbrales bajos de LVPS-DP1.

- Monitoreo de LVPS2-DP.
- Depuración del monitoreo de LVPS-DP2 por el puerto serie.
- Comparación de umbrales altos de LVPS-DP2.
- Comparación de umbrales bajos de LVPS-DP2.

Figura 3.112. Librería del monitoreo analógico por sistema.

Para la comunicación con los subsistemas SPI se crearon librerías que realizaban las siguientes funciones:

- Configuración del bus SPI
- Interacción con el sistema HVPS
- Generación de alarmas
- Almacenamiento en la memoria microSD

En la figura 3.113 se pueden ver las funciones de la librería que configuran la transmisión y recepción de datos en SPI.

Librería de SPI

Funciones de alto nivel:

- Inicializa la comunicación SPI.
- Envía un reloj para recibir datos por la línea MISO.
- Enciende *Chip Select* de un dispositivo dado.

Funciones de bajo nivel:

- Escribe valores en el registro de comunicación SPI para prescalar la velocidad de operación a 2 MHz.
- Escribe el modo de operación 0 en registros de comunicación SPI.
- Envía un comando por SPI usando las banderas de interrupción de transmisión completa.
- Configura puertos para salida de *Chip Select*.

Figura 3.113. Librería general de la comunicación SPI.

El *Housekeeping* realiza el control directo de los dispositivos de HVPS (expansores de puertos y circuitos DAC). Debido a que los convertidores requieren una configuración especial y los expansores de puertos funcionan con interrupciones, se vio la necesidad de crear una librería exclusiva para interactuar con HVPS. En la figura 3.114 se muestran sus funciones.

Librería de HVPS

** Hace uso de la Librería SPI*

Funciones de alto nivel:

- Inicializa la comunicación SPI con los DACs.
- Envía una palabra de 24 *bits* a un DAC en específico.
- Encender multiplicadores de voltaje.

Funciones de bajo nivel:

- Configura las interrupciones de los expansores de puertos.
- Configura puerto y manda salida de pulsos para encender multiplicadores.
- Lectura de las terminales del expansor de puertos.
- Lectura de las terminales de interrupción del expansor de puertos.
- Configura una conexión *daisy chain* en los DAC.

Figura 3.114. Librería para interactuar con el sistema HVPS.

Para la generación de alarmas provocadas por los datos que se obtienen por SPI, se hizo otra librería, debido al manejo de las estructuras donde se almacenan los datos por subsistema. Además de manejar las líneas dedicadas para reprogramación de subsistemas con FPGA. Las funciones se muestran en la figura 3.115.

Librería de AlarmasSPI

** Hace uso de la Librería SPI*

Funciones de alto nivel:

- Detección de líneas dedicadas de SPI para envío de *Softreset* al subsistema.
- Detección de líneas dedicadas de SPI, para envío de *Hardreset* al subsistema.
- Almacenamiento de valores en estructuras de subsistemas.
- Comparación con umbrales para la generación de alarmas.

Funciones de bajo nivel:

- Lectura de puertos de entrada del microcontrolador.
- Enmascaramiento de señales y comparación de estados para generar acción sobre líneas dedicadas.

Figura 3.115. Librería para almacenamiento y generación alarmas de SPI.

Para el almacenamiento de datos se utilizó una API para guardarlos en la memoria microSD. Se muestra en la figura 3.116.

Librería de almacenamiento en microSD

** Hace uso de la Librería SPI*

Funciones de alto nivel:

- Detección de la memoria *microSD*.
- Activación del *Chip Select* del dispositivo.
- Generación de archivos por subsistema.
- Asignación de parámetros por archivo.
- Abre un documento y agrega valores por subsistema.
- Creación de documentos nuevos por primera ocasión.

Figura 3.116. Librería para almacenamiento en la *microSD*.

Para el control del encendido y apagado de sistemas se creó una librería que controla la activación de relevadores. Esta incluye el control de las líneas de selección del decodificador, verificación de los estados de cierre de relevadores, y el tiempo de activación de estos últimos. En la figura 3.117 se muestra el listado de funciones de esta librería.

Librería High Level Command

Funciones de alto nivel:

- Encender un relevador.
- Apagar un relevador.
- Verificar estado de cierre de relevadores (*Contact Closure*).

Funciones de bajo nivel:

- Control de puertos de microcontrolador para seleccionar las 14 salidas del decodificador.
- Habilitación encendida del decodificador.
- Habilitación apagada del decodificador.
- Control de Latch y duración de selección.
- Estado cero de selección del decodificador.
- Lectura de puertos destinados a cierre de relevadores (*Contact Closure*).

Figura 3.117. Librería para control de relevadores de las LVPS.

Una tarea fundamental para el funcionamiento del *Housekeeping* es la inicialización y variables, para esto se creó una librería encargada de realizar esta tarea. En la figura 3.118 se muestran sus características.

Librería de Startup

Funciones de alto nivel:

- Configura puertos
- Inicializa Interrupciones

Funciones de bajo nivel:

- Escribe valores al registro de puertos para asignar un comportamiento de entrada o salida.
- Escribe valores al registro de puertos para asignar resistencia de *pull-up*.
- Desactiva y configura interrupciones externas.
- Declaración de variables globales.
- Declaración de estructuras globales para almacenamiento de datos.
- Inicialización de variables globales.
- Inicialización de estructuras globales.

Figura 3.118. Configuración inicial en el *Housekeeping*.

MEMORIA EEPROM

Una característica del *Housekeeping* es la capacidad que tiene para generar alarmas con base en los datos que adquiere de los diferentes subsistemas del DP. Para lograr esto, deben almacenarse valores de configuración en donde se establezcan los umbrales contra los cuales serán comparados los datos adquiridos con los subsistemas.

Cuando el *Housekeeping* inicia su operación, es necesario descargar valores de la memoria EEPROM a la memoria *flash* para poder interactuar con ellos en funcionamiento. Así mismo, si sucediera algún cambio en los umbrales se debe hacer lo mismo en la memoria EEPROM. Es necesario asegurar que estos valores se mantengan aún incluso cuando el *Housekeeping* se recupere de un reinicio inesperado, por tal motivo se hace uso de la memoria EEPROM del microcontrolador. Esta memoria cuenta con celdas unitarias que pueden ser programadas de forma independiente, la celda es programada cuando es descargada, en este caso se dice que el valor lógico es '0', en cambio, cuando la celda es borrada, el valor lógico de es '1'. Esta memoria se encuentra organizada en localidades de 8 *bits*, para acceder a las localidades se emplea un registro de 8 *bits*. Debido a esto y que los datos que se tienen que almacenar son de tipo entero de 16 *bits*, la única forma de almacenarlos es separando el dato entero en dos datos de 8 *bits*. Una vez hecho esto, las dos partes que componen el dato entero son almacenadas en localidades contiguas. Para recuperar un dato, se direccionan los dos datos contiguos para ser concatenados. Cabe mencionar que se debe de tener cuidado al direccionar la memoria, ya que si se hace una mal direccionamiento, se puede alterar la información de alguna otra localidad de la memoria EEPROM u obtener un dato erróneo.

La EEPROM tiene tres modos de operación [26][27] los cuales, son las siguientes:

- Limpiar
- Escribir
- Limpiar y Escribir

Cuando se requiere escribir un nuevo dato en la memoria EEPROM, se debe borrar primero la localidad de memoria correspondiente y posteriormente escribir en ella. En ocasiones, para escribir un nuevo valor simplemente es necesario limpiar la localidad y algunas veces no es necesario ejecutar ninguna operación, por lo que si se implementa un algoritmo de escritura eficiente, se pueden reducir los tiempos de este proceso. Su funcionamiento se muestra en la figura 3.119.

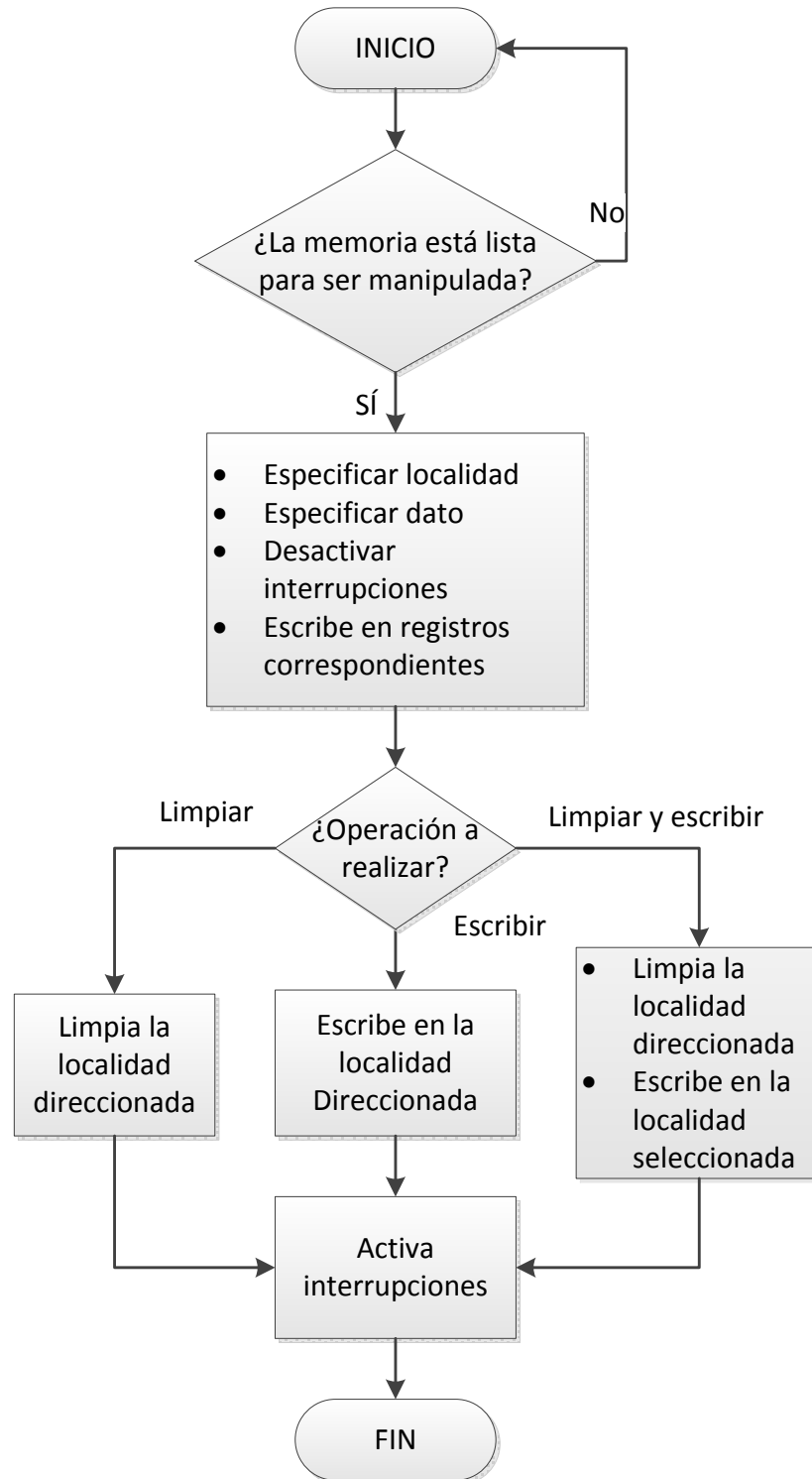


Figura 3.119. Diagrama de almacenamiento en EEPROM del *Housekeeping*.

Para hacer más eficiente el almacenamiento de la memoria EEPROM del *Housekeeping*, se creó una librería para manipular la memoria EEPROM, la cual se muestra en la figura 3.120.

Librería EEPROM

Funciones de alto nivel:

- Guarda un *byte* en la memoria EEPROM.
- Guarda un dato *int* en la memoria EEPROM.

Funciones de bajo nivel:

- Lee los registros para comparar si el valor a sobrescribir se ha actualizado.
- Guarda un *byte* en una dirección de la EEPROM seleccionada.
- Separa un dato *int* en 2 variables *byte* para almacenarlas en la memoria EEPROM de acuerdo a la dirección seleccionada.

Figura 3.120. Almacenamiento en EEPROM del *Housekeeping*.

Durante la operación del *Housekeeping*, es de vital importancia asegurar un funcionamiento continuo, de manera que si se presenta un problema de comunicación o de otro tipo y que genere que el microcontrolador quede fuera de operación o dicho de otra manera “congelado”, se cuente con un método para generar un reinicio del sistema y continuar con el funcionamiento normal. Debido a esto, se generaron librerías para manipular el módulo *Watchdog* del microcontrolador, ésta se muestra en la figura 3.121.

Librería de Watchdog

Funciones de alto nivel:

- Desactiva el *Watchdog*.
- Función de encendido del *Watchdog* seleccionando el tiempo de interrupción, tiempos desde 16 ms hasta 8 segundos.
- Resetea el contador del *Watchdog*.

Figura 3.121. Librería para el control del *Watchdog* en el *Housekeeping*.

El *Watchdog Timer* tiene como función principal resetear el microcontrolador al presentarse algún problema de *software*. Éste consiste en un contador, que al ser activado requiere ser reseteado o desactivado antes de desbordarse, para que el programa siga corriendo normalmente, si es que no existe problema alguno o si éste ha sido corregido. Este contador se puede escalar para manipular el tiempo que tarda en desbordarse, este tiempo puede ser de unos pocos milisegundos hasta unos cuantos segundos.

Comunicación entre CPU y el *Housekeeping*

El *Housekeeping* es un sistema que es totalmente controlado por la CPU, y en algunos casos muy específicos tiene la tarea de responder de forma autónoma a eventos esperados. Para cada requerimiento se han implementado comandos para las siguientes tareas:

- Encender o apagar un sistema.
- Lectura de los parámetros de cualquier subsistema que interactúa con *Housekeeping*.
- Configurar umbrales de comparación para valores altos y bajos en la generación de alarmas.
- Generar alarmas de baja prioridad, a partir de la comparación de los valores leídos, con rangos de los umbrales previamente establecidos.
- Generar alarmas de alta prioridad para casos de extrema precaución, donde la CPU debe tomar acción en la situación presentada.
- Realiza descargas de información de modo periódico a la CPU o SIREN según sea el caso.

Para establecer comunicación entre la CPU y el *Housekeeping*, por requerimiento de la misión y para lograr una comunicación más eficiente y capaz de detectar errores, se propuso un protocolo de comunicación.

Este protocolo debía ser capaz de comunicar ambos subsistemas para ejecutar todas las acciones antes mencionadas, con la posibilidad de ser flexible y eficiente en su transmisión. Para esta tarea se presentaron varias propuestas de comunicación, las cuales, como resultado de las integraciones, fueron cambiando, de tal forma que se llegó a una propuesta concreta. En este apartado se explicará la versión que finalmente opera en el *Housekeeping*.

Con el fin de distinguir las tareas que tiene que realizar el *Housekeeping*, se propusieron 5 tramas con 5 *headers* o cabeceras diferentes, éstas son los *bytes* que al inicio de la trama funcionan como indicadores de la tarea a realizar. El contenido de la trama, así como el tamaño de ésta, depende del tipo de tarea que CPU le solicite al *Housekeeping*; además se propuso que el *byte* de fin de trama, tuviera el valor hexadecimal 0x17h, cuyo valor en ASCII indica fin de un bloque de transmisión. En la tabla 3.14 se muestra la estructura general de cada trama propuesta.

Tabla 3.14. Tipos de tramas para la comunicación con CPU.

| | Cabecera | Descripción/Sentido | Fin |
|---|-----------------|--|------------|
| 1 | 0x31 | Ejecutar acción/ CPU a HK | 0x17 |
| 2 | 0x32 | Hacer una lectura/ CPU a HK y HK a CPU | 0x17 |
| 3 | 0x33 | Configurar umbral de Parámetro/ CPU a HK | 0x17 |
| 4 | 0x34 | Alarma de baja prioridad/ HK a CPU | 0x17 |
| 5 | 0x35 | Alarma de alta prioridad HK a CPU | 0x17 |

Se explicará cada una de las tramas de comunicación en el orden que se muestran en la tabla anterior.

Ejecutar acción

Cuando el *Housekeeping* recibe como cabecera el hexadecimal conteniendo el valor 0x31h, se interpreta como una acción que debe ejecutar. La trama tiene una longitud definida de 5 *bytes*. Su estructura se muestra en la tabla 3.15:

Tabla 3.15. Trama para encender el subsistema CLKB.

| Cabecera | Comando | Fin de Bloque | CRC16H | CRC16L |
|-----------------|----------------|----------------------|---------------|---------------|
| 0x31 | 0x65 | 0x17 | 0xFE | 0x68 |

- Cabecera. *Byte* cuyo valor en hexadecimal es 0x31, indica una acción.
- Comando. Alguna de las tareas que el *Housekeeping* tiene que realizar. Éstas incluyen principalmente acciones que llevan al *Housekeeping* a actuar sobre los subsistemas, como el encendido y la reprogramación de subsistemas, o bien en el monitoreo de variables. Los comandos son listados en la tabla 3.16.
- Fin. *Byte* que indica el fin de bloque transmitido
- Comprobación de redundancia cíclica. 2 *bytes* que son el resultado del CRC.

Entre las tareas que puede realizar el *Housekeeping* se encuentra el monitoreo por subsistemas. Cuando el *Housekeeping* recibe este comando de la CPU, inmediatamente se enviarán a esta última todos los parámetros de lectura de forma individual del subsistema especificado.

Tabla 3.16. Tareas que el *Housekeeping* puede realizar.

| Acción a ejecutar | Comando Hex | Acción a ejecutar | Comando Hex |
|--------------------------|--------------------|--------------------------|--------------------|
| Turn On CLKB | 65h | Prog_B CCB | 48h |
| Turn On CCB | 71h | Prog_B CLKB | 4Ah |
| Turn On GPS | 74h | Prog_B PDM | 49h |
| Turn On PDM | 7Ah | Reset CCB | 4Fh |
| Turn On EC | 63h | Reset CLKB | 42h |
| Turn On DST | 64h | Reset PDM | 4Eh |
| Turn On CPU | 61h | | |
| | | Monitoreo HK | 11h |
| Turn Off CLKB | 72h | Monitoreo LVPS1-DP | 22h |
| Turn Off CCB | 77h | Monitoreo LVPS2-DP | 33h |
| Turn Off GPS | 79h | Monitoreo LVPS-HK | 44h |
| Turn Off PDM | 78h | Monitoreo LVPS-PDM | 55h |
| Turn Off EC | 76h | Monitoreo CCB | 66h |
| Turn Off DST | 66h | Monitoreo CLKB | 77h |
| Turn Off CPU | 73h | Monitoreo GPS | 88h |
| | | Monitoreo PDM | 99h |
| | | Monitoreo LENSES | AAh |

Hacer una lectura

Cuando el *Housekeeping* recibe como cabecera el hexadecimal conteniendo el valor 0x32h, se interpreta como una petición para realizar la lectura de un parámetro determinado de cualquier sistema. Una petición de lectura tiene una longitud definida de 6 *bytes*. Su estructura se muestra en la tabla 3.17:

Tabla 3.17. Lectura de la temperatura 1 del *Housekeeping*.

| <i>Cabecera</i> | <i>Sistema</i> | <i>Parámetro</i> | <i>Fin de Bloque</i> | <i>CRC16H</i> | <i>CRC16L</i> |
|-----------------|----------------|------------------|----------------------|---------------|---------------|
| 0x32 | 0x11 | 0x81 | 0x17 | 0x37 | 0x78 |

- *Cabecera*. *Byte* cuyo valor en hexadecimal es 0x32, indica una petición de lectura.
- *Sistema*. En este *byte* se indicará el tipo de sistema que puede ser seleccionado para lectura.
- *Parámetro*. El *Housekeeping* se encargará de medir voltajes, corrientes y temperaturas de los sistemas, pero cada uno de los sistemas tiene parámetros específicos. En la tabla 3.18 se puede observar qué parámetros aplican para cada sistema y su valor hexadecimal.
- *Fin*. *Byte* que indica el fin de bloque transmitido.
- Comprobación de redundancia cíclica. 2 *bytes* que son el resultado del CRC.

En la tabla de parámetros 3.18 se puede observar en azul la dirección en hexadecimal correspondiente a cada sistema y en verde la dirección en hexadecimal que corresponde a uno de los parámetros del sistema. Sólo las tramas válidas son para los casos que contemplen lecturas. Los no contemplados no serán reconocidos como lecturas válidas. En total son 63 parámetros que pueden ser leídos de forma individual por CPU/SIREN o agrupados por subsistema.

Tabla 3.18. Parámetros que pueden ser leídos por subsistema.

| Parámetro | Hex | Sistema | HK | LVPS1-DP | LVPS2-DP | LVPS-HK | LVPS-PDM | CCB | GPS | CLKB | PDM | HVPS | LENSES |
|---------------------|------|---------|----------|----------|----------|----------|----------|--------|--------|--------|---------|-------|--------|
| | Hex | Hex | 0x11 | 0x22 | 0x33 | 0x44 | 0x55 | 0x66 | 0x77 | 0x88 | 0x99 | 0xAA | 0xBB |
| Voltaje | 0x61 | V1 | - | V 5 CCB | V 12 CPU | V 12 HK | V 5 PDM | V 3.3 | - | Vc Aux | V Asic1 | DAC 1 | - |
| | 0x62 | V2 | - | V 5 CLKB | V 5 DST | V 3.3 HK | V 5 EC | V 2.5 | - | Vc Int | V Asic2 | DAC 2 | - |
| | 0x63 | V3 | - | V 5 GPS | - | - | - | V 1.2 | - | - | V Asic3 | DAC 3 | - |
| | 0x64 | V4 | - | - | - | - | - | V 5 | - | - | V fpga1 | DAC 4 | - |
| | 0x65 | V5 | - | - | - | - | - | v 1.8 | - | - | V fpga2 | DAC 5 | - |
| | 0x66 | V6 | - | - | - | - | - | - | - | - | - | DAC 6 | - |
| | 0x67 | V7 | - | - | - | - | - | - | - | - | - | DAC 7 | - |
| | 0x68 | V8 | - | - | - | - | - | - | - | - | - | DAC 8 | - |
| | 0x69 | V9 | - | - | - | - | - | - | - | - | - | DAC 9 | - |
| Corriente | 0x71 | I1 | - | I 5 CCB | I 12 CPU | I 12 HK | I 5 PDM | I 3.3 | - | - | - | - | - |
| | 0x72 | I2 | - | I 5 CLKB | I 5 DST | I 3.3 HK | I 5 EC | I 2.5 | - | - | - | - | - |
| | 0x73 | I3 | - | I 5 GPS | - | - | - | I 1.2 | - | - | - | - | - |
| | 0x74 | I4 | - | - | - | - | - | - | - | - | - | - | - |
| | 0x75 | I5 | - | - | - | - | - | - | - | - | - | - | - |
| Temperatura | 0x81 | T1 | Temp 1 | - | - | - | - | T max | Temp 1 | Temp 1 | Temp 1 | - | Temp 1 |
| | 0x82 | T2 | Temp 2 | - | - | - | - | T fpga | ADC | - | - | - | Temp 2 |
| | 0x83 | T3 | Temp 3 | - | - | - | - | - | - | - | - | - | Temp 3 |
| | 0x84 | T4 | Temp 4 | - | - | - | - | - | - | - | - | - | Temp 4 |
| | 0x85 | T5 | Temp 5 | - | - | - | - | - | - | - | - | - | Temp 5 |
| | 0x86 | T6 | Temp SPI | - | - | - | - | - | - | - | - | - | Temp 6 |
| | 0x87 | T7 | - | - | - | - | - | - | - | - | - | - | Temp 7 |
| | 0x88 | T8 | - | - | - | - | - | - | - | - | - | - | Temp 8 |
| | 0x89 | T9 | - | - | - | - | - | - | - | - | - | - | Temp 9 |
| Total de Parámetros | | 63 | 6 | 6 | 4 | 4 | 4 | 10 | 2 | 3 | 6 | 9 | 9 |

Respuesta a una lectura

Cuando el *Housekeeping* recibe la solicitud de lectura de un parámetro dado, éste debe ser capaz de responder con el valor solicitado de forma correcta. Una respuesta a la petición de lectura tiene una longitud de 8 *bytes*, conformada por la trama de lectura más 2 *bytes* que contienen la información a enviar. Su estructura se muestra en la tabla 3.19, considerando como el dato a enviar 0x0100.

Tabla 3.19. Envío de la lectura de la temperatura 1 del *Housekeeping*.

| Cabecera | Sistema | Parámetro | Data H | Data L | Fin de Bloque | CRC16H | CRC16L |
|----------|---------|-----------|--------|--------|---------------|--------|--------|
| 0x32 | 0x11 | 0x81 | 0x01 | 0x00 | 0x17 | 0x56 | 0xAA |

Configurar umbral de comparación

Cuando el *Housekeeping* recibe como cabecera el hexadecimal conteniendo el valor 0x33, se interpreta como una petición para configurar un umbral de comparación. Una petición de configuración tiene una longitud de 9 *bytes*. Las estructuras de umbral alto y bajo se muestran en las figuras 3.20 y 3.21, respectivamente, configurando el parámetro Temperatura 1.

Tabla 3.20. Configuración del umbral alto con el valor 0x03FFh.

| <i>Cabecera</i> | <i>Sistema</i> | <i>Parámetro</i> | <i>Umbral Alto</i> | <i>DATA H</i> | <i>DATA L</i> | <i>Fin de Bloque</i> | <i>CRC16H</i> | <i>CRC16L</i> |
|-----------------|----------------|------------------|--------------------|---------------|---------------|----------------------|---------------|---------------|
| 0x33 | 0x11 | 0x81 | 0x48 | 0x03 | 0xFF | 0x17 | 0x60 | 0xDF |

Tabla 3.21. Configuración del umbral bajo con el valor 0x0080h.

| <i>Cabecera</i> | <i>Sistema</i> | <i>Parámetro</i> | <i>Umbral Bajo</i> | <i>DATA H</i> | <i>DATA L</i> | <i>Fin de Bloque</i> | <i>CRC16H</i> | <i>CRC16L</i> |
|-----------------|----------------|------------------|--------------------|---------------|---------------|----------------------|---------------|---------------|
| 0x33 | 0x11 | 0x81 | 0x4C | 0x00 | 0x80 | 0x17 | 0x28 | 0xCB |

- *Cabecera*. *Byte* cuyo valor en hexadecimal es 0x33, indica una petición de configuración.
- *Sistema*. En este *byte* indicará el tipo de sistema que puede ser seleccionado para configurar.
- *Parámetro*. Indica que parámetro será modificado para el umbral.
- *Umbral*: Puede escogerse el umbral alto con 0x48h o umbral bajo con 0x4Ch.
- *Data*. El valor numérico para configurar el umbral, está compuesto por 2 *bytes* con un valor máximo de 65535. El valor se manda sin punto decimal.
- *Fin*. *Byte* que indica el fin de bloque transmitido.
- *Comprobación de redundancia cíclica*. 2 *bytes* que son el resultado del CRC.

No todos los parámetros son configurables, en este caso para los parámetros de HVPS no se configuran umbrales altos o bajos, sino voltajes de funcionamiento para cada salida de las 9 disponibles. En este caso no importa qué valor reciba de alto o bajo de umbral será aceptado como un cambio de voltaje válido. Los parámetros configurables son 54 por cada umbral y se muestran en la tabla 3.22.

Tabla 3.22. Parámetros de configuración del *Housekeeping*.

| Parámetro | Valor Bajo | Valor Alto | HK | LVPS1-DP | LVPS2-DP | LVPS-HK | LVPS-PDM | CCB | GPS | CLKB | PDM | HVPS | LENSES |
|-----------|------------|------------|----------|----------|----------|----------|----------|--------|--------|--------|---------|-------|--------|
| | | | 0x11 | 0x22 | 0x33 | 0x44 | 0x55 | 0x66 | 0x77 | 0x88 | 0x99 | 0xAA | 0xBB |
| Voltaje | 61-69 | 0x48 | - | V 5 CCB | V 12 CPU | V 12 HK | V 5 PDM | V 3.3 | - | Vc Aux | V Asic1 | DAC 1 | - |
| | | | - | V 5 CLKB | V 5 DST | V 3.3 HK | V 5 EC | V 2.5 | - | Vc Int | V Asic2 | DAC 2 | - |
| | | | - | V 5 GPS | - | - | - | V 1.2 | - | - | V Asic3 | DAC 3 | - |
| | | | - | - | - | - | - | V 5 | - | - | V fpga1 | DAC 4 | - |
| | | | - | - | - | - | - | v 1.8 | - | - | V fpga2 | DAC 5 | - |
| | | | - | - | - | - | - | - | - | - | - | DAC 6 | - |
| | | | - | - | - | - | - | - | - | - | - | DAC 7 | - |
| | | | - | - | - | - | - | - | - | - | - | DAC 8 | - |
| | | | - | - | - | - | - | - | - | - | - | DAC 9 | - |
| Corriente | 0x4C | 0x48 | - | I 5 CCB | I 12 CPU | I 12 HK | I 5 PDM | I 3.3 | - | - | - | - | - |
| | | | - | I 5 CLKB | I 5 DST | I 3.3 HK | I 5 EC | I 2.5 | - | - | - | - | |
| | | | - | I 5 GPS | - | - | - | I 1.2 | - | - | - | - | |
| | | | - | - | - | - | - | - | - | - | - | - | |
| | | | - | - | - | - | - | - | - | - | - | - | |
| Temp | 81-89 | 0x48 | Temp 1 | - | - | - | - | T max | Temp 1 | Temp 1 | Temp 1 | - | Temp 1 |
| | | | Temp 2 | - | - | - | - | T fpga | ADC | - | - | - | Temp 2 |
| | | | Temp 3 | - | - | - | - | - | - | - | - | - | Temp 3 |
| | | | Temp 4 | - | - | - | - | - | - | - | - | - | Temp 4 |
| | | | Temp 5 | - | - | - | - | - | - | - | - | - | Temp 5 |
| | | | Temp SPI | - | - | - | - | - | - | - | - | - | Temp 6 |
| | | | - | - | - | - | - | - | - | - | - | - | Temp 7 |
| | | | - | - | - | - | - | - | - | - | - | - | Temp 8 |
| | | | - | - | - | - | - | - | - | - | - | - | Temp 9 |
| | 54 | 54 | 6 | 6 | 4 | 4 | 4 | 10 | 2 | 3 | 6 | 0 | 9 |
| Total | 108 | | | | | | | | | | | | |

Manejo de alarmas de baja prioridad

Las alarmas de baja prioridad se definen como todas las que son detectadas al momento de hacer la lectura de algún parámetro y sobrepasar un umbral bajo o alto según sea el caso. Son llamadas de baja prioridad porque solamente registran eventos pero no provocarán una acción sobre el subsistema HK afectado, como consecuencia de ser detectadas.

Las alarmas de baja prioridad se almacenan por sistema y son enmascaradas en cuatro *bytes*, donde dos son destinados a alarmas de umbral alto y los otros dos para alarmas de umbral bajo. Así, cada parámetro es enmascarado por un *bit* dependiendo su disponibilidad a ser leído y comparado en cada sistema. Al envío de alarmas de baja prioridad se le ha asignado la

cabecera 0x35, para indicarle a la CPU que se le está enviando una trama conteniendo esta información. La trama tiene longitud definida por 9 *bytes*, la cual se muestra en la tabla 3.23:

Tabla 3.23. Trama para el envío de alarmas a CPU.

| <i>Cabecera</i> | <i>Sistema</i> | <i>Umbral Alto H</i> | <i>Umbral Alto L</i> | <i>Umbral Bajo H</i> | <i>Umbral Bajo L</i> | <i>Fin de Bloque</i> | <i>CRC16H</i> | <i>CRC16L</i> |
|-----------------|----------------|----------------------|----------------------|----------------------|----------------------|----------------------|---------------|---------------|
| 0x34 | 0x11 | 0x00 | 0x00 | 0x00 | 0x01 | 0x17 | 0xEA | 0xA8 |

- *Cabecera*. *Byte* cuyo valor en hexadecimal es 0x34, indica que se envían alarmas de baja prioridad.
- *Sistema*. Este *byte* indicará el sistema que ha generado la alarma. Cambia, dependiendo de si se trata de lectura o configuración. En las siguientes tablas se observa el valor asignado por sistema.
- *Umbral Alto*: Lo conforman 2 *bytes* conteniendo *bits* de enmascaramiento si el parámetro del sistema ha pasado un nivel de umbral alto. En la tabla 3.24 se puede observar los *bits* que enmascara cada sistema.
- *Umbral Bajo*: Lo conforman 2 *bytes* conteniendo *bits* de enmascaramiento si el parámetro del sistema ha bajado de un nivel de umbral bajo. En la tabla 3.24 se puede observar los *bits* que enmascara cada sistema.
- *Fin*. *Byte* que indica el fin de bloque transmitido.
- *Comprobación de redundancia cíclica*. 2 *bytes* que son el resultado del CRC.

En el caso de las alarmas para HVPS no se utilizan umbrales altos o bajos, sino 2 *bytes* para las condiciones de estado de los expansores de puertos que se controlan y otros dos *bytes* para el estado de encendido o apagado de cada uno.

Tabla 3.24. Enmascaramiento de alarmas de baja prioridad.

| | | Byte 2 | | | | | | | | Byte 1 | | | | | | | |
|---------------------|------------------|--------|----|----|----|----|----|--------|--------|--------|--------|---------|----------|----------|----------|----------|----------|
| | | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| HK 0x11 | Umbral Alto | | | | | | | | | | | T digit | Temp 5 | Temp 4 | Temp 3 | Temp 2 | Temp 1 |
| | Umbral Bajo | | | | | | | | | | | | | | | | |
| LVPS Voltaje 0x22 | Umbral Alto | | | | | | | | EC | PDM | HK 33 | HK 12 | DST | CPU | GPS | CLKB | CCB |
| | Umbral Bajo | | | | | | | | | | | | | | | | |
| LVPS Corriente 0x33 | Umbral Alto | | | | | | | | EC | PDM | HK 33 | HK 12 | DST | CPU | GPS | CLKB | CCB |
| | Umbral Bajo | | | | | | | | | | | | | | | | |
| CCB 0x66 | Umbral Alto | | | | | | | T fpga | T max | i1.2 | i2.5 | i3.3 | v1.8 | v5 | v1.2 | v2.5 | v3.3 |
| | Umbral Bajo | | | | | | | | | | | | | | | | |
| GPS 0x77 | Umbral Alto | | | | | | | | | | | | | | | ADC | Temp 1 |
| | Umbral Bajo | | | | | | | | | | | | | | | | |
| CLKB 0x88 | Umbral Alto | | | | | | | | | | | | | | Temp 1 | VcInt | VcAux |
| | Umbral Bajo | | | | | | | | | | | | | | | | |
| PDM 0x99 | Umbral Alto | | | | | | | | | | Temp2 | Temp1 | V Asic 5 | V Asic 4 | V Asic 3 | V Asic 2 | V Asic 1 |
| | Umbral Bajo | | | | | | | | | | | | | | | | |
| HVPS 0xAA | Condición estado | | | | | | | | STA9 | STA8 | STA7 | STA6 | STA5 | STA4 | STA3 | STA2 | STA1 |
| | ON/OFF Alarma | | | | | | | | ON9 | ON8 | ON7 | ON6 | ON5 | ON4 | ON3 | ON2 | ON1 |
| LENSES 0xBB | Umbral Alto | | | | | | | | Temp 9 | Temp 8 | Temp 7 | Temp 6 | Temp 5 | Temp 4 | Temp 3 | Temp 2 | Temp 1 |
| | Umbral Bajo | | | | | | | | | | | | | | | | |

Envío de alarmas de alta prioridad

Las alarmas de alta prioridad se definen como todas las que son detectadas al momento ejecutar una acción y no recibir el resultado esperado. También se nombran de alta prioridad a las que se registran en eventos de comunicación no exitosa, en los buses de comunicación serie. Estas alarmas provocarán una acción sobre el subsistema afectado por parte de la CPU. Cuando el *Housekeeping* envía como cabecera el hexadecimal conteniendo el valor 0x35h, la CPU deberá interpretar esta trama como una alarma de alta prioridad. Se tiene una longitud definida de la trama que consiste en 5 bytes. Su estructura se muestra en la tabla 3.25:

Tabla 3.25. Trama ejemplo de alarma de alta prioridad.

| Cabecera | Código de alarma | Fin de Bloque | CRC16H | CRC16L |
|----------|------------------|---------------|--------|--------|
| 0x35 | A definir | 0x17 | --- | --- |

Los códigos de alarma son muy específicos de cada subsistema, por lo que se utilizará un código hexadecimal para cada caso. Esto da la posibilidad de detectar por el momento hasta 255 alarmas de alta prioridad.

Una vez que se desarrolló un protocolo más completo, se tuvo la necesidad de implementar mejoras en el cálculo del CRC, con el fin de eliminar la restricción del cálculo de CRC para 2 *bytes* que se tenía implementado.

Para hacer el CRC de forma más óptima, se ha hecho el cálculo a través de tablas con valores pre-calculados de CRC. Esta tabla se obtuvo realizando un barrido de todos los posibles resultados de la combinación de 0 a 255, el rango comprendido en un *byte* para agilizar el cálculo. En el apéndice B se muestra la tabla con los valores generados. La ventaja principal de tener valores calculados de CRC se ve reflejada principalmente en reducir los tiempos de procesamiento, debido a que la obtención del CRC es más rápida pero se logra a costa de ocupar más memoria *flash* en el microcontrolador.

Al contar con un protocolo de comunicación más completo, se tuvo la necesidad de modificar el banco de pruebas para agregar más funcionalidades y adaptarlo al nuevo protocolo.

Mejoras al banco de pruebas de CPU / SIREN

De acuerdo a la propuesta del protocolo de comunicación final para el *Housekeeping* y la CPU / SIREN, se vio la necesidad de implementar en la terminal del banco de pruebas una interfaz de usuario interactiva, para poder enviar todos los comandos, de acción, lectura y configuración de umbrales dentro del *Housekeeping*. Se trabajó en una interfaz de usuario que funciona a través de un menú interactivo, que manda instrucciones desde la computadora, al banco de pruebas para enviar las tramas de comunicación con el *Housekeeping*. Esta interfaz está compuesta por un menú con una serie de opciones que conforme el usuario va seleccionando, permiten consumir el envío de una trama codificada con el protocolo propuesto. Debido a la gran cantidad de comandos elegibles por el usuario, se ha creado este menú que navega entre sus opciones para seleccionar el tipo de acción que necesita ser enviada como petición. En la figura 3.122 se muestra una captura de pantalla de la interfaz, donde se envía como ejemplo, una trama de configuración del parámetro “termistor 2” del *Housekeeping*, con un valor de umbral alto de 5200 en decimal, que para el caso de temperatura, corresponde a 52.000° C. Los pasos que se siguieron fueron los siguientes:

1. Selección de encabezado, entre las opciones se encuentran: función, lectura, configuración.
2. Seleccionar la dirección de destino, en este caso será un subsistema.
3. Seleccionar el parámetro, este menú cambia dependiendo del subsistema elegido en la opción anterior.
4. Seleccionar un umbral.

5. Seleccionar un signo “-” en caso de requerir un valor negativo, en caso contrario seleccionar simplemente el signo de “+”.
6. Seleccionar una carácter que multiplicará por 100 el valor de su correspondiente en ASCII, en el ejemplo se utilizó el 52 que se multiplicó por 100 dando como resultado 5200.

Al finalizar la configuración la trama es enviada al *Housekeeping*.

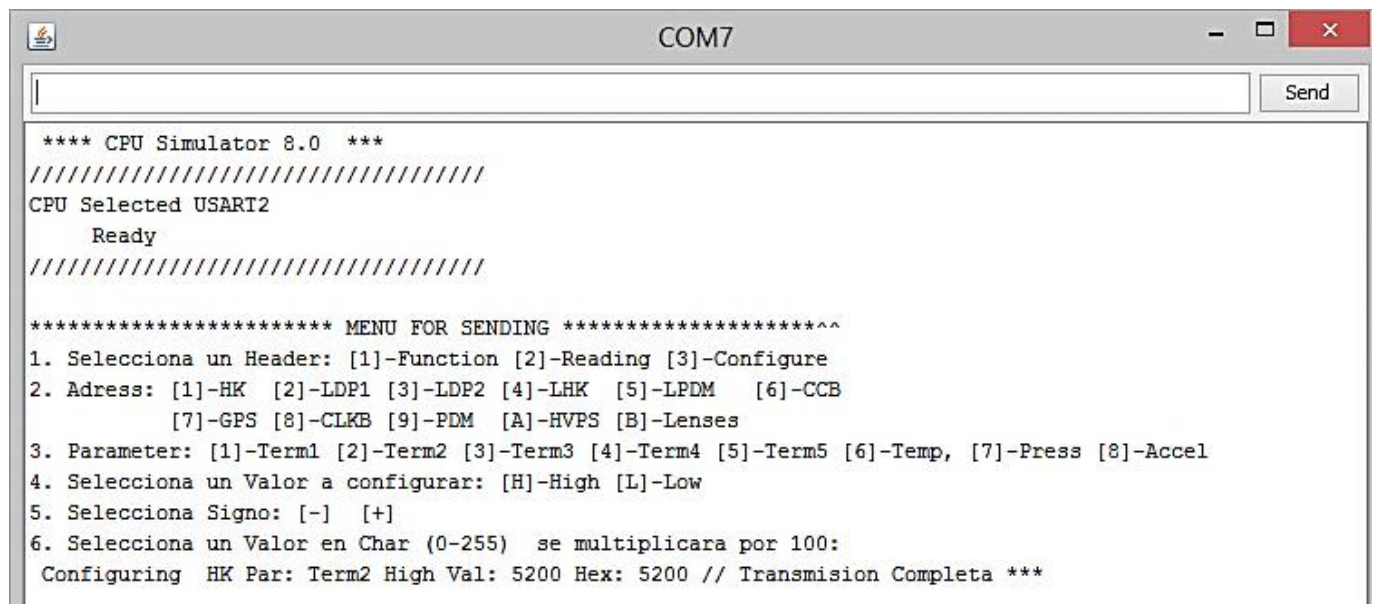


Figura 3.122. Captura de la interfaz de comunicación.

La importancia de la implementación de esta mejora en el banco de pruebas de CPU/SIREN, permitió interactuar de forma fácil con el *Housekeeping*, de forma que se pudieran probar todos las interfaces que tienen conexión con otros subsistemas de forma fácil y de forma individual, permitiendo avances en la forma de operación del *Housekeeping*.

En el diagrama de árbol de la figura 3.123 se representa la forma en que se puede navegar por sus opciones, donde las opciones elegibles por el usuario se encuentran sombreadas, las demás son previamente determinadas.

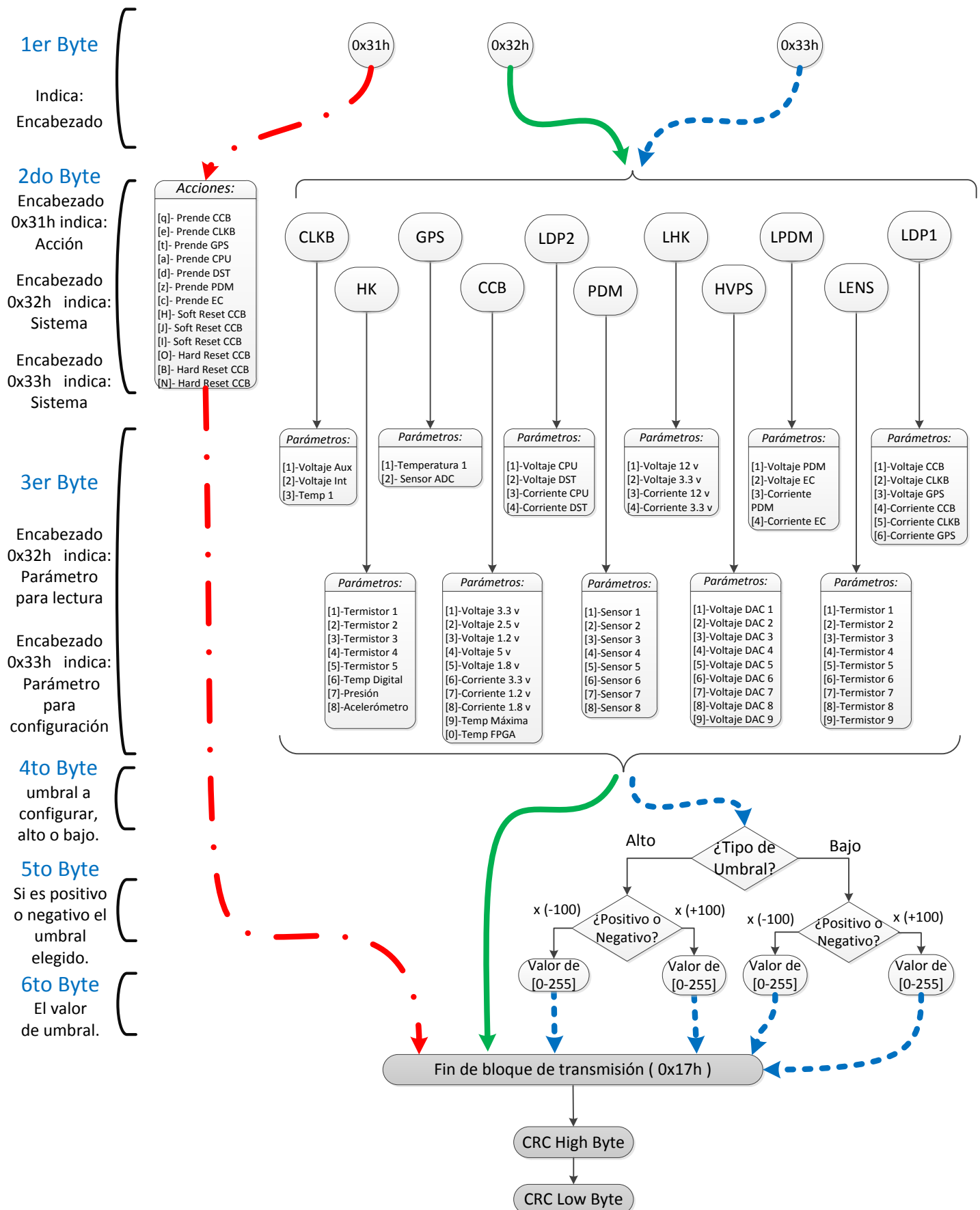


Figura 3.123. Diagrama de flujo de la interfaz de software.

Depurador y control por medio de SIREN

Esta parte del banco de pruebas no estuvo habilitada hasta que el *Housekeeping* fue capaz de recibir comandos por las 2 USART del microcontrolador, primero se enfocó el desarrollo en lograr la forma óptima de funcionamiento usando únicamente la conexión con la CPU, posteriormente se habilitó la USART que corresponde a la conexión con SIREN. Además de realizar un control similar al de la CPU, una característica esencial entre la interacción del *Housekeeping* y SIREN, se encuentra en el intercambio de información de sensores térmicos antes del encendido de la CPU, y posteriormente a su encendido, además del intercambio de información del funcionamiento de todos los parámetros leídos por el *Housekeeping*. Recordando que SIREN es el sistema encargado de la telemetría, todos los datos enviados a este sistema serán transmitidos a la estación en tierra, encargada de revisar el estado del vuelo. Estos datos serán analizados en tiempo real y permitirá tomar decisiones a bordo del globo. De esta forma se implementó en el *Housekeeping* un sistema de depuración en donde se envía en caracteres de tipo ASCII, todos los parámetros, tareas y errores que se presentan. La función de depuración realiza lo siguiente:

- Mostrar el estado del *Housekeeping*, con la posibilidad de poder depurar cada interfaz y encontrar errores en *firmware*, y finalmente ayudar a resolver un mal funcionamiento del *Housekeeping*.
- Simular el comportamiento de SIREN para la detección de parámetros antes de encender la CPU, y posteriormente a su encendido, la posibilidad de comandar el *Housekeeping* de forma redundante a la CPU.

Este sistema fue de gran importancia para entregar un *firmware* más cercano a la versión de vuelo del subsistema. En la figura 3.124 se puede ver una captura de pantalla de una terminal receptora de caracteres ASCII, donde se pueden observar los datos de depuración del *Housekeeping*. La opción para seleccionar la USART de control entre CPU y SIREN, se realiza en el banco de pruebas por medio de un selector o *Jumper* puesto en la tarjeta de circuito impreso.

```

COM8
Enviar

LA SD VA AQUI
Termistores LENSES: 27.19, -59.28, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,

***HouseKeeping***
  Temperature Board: 33.59
  HK internal thermistor: 0.00
  CPU internal thermistor: 0.00          0.00          0.00          0.00

** LVPS MONITORING **
  LVPS HK:      10.088v 3118mA  13.777v 659mA
  LVPS-DP1:    0.000v 112mA   0.000v 537mA  0.000v 542mA
  LVPS-DP2:    15.081v 586mA   14.467v 1534mA
  LVPS-PDM:    15.067v 4506mA  0.000v 4721mA

***CLKB values***
  Temperature: -49.450  Vcint: 0.750  Vcaux: -3.290
DONE Signal detected from CLKB
No ALARM from CLKB

***CCB values ***  Test CCB:  b0b   Volt 3.3v:   3.299  Curr 3.3v:   0.090
Volt 2.5v:  2.500  Curr 2.5v:  0.900  Volt 1.2v:   1.200  Curr 1.2v:   0.239
Volt 5v:    5.000  Volt 1.8v:  1.799  Board Temp:  70.000  FPGA Temp:   60.000
# Error: 0    # Times: 0
DONE Signal detected from CCB
No ALARM from CCB

***GPS values***
  Temperature: 0.15
  ADC (Dec Value): 0

LA SD VA AQUI
Termistores LENSES: 27.18, -59.28, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,

***HouseKeeping***
  Temperature Board: 33.52
  H
  
```

Desplazamiento automático No hay fin de línea 38400 baudio

Figura 3.124. Interfaz de depuración del *Housekeeping*.

Diagrama de flujo general

Dentro de la filosofía de operación del *Housekeeping* se determinó que se debe seguir un protocolo de operación. Este consiste en el diagrama de flujo de la figura 3.125.

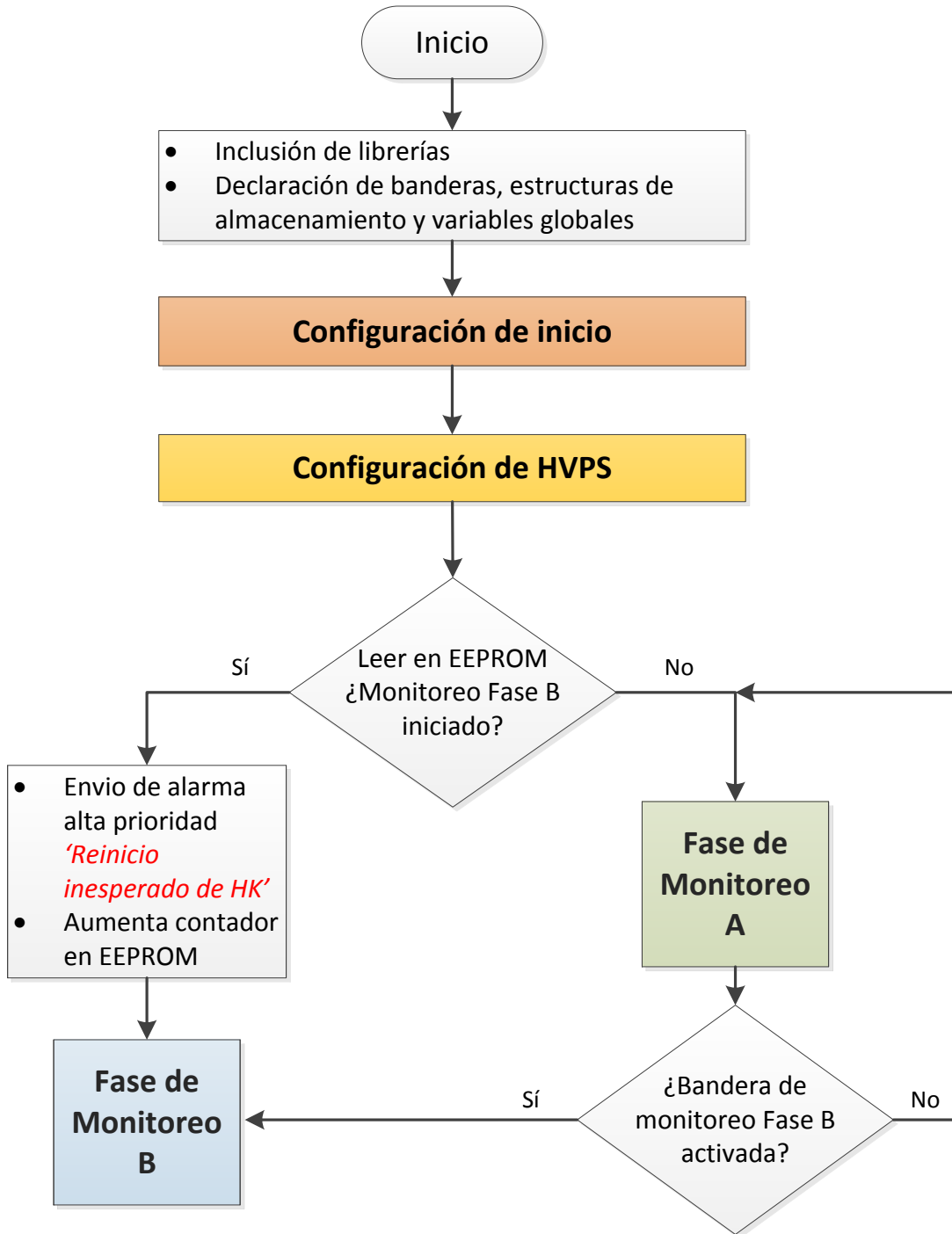


Figura 3.125. Diagrama de flujo de la operación del *Housekeeping*.

Este diagrama consta de 4 bloques importantes que han sido resumidos y se explican a continuación. Los primeros 2 bloques son de configuración y los 2 siguientes engloban secuencias de monitoreo que llamamos fase A y fase B.

La configuración de inicio, figura 3.126, llama todas las librerías encargadas de configurar el microcontrolador del *Housekeeping*. Esto incluye la declaración de puertos de entrada y salida, la habilitación de resistencias *pull-up* para los puertos que lo necesiten, como consecuencia los circuitos integrados quedan en espera de recibir señales de control para ser utilizados. Continúa la configuración haciendo uso de las librerías para configurar los módulos USART 0, 2 y 3. Configura la comunicación SPI e inicia los módulos para retardos, la adquisición analógica y la habilitación de interrupciones. Posteriormente se carga en la memoria *flash* todos los valores almacenados en EEPROM de los umbrales, para ser utilizados a mayor velocidad.

Configuración de inicio:

- Configuración de puertos E/S.
- Inicialización de los umbrales de los parámetros desde la EEPROM.
- Activación de interrupciones.
- Configuración de módulos (SPI, RS422).
- Inicialización de contadores para retardos.

Figura 3.126. Función de configuración de inicio.

El siguiente bloque es importante, ya que incluye la configuración del sistema HVPS por el protocolo SPI, estas configuraciones se muestran en la tabla 3.127.

Configuración de HVPS:

- Inicialización de expansores de puertos.
- Envío de pulsos de encendido.
- Configuración de interrupciones en expansores.
- Habilitación de la línea de interrupción de HVPS.
- Inicialización de convertidores digitales analógicos.

Figura 3.127. Función de configuración de HVPS.

Siguiendo el diagrama, se pregunta por la bandera EEPROM para iniciar el monitoreo fase B, en caso de que esta bandera no esté activada se procederá a continuar con el monitoreo fase A. En caso de que ya esté activada indicará un reinicio no inesperado y se mandará una alarma a la CPU.

El monitoreo fase A se encarga principalmente de reportar los datos de temperatura a SIREN, con el fin de que este sistema decida cuáles serán las condiciones de temperatura apropiadas para el encendido de la CPU y DST. Estos últimos subsistemas no deben ser encendidos en condiciones de temperatura menor a los cero grados centígrados. Cuando SIREN encuentre las condiciones apropiadas, mandará comandos para encender la CPU y de ahí espera un tiempo en lo que la CPU toma el control para encender los demás subsistemas. Esto será realizado en el monitoreo fase B. Si llegara a ocurrir un reinicio en el *Housekeeping*, una bandera almacenada en la EEPROM evitará que se vuelva a iniciar el monitoreo fase A, enviando una alarma a la CPU para informar de un reinicio no esperado y continuará al ciclo de monitoreo fase B. El diagrama de flujo del monitoreo fase A se muestra en la figura 3.128.

El monitoreo fase B es el ciclo en donde se espera que el *Housekeeping* esté operando. En este ciclo, el *Housekeeping* se encargará de esperar instrucciones para encender los sistemas restantes, y con esto, iniciar la lectura de parámetros por sistema. Esto lo hace preguntando por banderas de encendido, las cuales se activan únicamente después de contar con el cierre de un relevador que indica el encendido de un subsistema. Posteriormente se pregunta por el número de ciclos que realiza, esto sirve principalmente para controlar los tiempos en que se envía la información a SIREN y CPU. SIREN también servirá para depurar el *firmware del Housekeeping*, éste recibe información cada 10 ciclos, mientras que la CPU, que sólo almacenará parámetros cierto tiempo, recibe esta información cada 100 ciclos. El monitoreo fase B se muestra en la figura 3.129.

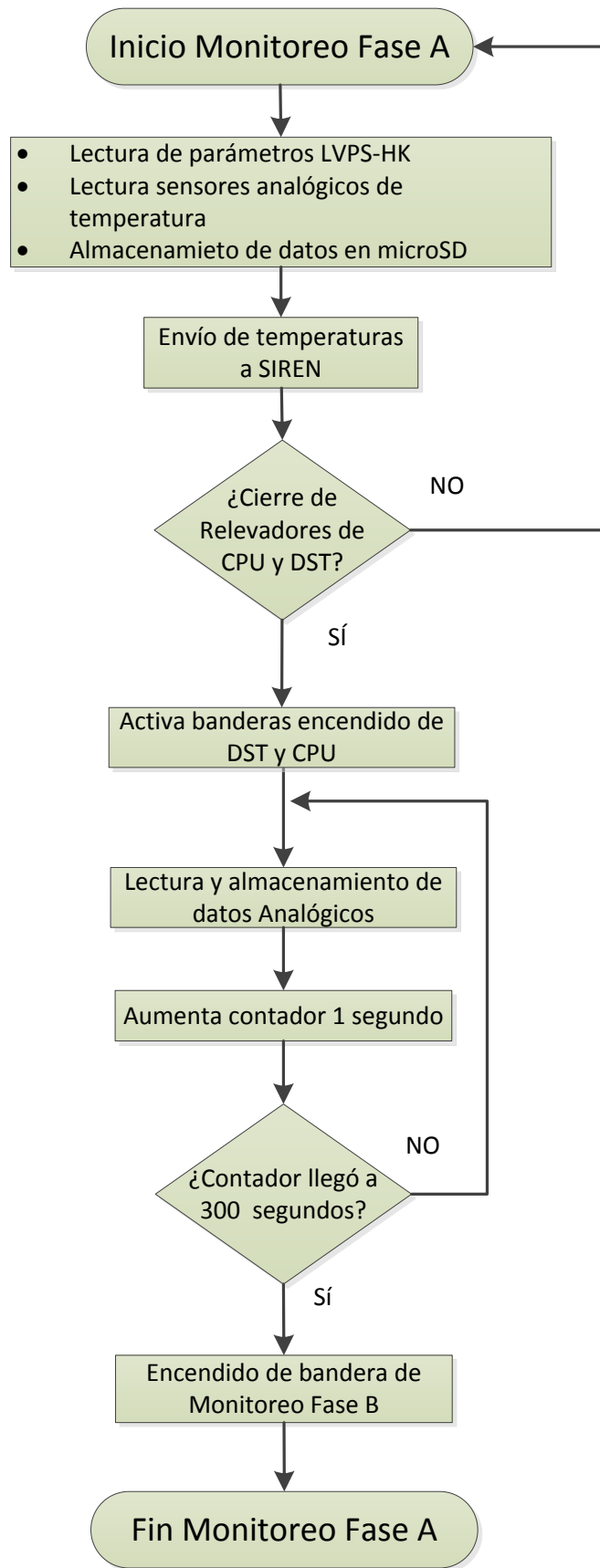


Figura 3.128. Monitoreo fase A.

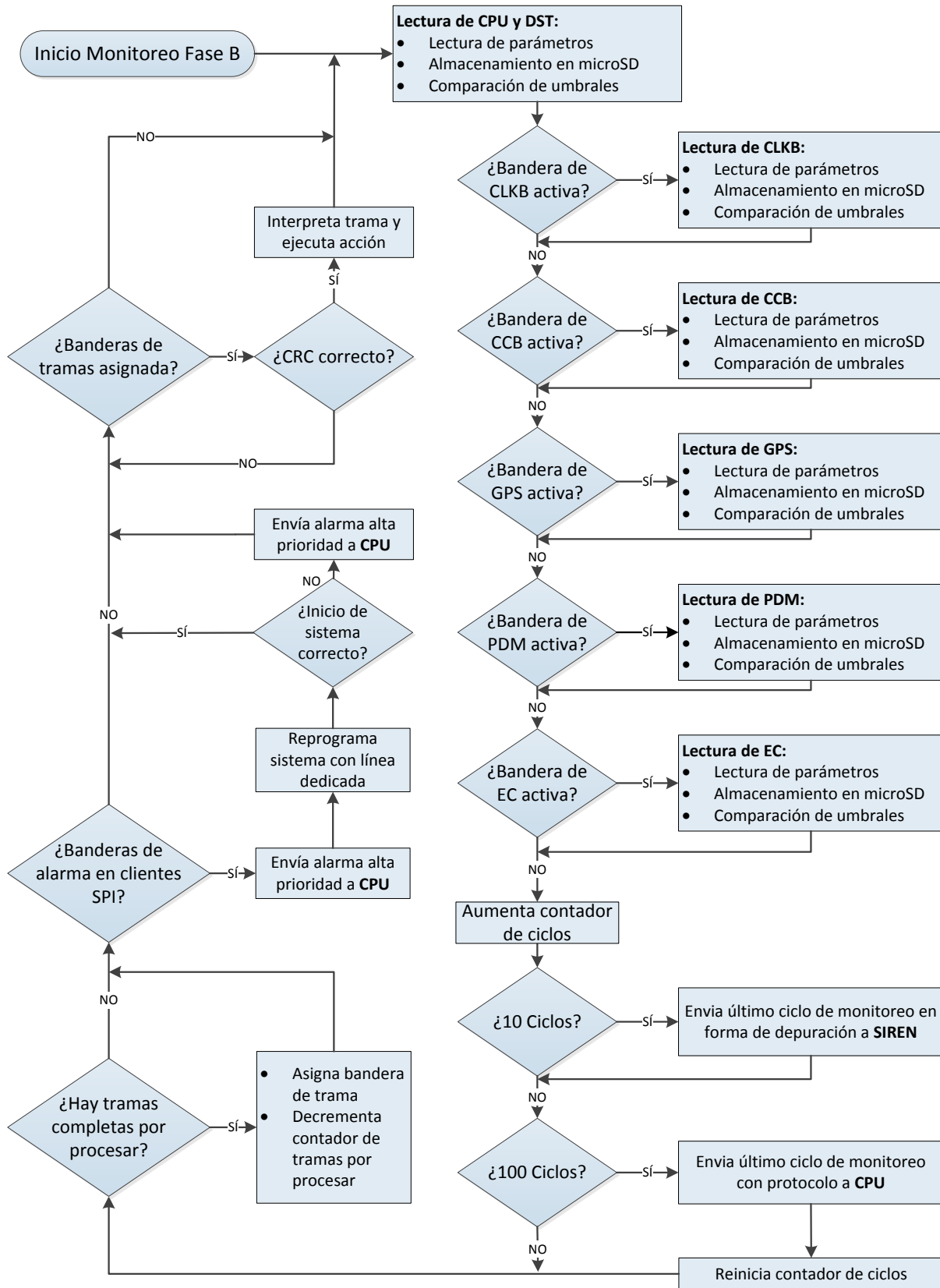


Figura 3.129. Monitoreo fase B.

En el ciclo de monitoreo fase B se puede observar que se pregunta por las tramas completas por procesar, ya que se implementó un *buffer* capaz de recibir varias tramas. Debido a que las tramas no tienen longitud fija, el *buffer* se comporta de manera dinámica y lo hace por una implementación de una máquina de estados en la interrupción de la USART 2 y 3 que corresponden a la CPU y SIREN. Esta máquina de estados se muestra en la figura 3.130.

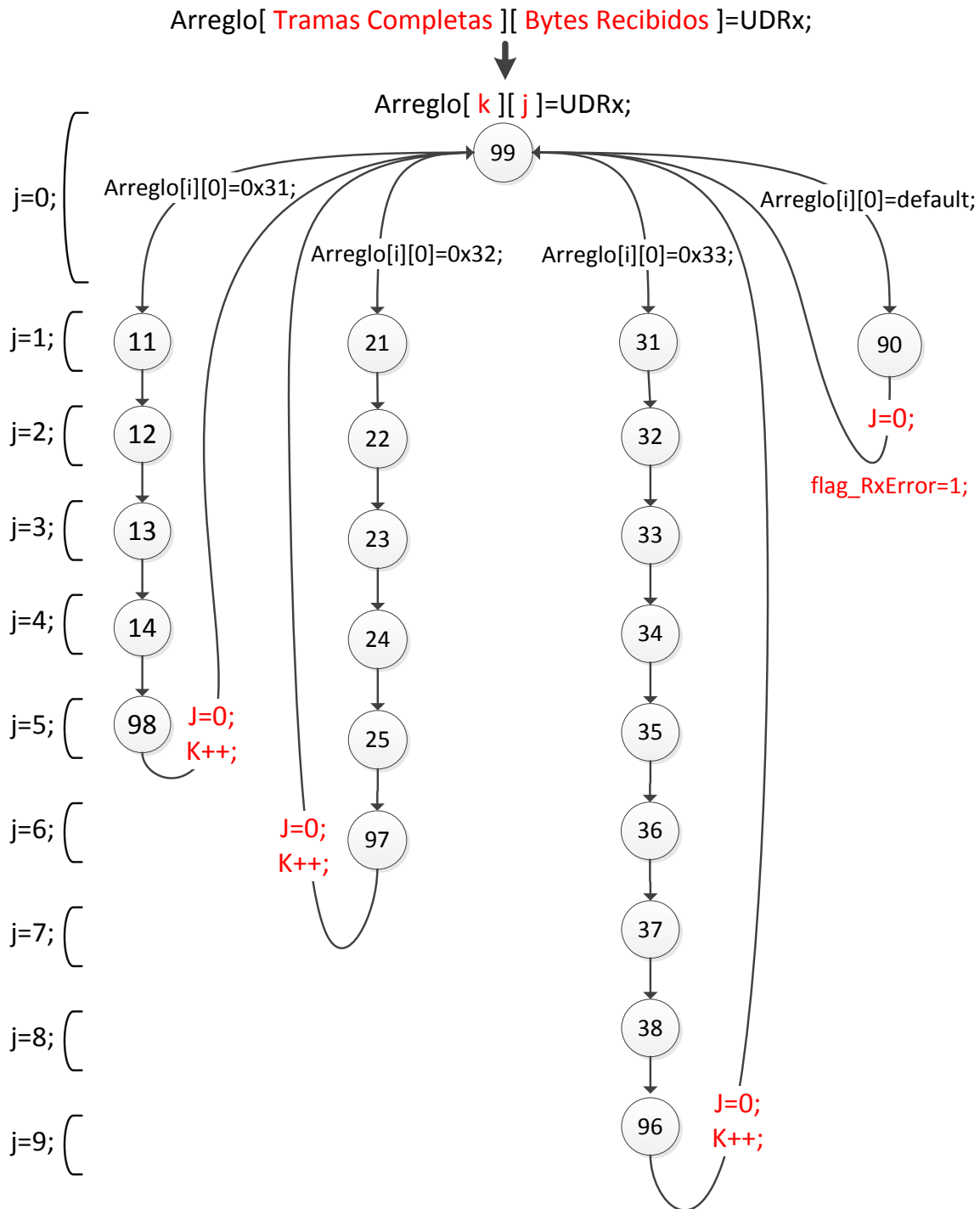


Figura 3.130. Control del buffer dinámico para recibir tramas.

En el ciclo B se pregunta si hay tramas para procesar, en caso de que hubiera tramas se les asigna una bandera dependiendo el tipo de encabezado y posteriormente se ejecuta la instrucción. También se preguntan por las alarmas de subsistemas SPI y en caso de presentarse, se enviará una alarma de alta prioridad a la CPU dependiendo el caso de ésta.

3.5.3. Ensamble del modelo de vuelo

Montaje en gabinete

En esta etapa se usó un gabinete o *case* de tipo comercial para montar el *Housekeeping*. De esta manera se asegura un correcto montaje dentro del *subrack* del DP al usar un modelo que cumple con las medidas estándar, además de que con el uso de este gabinete se reduce el peso del *Housekeeping*, en comparación con el anterior gabinete que se fabricó en México. En la figura 3.131 se muestra el gabinete a utilizar.

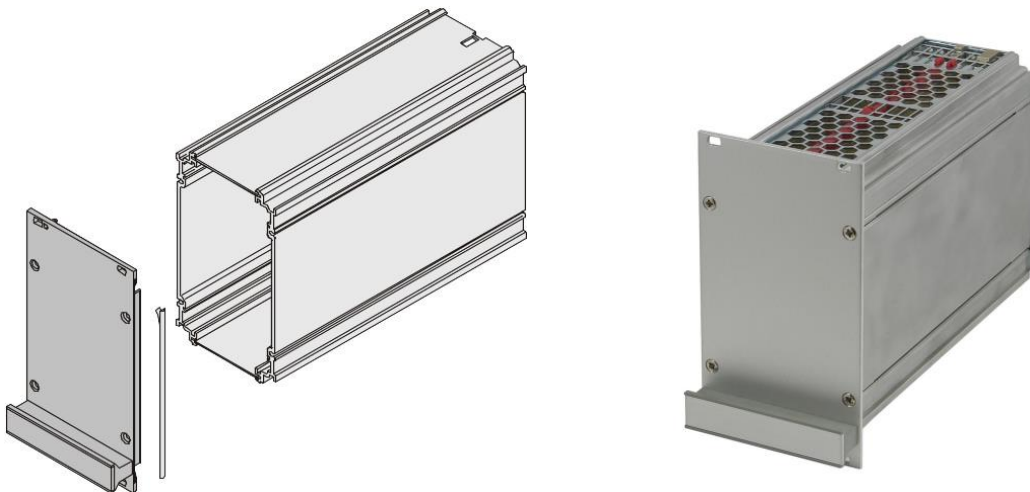


Figura 3.131. Gabinete para montar el *Housekeeping*.

Fue necesario hacer adecuaciones en este nuevo gabinete para montar el *Housekeeping*:

- Se hicieron las perforaciones necesarias al panel frontal para colocar los conectores de tipo D-Sub. Se generó un modelo CAD del panel frontal con las huellas de los conectores.
- Fue necesario modificar la base del gabinete para poder sujetar la estructura del *Housekeeping*. Se realizaron las perforaciones necesarias y se agregaron soleras de aluminio a la placa para dar altura a la estructura montada.

En la figura 3.132 se muestra una imagen del panel frontal con las perforaciones necesarias para colocar los conectores.

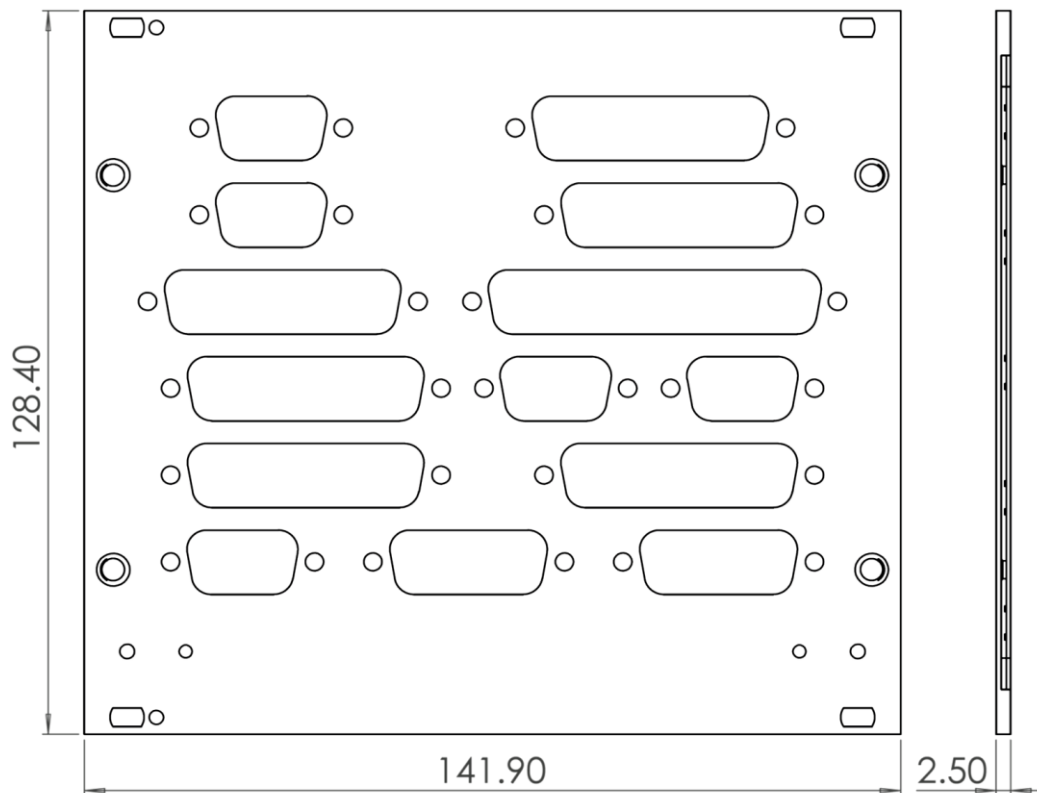


Figura 3.132. Panel frontal del gabinete del *Housekeeping*.

En la figura 3.133 se muestra una imagen del *Housekeeping* sobre la placa inferior del gabinete.

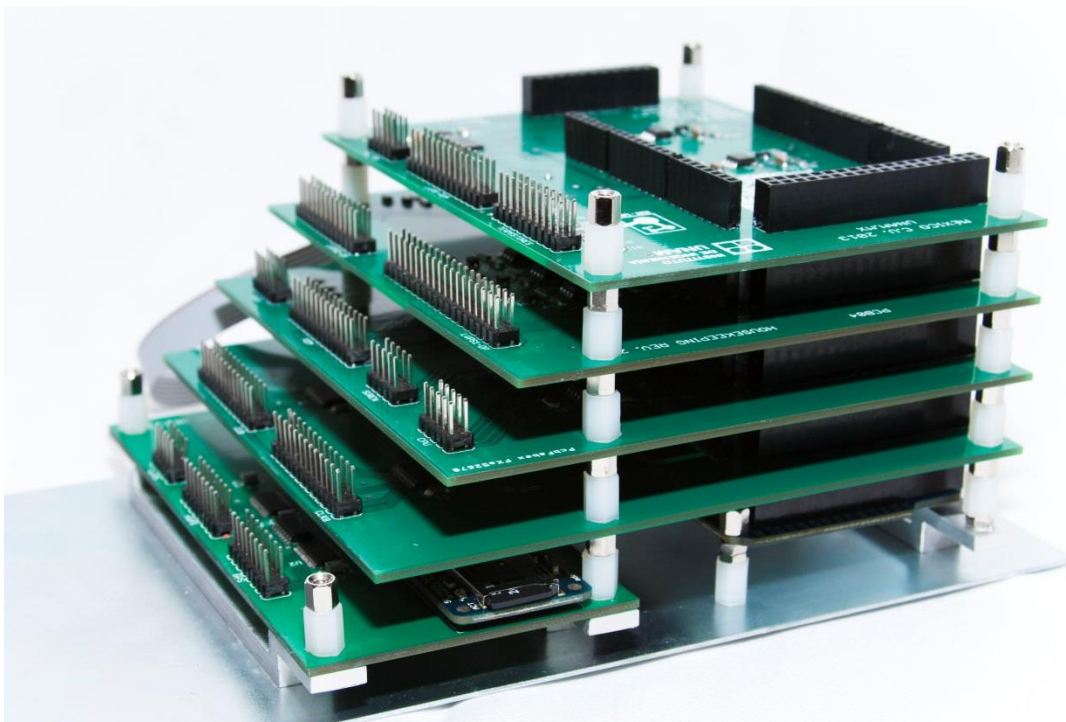


Figura 3.133. *Housekeeping* sobre la placa inferior del gabinete.

En las figuras 3.134 y 3.135 se muestra el *Housekeeping* dentro del gabinete sin cerrar por completo y el gabinete cerrado completamente, respectivamente.



Figura 3.134. *Housekeeping* dentro del gabinete.



Figura 3.135. *Housekeeping* dentro del gabinete.

4. Evaluación y puesta a punto

En este capítulo se abordarán las pruebas que se hicieron para validar las dos versiones del *Housekeeping*. Para comprobar el correcto funcionamiento de las interfaces externas de la primera versión del *Housekeeping*, se llevaron a cabo pruebas básicas de *hardware* y *software*. En cambio, para la versión de vuelo se vio la necesidad de realizar pruebas de forma más completa, tanto en *software* como en *hardware*, por este motivo se construyeron los bancos de prueba de los subsistemas.

Se realizaron pruebas de vacío y ciclado térmico para las dos versiones del *Housekeeping*, en la cámara de vacío que se encuentra en el Laboratorio de Detectores del ICN.

4.1. Validación del microcontrolador en condiciones de vacío

Una de las pruebas más importantes fue la validación del microcontrolador como componente principal del *Housekeeping*. Para esto se realizaron pruebas dentro de la cámara de vacío para someterlo a condiciones de alto vacío y temperaturas que oscilaron entre $-37\text{ }^{\circ}\text{C}$ y $45\text{ }^{\circ}\text{C}$.

En la prueba realizada se buscó que el microcontrolador tuviera una carga de procesos considerable, además de adicionarle carga en las terminales de salida para aumentar su consumo, con el fin de verificar que operara con normalidad en las condiciones de la prueba.

Las tareas que realizó el microcontrolador fueron las siguientes:

- Adquirir datos de dos termistores, uno de ellos mide la temperatura del microcontrolador de la tarjeta de desarrollo y el otro mide la temperatura ambiente
- Lee un sensor de temperatura por medio del estándar I2C
- Lee el valor de cinco potenciómetros por medio de los canales del ADC
- Manipular varios LEDs con el PWM del microcontrolador
- Controlar un contador en un *display* de 7 segmentos
- Resolver una operación matemática de prueba:

$$y = 99.99 \times \sqrt{x}.$$

Ecuación 11

- Almacenar y leer datos de la memoria EEPROM
- Enviar información a una PC por medio del puerto USB de la tarjeta

En la figura 4.1 se muestra la tarjeta de desarrollo con las interfaces de prueba.

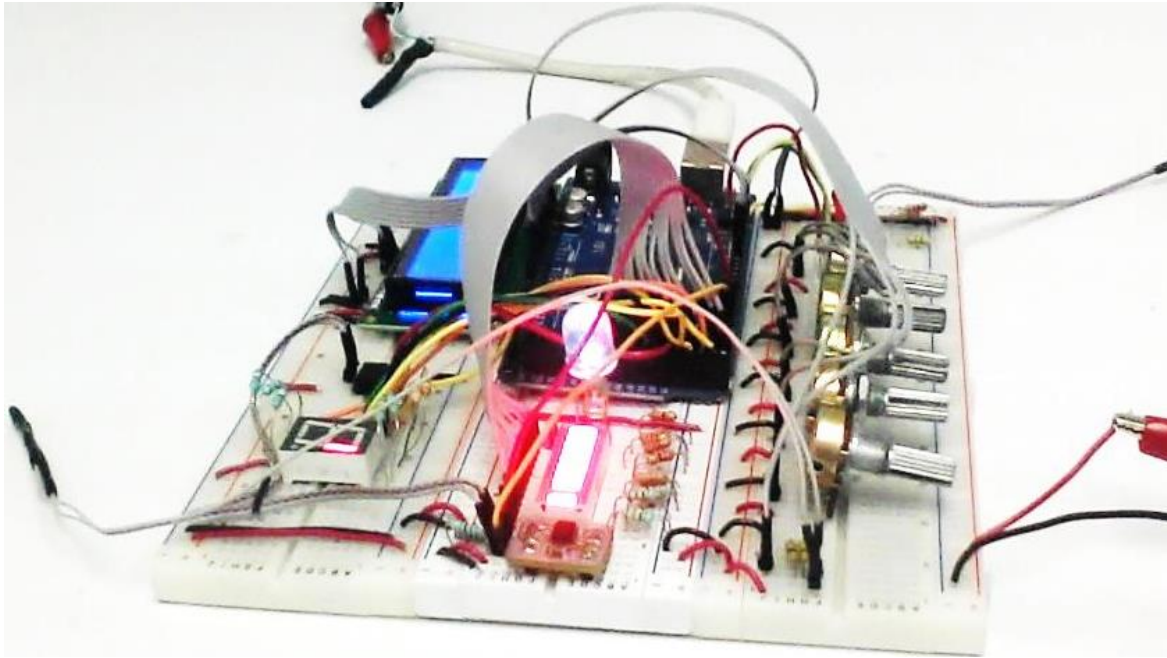


Figura 4.1. Circuito en *protoboard* con los dispositivos de prueba.

Las pruebas de presión y temperatura se llevaron a cabo paralelamente, ya que la cámara de vacío permite variar la temperatura de su plancha. Ésta tiene contacto con cualquier sistema alojado dentro de ella. Así mismo, la presión puede variar una vez cerrada la campana de la cámara y se puede mantener a una presión constante menor a la del ambiente. Se realizaron pruebas con los siguientes parámetros de temperatura y presión:

- Temperatura y presión ambiente, durante 6 horas
- Temperatura de 45 °C con presión de 1.4 mbar a 1.99 mbar, durante 8 horas
- Temperatura de -36 °C con presión de 1.33 mbar, durante 8 horas
- Temperatura de 21 °C con presión de 1.33×10^{-5} mbar, alto vacío, durante 8 horas
- Temperatura de -37 °C con presión de 1.33×10^{-5} mbar, alto vacío, durante 8 horas

En todas las pruebas realizadas, la temperatura de la plancha fue obtenida del sensor interno de la cámara de vacío. En la imagen 4.2 se observa el diagrama del circuito dentro de la cámara de vacío.

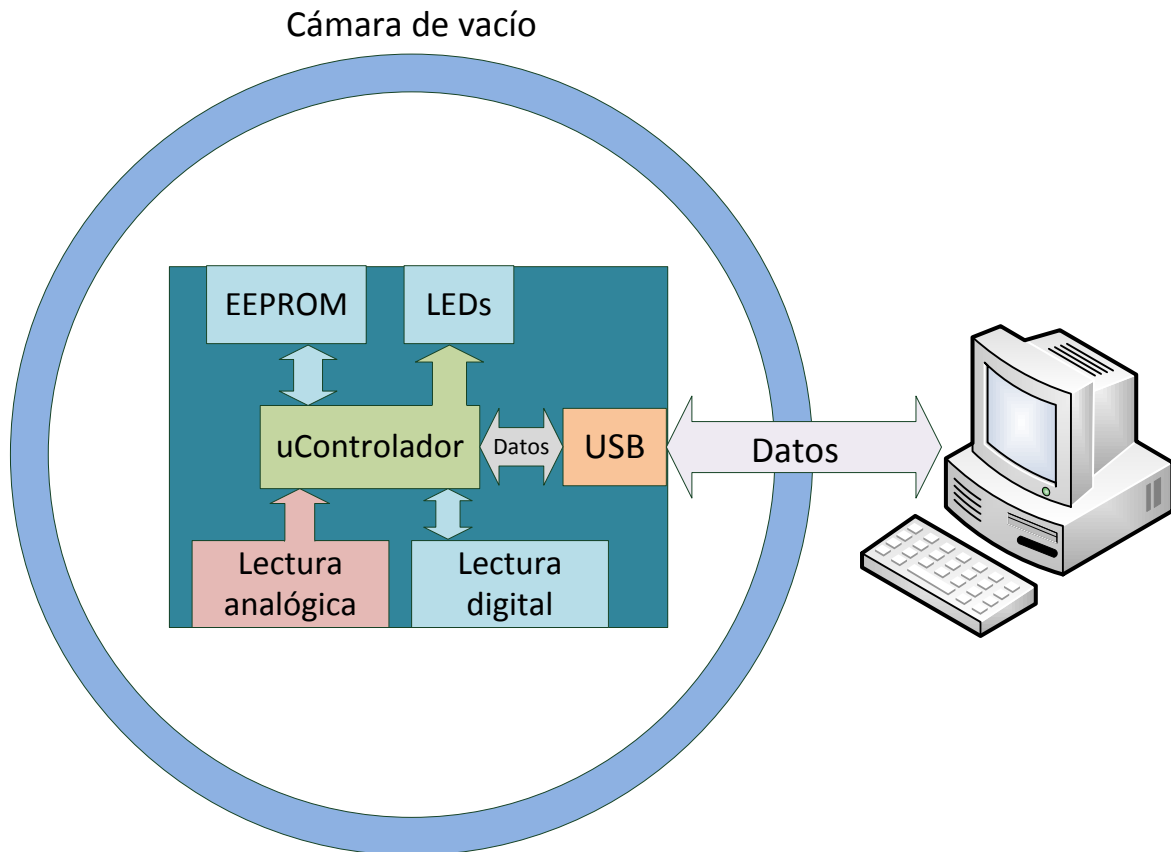


Figura 4.2. Diagrama del circuito dentro de la cámara de vacío.

Las operaciones aritméticas realizadas con el microcontrolador fueron previamente resueltas con ayuda de MATLAB, para comparar resultados. El valor de la variable independiente 'x' de la ecuación 1, fue elegido por el usuario a través de la comunicación por USB. Durante la prueba, se estuvo comprobando que el valor de salida fuera el esperado, con respecto al valor calculado por MATLAB.

La escritura de la memoria EEPROM consistía en almacenar valores constantes en un bloque de memoria EEPROM, esto con el fin de que al ser leídas las localidades de memoria, se pudiera detectar algún error o un cambio de información que podría llegar a presentarse debido a las condiciones a las que estaba sometido el microcontrolador. Durante las pruebas, se verificó que el valor almacenado en la memoria EEPROM se mantuvo constante.

Resultados de la prueba a presión ambiente

En esta prueba se verificó el funcionamiento del microcontrolador antes de cerrar la campana de la cámara de vacío, en la tabla 4.1 se muestran las mediciones. Cabe resaltar que la tarjeta de desarrollo del microcontrolador se alimentó con un voltaje de 6.5 V, debido a que cuenta con un regulador de voltaje interno que proporciona los 5 V que necesita el microcontrolador.

Tabla 4.1. Mediciones realizadas a temperatura y presión ambiente.

| Presión [mbar] | Temperatura μ controlador [°C] | Temperatura de la plancha [°C] | Corriente máxima [A] | Voltaje [V] | Prueba de Memoria EEPROM | Operación matemática |
|----------------|------------------------------------|--------------------------------|----------------------|-------------|--------------------------|----------------------|
| 750 | 35.7 | 24.2 | 0.27 | 6.5 | Aprobada | Aprobada |
| 750 | 36 | 24.1 | 0.27 | 6.5 | Aprobada | Aprobada |
| 750 | 36.1 | 24.4 | 0.27 | 6.5 | Aprobada | Aprobada |

En esta prueba el consumo de potencia del microcontrolador se mantuvo constante. Se puede concluir que el microcontrolador se comportó como se esperaba.

Resultados de la prueba con una presión de 1.43 a 1.995 mbar

En esta prueba la presión dentro de la cámara de vacío varió entre 1.43 mbar y 1.995 mbar. Esto se hizo para simular las condiciones de presión que se presentan a alturas que van de 30 km a 45 km, a las cuales será sometido el instrumento.

Primero se realizó esta prueba ajustando la temperatura de la plancha a aproximadamente 45 °C. En la tabla 4.2 se muestran las mediciones de la prueba.

Tabla 4.2. Mediciones en vacío con temperatura cercana a 45 °C.

| Presión [mbar] | Temperatura μ controlador [°C] | Temperatura de la plancha [°C] | Corriente máxima [A] | Voltaje [V] | Prueba de Memoria EEPROM | Operación matemática |
|----------------|------------------------------------|--------------------------------|----------------------|-------------|--------------------------|----------------------|
| 1.43-1.86 | 57.8 | 44.2 | 0.27 | 6.5 | Aprobada | Aprobada |
| 1.86 | 61.9 | 45.3 | 0.27 | 6.5 | Aprobada | Aprobada |
| 1.995 | 64.5 | 45.7 | 0.27 | 6.5 | Aprobada | Aprobada |

Los resultados fueron los siguientes:

- La temperatura máxima que alcanzó el microcontrolador fue de 64.5 °C, la cual está por debajo de la temperatura máxima de operación que proporciona el fabricante (85 °C)
- La corriente que proporcionaba la fuente se mantuvo constante
- No se presentó ninguna caída en el voltaje de alimentación
- La operación de lectura y escritura en la EEPROM se realizó de forma correcta
- La expresión matemática se resolvió de manera correcta

Al terminar las mediciones de la prueba anterior, se procedió a disminuir la temperatura de la plancha de la cámara a aproximadamente -36 °C. En la tabla 4.3 se muestran las mediciones realizadas.

Tabla 4.3. Mediciones en vacío con temperatura cercana a -36 °C.

| Presión [mbar] | Temperatura μ controlador [°C] | Temperatura de la plancha [°C] | Corriente máxima [A] | Voltaje [V] | Prueba de Memoria EEPROM | Operación matemática |
|----------------|------------------------------------|--------------------------------|----------------------|-------------|--------------------------|----------------------|
| 1.33 | -0.6 | -36.6 | 0.27 | 6.5 | Aprobada | Aprobada |
| 1.33 | -1.3 | -36.5 | 0.27 | 6.5 | Aprobada | Aprobada |
| 1.33 | -2.4 | -36.5 | 0.27 | 6.5 | Aprobada | Aprobada |

Los resultados fueron los siguientes:

- La temperatura mínima que alcanzó el microcontrolador fue de -2.4 °C, la cual es superior a la temperatura mínima de operación que proporciona el fabricante (-40 °C)
- La corriente que proporcionaba la fuente se mantuvo constante
- No se presentó ninguna caída en el voltaje de alimentación
- La operación de lectura y escritura en la EEPROM se realizó de forma correcta
- La expresión matemática se resolvió de manera correcta

Resultados de la prueba en alto vacío

Para finalizar la prueba sobre el microcontrolador en la cámara de vacío, se procedió a ajustar la presión al valor mínimo posible. Se logró un valor de 1.33×10^{-5} mbar. Una vez que la cámara había llegado al alto vacío, la temperatura de la plancha se ajustó a 21 °C. En la tabla 4.4 se muestran las mediciones realizadas.

Tabla 4.4. Mediciones en alto vacío con temperatura cercana a 21 °C.

| Presión [mbar 10^{-5}] | Temperatura μ controlador [°C] | Temperatura de la plancha [°C] | Corriente máxima [A] | Voltaje [V] | Prueba de Memoria EEPROM | Operación matemática |
|---------------------------|------------------------------------|--------------------------------|----------------------|-------------|--------------------------|----------------------|
| 3.59 | 31.3 | 21.4 | 0.27 | 6.5 | Aprobada | Aprobada |
| 3.19 | 31.4 | 21.5 | 0.27 | 6.5 | Aprobada | Aprobada |
| 3.73 | 31.8 | 21.8 | 0.27 | 6.5 | Aprobada | Aprobada |

Los resultados fueron los siguientes:

- La máxima temperatura que alcanzó el microcontrolador fue de 31.8 °C, que también está por debajo de la temperatura máxima de operación que proporciona el fabricante
- La corriente que proporcionaba la fuente se mantuvo constante
- No se presentó ninguna caída en el voltaje de alimentación
- La operación de lectura y escritura en la EEPROM se realizó de forma correcta
- La expresión matemática se resolvió de manera correcta

La última prueba que se realizó fue ajustar la temperatura de la cámara de vacío a $-37\text{ }^{\circ}\text{C}$, conservando la presión a 1.33×10^{-5} mbar. En la tabla 4.5 se muestran las mediciones.

Tabla 4.5. Mediciones en alto vacío con temperatura cercana a $-37\text{ }^{\circ}\text{C}$.

| Presión [mbar 10^{-5}] | Temperatura μ controlador [$^{\circ}\text{C}$] | Temperatura de la plancha [$^{\circ}\text{C}$] | Corriente máxima [A] | Voltaje [V] | Prueba de Memoria EEPROM | Operación matemática |
|---------------------------|--|--|----------------------|-------------|--------------------------|----------------------|
| 2.13 | 7.7 | -37.3 | 0.27 | 6.5 | Aprobada | Aprobada |
| 2.13 | 7.5 | -31.7 | 0.27 | 6.5 | Aprobada | Aprobada |
| 2.26 | 7.3 | -31.7 | 0.27 | 6.5 | Aprobada | Aprobada |

Los resultados fueron los siguientes:

- La temperatura mínima que alcanzó el microcontrolador fue de $7.3\text{ }^{\circ}\text{C}$, que es superior a la temperatura mínima de operación que proporciona el fabricante
- La corriente que proporcionaba la fuente se mantuvo constante
- No se presentó ninguna caída en el voltaje de alimentación
- La operación de lectura y escritura en la EEPROM se realizó de forma correcta
- La operación matemática se realizó de manera correcta

Se midió y registró el consumo de potencia en la tarjeta de desarrollo. El voltaje de alimentación del circuito fue de 6.5 V. Las mediciones se muestran en la tabla 4.6, donde se observa cómo aumenta el consumo de potencia al agregar más procesos durante la operación del microcontrolador.

Tabla 4.6. Potencia consumida por el circuito de prueba.

| Operación | Potencia [W] |
|---|--------------|
| Sólo tarjeta de desarrollo, sin realizar procesos | 0.845 |
| Resolución de la expresión matemática | 0.91 |
| Comunicación con una PC | 1.365 |
| Escritura en memoria EEPROM | 1.56 |
| Conversión digital | 1.755 |

El valor de potencia más alto calculado fue de 1.755 W, en ese momento el microcontrolador se encontraba realizando todas las tareas descritas anteriormente.

Al terminar la prueba, el microcontrolador fue probado fuera de la cámara de vacío y éste funcionó con normalidad. Con los resultados obtenidos en las pruebas térmicas y de vacío, se comprueba que el microcontrolador propuesto puede operar sin ningún problema en las condiciones requeridas para la misión.

4.2. Pruebas al primer prototipo del *Housekeeping*

En el capítulo 3 se mencionó como se probaron los diferentes módulos del *Housekeeping* de manera separada, a continuación se explicará el procedimiento para validar las interfaces externas del primer prototipo del *Housekeeping*, así como las pruebas realizadas de vacío y ciclado térmico para asegurar su correcto funcionamiento.

4.2.1. Pruebas de interfaces externas

Para validar de manera íntegra las interfaces externas y el *software* del primer prototipo del *Housekeeping*, se realizaron pruebas con las siguientes interfaces:

- Comunicación Serie RS-422
 - Interacción entre la CPU y el *Housekeeping*
- Comunicación SPI e interfaz LVDS
 - Interacción entre el *Housekeeping* y los subsistemas
- Adquisición analógica
 - Lectura de voltajes fijos en las entradas de los multiplexores analógicos
- Activación de subsistemas
 - Encendido y apagado de relevadores usando el circuito de *High Level Command*

Estas pruebas se realizaron con circuitos armados en *protoboard*. En la imagen 4.3 se muestra la conexión del *Housekeeping* con los circuitos ya mencionados. Estos consisten en una *protoboard* con relevadores para simular el encendido y apagado de subsistemas, una *protoboard* con integrados LVDS para probar la comunicación SPI diferencial, así como una aproximación del subsistema HVPS, con el cual también se establece comunicación SPI. Finalmente se usaron divisores de voltaje para simular lecturas analógicas en las entradas de los multiplexores analógicos.

1. *Housekeeping*
2. *Protoboard* con circuitos Integrados LVDS
3. Aproximación al subsistema HVPS
4. *Protoboard* con relevadores

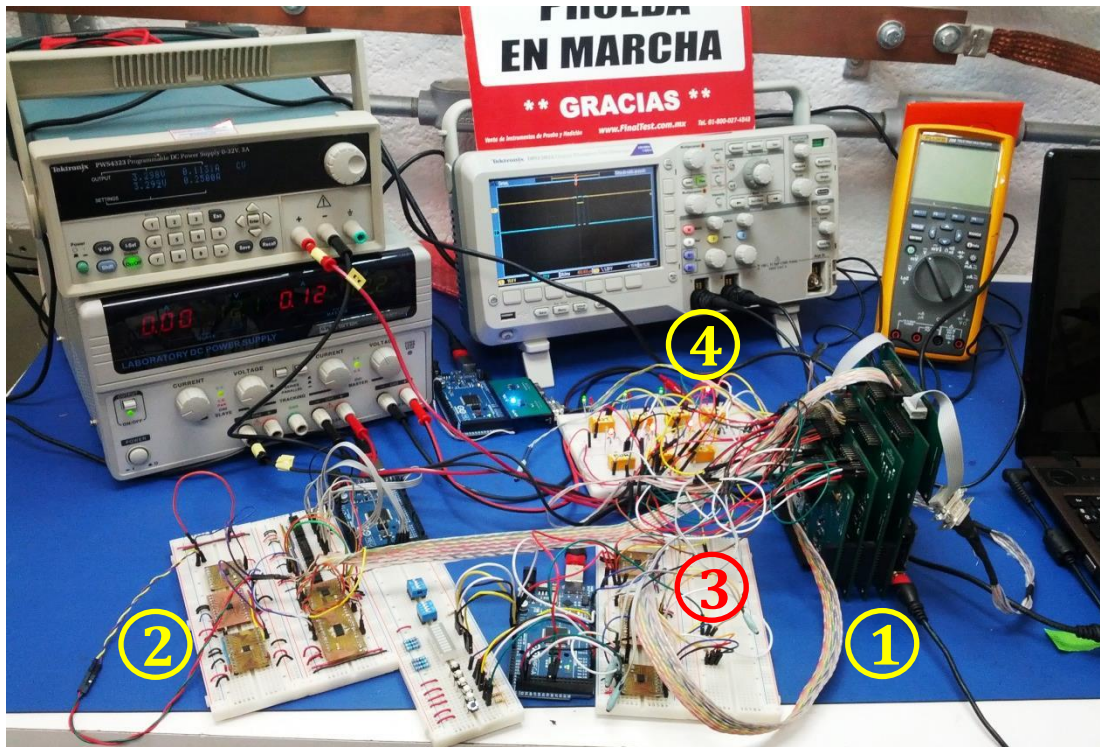


Figura 4.3. Pruebas a la primera versión del *Housekeeping*.

Comunicación serie RS-422

Para validar la interfaz RS-422, se realizaron pruebas para verificar la transmisión y la recepción de datos a través de este módulo en el *Housekeeping*, utilizando para ello un protocolo sencillo. La primera prueba se realizó fue verificar la correcta transmisión de una cadena de *bytes*. Se conectó un osciloscopio en la terminal transmisora de uno de los conectores del *Housekeeping*, el cual se programó para enviar una cadena de *bytes* continuamente. En la figura 4.4 se muestra un esquema de la prueba.

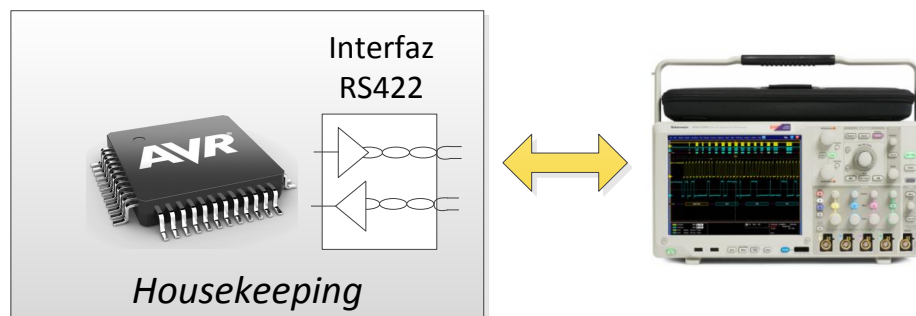


Figura 4.4. Prueba para la interfaz RS-422 en el *Housekeeping*.

Esta cadena de datos está compuesta por 9 *bytes*. La cadena de datos se pudo observar con ayuda de un osciloscopio digital, el cual tiene integrado un módulo de comunicaciones seriales. El *byte* de cabecera tiene como valor 0xAA y el *byte* de final de trama tiene valor de

0xFF. Esta prueba fue crucial porque se pudo verificar que *Housekeeping* es capaz de enviar tramas compuestas por varios bytes de forma correcta. En la figura 4.5 se puede observar una captura de osciloscopio.

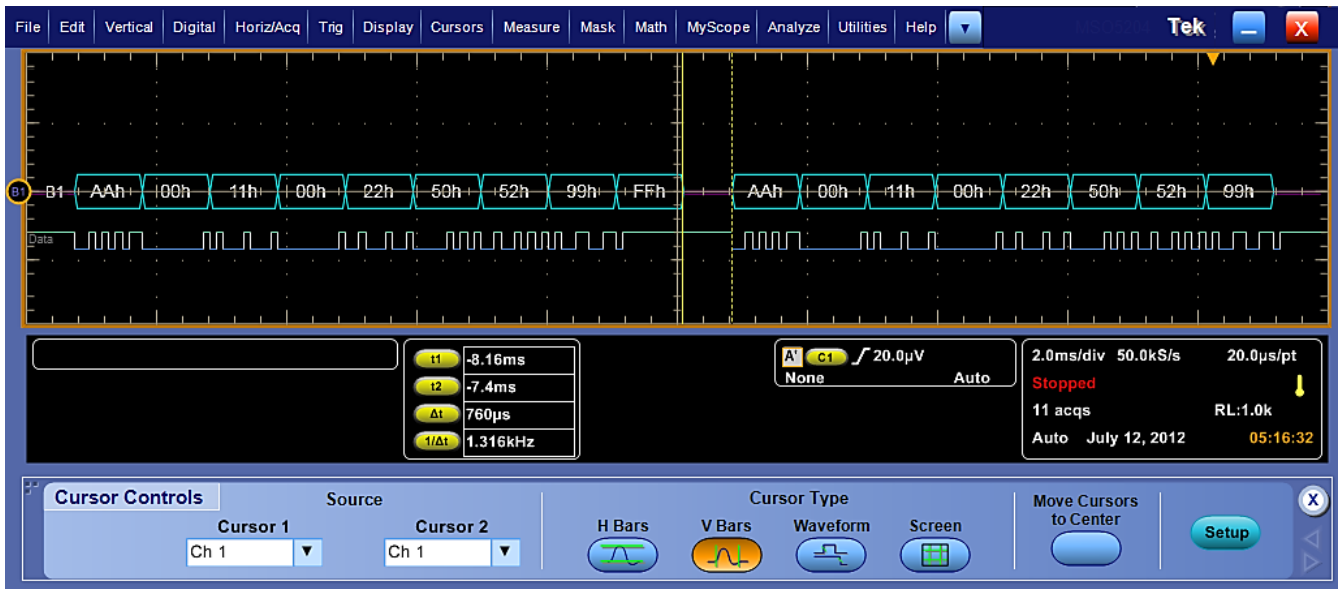


Figura 4.5. Trama enviada por el *Housekeeping*.

Para verificar la correcta recepción de datos a través de la interfaz RS-422, se realizó una prueba donde un usuario enviaba comandos en una PC que daba instrucciones al banco de pruebas de CPU, el cual enviaba un carácter al *Housekeeping*. Este carácter es interpretado y dependiendo de su valor activa uno de los circuitos de *High Level Command*. Cada circuito de *High Level Command* está conectado a un relevador, al cual se le conectaron LEDs para indicar el estado de encendido/apagado. En la figura 4.6 se muestra el esquema de esta prueba.

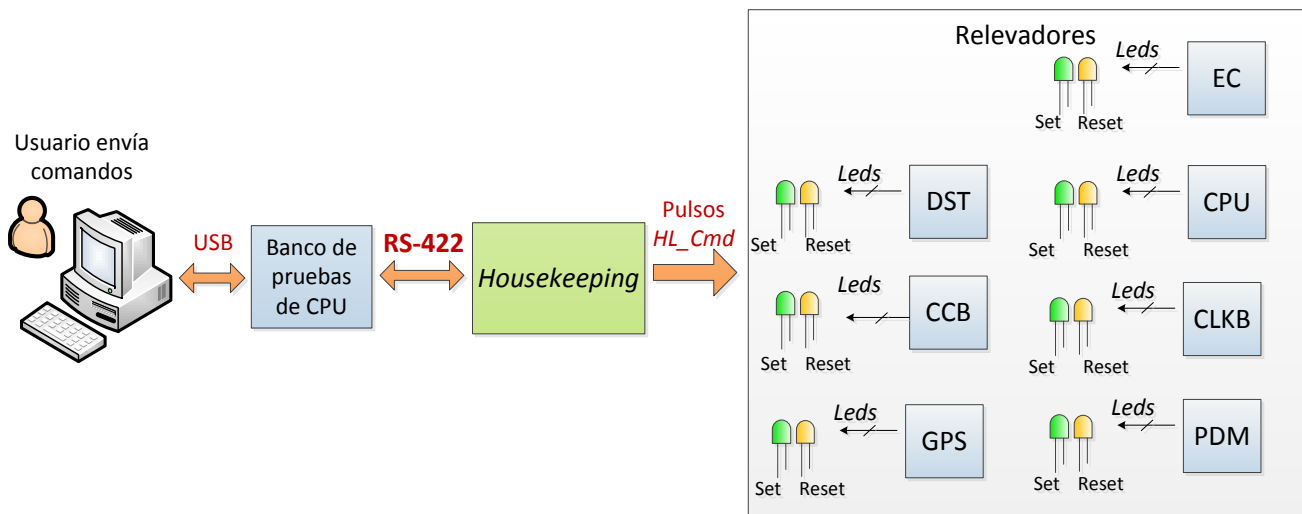


Figura 4.6. Esquema de pruebas para *High Level Command*.

Comunicación SPI e interfaz LVDS

Para validar la interfaz LVDS para el bus SPI del *Housekeeping*, se conectaron transmisores y receptores LVDS externos para realizar las siguientes tareas:

- Intercambio de tramas de datos entre el *Housekeeping* y la interfaz de subsistemas como CLKB, CCB, PDM.
- Recepción de alarmas a través de interrupciones generadas por pulsadores, simulando la interfaz de alarmas de subsistemas.
- Control básico de los dispositivos de HVPS (configuración de entrada/salida de expansores y establecimiento de voltajes en los convertidores analógicos digitales).

Se usó un microcontrolador para generar las tramas de datos que se intercambiaron con el *Housekeeping* y leer los expansores de puertos de HVPS. En la figura 4.7 se muestra un diagrama con la configuración usada en las pruebas.

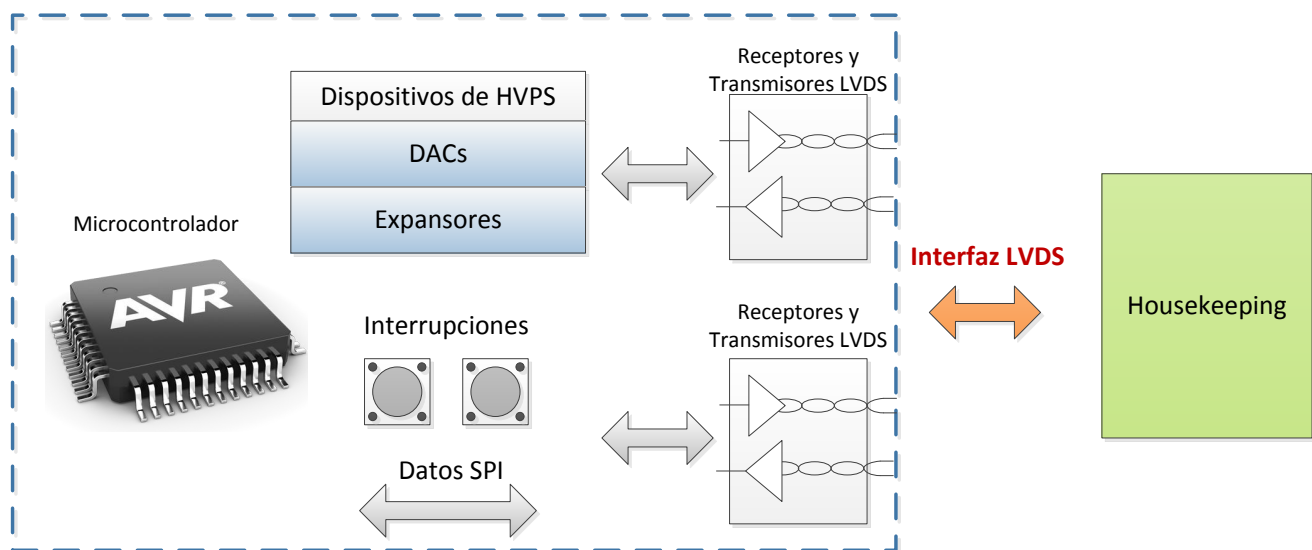


Figura 4.7. Pruebas de interfaz SPI LVDS del *Housekeeping*.

Respaldo de información del *Housekeeping*

Como parte del diseño del *Housekeeping*, se ha implementado un sistema de respaldo de información en una memoria *microSD*, debido a la flexibilidad que presenta para la extracción de su contenido, ya que la información almacenada se puede consultar en prácticamente cualquier dispositivo que tenga un lector de este tipo de memorias. Esta información es almacenada en archivos con un formato llamado “valores separados por comas”, (CSV). Los valores de cada subsistema son almacenados en un archivo diferente, en la figura 4.8 se observan los diferentes archivos generados por el *Housekeeping*.

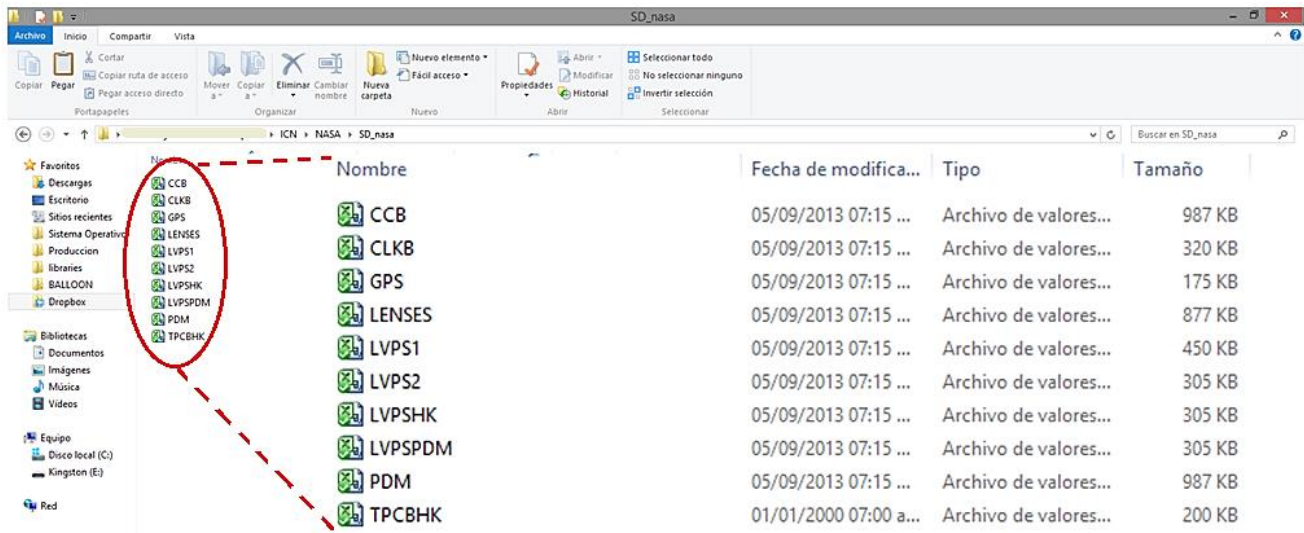


Figura 4.8. Archivos generados en la memoria SD.

El formato CSV es de gran utilidad, ya que al ser consultado en una computadora, el contenido se puede visualizar en una tabla de datos, por lo que se pueden obtener representaciones gráficas de una manera sencilla. En la figura 4.9 se muestra el contenido de un archivo del subsistema LENSES visualizado en una tabla de datos.

| | A | B | C | D | E | F | G | H | I | J | K |
|----|-------|-------|-------|-------|-------|-------|-------|-------|--------|-------|-------|
| 1 | 22.74 | 24.06 | 24.22 | 24.15 | 23.95 | 23.88 | 24.22 | 24.29 | -59.58 | 42055 | 36356 |
| 2 | 23.84 | 26.65 | 24.4 | 26.67 | 28.98 | 24.2 | 24.3 | 24.32 | -59.56 | 0 | 256 |
| 3 | 23.95 | 26.78 | 24.43 | 26.69 | 29.1 | 24.23 | 24.24 | 24.35 | -59.55 | 0 | 256 |
| 4 | 23.99 | 26.97 | 24.37 | 26.8 | 29.24 | 24.24 | 24.3 | 24.33 | -59.55 | 0 | 256 |
| 5 | 24.04 | 27.17 | 24.42 | 26.77 | 29.5 | 24.29 | 24.27 | 24.33 | -59.53 | 0 | 256 |
| 6 | 24.09 | 27.33 | 24.45 | 26.87 | 29.68 | 24.29 | 24.3 | 24.35 | -59.6 | 0 | 256 |
| 7 | 24.14 | 27.5 | 24.41 | 26.92 | 29.83 | 24.32 | 24.29 | 24.33 | -59.58 | 0 | 256 |
| 8 | 24.15 | 27.67 | 24.41 | 26.98 | 30.03 | 24.34 | 24.28 | 24.37 | -59.55 | 0 | 256 |
| 9 | 24.23 | 27.82 | 24.44 | 27.04 | 30.23 | 24.38 | 24.32 | 24.32 | -59.56 | 0 | 256 |
| 10 | 24.37 | 27.96 | 24.42 | 27.09 | 30.43 | 24.4 | 24.32 | 24.39 | -59.53 | 0 | 256 |
| 11 | 24.38 | 28.15 | 24.46 | 27.1 | 30.59 | 24.4 | 24.35 | 24.39 | -59.56 | 0 | 256 |
| 12 | 24.51 | 28.33 | 24.44 | 27.17 | 30.8 | 24.45 | 24.38 | 24.33 | -59.58 | 0 | 256 |
| 13 | 24.54 | 28.41 | 24.44 | 27.18 | 30.95 | 24.53 | 24.33 | 24.43 | -59.51 | 0 | 256 |
| 14 | 24.62 | 28.57 | 24.45 | 27.24 | 31.11 | 24.49 | 24.36 | 24.39 | -59.58 | 0 | 256 |
| 15 | 24.68 | 28.73 | 24.47 | 27.25 | 31.25 | 24.53 | 24.36 | 24.38 | -59.55 | 0 | 256 |
| 16 | 24.68 | 28.84 | 24.49 | 27.31 | 31.41 | 24.58 | 24.38 | 24.32 | -59.56 | 0 | 256 |
| 17 | 24.7 | 29.07 | 24.5 | 27.34 | 31.63 | 24.6 | 24.4 | 24.36 | -59.58 | 0 | 256 |

Figura 4.9. Datos obtenidos del subsistema LENSES.

4.2.2. Pruebas de vacío y ciclado térmico

Se realizaron pruebas de vacío y ciclado térmico al primer prototipo del *Housekeeping*, simulando las condiciones más cercanas a las esperadas durante el vuelo.

Las condiciones de temperatura y presión que se configuraron en la cámara de vacío fueron las siguientes:

- Presión de 3 mbar
- Temperatura máxima: 50°C
- Temperatura mínima: -30°C

En la figura 4.10 se muestra la cámara de vacío del ICN y la computadora para la adquisición de datos enviados por USB, a dispositivos que cuenten con ésta conexión dentro de la cámara.



Figura 4.10. Cámara de vacío del ICN.

Las pruebas del *Housekeeping* consistieron en:

- Comprobar la correcta comunicación entre la CPU y el *Housekeeping*
- Comprobar la correcta adquisición analógica del *Housekeeping*

Comunicación serie entre la CPU y el *Housekeeping*

Para esta prueba se implementó el protocolo final propuesto para la CPU, se realizaron las siguientes pruebas:

- Se hizo el control de los relevadores de las LVPS por medio del *High Level Command*
- Se verificó el estado de los *Contact Closure*
- Se solicitaron parámetros que el *Housekeeping* adquirió en la prueba

Gracias a esta prueba se pudo verificar el correcto funcionamiento de los *High Level Command*, así como la lectura correcta del cierre de relevadores del *Contact Closure*.

Los datos que CPU podía a solicitar al *Housekeeping* principalmente son:

- Temperaturas de termistores
- Voltajes que se monitorean de las LVPS
- Corrientes que se monitorean de las LVPS
- Parámetros que *Housekeeping* intercambia con sus esclavos por SPI

Cabe mencionar que un punto muy importante fue que se pudo verificar el correcto funcionamiento del CRC propuesto.

Adquisición analógica

En esta prueba el *Housekeeping* se encargó de adquirir los valores de los monitores de voltaje y corriente de las LVPS, con esta prueba se comprobó que este módulo funcionaba de manera correcta.

La prueba se llevó a cabo de la siguiente manera:

- Al iniciar la prueba, la temperatura de la cámara de vacío era de 25 °C (temperatura ambiente), posteriormente se elevó a 50 °C. Al llegar a este punto la temperatura se mantuvo aproximadamente por 12 horas.
- Una vez que transcurrieron 12 horas en las condiciones anteriores, se procedió bajar la temperatura hasta -30°C, manteniendo esta condición por el mismo tiempo.
- Este procedimiento se repitió una vez más y al llegar al final de la prueba se llevó la temperatura interior a temperatura ambiente.

Cabe mencionar que la temperatura que se configuró en la cámara de vacío es la temperatura de la plancha donde se colocó el *Housekeeping*, que es el único medio de transmisión de calor por contacto con el *Housekeeping*. Se implementó una cubierta temporal simulando la caja en la que estaría ubicado el *Housekeeping*.

En la figura 4.11 se muestra el perfil de la prueba realizada.

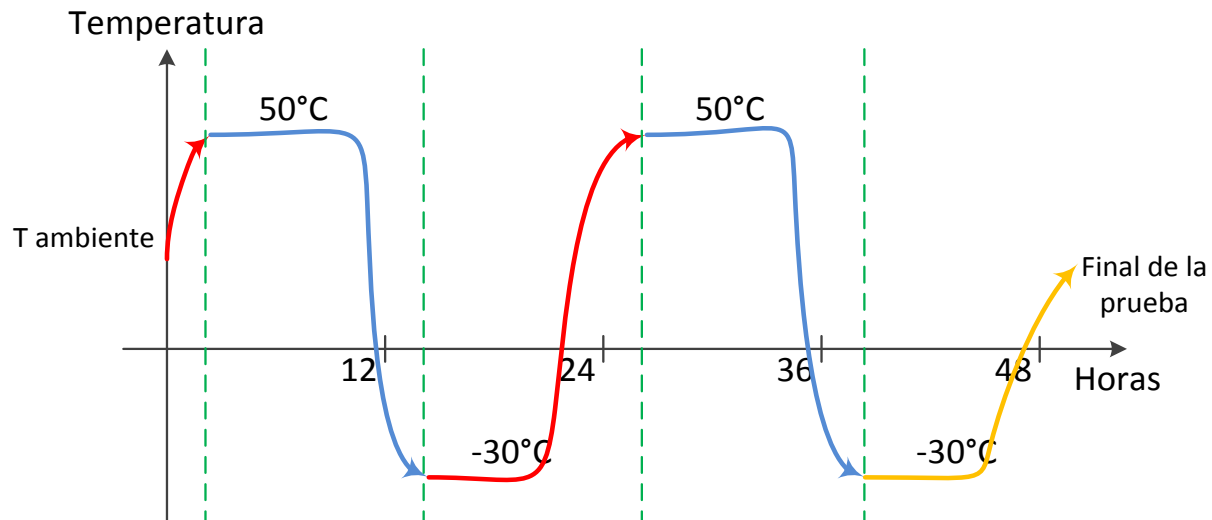


Figura 4.11. Perfil de temperatura aplicado para las pruebas.

Durante la prueba se utilizaron los termistores del *Housekeeping*, para medir la temperatura de los componentes más críticos que lo componen, así como en puntos de importancia dentro de la campana de la cámara de vacío. En la siguiente lista se menciona en que puntos se midió la temperatura:

- Microcontrolador
- Regulador de voltaje dentro de la tarjeta de desarrollo
- Superficie de la tarjeta número cinco
- Soporte de la estructura del *Housekeeping*
- Placa de un gabinete colocado sobre el *Housekeeping*
- Plancha de la cámara de vacío
- Ambiente en vacío

Durante la prueba el *Housekeeping* estuvo adquiriendo y almacenando los datos de temperatura registrada en los diferentes termistores; además, utilizó un puerto USB para mandar esta información a una computadora a través de un puerto serial virtual. Esta computadora capturó los datos del puerto serie y los almacenó para su análisis. Los datos procesados de cada termistor se muestran en las siguientes gráficas.

En la figura 4.12 se muestra el *Housekeeping* sobre la plancha de la cámara de vacío antes del inicio de la prueba.

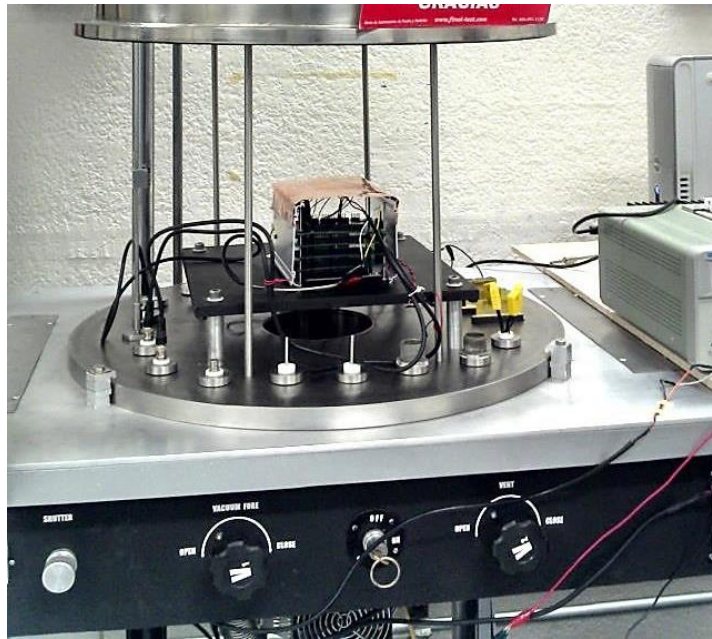


Figura 4.12. Housekeeping sobre la plancha de la cámara de vacío.

En la imagen 4.13 se observa el comportamiento de la temperatura en el microcontrolador durante la prueba.

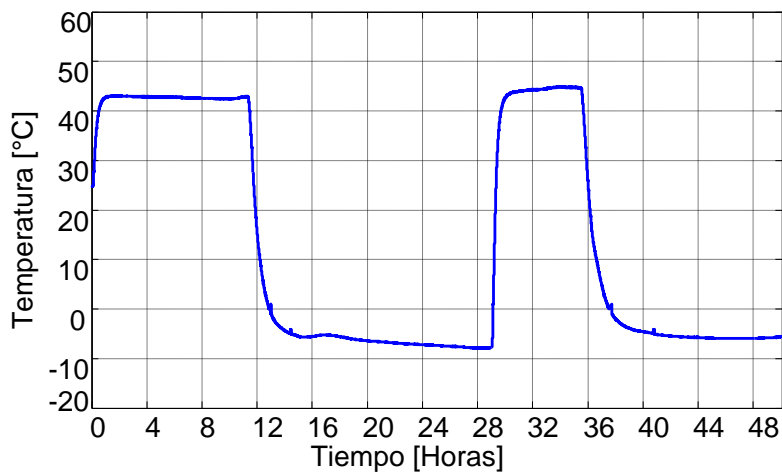


Figura 4.13. Temperatura en el microcontrolador.

- La temperatura máxima que alcanzó fue de 45 °C, esto ocurrió en el segundo ciclo de calentamiento.
- La temperatura mínima que reportó fue de -8 °C, aproximadamente, esto sucedió en el primer ciclo de enfriamiento.
- En esta prueba no se sobrepasó las temperaturas máxima (85 °C) y mínima (-40 °C) de operación que proporciona el fabricante para el microcontrolador.

En la figura 4.14 se muestra el comportamiento de la temperatura en el regulador de voltaje de la tarjeta de desarrollo.

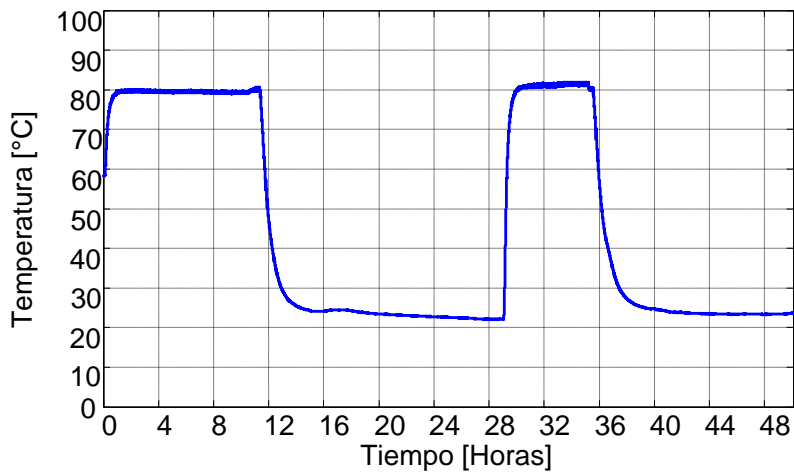


Figura 4.14. Temperatura en el regulador de voltaje.

- La temperatura máxima que alcanzó el regulador fue de 81 °C, esto ocurrió en el segundo ciclo de calentamiento.
- La temperatura mínima que alcanzó este componente fue de aproximadamente 21 °C, esta temperatura fue alcanzada en el primer ciclo de enfriamiento.
- Nuevamente se comprobó que en este componente no se sobrepasa la temperatura máxima de operación (125 °C).

En la figura 4.15 se muestra el comportamiento de la temperatura sobre la superficie de la PCB05.

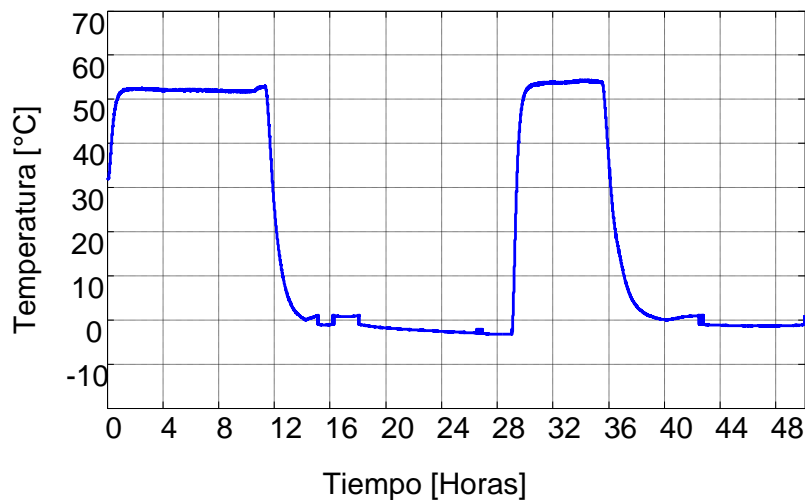


Figura 4.15. Temperatura en la superficie de la PCB05.

- La temperatura máxima que se midió en la superficie de esta tarjeta fue de 53 °C aproximadamente, esto ocurrió en el segundo ciclo de calentamiento.
- La temperatura mínima que se midió en esta tarjeta fue de -2 °C aproximadamente, esta temperatura fue alcanzada en el primer ciclo de enfriamiento.

En la figura 4.16 se muestra el comportamiento de la temperatura registrada en uno de los soportes de la estructura del *Housekeeping*.

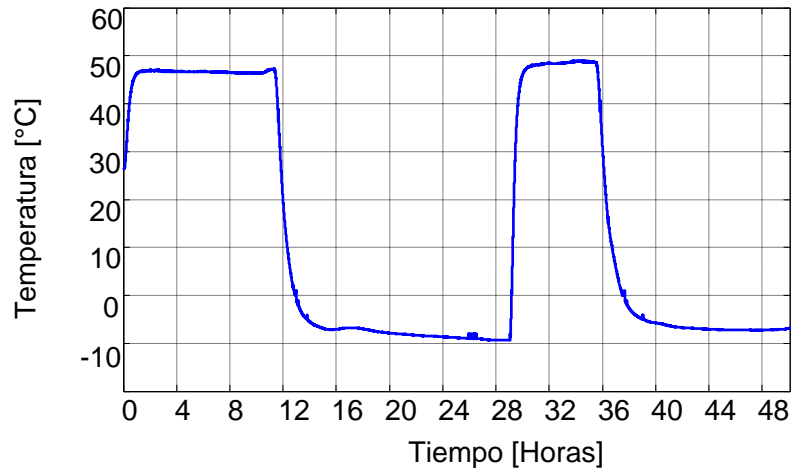


Figura 4.16. Temperatura en uno de los soportes del *Housekeeping*.

- La temperatura máxima que se registró fue de 49 °C aproximadamente, esto ocurrió en el segundo ciclo de calentamiento.
- La temperatura mínima que se midió en el soporte fue de -10 °C aproximadamente, esta temperatura fue alcanzada en el primer ciclo de enfriamiento.

En la figura 4.17 se muestra el comportamiento de la temperatura sobre una pared del gabinete que se colocó sobre el *Housekeeping*.

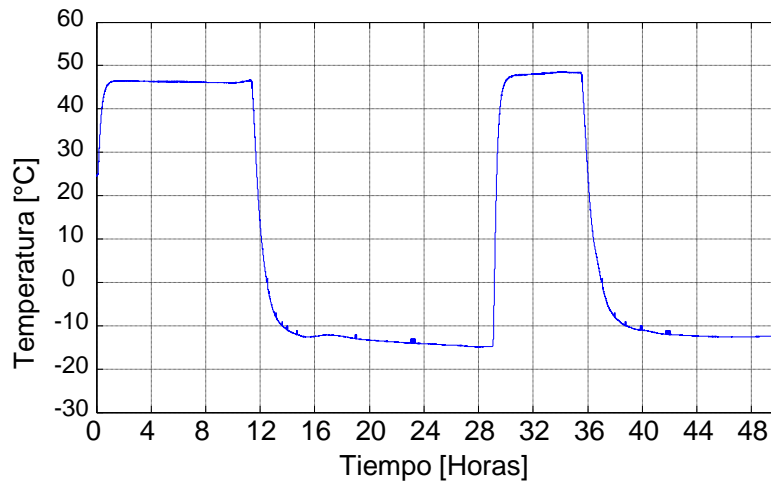


Figura 4.17. Temperatura en el gabinete sobrepuesto.

- La temperatura máxima que se midió en el gabinete fue de 49 °C, aproximadamente, esto ocurrió en el segundo ciclo de calentamiento.
- La temperatura mínima que se midió en el gabinete fue casi de -15 °C, esta temperatura fue alcanzada en el primer ciclo de enfriamiento.

En la figura 4.18 se muestra el comportamiento de la temperatura en la plancha de la cámara de vacío.

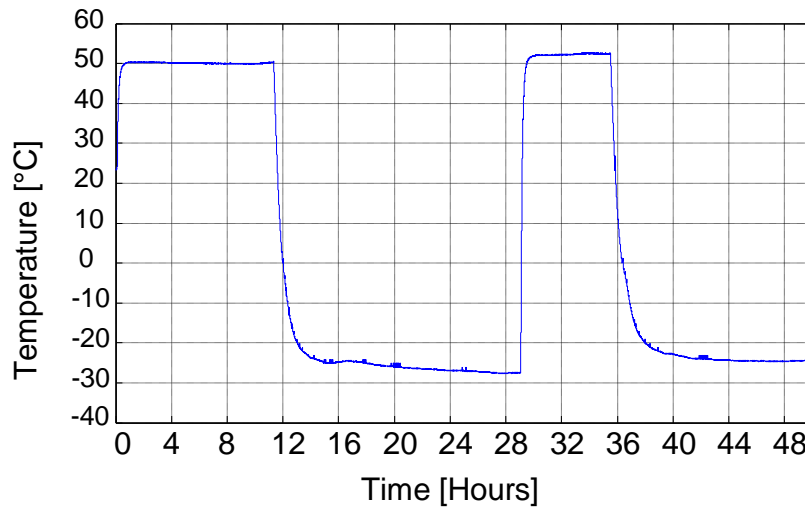


Figura 4.18. Temperatura en la plancha de la cámara de vacío.

- La temperatura máxima que se midió en la plancha fue de 51 °C aproximadamente, esto ocurrió en el segundo ciclo de calentamiento.
- La temperatura mínima que se midió en la plancha de la cámara de vacío fue casi de - 29 °C, esta temperatura fue alcanzada en el primer ciclo de enfriamiento.
- Se puede observar que se midieron valores muy parecidos a los que fueron configurados en la cámara de vacío.

En la figura 4.19 se muestra el comportamiento de la temperatura del ambiente dentro de la cámara de vacío.

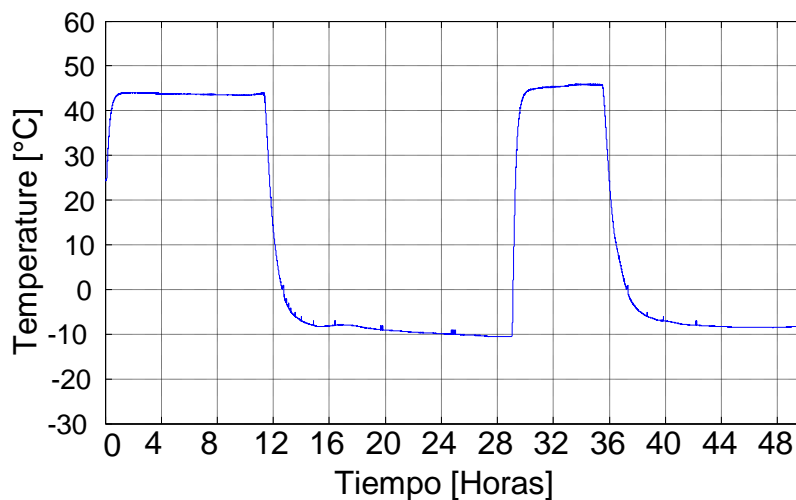


Figura 4.19. Temperatura de la cámara de vacío.

- La temperatura máxima que se midió fue de 45 °C aproximadamente, esto ocurrió en el segundo ciclo de calentamiento.
- La temperatura mínima que se midió en la plancha de la cámara de vacío fue de -10 °C, esto ocurrió en el primer ciclo de enfriamiento.

4.3. Pruebas a la versión de vuelo del *Housekeeping*

Las pruebas a la versión de vuelo del *Housekeeping* fueron realizadas utilizando los bancos de pruebas desarrollados en el capítulo 3, además se utilizaron las LVPS reales, es decir, las que serán utilizadas durante el vuelo. Para asegurar el correcto funcionamiento del *Housekeeping* fue necesario implementar nuevamente las pruebas de vacío y ciclado térmico.

4.3.1. Integración con bancos de prueba

Con el fin de validar de manera íntegra las interfaces externas y el *software* del *Housekeeping*, se realizó la integración con todos los bancos de prueba que se fabricaron. En la figura 4.20 se muestra la conexión entre el *Housekeeping* y los bancos de prueba de los subsistemas.

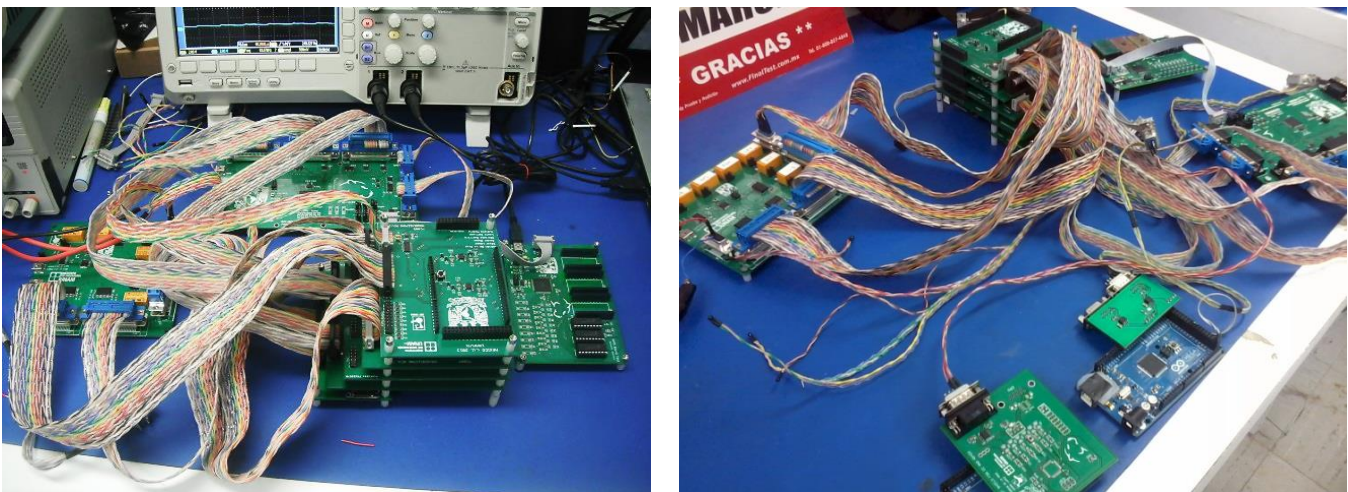


Figura 4.20. *Housekeeping* y los bancos de pruebas.

Con esta integración se verificó lo siguiente:

- **CPU y SIREN.** Se comprobó la correcta comunicación usando el protocolo definitivo para RS422.
 - Se enviaron comandos de encendido/apagado de subsistemas. Esto se verificó con los relevadores del banco de pruebas de LVPS.
 - Obtención de parámetros de monitoreo analógico. Se hizo la petición de valores de monitoreo analógico y se verificó que se recibieran de manera correcta por parte de la CPU.

- Configuración de umbrales para generación de alarmas. Esto se verificó con la lectura de termistores, forzando alarmas al establecer valores de los umbrales a temperatura ambiente.
- Recepción de alarmas.
- **SPI**
 - Se verificó el correcto intercambio de información entre *Housekeeping* y los subsistemas a través de la interfaz LVDS.
 - Se verificó la correcta generación de alarmas a través del banco de pruebas SPI, y que el *Housekeeping* detectara y atendiera correctamente la alarma.
 - Se comprobó el correcto envío de los comandos de *Reset* y *Program_B* por parte del *Housekeeping*.
 - Se verificó que se hiciera de manera correcta el control de los dispositivos de HVPS a través de su banco de pruebas.
- **LVPS**
 - Se comprobó que se recibieran de manera correcta los comandos de encendido/apagado de subsistemas, a través de los LEDs indicadores de cada relevador y de la lectura de las señales *Contact Closure*.
 - Se comprobó que el *Housekeeping* hiciera de manera correcta la lectura de voltajes para el monitoreo de voltajes y corrientes de LVPS.
- **Lectura de termistores.** Se comprobó que se leyera de manera correcta cada termistor y que se generaran alarmas cuando se excediera la temperatura máxima configurada.
- **Almacenamiento SD.** Se verificó que se almacenaran de manera correcta los datos de monitoreo en la memoria SD.
- **Verificación del consumo de energía.** Se verificó que el *Housekeeping* no excediera el consumo máximo permitido.

4.3.2. Integración con las LVPS

La primera integración que se realizó con los subsistemas reales fue con las LVPS, las cuales fueron diseñadas y fabricadas también en México. Se realizó el montaje del *Housekeeping* y de las LVPS dentro de sus gabinetes correspondientes, además de que fue posible contar con un gabinete (*rack*) parecido al que se usará en DP, donde fue posible introducir todos los módulos de LVPS junto con el *Housekeeping*. En la figura 4.21 se muestra el gabinete y los subsistemas que lo conforman.

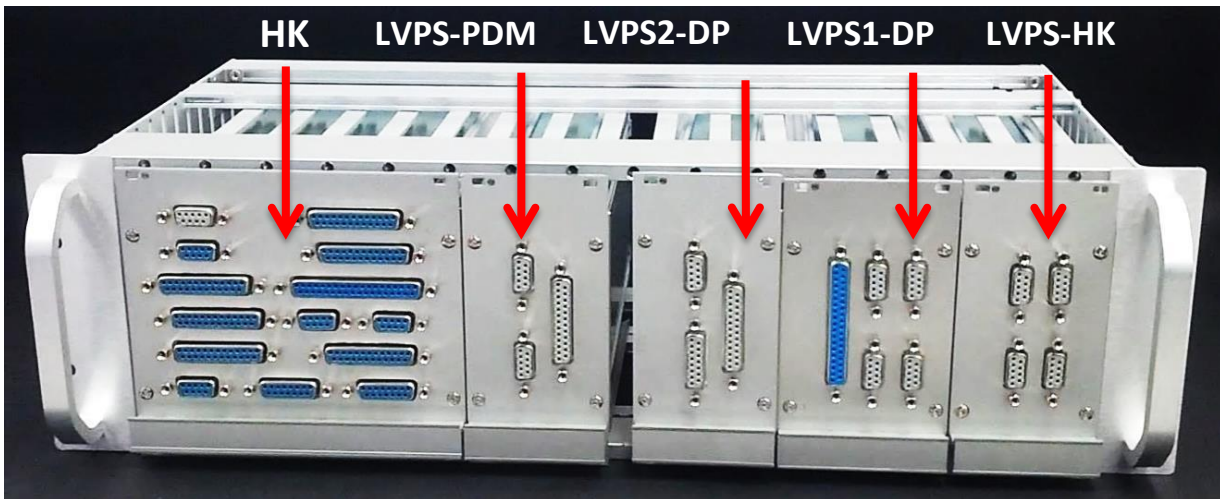


Figura 4.21. Housekeeping y LVPS dentro del rack.

En esta prueba se conectó la alimentación del *Housekeeping*, con lo que se pudo comprobar el consumo real del *Housekeeping* en operación. Además, al conectarse a los demás módulos de las LVPS, se pudo comprobar la correcta adquisición de los valores de los monitores de voltaje y corriente, el control ON/OFF de los relevadores de las LVPS y la verificación de las señales *Contact Closure* de las LVPS.

En la figura 4.22 se muestra la conexión entre el *Housekeeping* y las LVPS dentro del rack.

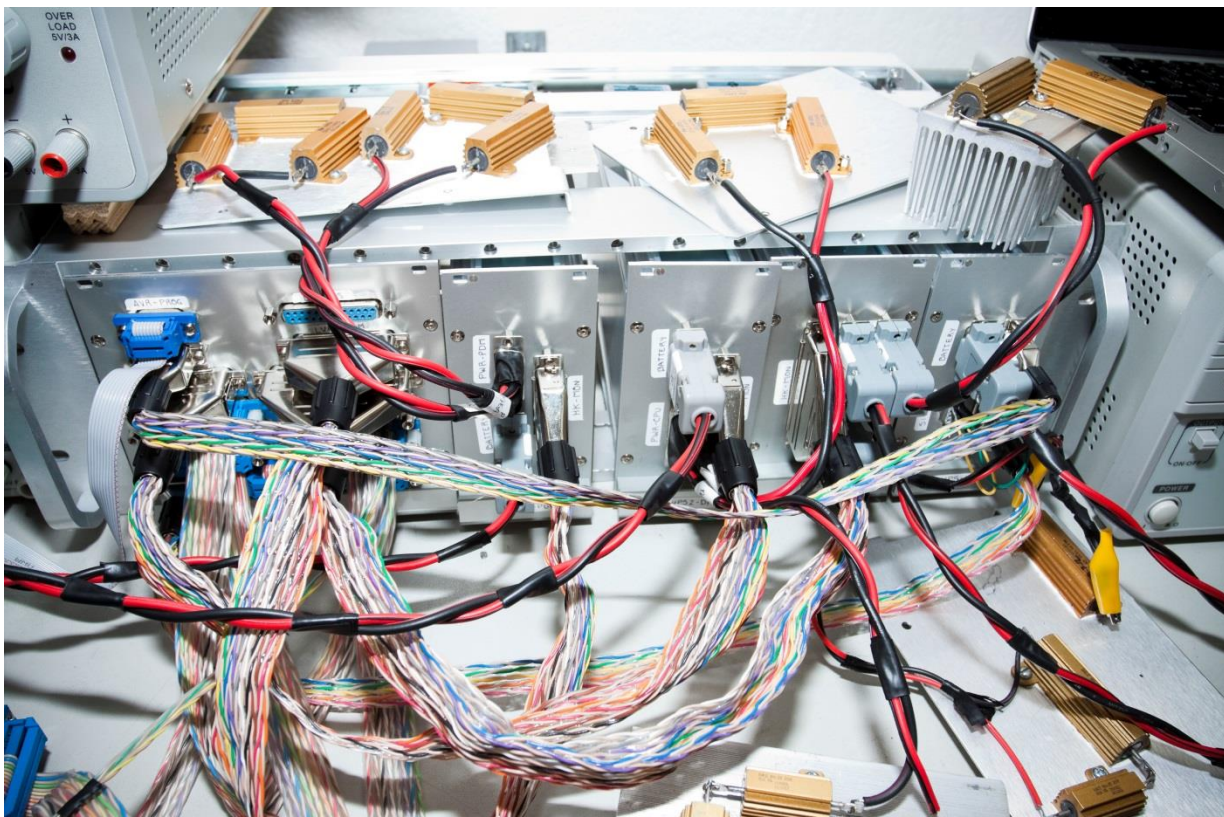


Figura 4.22. Conexión entre el *Housekeeping* y las LVPS.

En la figura anterior se muestra el gabinete con los módulos montados en su interior. En la parte superior de la imagen se observan las cargas estáticas que se usaron para simular el consumo de los subsistemas de las LVPS. Estas cargas generaron valores de monitoreo de voltaje y corriente que fueron verificados con las lecturas del *Housekeeping*. Durante la prueba se registró el consumo en operación del *Housekeeping*, los datos se muestran en la tabla 4.7.

Tabla 4.7. Consumos del *Housekeeping*.

| Parámetro | Lectura |
|--------------------|---------|
| Voltaje de 12 V | 11.82 V |
| Corriente de 12 V | 300 mA |
| Voltaje de 3.3 V | 3.29 V |
| Corriente de 3.3 V | 348 mA |

La potencia consumida para la fuente de 12 V es del orden de **3.6 W**.

La potencia consumida para la fuente de 3.3 V es del orden de **1.14 W**.

Los valores obtenidos de consumo cumplen con el límite de potencia establecido para el *Housekeeping*, menor a **5 W**. En la siguiente figura 4.23 se observa toda la conexión incluyendo las LVPS y los bancos de pruebas de SPI y de CPU.

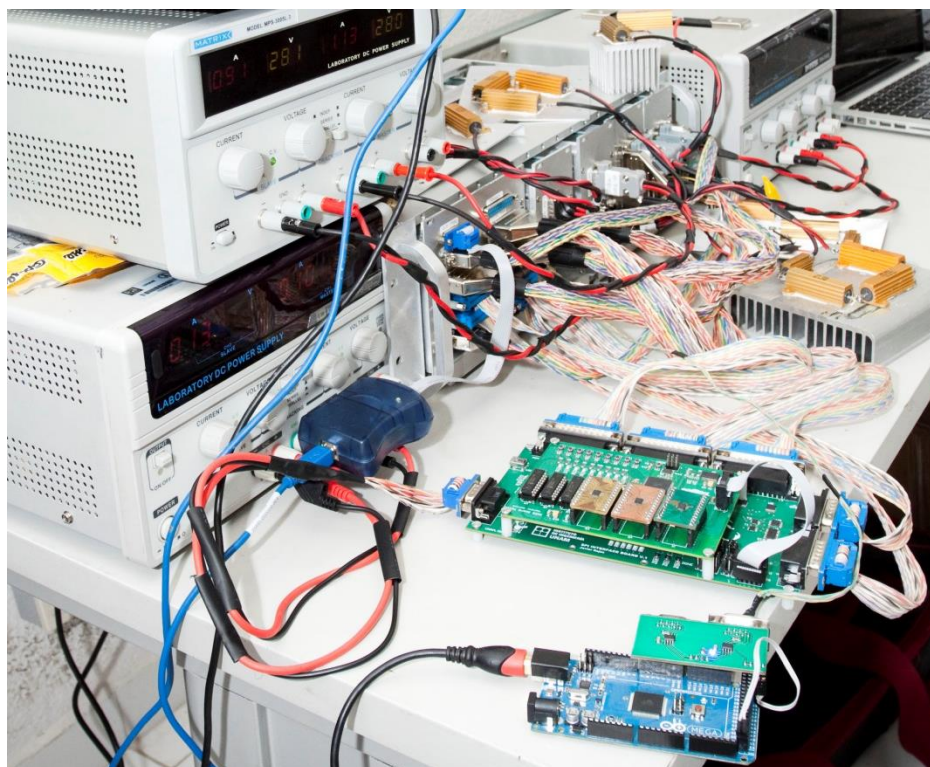


Figura 4.23. *Housekeeping*, módulos LVPS y los bancos de prueba.

4.3.3. Condiciones de vacío y ciclado térmico

Una vez que se llevaron a cabo las pruebas de todo el conjunto a temperatura ambiente, se procedió a realizar pruebas de vacío y de ciclado térmico. Debido a la presencia de las LVPS reales el banco de pruebas de las LVPS no fue utilizado.

Las pruebas que se realizaron fueron las siguientes:

- Medición de temperatura en componentes críticos de *Housekeeping* y algunos puntos del *rack*, así como en componentes críticos de las LVPS.
- Comunicación serie entre la CPU y el *Housekeeping*.
- Comunicación SPI entre todos los subsistemas y el *Housekeeping*.
- Monitoreo analógico de las LVPS con carga estática.
- Encendido y apagado de los subsistemas.
- Verificación de estado ON/OFF de subsistemas.

Los componentes críticos del *Housekeeping* en los que se midió temperatura fueron: el microcontrolador y el regulador de la tarjeta de desarrollo.

Las condiciones de la prueba fueron:

- Presión de 3 mbar
- Temperatura máxima de 50 °C
- Temperatura mínima de -20 °C

En la figura 4.24 se muestra la pantalla de la cámara de vacío con la presión requerida; las unidades están en mbar y se indican en la parte inferior de la pantalla.



Figura 4.24. Presión de 3 mbar durante la prueba.

Esta prueba fue similar a la prueba que se le realizó a la primera versión del *Housekeeping*, debido a esto, sólo se hizo un ciclo de calentamiento y enfriamiento.

En la figura 4.25 se puede observar el *rack* con el *Housekeeping*, las LVPS, los bancos de pruebas y las cargas estáticas sobre la plancha de la cámara de vacío, antes de comenzar la prueba.

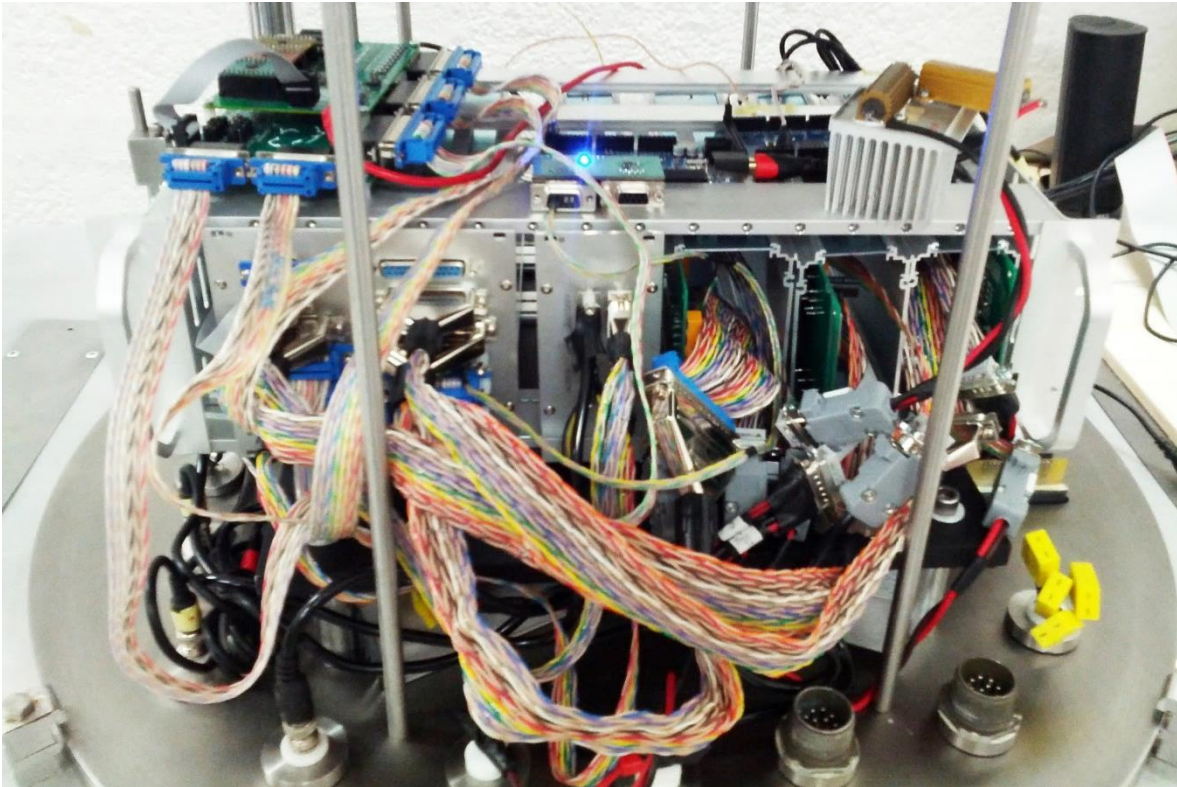


Figura 4.25. *Housekeeping* y demás módulos en la cámara de vacío.

Los termistores de *Housekeeping* fueron colocados en los siguientes puntos:

- Microcontrolador
- Regulador de la tarjeta de desarrollo
- Gabinete de *Housekeeping*
- Interior del *rack*
- Parte inferior del *rack*

Los datos de temperatura fueron almacenados por el *Housekeeping* en la memoria *microSD*. Posteriormente, mediante el uso de MATLAB, estos datos fueron graficados. De esta manera se puede observar el comportamiento de la temperatura en los dispositivos y en los puntos que fueron elegidos. En la figura 4.26 se muestra el comportamiento de la temperatura en el microcontrolador del *Housekeeping*.

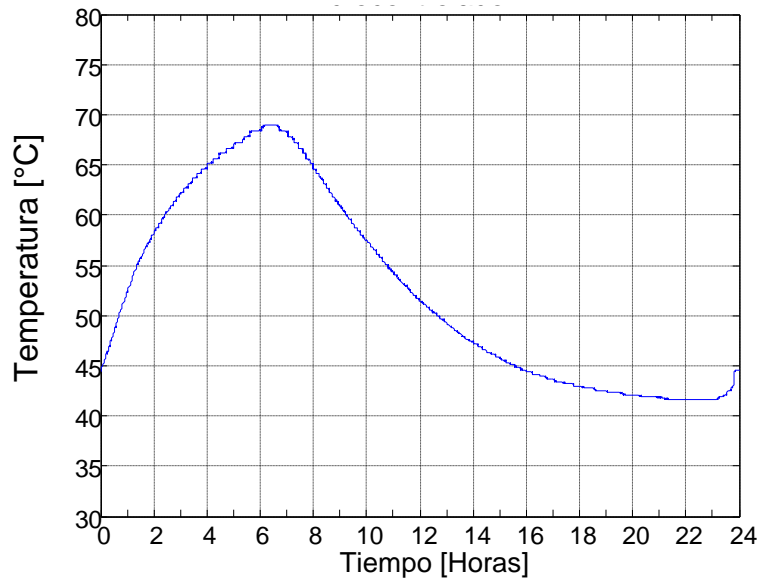


Figura 4.26. Temperatura en el microcontrolador.

- La temperatura máxima que se alcanzó fue de 69 °C, esto ocurrió a las 6 horas de iniciada la prueba.
- La temperatura mínima que se alcanzó fue de 42 °C, esto ocurrió a las 23 horas de que inició la prueba, una hora antes de terminar la prueba.
- Se puede observar que la temperatura de este componente no alcanzó la temperatura máxima especificada por el fabricante que es de 85 °C.

En la figura 4.27 se muestra el comportamiento de la temperatura en el regulador de la tarjeta de desarrollo, éste es el componente más crítico, debido a que es el componente que más calor disipa.

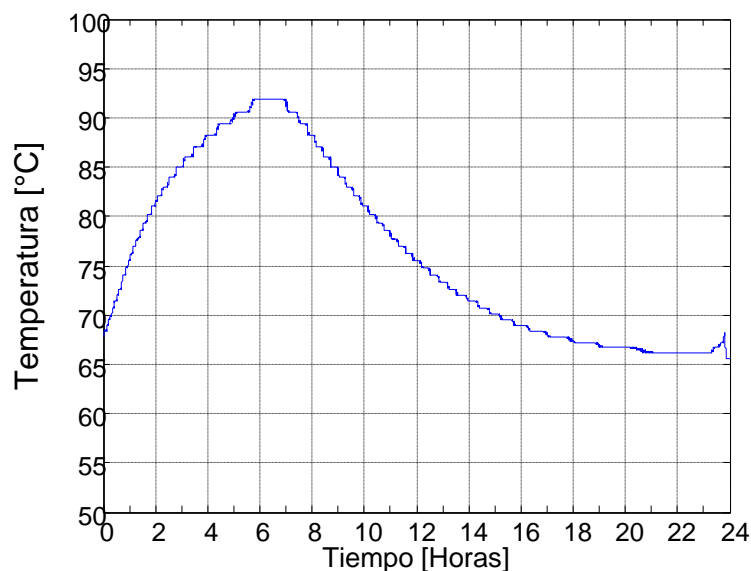


Figura 4.27. Temperatura en el regulador de la tarjeta de desarrollo.

- La temperatura máxima que se alcanzó fue de 92 °C, esto ocurrió a las 6 y media horas de iniciada la prueba.
- La temperatura mínima que se alcanzó fue de 66 °C, esto ocurrió a las 23 horas de que inició la prueba, una hora antes de terminar la prueba.
- Se puede observar que la temperatura de este componente no alcanzó la temperatura máxima especificada por el fabricante que es de 125 °C. También se puede observar que al enfriar la cámara de vacío, la temperatura del componente estuvo muy lejos de la temperatura mínima que especifica el fabricante (0 °C).

En la figura 4.28 se muestra el comportamiento de la temperatura sobre el gabinete del *Housekeeping*.

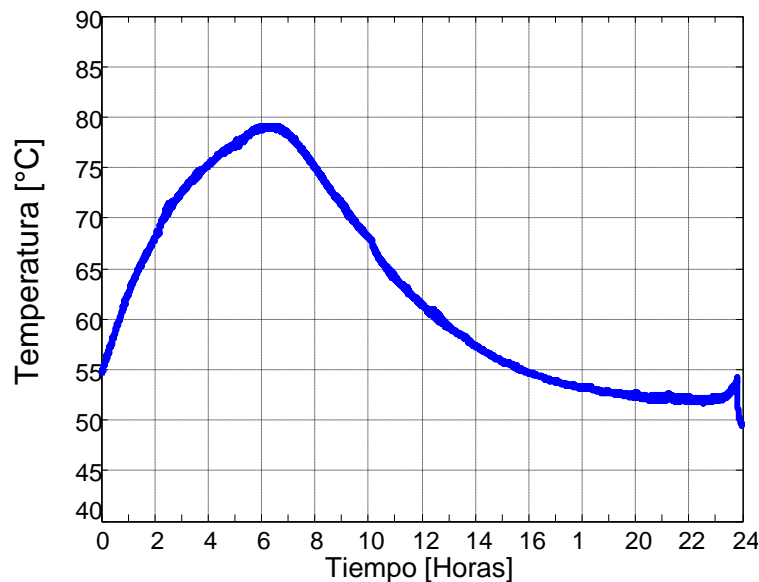


Figura 4.28. Temperatura en el gabinete del *Housekeeping*.

- La temperatura máxima que se alcanzó fue de 79 °C, esto ocurrió a las 6 horas de iniciada la prueba.
- La temperatura mínima que se alcanzó fue de 52 °C, aproximadamente, esto ocurrió a las 23 horas de que inició la prueba; una hora antes de terminar la prueba.
- La temperatura mínima era un resultado esperado debido al calor emitido por las cargas estáticas de las LVPS, cada una de 15 W, aproximadamente.

En la figura 4.29 se muestra el comportamiento de la temperatura dentro del *rack* que contiene a *Housekeeping* y las LVPS.

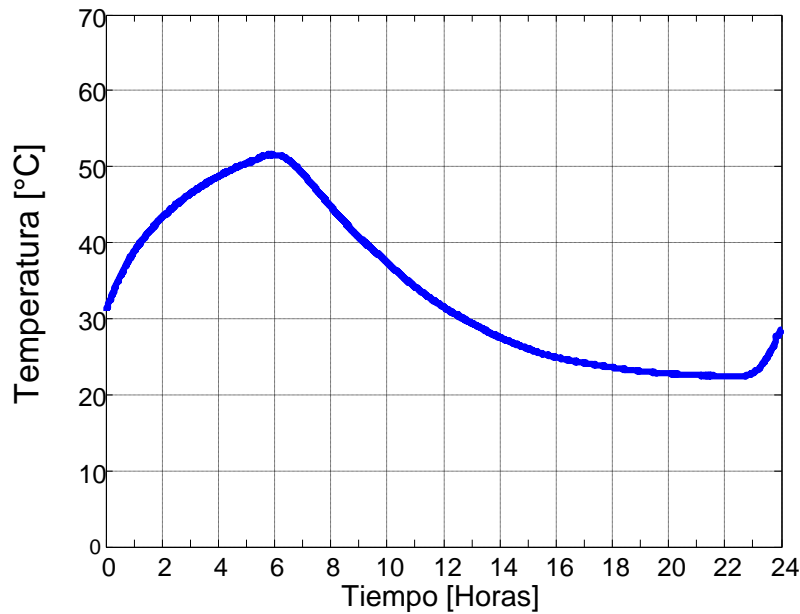


Figura 4.29. Temperatura en el interior del *rack*.

- La temperatura máxima que se alcanzó fue de 51 °C, esto ocurrió a las 6 horas de iniciada la prueba.
- La temperatura mínima que se alcanzó fue de 23 °C, aproximadamente, esto ocurrió a las 23 horas de que inició la prueba; una hora antes de terminar la prueba.
- La temperatura mínima también se vio afectada por las cargas de las LVPS.

En la figura 4.30 se muestra el comportamiento de la temperatura en un punto fuera del *rack* que contiene al *Housekeeping* y las LVPS.

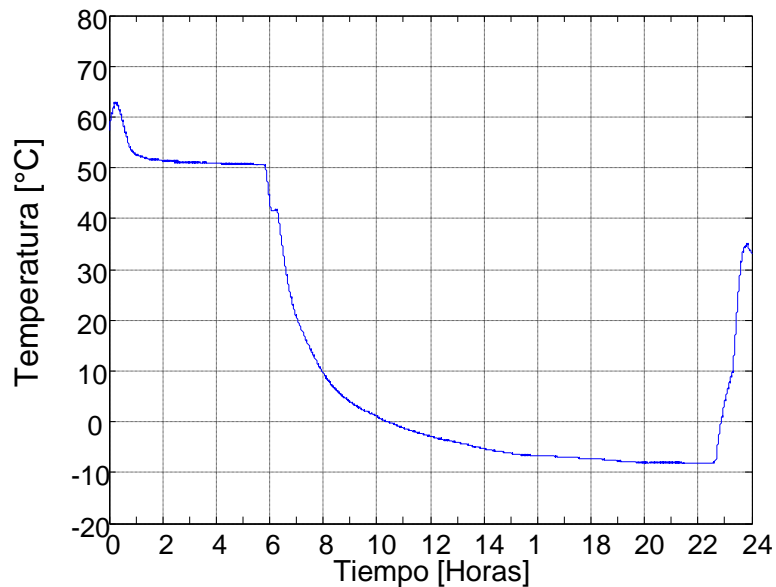


Figura 4.30. Temperatura en un punto fuera del *rack*.

- La temperatura máxima que se alcanzó fue de 50 °C, esto ocurrió durante las 6 primeras horas de iniciada la prueba.
- La temperatura mínima que se alcanzó fue de -10 °C aproximadamente.
- Como se observa en la gráfica, después de las 6 primeras horas de prueba, la temperatura comenzó a descender hasta que llegó a su punto mínimo a las 23 horas de iniciada la prueba.
- El comportamiento de la temperatura fue muy parecido a la temperatura a la que se configuró la cámara de vacío.

Los resultados de estas pruebas fueron principalmente para caracterizar el comportamiento de las LVPS interactuando con el *Housekeeping* en condiciones de vacío.

El *Housekeeping* ya había sido sometido a pruebas de este tipo y las soportó, pero se aprovechó para probar algunas interfaces que no se habían utilizado.

5. Integraciones internacionales y vuelo con NASA

En este capítulo se describen las integraciones que se realizaron con la colaboración europea, realizando pruebas de comunicación entre el *Housekeeping* y los demás subsistemas del DP, además se presentan los resultados de la integración entre el DP y PDM, se describe también la prueba de vuelo en un globo estratosférico de la NASA, como parte del proyecto Pixqui.

5.1. Integración del *Housekeeping* en *Data Processor*

Como parte del cronograma propuesto por parte del CNES para la integración del instrumento, se espera que para finales de 2013 se encuentre terminado y ensamblado, listo para ser probado en una serie de vuelos durante 2014. Para cumplir con el objetivo fueron propuestas dos integraciones para ensamblar el DP, de manera que se pudieran realizar las primeras pruebas de comunicación entre los diferentes subsistemas que lo componen.

Esta serie de integraciones se realizaron en el Instituto Nacional de Física Nuclear (INFN), en Nápoles, Italia, en las cuales hubo participación de la colaboración mexicana.

5.1.1. Primera integración

En esta etapa se esperaba que la UNAM entregara el primer prototipo, con las características de *hardware* necesarias para establecer una comunicación básica con los demás subsistemas. Tuvo lugar en las últimas semanas de noviembre de 2012.

La integración consistió en hacer una primera prueba de compatibilidad con todas las interfaces propuestas, para lograr el correcto intercambio de información entre los diferentes subsistemas.

En esta primera integración estuvieron presentes los siguientes subsistemas:

- CPU
- GPS
- CLKB
- CCB
- LVPS-HK

- LVPS1-DP
- LVPS2-DP

La primera parte de la integración consistió en alimentar al *Housekeeping* con la LVPS-HK y verificar su consumo. Dado que esta versión de la LVPS no contaba con los monitores de voltaje y corriente de los convertidores que alimentan al *Housekeeping*, se registraron los valores de voltaje y corriente a la entrada de la LVPS. En la tabla 5.1 se muestran los valores registrados, los cuales cumplen con las especificaciones de potencia del *Housekeeping* descritas en el capítulo 3.

Tabla 5.1. Consumo registrado por la LVPS-*Housekeeping*.

| | Voltaje [V] | Corriente [I] | Potencia [W] |
|---|-------------|---------------|--------------|
| LVPS-<i>Housekeeping</i> sin carga | 28 | 0.01 | 0.28 |
| LVPS-<i>Housekeeping</i> con carga | 28 | 0.169 | 4.732 |

Una vez que *Housekeeping* funcionaba con la LVPS-HK. Se procedió a realizar la conexión con la CPU y se comprobó lo siguiente:

- Que el cable de conexión coincidiera en la asignación de señales, para la comunicación RS-422.
- Se midió en el osciloscopio que las señales generadas por cada sistema cumplieran con el estándar RS-422.
- El correcto envío y recepción de *bytes* de forma bidireccional.

Con base en las pruebas anteriores se concluyó la primera integración con la CPU. En la figura 5.1 se observa: 1, CPU; 2, LVPS; 3, *Housekeeping*.

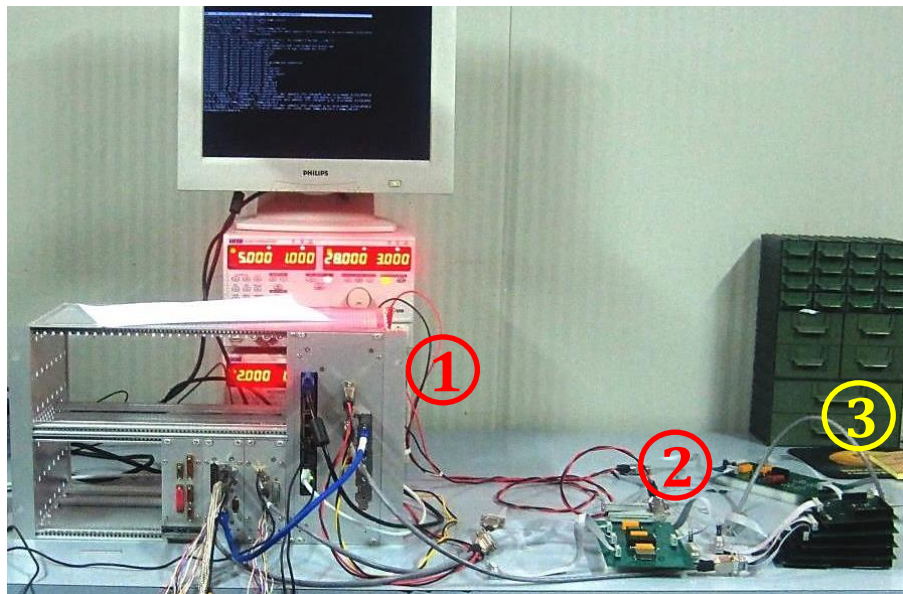


Figura 5.1. Conexión entre el *Housekeeping*, la CPU y la LVPS.

Pruebas con SPI

Las pruebas que se realizaron con los subsistemas que utilizan el *bus* SPI fueron las siguientes:

- Se verificaron las terminales de los conectores y la correcta asignación de señales.
- Se realizó el intercambio de información con cada subsistema de forma independiente, estableciendo lo siguiente:
 - Que el intercambio de datos sería de 16 *bits* en 2 paquetes de 8 *bits*.
 - Los comandos necesarios para interactuar con cada subsistema, acordando usar los mismos comandos de lectura para CLKB y CCB.
 - Los comandos para la lectura de los sensores de GPS, tratándose de sensores comerciales.

En cuanto a la parte mecánica, no se logró una buena integración debido a que las cajas construidas eran de mayor peso al que estaba destinado para el *Housekeeping* y las LVPS. Por lo que se decidió utilizar los módulos comerciales para la versión de vuelo. En la figura 5.2 se muestra como fueron colocados los gabinetes construidos en México al *rack* donde finalmente se colocará el DP.

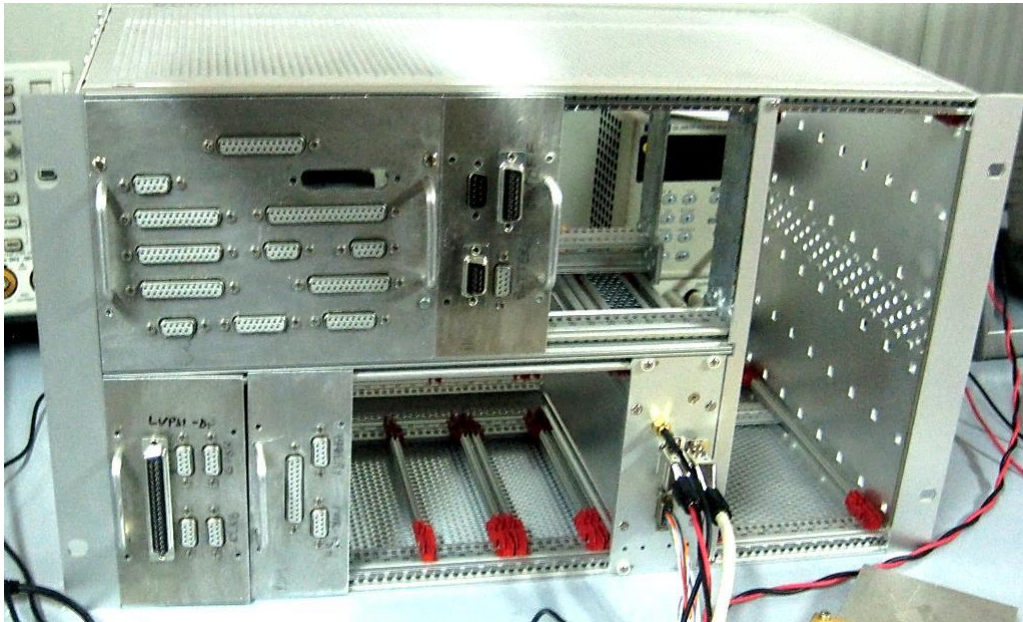


Figura 5.2. Integración del *Housekeeping* en el *rack*.

Modificaciones y mejoras

Gracias a las pruebas realizadas durante esta integración, se detectaron algunos detalles que era preciso modificar en la configuración del *hardware* del *Housekeeping*, esto para corregir algunos problemas y lograr algunas mejoras en las interfaces de comunicación.

- Se detectó un problema con el sistema de encendido/apagado de subsistemas, éste generaba pulsos no deseados a la salida del decodificador que controla las señales de *High Level Command*, lo cual provocaba que se encendiera más de un subsistema al mandar un pulso de encendido. Se decidió controlar el habilitador del decodificador directamente desde el microcontrolador, y utilizar una resistencia de *pull down* para la línea de cambio de estado en el decodificador.
- Con el fin de aumentar la confiabilidad de las líneas de reprogramación de los subsistemas, se agregó una resistencia externa de *pull up* a todas las señales de *Program_B*. Estas últimas ya contaban con una resistencia habilitada por *software*, pero en el caso de que el microcontrolador llegara a fallar, es de suma importancia contar con una resistencia de respaldo sujeta a un *bus* de alimentación separado al del microcontrolador.
- En el caso de la comunicación LVDS con SPI, se tuvo la necesidad de asignar un control de los habilitadores de los dispositivos LVDS, debido a que algunos subsistemas no aseguraban un estado de alta impedancia en sus líneas de comunicación, cuando eran deshabilitados como esclavos. Con este habilitador se asegura una alta impedancia para iniciar una nueva comunicación con un subsistema y asegurar la integridad en los datos leídos.
- Se implementó de forma improvisada un conector en el panel frontal del gabinete para el programador del microcontrolador del *Housekeeping*, esto permite cargar nuevo *firmware* de manera fácil, sin la necesidad de que algún encargado directo del *Housekeeping* se encuentre presente para realizar alguna actualización. De esta manera se dejó un sistema funcionando y que era capaz de recibir actualizaciones de *firmware*.

Conclusión

Como resultado del éxito logrado en esta primera integración, se entregó el primer prototipo parcialmente ensamblado del DP, para ser utilizado de forma autónoma en Riken, Japón, para realizar una integración con el sistema PDM. Con esto, la colaboración “congeló” las modificaciones en *hardware* de los distintos prototipos, para evitar que algún cambio sustancial pudiera comprometer el ajustado cronograma de la misión. Además, se solicitó al equipo mexicano la construcción de 2 módulos más con el fin de ser entregados como versiones de vuelo, agregando las mejoras de confiabilidad propuestas y colocados en gabinetes comerciales, con protección para lograr buena compatibilidad electromagnética (EMC) con otros dispositivos.

Se logró el objetivo de construir el dispositivo en tiempo para ser probado con otros subsistemas, cumpliendo los requerimientos iniciales. Esto dio pie a una nueva fase de desarrollo, basada principalmente en *software*. La UNAM se comprometió a construir bancos de pruebas de los subsistemas del *Housekeeping*, con el fin de desarrollar un *firmware* más enfocado a la operación acordada entre todos los sistemas del DP. En la tabla 5.2 se puede observar un resumen de los resultados obtenidos en la primera integración.

Tabla 5.2. Resultados de la primera integración en el DP con el *Housekeeping*.

| Sistemas | Conexión Eléctrica | Líneas probadas | Resultados |
|------------|-------------------------|--|-------------------------------|
| HVPS | - | No | No presente |
| GPS | Comunicación LVDS | Bus SPI | Incompatibilidad con sensores |
| | Asignación Cable | Correcta | |
| CCB | Comunicación LVDS | Dedicadas a: Alarmas, <i>Soft Reset, Hard Reset.</i> | Comunicación básica |
| | | Bus SPI | |
| | Compatibilidad de Cable | Correcta | |
| CLKB | Comunicación LVDS | Dedicadas a: Alarmas, <i>Soft Reset, hard Reset.</i> | Comunicación básica |
| | | Bus SPI | |
| | Compatibilidad de Cable | Correcta | |
| CPU | Estándar RS-422 | Envío de 1 <i>Byte</i> | Control Básico |
| | Compatibilidad de Cable | Correcta | |
| SIREN | Estándar RS-422 | No | No presente |
| | Compatibilidad de Cable | Correcta | |
| PDM | - | No | No presente |
| LVPS-DP1 | Lectura Analógica | Monitores | Faltantes |
| | HL_CMD | Encendido / Apagado | Control Básico |
| | <i>Contact Closure</i> | Probado | |
| | Compatibilidad de Cable | Correcta | |
| LVPS-DP2 | Lectura Analógica | Monitores | Faltantes |
| | HL_CMD | Encendido / Apagado | Control Básico |
| | <i>Contact Closure</i> | Probado | |
| | Compatibilidad de Cable | Correcta | |
| LENSES | - | - | No presente |
| HK-MONITOR | Lectura Analógica | Monitores | Faltantes |
| | Compatibilidad de Cable | Correcta | Control Básico |
| LVPS-PDM | Lectura Analógica | Monitores | Faltantes |
| | HL_CMD | Encendido / Apagado | No utilizada |
| | <i>Contact Closure</i> | Probado | |
| | Compatibilidad de Cable | Correcta | |

5.1.2. Segunda integración

Para esta segunda integración con el DP, se desarrollaron los bancos de prueba para simular a los subsistemas con los que interactúa el *Housekeeping*, por lo que se realizaron mejoras y modificaciones en el *Housekeeping* respecto a la primera integración, éstas se muestran en la tabla 5.3.

Tabla 5.3. Mejoras alcanzadas con los bancos de prueba.

| Sistemas ya Integrados | Sistema por simular | Conexión Eléctrica | Líneas probadas | Software a Desarrollar |
|------------------------|---------------------|-------------------------|--------------------------|---|
| CPU | CPU | Estándar RS422 | Comunicación Serie | <i>Firmware</i> (alarmas, monitoreo, almacenamiento, depuración) |
| | | Compatibilidad de Cable | Asignación de Terminales | |
| SIREN | SIREN | Estándar RS422 | Comunicación Serie | |
| | | Compatibilidad de Cable | Asignación de Terminales | |
| PDM | PDM | LVDS | Comunicación SPI | Propuesta de funcionamiento |
| | | Compatibilidad de Cable | Asignación de Terminales | |
| LVPS-DP1 | LVPS-DP1 | Lectura Analógica | Monitores | <i>Firmware</i> (alarmas, monitoreo, almacenamiento, depuración) |
| | | HL_CMD | Apagado / Encendido | |
| | | <i>Contact Closure</i> | Cierre de Relevador | |
| | | Compatibilidad de Cable | Asignación de Terminales | |
| LVPS-DP2 | LVPS-DP2 | Lectura Analógica | Monitores | |
| | | HL_CMD | Apagado / Encendido | |
| | | <i>Contact Closure</i> | Cierre de Relevador | |
| | | Compatibilidad de Cable | Asignación de Terminales | |
| LENSES | LENSES | Lectura Analógica | Monitores | |
| HK-MONITOR | HK-MONITOR | Lectura Analógica | Monitores | |
| | | Compatibilidad de Cable | Asignación de Terminales | |
| LVPS-PDM | LVPS-PDM | Lectura Analógica | Monitores | |
| | | HL_CMD | Apagado / Encendido | |
| | | <i>Contact Closure</i> | Cierre de Relevador | |
| | | Compatibilidad de Cable | Asignación de Terminales | |

Con ayuda de los bancos de prueba se logró desarrollar un *firmware* que contiene la mayoría de las funciones del *Housekeeping*, además se incluyó un proceso de depuración que permitió observar su correcto funcionamiento, el cual consiste en visualizar toda la información que recibe el *Housekeeping* en una pantalla.

En esta segunda integración se tuvieron presentes los siguientes subsistemas, en sus versiones de vuelo:

- CPU
- GPS
- CLKB
- CCB
- LVPS-HK
- LVPS-DP1
- LVPS-DP2

En esta etapa, la primera serie de pruebas se realizó en el laboratorio de detectores del ICN, con las LVPS. Ya se contaba con los monitores de voltaje y corriente para los convertidores que alimentan al *Housekeeping*, además de que se contaba también con el sistema de almacenamiento interno de datos en la *microSD*, por lo que los datos monitoreados y posteriormente almacenados se pudieron analizar después de la prueba. La integración con los demás subsistemas del DP se realizó nuevamente en el INFN, en Nápoles, Italia. En la figura 5.3 se puede observar el *Housekeeping* montado en el DP.

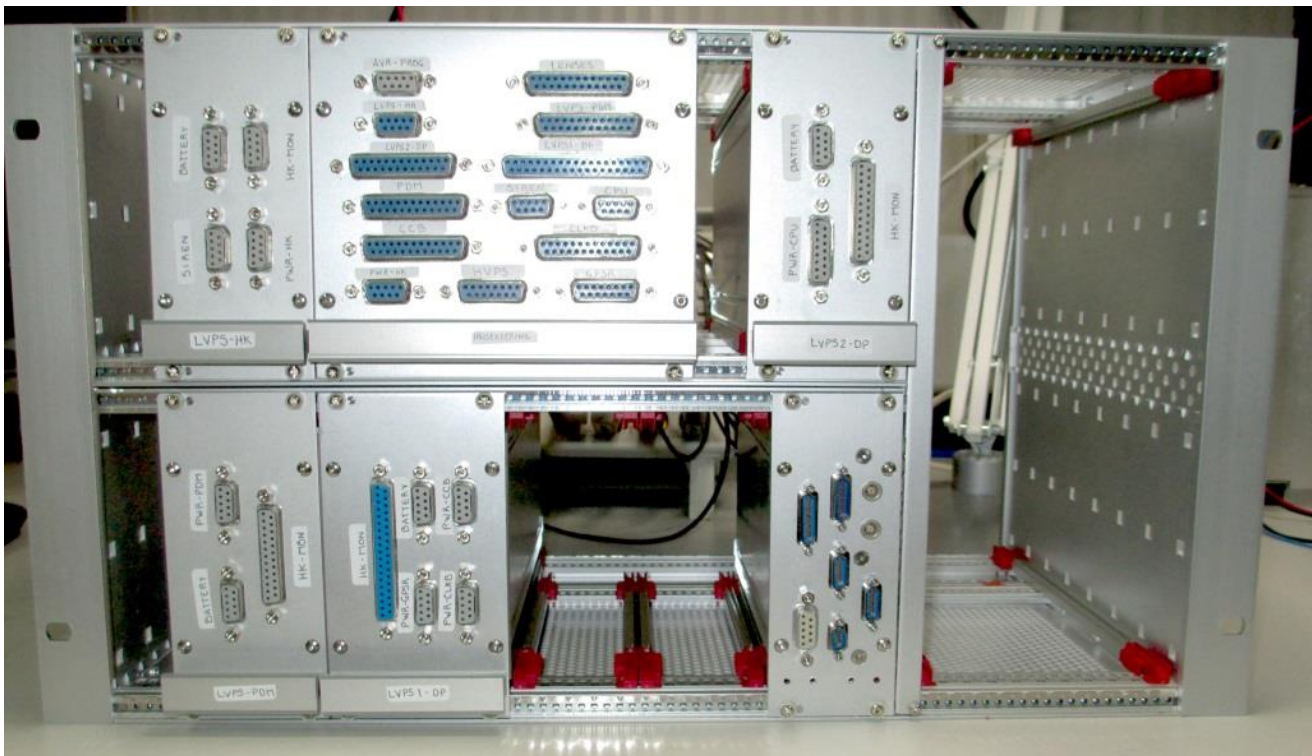


Figura 5.3. *Housekeeping* y LVPS montadas en rack de DP.

Para iniciar esta integración se verificó que las interfaces de *hardware* fueran compatibles, de la misma manera en que se realizó en la primera integración.

El intercambio de información de todos los colaboradores hizo más fácil la integración, debido que ya se había acordado la forma de operación de los demás sistemas, por lo que la integración consistía en evaluar las operaciones programadas por cada subsistema con sus subsistemas reales.

Se realizaron las siguientes pruebas con la CPU:

- Se verificó que el cálculo del CRC fuera el mismo por ambas partes, una vez hecho esto, se comenzó a utilizar todos los comandos para operar al *Housekeeping*, por lo que el banco de pruebas de CPU ya no fue utilizado.
- Se comenzó a trabajar en la generación de alarmas y en las acciones que se llevarán a cabo por la activación de éstas. También en esta etapa, la interacción con CPU se hizo más compleja, al agregar situaciones de alarmas de baja y alta prioridad.

Debido a las especificaciones técnicas del sistema de almacenamiento de la CPU, no puede ser operado a temperaturas menores a 0 °C, por lo que se implementó la conexión de algunos de los termistores del *Housekeeping* en el módulo de CPU, para medir la temperatura de sus componentes antes de ser operados. Esta conexión se llevó a cabo en el conector destinado para LENSES.

También se probó la interacción con los subsistemas SPI de la siguiente manera:

- CPU solicitaba una reprogramación por *Program_B* en los subsistemas, con el fin de verificar que estas líneas funcionaban correctamente.
- Para verificar el correcto funcionamiento de los subsistemas se realizó un monitoreo de los subsistemas cada 1.2 segundos, en periodos de 10 horas, durante tres semanas, que duró la integración.

En la figura 5.4 se muestra la integración del *Housekeeping* en el DP.

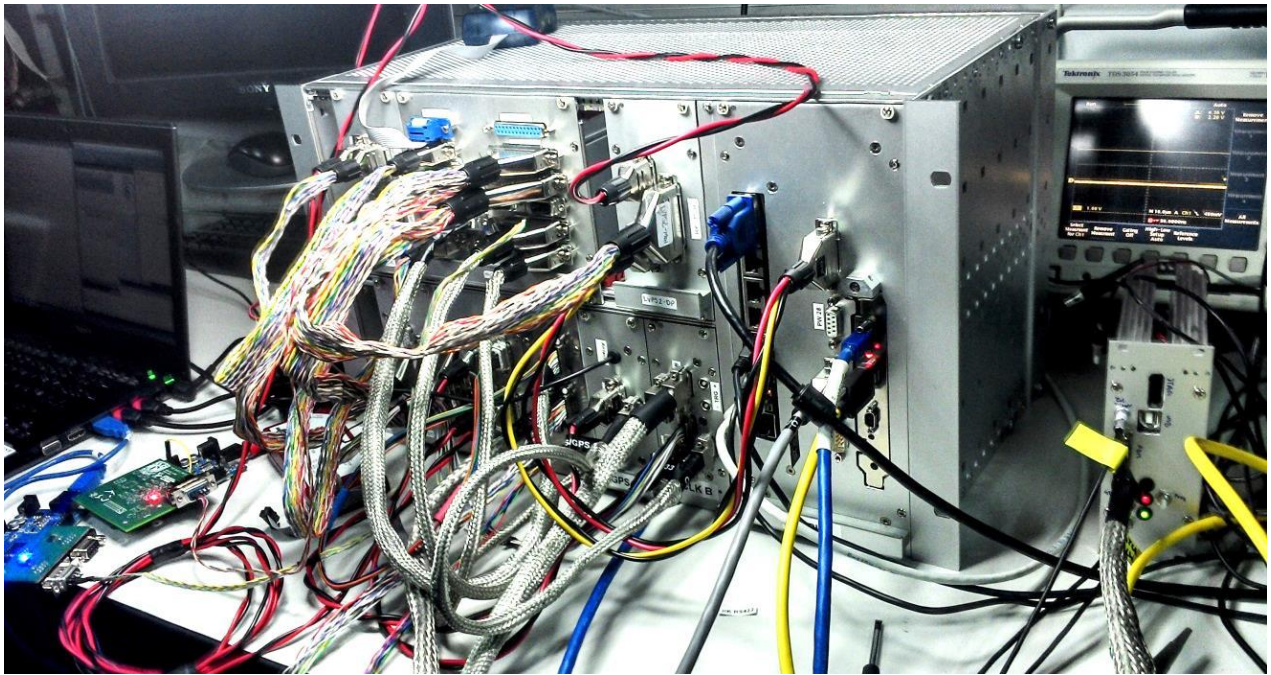


Figura 5.4. Subsistemas durante la segunda integración.

Como resultado de esta integración, el sistema DP cumplió casi en su totalidad con su etapa de desarrollo, lo que permitiría la integración con los demás sistemas que conforman el instrumento (PDMB, HVPS, LENSES y SIREN). Debido a este avance, comenzó la etapa de integración dentro de la góndola, en donde se definió la posición de los sistemas dentro de la estructura, la forma en que estarían fijos dentro de ella y la forma de acceder a éstos desde el exterior. Para esto último, se diseñaría la tapa superior de la góndola, en donde se colocarían conectores para los subsistemas que lo requieran. En el caso del *Housekeeping*, el conector de depuración para SIREN fue colocado, éste sería uno de los puntos de entrada al DP, para verificar el comportamiento del *Housekeeping*. En la figura 5.5 se muestra una vista superior de la góndola.

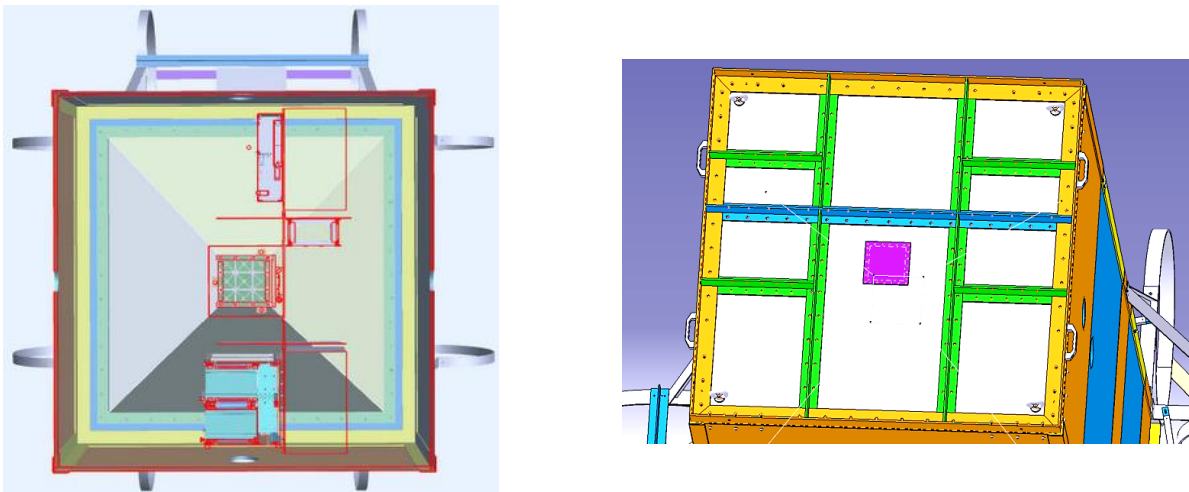


Figura 5.5. Vista superior de la góndola.

Modificaciones y mejoras

Durante esta integración se agregaron nuevas rutinas en la operación del *Housekeeping*, con las cuales se logró un *firmware* más cercano a la versión final. Algunas de las rutinas que se agregaron son las siguientes:

- Envío de la información colectada de los sensores de temperatura colocados en CPU a SIREN.
- Se agregaron rutinas de verificación al momento de encender cada subsistema.
- Si la comunicación inicial con un subsistema falla, se enviaría una alarma de alta prioridad a la CPU.
- Notificaciones a CPU con alarmas de alta prioridad en cada reprogramación de un subsistema.
- Se agregaron rutinas para generar alarmas de baja prioridad, mismas que son enviadas a CPU.
- Se agregaron rutinas para la comunicación con SIREN, las cuales permiten la visualización en tiempo real del comportamiento del *Housekeeping*, además de que permiten efectuar sobre éste un control similar al que realiza CPU.

Conclusiones

Como resultado de esta integración, se logró obtener un *firmware* más completo del *Housekeeping*, ya que el *software* desarrollado para los bancos de prueba permitió un avance más significativo, al momento de realizar las pruebas de integración con los subsistemas reales. En la tabla 5.4 se puede observar un resumen de los resultados obtenidos en la segunda integración.

Tabla 5.4. Resultados de la segunda integración en el DP del *Housekeeping*.

| Sistemas | Conexión Eléctrica | Líneas probadas | Resultado |
|------------|-------------------------|---|----------------------------|
| HVPS | Comunicación LVDS | No | No presente |
| | Compatibilidad de Cable | No | |
| GPS | Comunicación LVDS | Bus SPI | Probado OK |
| | Asignación Cable | Correcta | Probado OK |
| CCB | Comunicación LVDS | Dedicadas a: Alarmas, <i>Soft Reset, hard Reset.</i> | Probado OK |
| | | Bus SPI | |
| | Compatibilidad de Cable | Correcta | Probado OK |
| CLKB | Comunicación LVDS | Dedicadas a: Alarmas, <i>Soft Reset, hard Reset.</i> | Probado OK |
| | | Bus SPI | |
| | Compatibilidad de Cable | Correcta | Probado OK |
| CPU | Estándar RS-422 | Protocolo | Probado OK |
| | Compatibilidad de Cable | Correcta | Probado OK |
| SIREN | Estándar RS-422 | Protocolo / Depuración | No presente |
| | Compatibilidad de Cable | Correcta | Probado OK |
| PDM | Comunicación LVDS | No | No presente |
| | Compatibilidad de Cable | Correcta | |
| LVPS-DP1 | Lectura Analógica | Monitores | Probado OK |
| | HL_CMD | Encendido / Apagado | Probado OK |
| | Contact Closure | Probado Ok | Probado OK |
| | Compatibilidad de Cable | Correcta | Probado OK |
| LVPS-DP2 | Lectura Analógica | Monitores | Probado OK |
| | HL_CMD | Encendido / Apagado | |
| | Contact Closure | Correcta | Probado OK |
| | Compatibilidad de Cable | Correcta | |
| LENSES | - | - | Ok / Faltó |
| HK-MONITOR | Lectura Analógica | Monitores | OK / Falla en monitor 12 V |
| | Compatibilidad de Cable | Correcta | Probado OK |
| LVPS-PDM | Lectura Analógica | Monitores | No utilizado |
| | HL_CMD | Encendido / Apagado | |
| | Contact Closure | Correcta | |
| | Compatibilidad de Cable | Correcta | |

5.2. Integración entre DP y PDM

Esta etapa de la integración fue dedicada a la interacción entre los subsistemas de DP, PDM y HVPS. En la integración entre el DP y PDM se utilizaron la mayoría de sus subsistemas, mientras que para el DP y HVPS sólo se utilizó el *Housekeeping* y una parte de HVPS. Para un mayor avance en la integración se decidió utilizar el respaldo del *Housekeeping* para interactuar con PDM y HVPS, mientras que la versión de vuelo permaneció en el *rack* del DP realizando sus funciones con los otros subsistemas.

Al inicio de esta integración se presentó el DP con modificaciones a su *rack*, estas modificaciones fueron realizadas con el fin de contar con un sistema de disipación de calor para los subsistemas que lo requerían, además de agregar un soporte de sujeción para la góndola. En la figura 5.6 se muestra el diseño final del *rack* del DP.

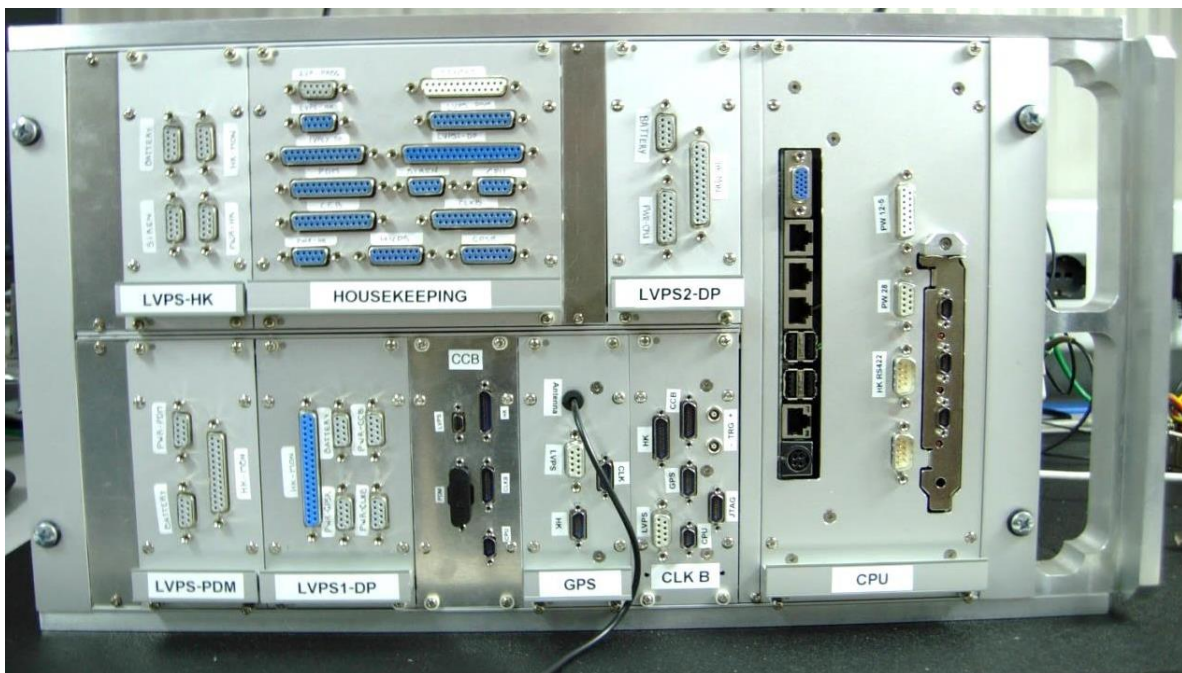


Figura 5.6. Rack del DP completo.

Para la integración entre el *Housekeeping* y PDM se fabricaron cables provisionales que posteriormente serán remplazados; se realizó el intercambio de información entre los dos subsistemas estableciendo lo siguiente:

- Los comandos necesarios para la interacción entre los subsistemas.
- Que el intercambio de datos sería de 32 *bits*, 16 *bits* para el comando/respuesta y 16 *bits* para el CRC.
- Se acordó utilizar el mismo CRC que utiliza el *Housekeeping* con la CPU, por lo que cada parte verificó que el cálculo fuera el mismo.

- Se intercambiaron comandos y se comprobaron los CRC de los datos enviados/recibidos en el *bus* SPI.
- Se verificaron las líneas dedicadas a las alarmas.

En la figura 5.7 se muestra una imagen de la conexión entre PDMB enmarcada en verde y el *Housekeeping* de respaldo con su respectiva LVPS enmarcadas en rojo.

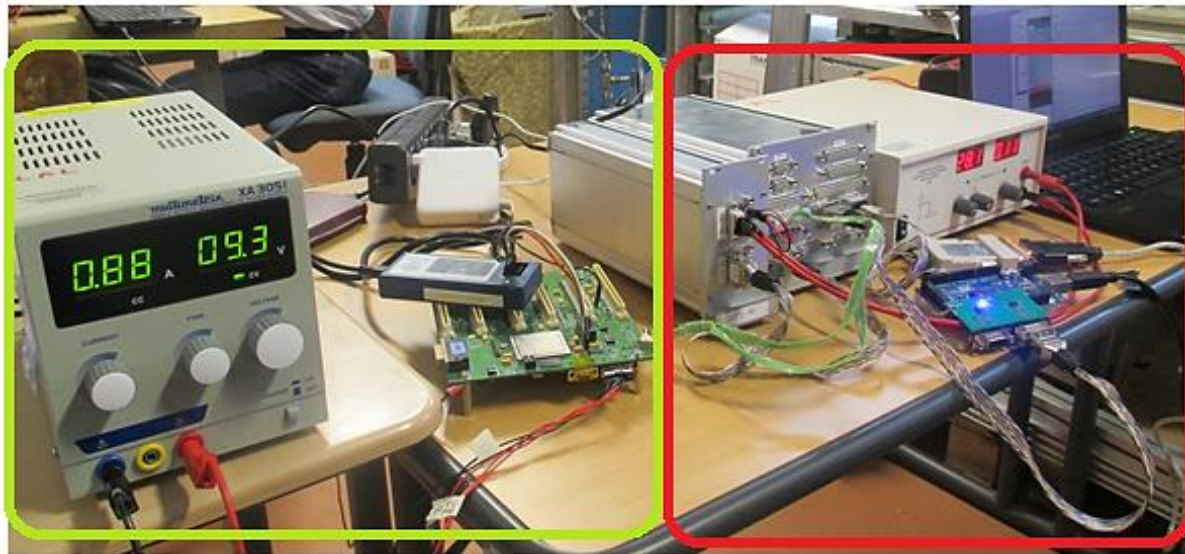


Figura 5.7. Conexión entre PDMB y el *Housekeeping*.

Para la integración entre HVPS y el *Housekeeping* también se ocuparon cables provisionales, además HVPS no fue colocado en su estructura mecánica con el objetivo de verificar señales en caso de que fuera necesario. Para el control de HVPS por parte del *Housekeeping* se verificó lo siguiente:

- Que el cable provisional tuviera la correcta asignación.
- Que los comandos enviados fueran los correctos, es decir, se verificó el correcto control de los dispositivos empleados en HVPS.
- Los valores a las salidas de los DACs que se encuentran en HVPS fueran los correctos.
- El encendido y apagado de los multiplicadores de voltaje empleados en HVPS mediante el *Housekeeping*.
- La detección de la interrupción generada por una falla en HVPS.

En la figura 5.8 se muestra una imagen de la interacción entre los subsistemas, el *Housekeeping* enmarcado en rojo y HVPS enmarcado en verde.

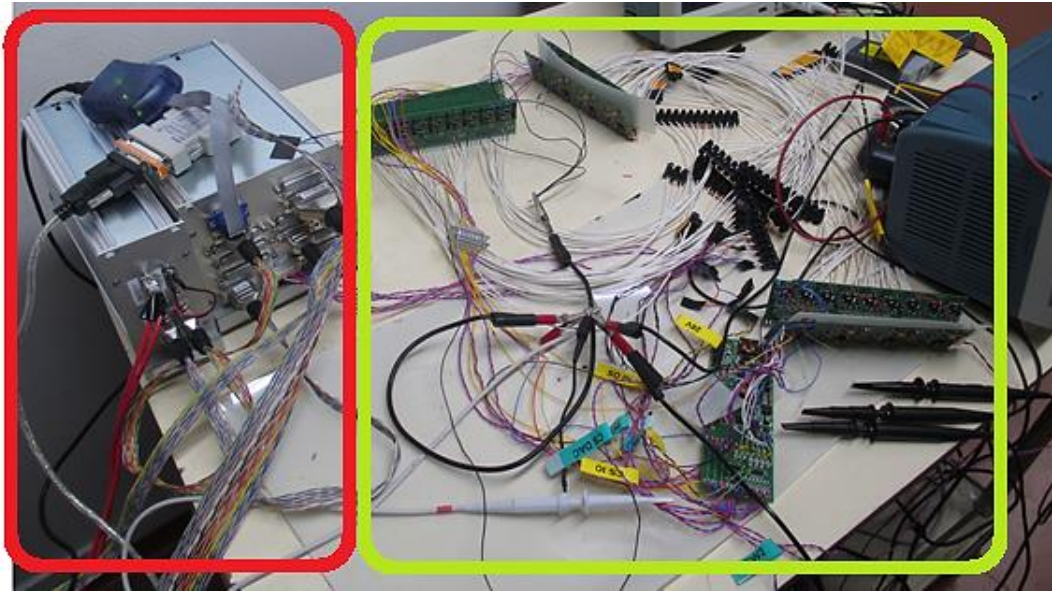


Figura 5.8. Conexión entre el *Housekeeping* y HVPS.

Modificaciones y mejoras

Durante esta integración se agregaron las rutinas para interactuar con PDM, sin embargo queda pendiente la generación de alarmas de alta y de baja prioridad que son transmitidas a la CPU. Con HVPS queda pendiente que en caso de presentarse una interrupción se debe implementar una rutina que detecte en donde se genera la falla y posteriormente la reconfiguración del subsistema.

Además se realizaron las siguientes actividades:

- Se realizaron modificaciones mínimas en el *software* entre la CPU y el *Housekeeping* siempre basadas en los resultados de la segunda integración del DP.
- Se corrigió una falla en el monitor de 12V de la LVPS-HK presentada en la segunda integración del DP.
- Se colocaron los sensores de temperatura que utilizará LENSES.
- Se colocaron de sensores de temperatura en los convertidores DC-DC de la LVPS2-DP.
- Se utilizó un convertidor RS232-422 similar al que será utilizado con SIREN.

Conclusiones

El *Housekeeping* ha logrado interactuar de forma exitosa con dos de los sistemas que son esenciales en la misión, por una parte se ha realizado el control de HVPS y el intercambio de comandos e información con PDM a través de la comunicación SPI. Se deben desarrollar algunas funciones de control y generación de alarmas para cumplir con las actividades descritas en el capítulo 3 en su totalidad.

5.3. Vuelo en plataforma de NASA

Como parte de un convenio entre la Agencia Espacial Mexicana y la NASA, para impulsar el desarrollo de la ciencia y tecnología aeroespacial en México, se construyó la plataforma suborbital mexicana de usos múltiples Pixqui.

Pixqui consiste en una caja metálica equipada con una computadora a bordo, con sistemas de energía, almacenamiento de datos y comunicación, que permitirá probar de manera rápida y a un bajo costo el funcionamiento de equipos y componentes espaciales en vuelos suborbitales. Dichos vuelos se realizan a una distancia cercana a los 40 km de altura sobre la superficie del planeta, en condiciones de vacío y temperatura bastante próximas a las del espacio exterior.

Con el objetivo de validar el *Housekeeping* en las condiciones de operación reales, antes de la primera prueba de vuelo para la misión EUSO-BALLOON, se consiguió integrar el *Housekeeping*, junto con sus bancos de prueba, dentro de la estructura de Pixqui. Como parte de Pixqui, *Housekeeping* estableció comunicación con la computadora a bordo de éste, para establecer un comportamiento similar al que se espera dentro del DP.

5.3.1. Desarrollo de la plataforma suborbital Pixqui

Pixqui es un dispositivo desarrollado por el Instituto de Ciencias Nucleares, el Instituto de Ingeniería y la Facultad de Ingeniería, todos de la UNAM. Entre sus características destaca que puede soportar aceleraciones de hasta 15 veces la gravedad de la superficie terrestre.

El *Housekeeping* fue montado en uno de los gabinetes de Pixqui, realizando las siguientes tareas:

- Comunicación periódica con los bancos de prueba.
- Comunicación con la CPU del sistema Pixqui, intercambiando:
 - Datos de monitoreo de los bancos de prueba.
 - Configuración de umbrales de generación de alarmas, para disparar alarmas en el *Housekeeping*.
 - Envío de alarmas a CPU, para su almacenamiento en la memoria interna de éste.
- Almacenamiento de datos de monitoreo.

Para la comunicación con la CPU del sistema Pixqui, se usó la interfaz RS-422 destinada para CPU del *Housekeeping*, usando el mismo protocolo de comunicación.

En la figura 5.9 se muestra la estructura de Pixqui.

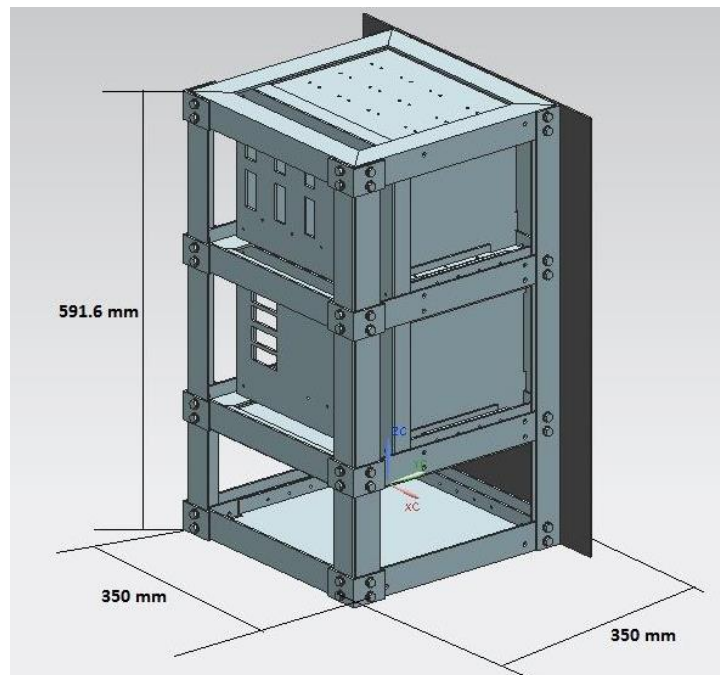


Figura 5.9. Estructura de Pixqui.

Antes de la prueba de vuelo, se montaron todas las tarjetas dentro de la estructura de Pixqui y se realizó una prueba preliminar de comunicación entre las tarjetas. En la figura 5.10 se muestra una sección de la estructura con algunas de las tarjetas montadas.

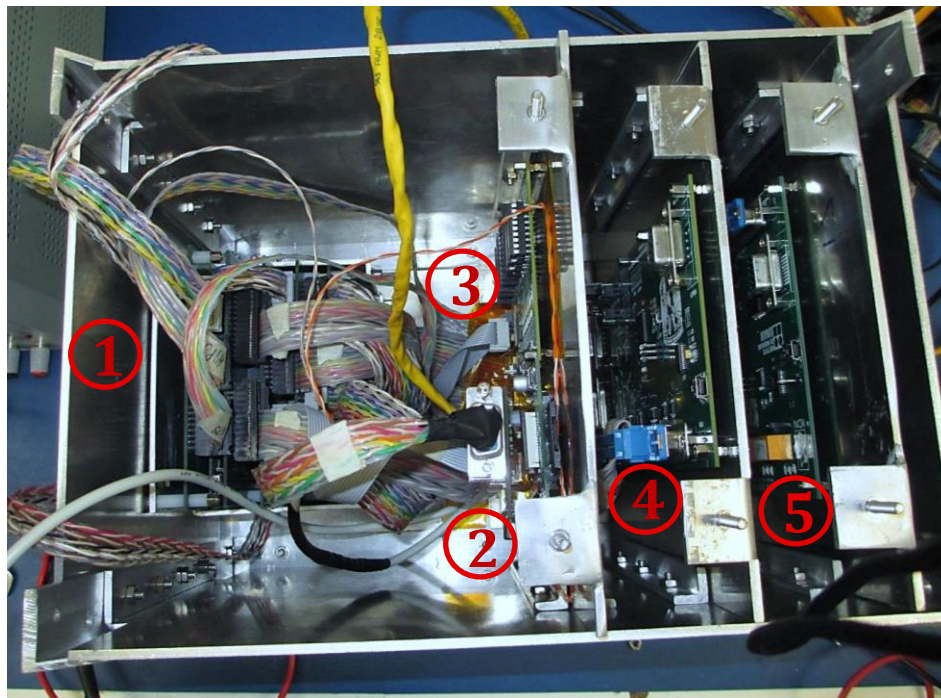


Figura 5.10. Tarjetas montadas en la estructura de Pixqui.

El orden en el que se montaron las tarjetas es el siguiente:

1. *Housekeeping*
2. CPU de Pixqui
3. Banco de pruebas de HVPS
4. Banco de pruebas SPI
5. Banco de pruebas de LVPS

En la figura 5.11 se muestra una prueba de alimentación de las tarjetas montadas en los gabinetes de Pixqui, en el orden siguiente: 1, módulo de alimentación; 2, módulo con *Housekeeping*; 3, Quetzal.

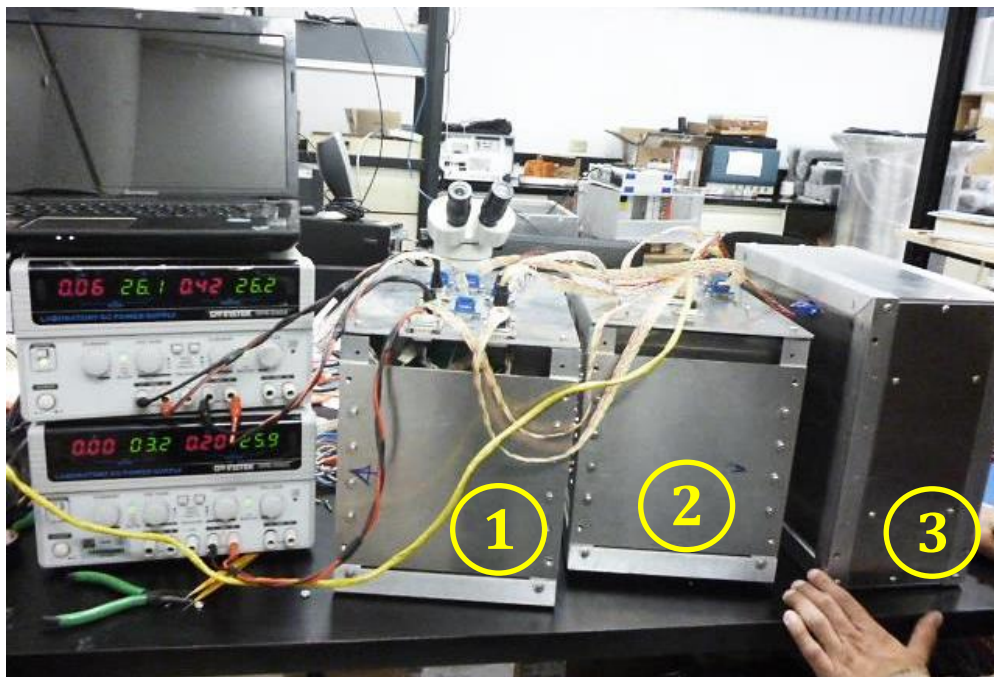


Figura 5.11. Prueba de alimentación.

Nuevamente se realizaron pruebas da vacío y ciclado térmico a las tarjetas montadas dentro de los gabinetes de Pixqui. En la figura 5.12 se muestran los gabinetes colocados en la plancha de la cámara de vacío.

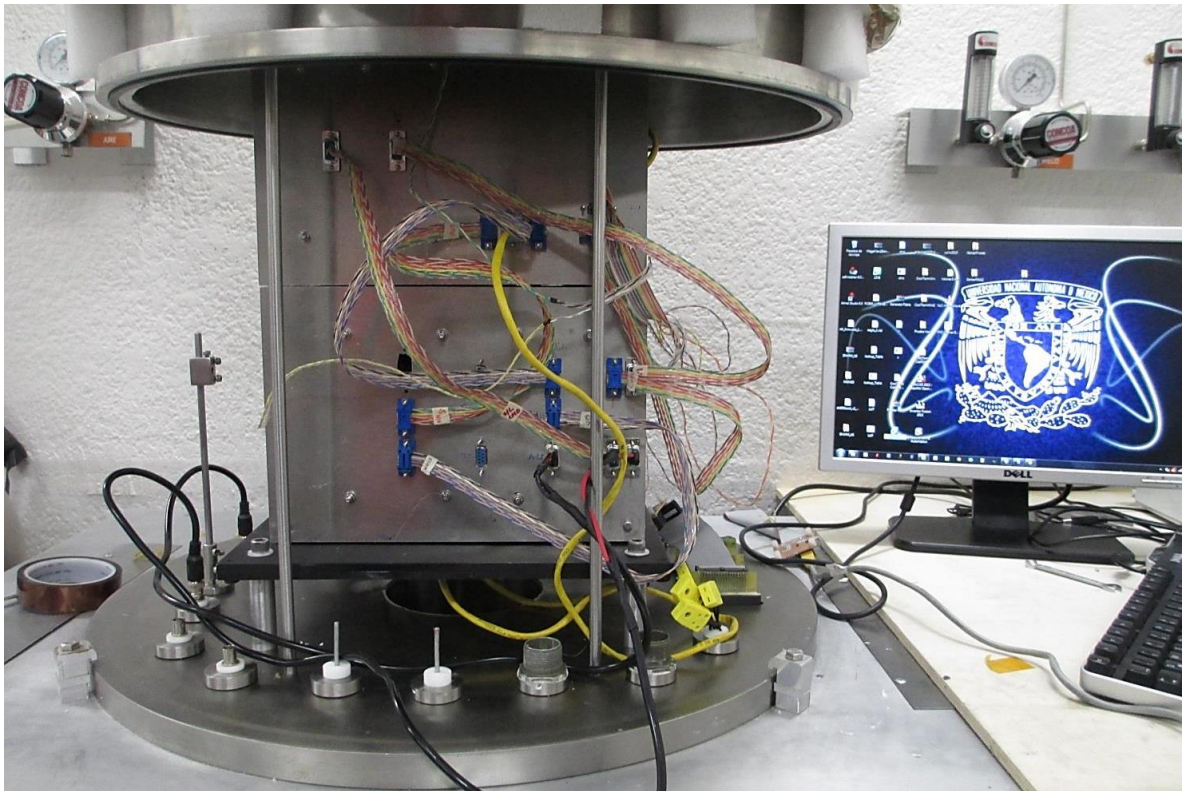


Figura 5.12. Prueba de vacío a los gabinetes de Pixqui.

5.3.2. Pruebas de compatibilidad electromagnética

La compatibilidad electromagnética es la habilidad de un dispositivo electrónico, unidad o sistema, para funcionar de manera correcta dentro de un ambiente electromagnético sin generar interferencia para otros equipos que se encuentren alrededor [28].

Como requerimiento para asegurar un correcto funcionamiento de los sistemas distribuidos dentro de la góndola de la NASA, fue necesario realizar una prueba de compatibilidad electromagnética (EMC por sus siglas en inglés), para garantizar que el sistema Pixqui no afectara el funcionamiento de los demás sistemas dentro de la góndola.

Para evaluar los niveles de EMC de un dispositivo existen diferentes estándares, los cuales establecen los niveles permitidos de EMC de un dispositivo. Éstos son aplicados dependiendo de la aplicación a la que va dedicado el dispositivo en cuestión. En la tabla 5.5 se muestra una lista de algunos estándares para EMC.

Tabla 5.5. Estándares para EMC[29].

| Prueba de EMC | |
|--|--------------------|
| Equipo de tecnología de la información | EN55022 Class A, B |
| Industrial, científico y médico | EN55011 Class A, B |
| Emisión por Armónicas de Corriente | EN61000-3-2 |
| Fluctuaciones de Voltaje | EN61000-3-3 |

La prueba se llevó a cabo para cumplir con el estándar EN55022 clase B, versión 2006 [30], fue realizada para todo el sistema Pixqui, incluido el *Housekeeping* y los bancos de pruebas junto con otros prototipos distribuidos en los compartimentos de Pixqui. Se tomaron las siguientes consideraciones:

- La prueba se llevó cabo en un rango de frecuencias de 30 MHz a 1 GHz.
- En esta prueba se midió el nivel de campo eléctrico que se produce cuando el equipo está en funcionamiento.
- De acuerdo a la clasificación del estándar, el equipo fue clasificado dentro de la clase A.

Los niveles de campo eléctrico son medidos en [dB μ V/m], tomando en cuenta que la distancia establecida por el estándar debe de ser de 10 m.

Se puede apreciar en la tabla 5.6 que el rango de frecuencias al que se realiza la prueba se divide en dos, el primero es de 30 MHz a 230 MHz y el segundo es de 230 MHz a 1 GHz. Los niveles de campo eléctrico son de 30 [dB μ V/m] y de 37 [dB μ V/m] respectivamente. Los demás parámetros no fueron medidos.

Tabla 5.6. Características del estándar [31].

| INFORMATION TECHNOLOGY EQUIPMENT EN55022:2006 EN55024:1998 | | | | | |
|--|---------------------------------------|--------------------|--|------------------------------|-------------------------------------|
| EMISSIONS EN55022 : 2006 | | | | | |
| | Port | Frequency Range | Limits | | Basic Standard |
| Radiated Emissions | Enclosure | 30 MHz - 230 MHz | 30 / 40 dBuV/m (10 m Measurment distance Class B / Class A) | | CISPR 22 |
| | | 230 MHz - 1 GHZ | 37 / 47 dBuV/m | | |
| Conducted Emissions | Telecommunications network ports (CM) | | CURRENT LIMITS | | Note: for products with clk > 9 kHz |
| | | 0.15 MHz - 0.5 MHz | 53 - 43 dBuA QP | 40 - 30 dBuA Av | |
| | | 0.5 MHz - 30 MHz | 43 dBuA QP | 30 dBuA Av | |
| Conducted Emissions | AC Mains Power Ports | 0.15 MHz - 0.5 MHz | 66-56 dBuV QP / 56 - 46 dBuV Average | 79 dBuV QP / 66 dBuV Average | |
| | | 0.5 MHz - 5 MHz | 56 dBuV QP / 46 dBuV Average | 73 dBuV QP / 60 dBuV Average | |
| | | 5 MHz - 30 MHz | 60 dBuV QP / 50 dBuV Average | 73 dBuV QP / 60 dBuV Average | |
| Mains Harmonics | | 0 - 2 kHz | EN61000-3-2 | | |
| Flicker | | | EN61000-3-3 | | |
| Inrush Current | | | EN61000-3-3 | | |

Esta prueba se realizó en una cámara semi-anechoica que se encuentra en la Facultad de Ciencias de la UNAM. En las figuras 5.13 y 5.14 se muestra el interior de la cámara semi-anechoica y la estructura Pixqui, respectivamente.



Figura 5.13. Antenas de la cámara semi-anechoica.

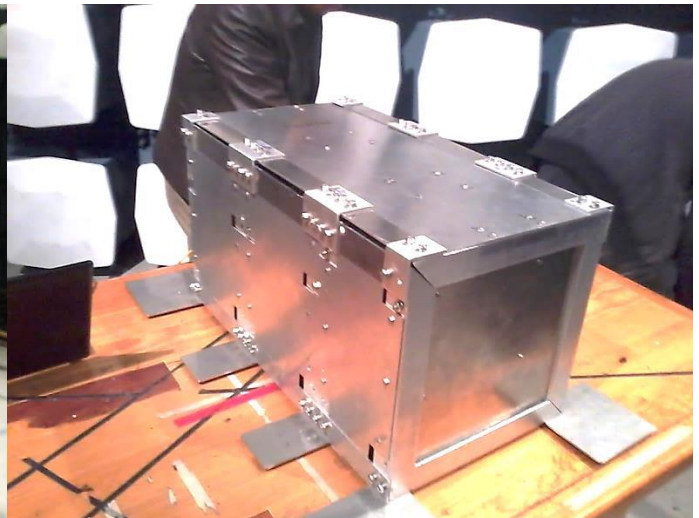


Figura 5.14. Pixqui dentro de la cámara.

Los resultados de la prueba se muestran se muestran en la figura 5.15.

SCAN TABLE: "EN 55022 Field-mod"

| Short Description: | | EN 55022 Field Strength | | | | |
|--------------------|---------|-------------------------|----------|------------|-----------|---------------------|
| Start | Stop | Step | Detector | Meas. Time | IF Bandw. | Transducer |
| 30.0 MHz | 1.0 GHz | 0.5 % | MaxPeak | 10.0 ms | 120 kHz | Biconilog3142 #1762 |

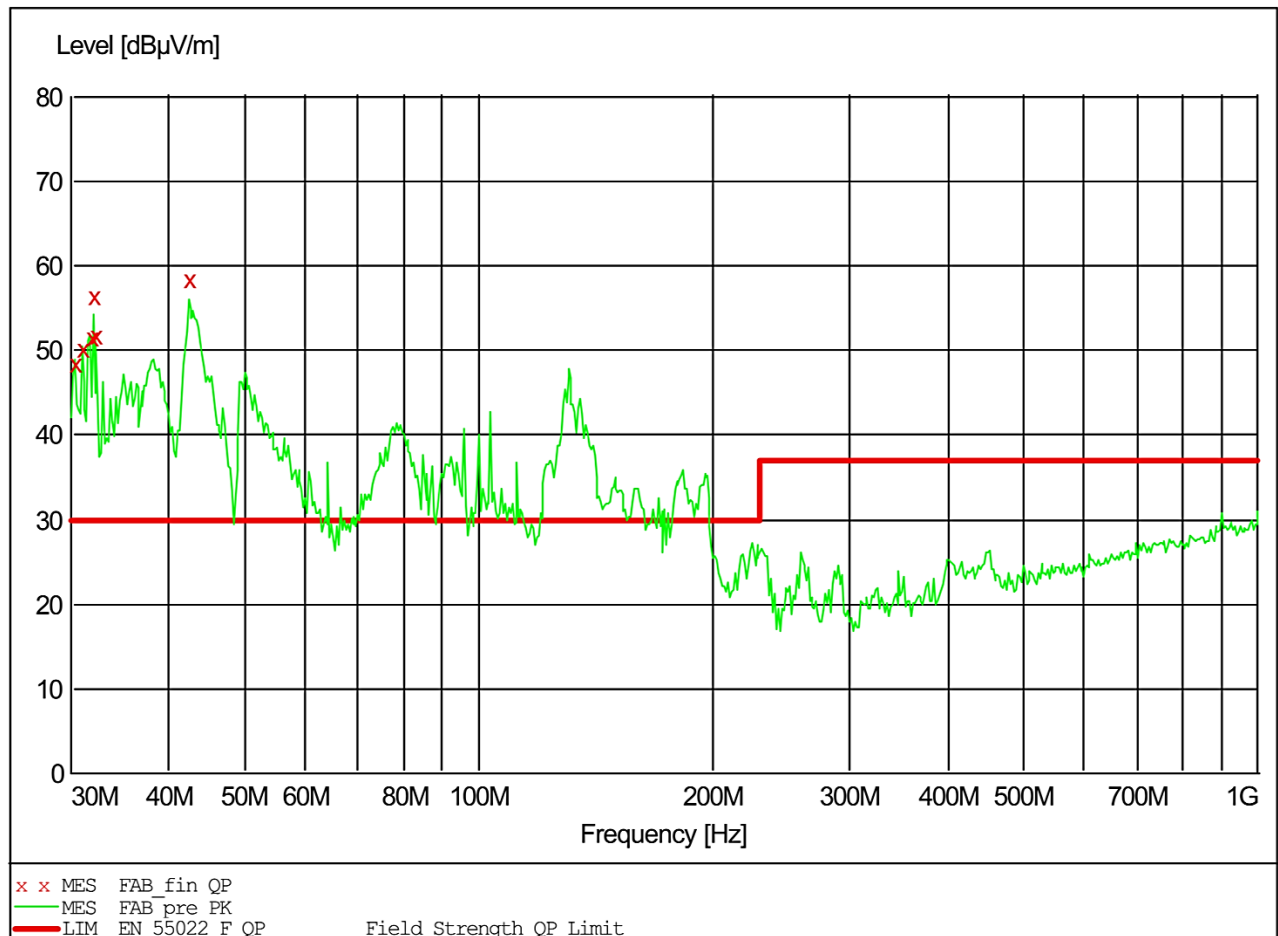


Figura 5.15. Gráfica resultante de las prueba de EMC.

La línea roja de la gráfica indica los niveles máximos de nivel de campo eléctrico que no se deben de rebasar. Se puede percibir que del rango de frecuencias de 30 MHz a 200 MHz, el sistema Pixqui en su totalidad, no cumplió con los requerimientos de este estándar. Para frecuencias altas, de 200 MHz a 1GHz, se puede observar el módulo no sobrepasa el nivel mínimo de campo eléctrico producido.

Debido a los resultados, la prueba no fue aprobada, ya que el sistema Pixqui sobrepasa los niveles mínimos de campo eléctrico propuesto por el estándar. Se debe de tener en cuenta que la estructura mecánica no estaba completamente sellada, posteriormente, antes de la prueba de vuelo se selló completamente la estructura y se realizó nuevamente otra prueba, misma que fue aprobada por el equipo de la NASA para el vuelo.

5.3.3. Vuelo en globo estratosférico

El vuelo de Pixqui se llevó a cabo en *Fort Sumner*, Nuevo México, el día 19 de Agosto de 2013. La adquisición de datos tuvo una duración aproximada de 9.5 horas. Se debe de tener en cuenta que el sistema Pixqui fue encendido una hora antes de iniciar el vuelo, por este motivo el tiempo efectivo de vuelo fue de 8.5 horas aproximadamente.

El sistema Pixqui fue montado en una góndola de la NASA, figura 5.16. El globo estratosférico se elevó aproximadamente 32 km de altura.



Figura 5.16. Góndola donde fue montado Pixqui.

En la figura 5.17 se muestra a la góndola durante el lanzamiento.



Figura 5.17. Lanzamiento del globo de NASA.

En la figura 5.18 se muestra la trayectoria que siguió el globo.



Figura 5.18. Recorrido del globo en el vuelo.

El *Housekeeping* realizó mediciones de temperatura de los componentes críticos con los que contaba el instrumento. Para esto utilizó sus trece termistores, los cuales fueron distribuidos en distintos puntos de la estructura mecánica. Además se utilizaron dispositivos digitales de temperatura que se comunican por medio de SPI.

En la tabla 5.7 se muestra la distribución de los sensores de temperatura.

Tabla 5.7. Distribución de los sensores de temperatura.

| Nombre asignado | Tipo | Ubicación |
|-----------------|---------------|---|
| Termistor 1 | Analógico | DC/DC FPGA |
| Termistor 2 | Analógico | DC/DC CPU |
| Termistor 3 | Analógico | Interior caja 1, frente a LVPS- <i>Housekeeping</i> |
| Termistor 4 | Analógico | Disipador de LVPS de <i>Housekeeping</i> |
| Termistor 5 | Analógico | DC/DC Quetzal |
| Termistor 6 | Analógico | Disipador FPGA |
| Termistor 7 | Analógico | Placa de aluminio LVPS de <i>Housekeeping</i> |
| Termistor 8 | Analógico | Caja 1 detrás de LVPS de FPGA |
| Termistor 9 | Analógico | Microcontrolador de <i>Housekeeping</i> |
| Termistor 10 | Analógico | Regulador de la tarjeta de desarrollo |
| Termistor 11 | Analógico | Microprocesador de CPU |
| Termistor 12 | Analógico | <i>Driver</i> USB de CPU |
| Termistor 13 | Analógico | Parte inferior caja 2 |
| Sensor PCB01 | Digital (SPI) | PCB01 de <i>Housekeeping</i> |
| Temp GPS1 | Digital (SPI) | Banco de pruebas SPI |
| Temp GPS2 | Digital (SPI) | Banco de pruebas SPI |

Toda la información fue registrada en archivos en la memoria *microSD* de la versión de vuelo del *Housekeeping*. El contenido de los archivos fue de 13,331 muestras, en 9.5 horas. Con este último dato y con el tiempo que duró la adquisición de datos, se puede calcular el tiempo que le tomó al *Housekeeping*, realizar una lectura de todos sus parámetros, de la siguiente manera:

$$t_{vuelo} = 9.5 \text{ hrs} = 34,200 \text{ seg}$$

El tiempo para hacer la lectura de todos los parámetros es:

$$t_{1 \text{ ciclo}} = \frac{t_{vuelo}}{\# \text{ muestras}} = \frac{34200 \text{ seg}}{13331 \text{ muestras}} = 2.56 \frac{\text{seg}}{\text{muestra}}$$

Dónde:

$t_{1 \text{ ciclo}}$, es el tiempo para realizar una lectura de todos los parámetros una vez

t_{vuelo} , es el tiempo de vuelo

#muestras, es el número de muestras que fueron almacenadas en la SD por parámetro

Con el resultado de la operación se sabe que el *Housekeeping* tardó 2.56 segundos para realizar un ciclo de lectura de todos sus parámetros.

5.3.4. Resultados

Una vez que finalizó el vuelo y se recuperó el Pixqui, se procedió a graficar los datos adquiridos, para conocer el comportamiento de los parámetros medidos y determinar si el *Housekeeping* operó de manera correcta, junto con los bancos de pruebas con los que interactuó.

En esta tesis solamente se retomaran los datos relacionados con el *Housekeeping*, los cuales se ordenan de la siguiente manera:

- Monitoreo de temperatura
 - Microcontrolador del *Housekeeping*
 - Regulador de voltaje de la tarjeta de desarrollo
 - Parte inferior caja 2 donde se montó el *Housekeeping*
- Sensores de temperatura SPI
 - Banco de pruebas de SPI
 - PCB01 del *Housekeeping*

Monitoreo de temperatura

Al someter el instrumento a condiciones reales de operación, un parámetro de suma importancia es la temperatura del microcontrolador. En la figura 5.19 se muestra el comportamiento de ésta durante el vuelo.

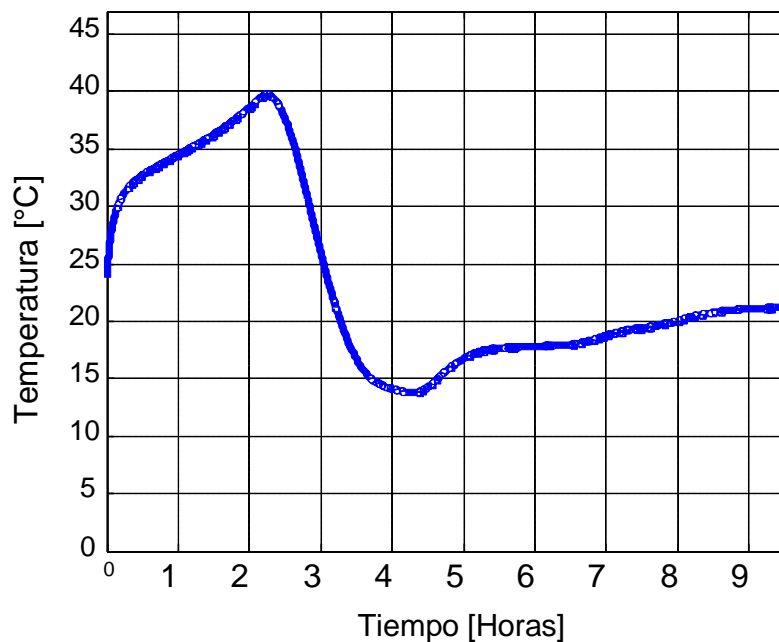


Figura 5.19. Temperatura en el microcontrolador.

En la gráfica se pueden observar los siguientes resultados:

- La temperatura máxima que alcanzó el microcontrolador durante el vuelo fue de 40 °C. Este valor está muy alejado de la temperatura máxima de operación recomendada por el fabricante que es de 85 °C.
- La temperatura mínima que alcanzó el microcontrolador fue cercana a 14 °C aproximadamente, este valor es mucho mayor a la temperatura mínima de operación que el fabricante recomienda que es de -40 °C.

Con los resultados anteriores se puede verificar que el microcontrolador seleccionado puede funcionar sin problemas, sometido a condiciones reales de vuelo.

El componente más crítico es el regulador de voltaje de la tarjeta de desarrollo del microcontrolador, debido a que es el que disipa más calor, como ya se ha mencionado en pruebas anteriores. En la figura 5.20 se muestra el comportamiento del regulador durante el vuelo.

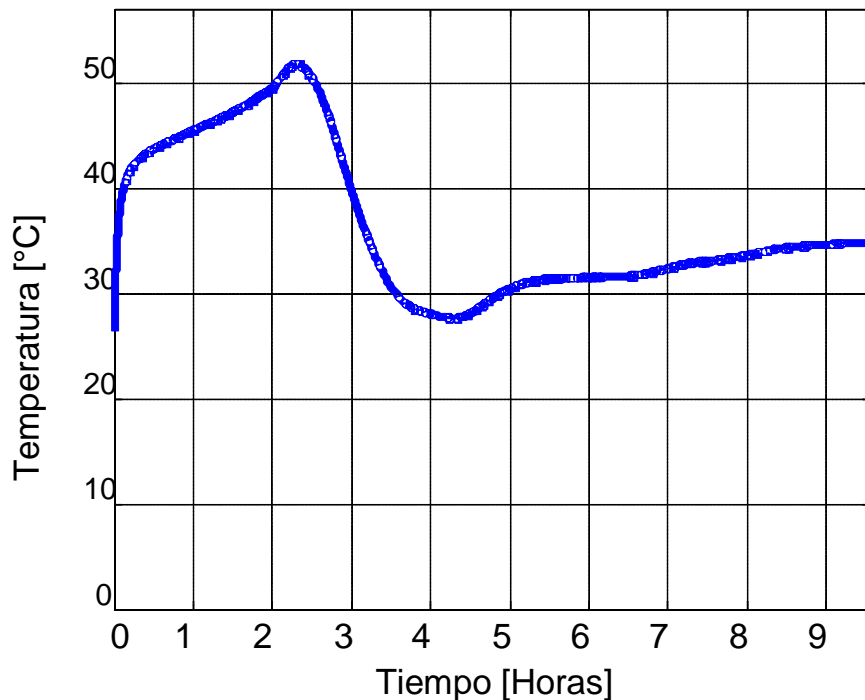


Figura 5.20. Temperatura del regulador de voltaje.

En la gráfica se pueden observar los siguientes resultados:

- La temperatura máxima que alcanzó el regulador de voltaje durante el vuelo fue de 52 °C. Este valor está muy alejado de la temperatura máxima de operación recomendada por el fabricante que es de 125 °C.

- La temperatura mínima que alcanzó el regulador de voltaje fue de 28 °C aproximadamente, este valor es mucho mayor a la temperatura mínima que el fabricante recomienda que es 0°C.

Con estos resultados se puede verificar que no existe ningún problema con este componente, aun siendo el más crítico, ya que durante el vuelo trabajó de manera normal.

Se consideró importante hacer hincapié al comportamiento de la temperatura que se midió en la caja 2 del sistema de Pixqui, en ésta se encontraba operando el *Housekeeping*. En la figura 5.21 se muestra la temperatura en la caja 2 del sistema Pixqui.

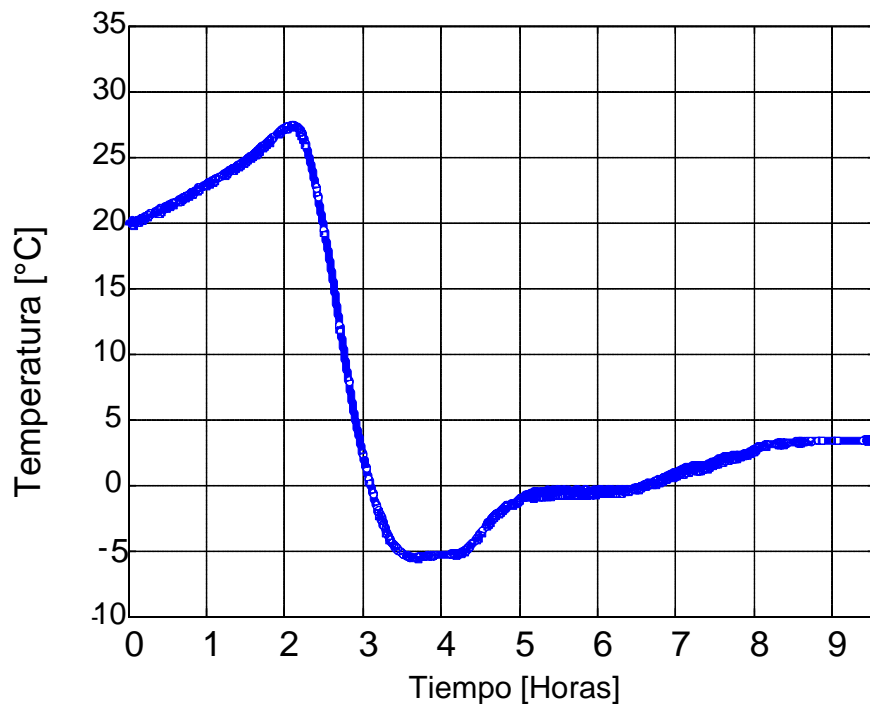


Figura 5.21. Temperatura en la caja 2 del sistema Pixqui.

En la gráfica se pueden observar los siguientes resultados:

- La temperatura máxima que se midió en la caja 2 durante el vuelo fue de 27 °C.
- La temperatura mínima que se midió en la caja 2 de -5 °C aproximadamente.

Este punto se consideró importante, ya que se encontraba en contacto directo con el *Housekeeping*, debido a su ubicación dentro de ésta y por medio de los tornillos con los que el *Housekeeping* se fijó dentro de la caja.

Temperatura SPI

Además de medir la temperatura con sensores analógicos, se hizo uso de sensores de temperaturas digitales SPI. En la figura 5.22 se muestra la temperatura sobre el banco de pruebas SPI.

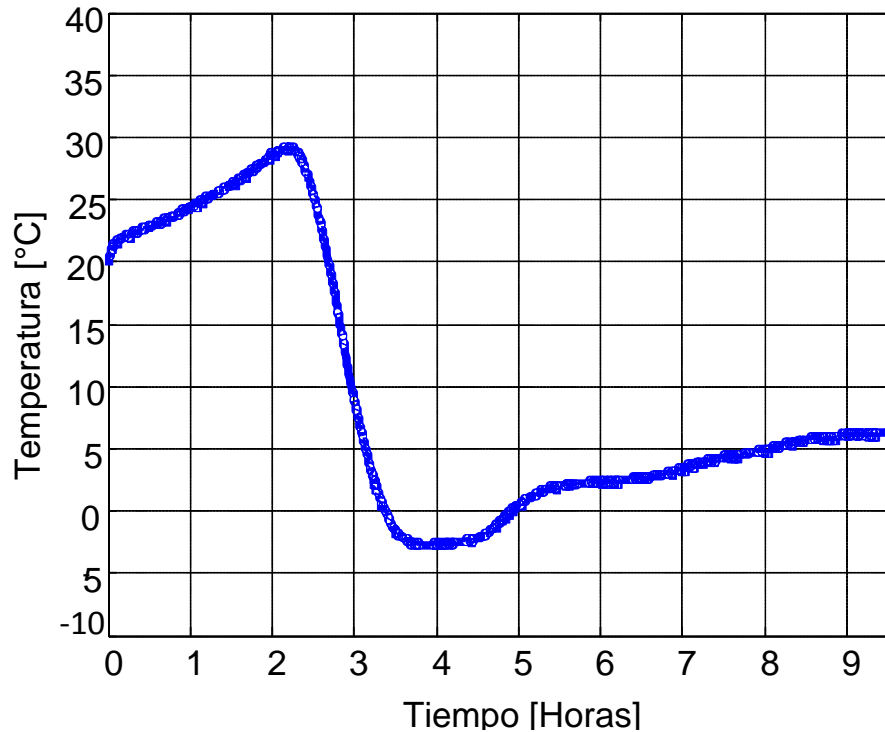


Figura 5.22. Temperatura en el banco de pruebas SPI.

En la gráfica se pueden observar los siguientes resultados:

- La temperatura máxima que se midió en el banco de pruebas SPI durante el vuelo fue de 30 °C.
- La temperatura mínima que se midió en el banco de pruebas SPI fue de -3 °C aproximadamente.

A pesar de la variación de la temperatura en el banco de pruebas de SPI durante el vuelo, éste se mantuvo operando de manera normal, además que la temperatura en éste no se salió del rango de temperatura de operación del microcontrolador.

Además de medir la temperatura en el banco de pruebas SPI, se midió también en la superficie de la PCB01 del *Housekeeping*. La gráfica de la figura 5.23 representa las mediciones que realizó el sensor de temperatura ubicado en la PCB01 del *Housekeeping*.

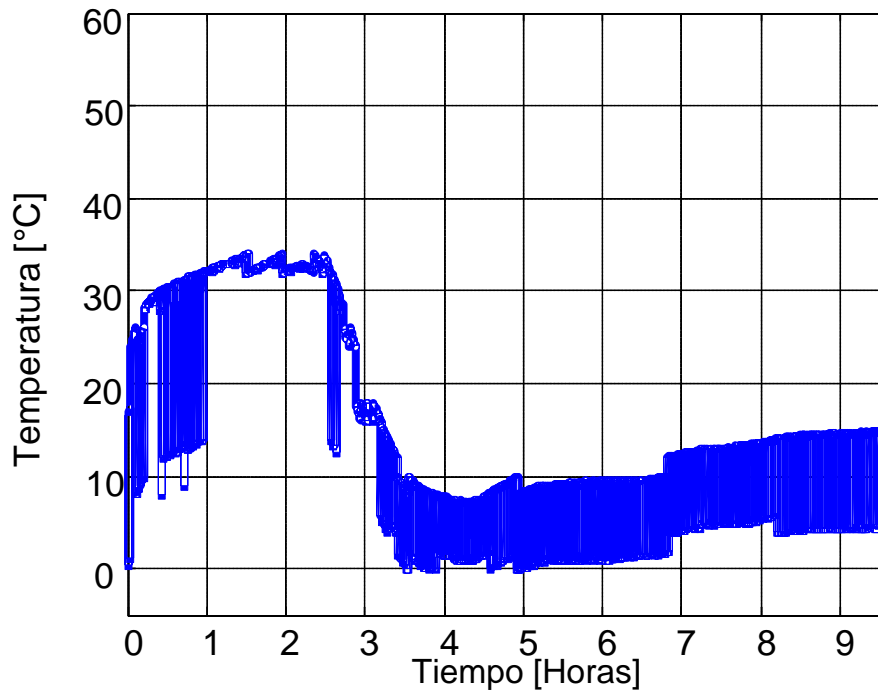


Figura 5.23. Temperatura de la PCB01.

Se puede observar que la temperatura sigue la tendencia de los demás sensores, tanto analógicos como digitales, pero tiene variaciones muy grandes, esto se atribuye a problemas de ruido encontrados en la comunicación.

Se puede apreciar una temperatura máxima de 33 °C aproximadamente, pero no se puede determinar una temperatura mínima en esta gráfica.

Comparación de datos de temperatura con la telemetría de NASA

Gracias al acuerdo logrado con la NASA, se tuvo acceso a la telemetría que utiliza para monitorear las condiciones de vuelo de sus globos, además ésta nos proporcionó un modelo de simulación térmica de toda la góndola del vuelo 642NT, figura 5.24.

La simulación de la figura 5.24 se hizo para observar las temperaturas máximas esperadas durante el vuelo, y así poder decidir si era necesario colocar un recubrimiento a la estructura de Pixqui. La ubicación de Pixqui en la góndola se encuentra detrás de los paneles solares en color rojo de la figura, los cuales muestran un calentamiento de 68 °C a 79 °C siendo las temperaturas máximas encontradas en toda la góndola. Por su parte, se esperaba que la temperatura máxima de Pixqui estuviera entre 21.75 °C a 33.39 °C.

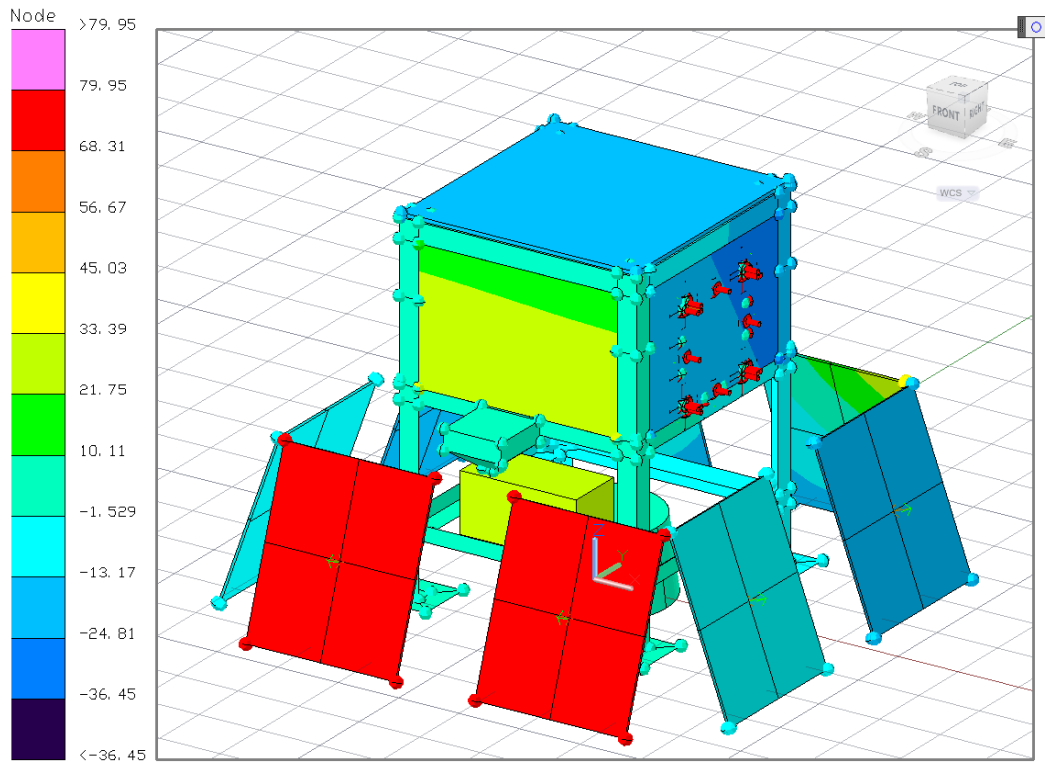


Figura 5.24. Simulación de temperaturas máximas esperadas.

Durante el vuelo la NASA registró la variación de temperatura en una parte cercana a donde Pixqui fue montado, figura 5.25. Estos datos fueron comparados con los datos registrados por el *Housekeeping* de la variación de temperatura en el gabinete de Pixqui.

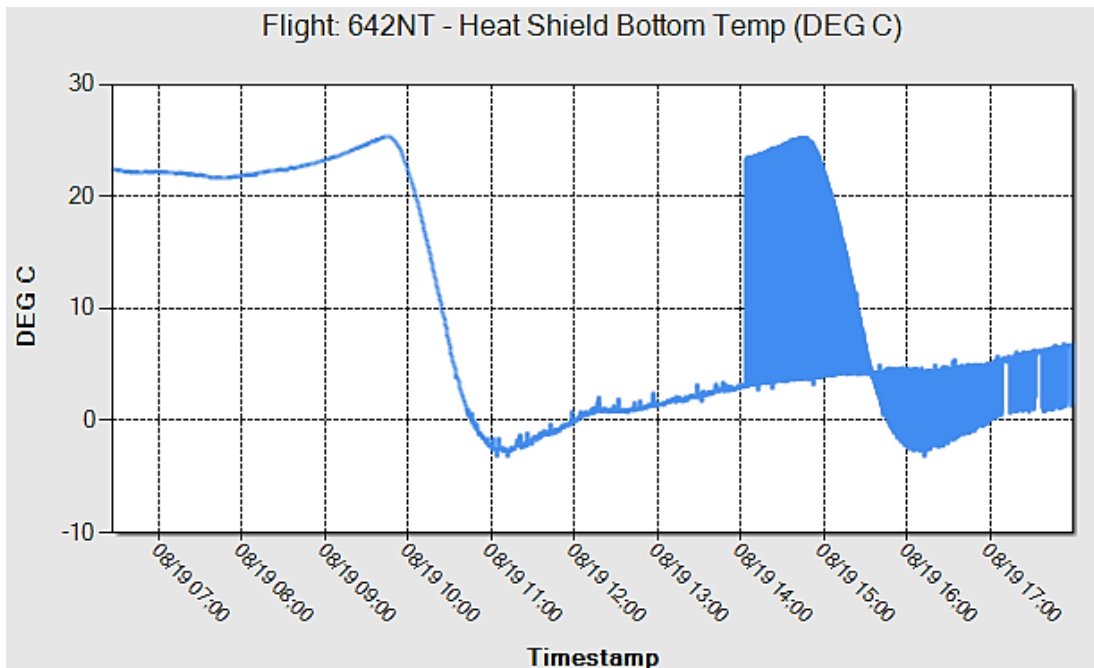


Figura 5.25. Telemetría cortesía de la NASA.

Se puede observar que el comportamiento de la temperatura en los datos registrados por la NASA, siguen el mismo comportamiento que los datos registrados por el *Housekeeping*, además, al comparar el valor máximo de la temperatura medida en la caja 2 del *Housekeeping* de la figura 5.16, que es de 27 °C, se puede notar que éste está dentro del intervalo de valores de temperatura máxima esperada en el modelo de simulación. De esta manera se comprueba que la medición de temperatura realizada durante el vuelo por el *Housekeeping*, se llevó a cabo de manera correcta.

Conclusión

El vuelo concluyó de forma exitosa y sirvió para agregar una prueba de validación más a los equipos entregados para EUSO-BALLOON, se espera que el *Housekeeping* se quede como un sistema permanente en futuros desarrollos para Pixqui, con una versión modificada para brindar servicios a clientes cuya finalidad sea la validación de componentes en condiciones cercanas a las del espacio, de forma accesible y confiable.

6. Resultados y conclusiones

En este capítulo se abordan los resultados obtenidos durante el desarrollo del *Housekeeping*, se presentan las conclusiones generales y se aportan sugerencias necesarias para mejorar el diseño del *Housekeeping* para versiones futuras.

6.1. Resultados logrados

De las metas logradas gracias al desarrollo del subsistema *Housekeeping*, se listan las siguientes:

1. Implementación de las interfaces de comunicación, necesarias para una correcta interacción con los diferentes subsistemas que componen la misión EUSO-BALLOON.
2. Selección adecuada del material necesario para el desarrollo del prototipo, basándose en una adecuada administración de los recursos disponibles, para obtener la mejor relación costo - prestaciones.
3. Diseño y fabricación de los circuitos impresos necesarios para el prototipo, así como el desarrollo de los bancos de pruebas.
4. Caracterización del *Housekeeping* mediante los bancos de prueba, con lo que se obtuvo un *firmware* más completo.
5. Desarrollo e implementación del *software* para cumplir con los requerimientos estipulados en el *Housekeeping*.
6. Montaje del sistema para cumplir los requerimientos de peso, potencia y dimensiones.
7. Validación de la tecnología desarrollada e implementada, que interactúa con cada sistema que conforma el instrumento del EUSO-BALLOON en condiciones muy severas en laboratorio, así como en una prueba de vuelo en condiciones reales.

8. Entrega de varios prototipos a la colaboración internacional, para su uso en la misión principal y en misiones secundarias que lo requieran.
9. Participación y publicación del trabajo realizado en el XXVIII Congreso de instrumentación del SOMI [32].
10. Publicación referente a la colaboración del desarrollo de la plataforma Pixqui en la Gaceta Digital de la Facultad de Ingeniería de la UNAM[33].
11. Publicación referente al desarrollo del Housekeeping en la misión EUSO-BALLOON en la Gaceta Digital de la Facultad de Ingeniería de la UNAM[34].

6.2. Conclusiones

Se mantuvo a la comunidad Mexicana de astro partículas en la frontera de investigación en el área, por medio de la participación en el desarrollo del sistema *Housekeeping*, como parte esencial en la nueva generación de detectores espaciales de rayos cósmicos de altas energías.

Se propició la generación de recursos humanos e infraestructura para la producción de tecnología espacial de alto nivel, mediante la participación en una colaboración internacional con socios de reconocida experiencia en esa área.

Finalmente y de manera más personal, este proyecto nos permitió desarrollar habilidades que complementan la formación de un ingeniero, como la gestión y planeación de proyectos limitados en tiempo, la participación multidisciplinaria que exige un proyecto de colaboración internacional, el uso de procesos de fabricación industrial y el manejo de equipo profesional para realizar pruebas a los prototipos diseñados. Todo esto concluyó en la implementación de un sistema de aplicación embebida que da solución a un problema de ingeniería, en este caso enfocado al campo de instrumentación espacial.

6.3. Trabajo a futuro

Después del desarrollo del primer prototipo del *Housekeeping*, el cual es capaz de operar en un ambiente cercano al espacio, los equipos de trabajo que se unan a la misión del JEM-EUSO deberán desarrollar versiones del *Housekeeping* que sean capaces de operar en condiciones espaciales, las cuales deberán ser utilizadas y caracterizadas en misiones que precedan a la

versión final que será utilizada en la ISS. Con la experiencia adquirida durante la elaboración de este trabajo se hacen algunas recomendaciones en el apartado siguiente.

6.4. Recomendaciones

Durante el desarrollo del *Housekeeping* se encontraron diversos aspectos que es posible mejorar, pero cuya implementación no se pudo finalizar debido a los reducidos tiempos de diseño y desarrollo. Entre estos aspectos se encuentran:

Hardware

- Migrar el diseño basado en un microcontrolador de Atmel a una tecnología endurecida contra radiación. Atmel cuenta con una variedad de dispositivos como memorias, procesadores, FPGAs y ASICs, específicos para este tipo de aplicaciones.
- Utilizar arquitecturas de procesamiento más rápidas, complejas y versátiles, como los FPGAs. Se recomienda utilizar además de los productos de Atmel, productos de Xilinx, ya que son reconocidos en el campo aeroespacial.
- Estudiar el uso de microcontroladores con sistemas de redundancia incluida en *hardware*. Tal es el caso de los microcontroladores de Texas Instruments, Hercules, los cuales son usados comúnmente en la industria de la aviación.
- Para agregar más confiabilidad en la adquisición de señales analógicas, es posible utilizar una configuración con filtros, así como mantener el uso de líneas diferenciales para evitar ruido no deseado.
- Usar *backshells* en todos los conectores, ya que éstos prolongan la vida de los cables, y de ser posible utilizar conectores *Micro-D*, ya que reducen espacio y tienen mejor acoplamiento mecánico que los conectores *Sub-D*.
- Utilizar malla metálica como recubrimiento de los cables, este blindaje reduce las probabilidades de cambios de estado no deseados en las señales digitales, y evita la inducción y emisión de señales electromagnéticas por los cables.
- Diseñar un sistema redundante de almacenamiento, con memorias endurecidas a radiación, para aumentar la confiabilidad del almacenamiento de datos.

- Hacer uso de técnicas de redundancia en *hardware*, como sistemas redundantes modulares tripes, o de auto purga.
- Referente al diseño de las tarjetas de circuito impreso, hacemos las siguientes recomendaciones:
 - Reducir el número de tarjetas de circuito impreso, utilizando técnicas de diseño multicapa, mientras el diseño lo permita, además que capas de planos de tierra ayudarán a evitar emisión electromagnética.
 - Para evitar problemas de *crosstalk* en las líneas de control, se recomienda separar los planos de tierra en circuitos digitales y analógicos. En el *Housekeeping* de este trabajo, estos problemas afectaron principalmente a los circuitos de *High Level Command*. El plano de tierra de alimentación puede presentar los mismos problemas de *crosstalk*.
 - Añadir puntos de prueba en las tarjetas de circuito impreso, con el fin de depurar señales eléctricas de forma sencilla y práctica en los PCBs, permitiendo el uso de osciloscopios o analizadores lógicos en las tarjetas.
 - Configuraciones *fail-safe* en cada terminal LVDS, esto aumentará la confiabilidad de los *buses* de transmisión al mantener un estado fijo en los receptores LVDS, además que se asegura el flujo de corriente en la resistencia terminadora de la línea de transmisión.
 - Mejorar la distribución de señales de reloj en el PCB, éste no debe tener interferencia con otras líneas, y en el caso de compartir el reloj para varios subsistemas, utilizar *buffers* y *drivers* especiales para la transmisión de señales de reloj en LVDS.

Software

- Mediante el uso de dispositivos más versátiles como FPGAs, se puede diseñar un protocolo orientado a *bits*, para una transmisión más eficiente de datos, banderas y alarmas.
- Hacer uso de redundancia en *software*, con un código de detección y corrección de errores en el protocolo de comunicación con cualquier sistema, ya que actualmente, sólo se detectan errores en la comunicación.

- El diseño de las librerías y módulos propios de procesamiento, pueden servir de base para describir *hardware* en dispositivos como los FPGAs, ya que actualmente algunos núcleos de procesadores de Atmel se pueden virtualizar en un FPGA.
- Diseñar un *firmware* que sirva para descargar la información que maneja el *Housekeeping* de forma eficiente y rápida para el sistema encargado de la telemetría.

APÉNDICES

A. Glosario de términos

Cámara semi anecoica. Sala diseñada para absorber las reflexiones producidas por ondas acústicas o electromagnéticas en cualquiera de las superficies que la conforman.

CCB. *Control Cluster Board.* Subsistema del DP, encargado de procesar y clasificar los datos obtenidos por el sistema de fotodetectores de PDM.

CLKB. *Clock Board.* Subsistema del DP, encargado de generar las bases de tiempo para la mayoría de los dispositivos del instrumento EUSO-BALLOON.

COTS. *Commercial Off-The-Shelf.* Una clase de componentes de *hardware* destinados a aplicaciones comerciales

CPU. *Central Processing Unit.* Es la unidad central de procesamiento de un sistema de cómputo.

CRC. *Cyclic Redundancy Code.* Código de detección de errores usado en redes digitales y dispositivos de almacenamiento de datos, usado para detectar cambios accidentales en los datos.

CSV. *Comma-Separated Values.* Formato para representar datos en forma de tabla, en las que las columnas están separadas por comas y las filas por saltos de línea.

DP. *Data Processor.* Sistema que controla toda la electrónica principal del instrumento de EUSO-BALLOON.

DST. *Data Storage.* Sistema de almacenamiento de datos del instrumento EUSO-BALLOON.

EMC. *Electromagnetic Compatibility.* Se refiere a la habilidad de un dispositivo electrónico, unidad o sistema, para funcionar de manera correcta dentro de un ambiente electromagnético, sin generar interferencia para otros equipos que se encuentren alrededor.

EUSO-BALLOON. *Extreme Universe Space Observatory on Balloon.* Versión reducida del telescopio JEM-EUSO, con un sistema mínimo de la electrónica y el sistema de detección de partículas que constituirá el instrumento final.

Fotomultiplicador. Tipo de detector óptico de vacío que responde a niveles muy bajos de luz.

GPS. *Global Positioning System.* Subsistema del DP, que provee el posicionamiento del instrumento de EUSO-BALLOON.

HL_CMD. *High Level Command.* Circuito usado para generar pulsos de encendido/apagado para los relevadores de las LVPS.

HVPS. *High Voltage Power Supply.* Sistema del instrumento EUSO-BALLOON, encargado de generar los voltajes necesarios para el sistema de fotodetectores de PDM.

JEM-EUSO. *Extreme Universe Space Observatory on board the Japanese Experiment Module.* Observatorio de radiación cósmica en la Estación Espacial Internacional, estudiará partículas de muy variada naturaleza.

LENSES. Sistema de lentes del instrumento EUSO-BALLOON.

LVDS. *Low-Voltage Differential Signaling.* Sistema diferencial de transmisión de datos de alta velocidad y de baja potencia.

LVPS. *Low Voltage Power Supply.* Subsistema que proporciona la alimentación para los diferentes subsistemas del instrumento EUSO-BALLOON.

NOYSCA. *New Operational System for the Control of Aerostat.* Estación de control en tierra de la misión EUSO-BALLOON.

NTC. *Negative Temperature Coefficient.* Se refiere a un tipo de termistores cuya resistencia disminuye cuando aumenta la temperatura.

PDM. *Photo Detector Module.* Módulo fotodetector del instrumento EUSO-BALLOON.

PTC. *Positive Temperature Coefficient.* Se refiere a un tipo de termistores cuya resistencia aumenta cuando aumenta la temperatura.

PWM. *Pulse-Width Modulation.* Técnica que modifica el ciclo de trabajo de una señal periódica.

SD. *Secure Digital.* Formato de tarjeta de memoria inventado por Panasonic, usado en dispositivos portátiles como cámaras digitales, teléfonos móviles y computadoras personales.

SIREN. *Network Interface System NOSYCA.* Sistema encargado de comunicar el instrumento EUSO-BALLOON con tierra.

SPI. *Serial Peripheral Interface.* Protocolo de comunicación serie entre un dispositivo maestro y uno o varios esclavos que utilizan cuatro terminales para transmitir información entre ellos.

USART. *Universal Synchronous/Asynchronous Receiver/Transmitter.* Periférico que incorporan muchos microcontroladores para comunicarse con dispositivos que soportan el estándar RS-232.

B. Tabla generada para el cálculo del CRC

| | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|--------|
| 0x0000 | 0x5935 | 0xb26a | 0xeb5f | 0x3de1 | 0x64d4 | 0x8f8b | 0xd6be |
| 0x7bc2 | 0x22f7 | 0xc9a8 | 0x909d | 0x4623 | 0x1f16 | 0xf449 | 0xad7c |
| 0xf784 | 0xaeb1 | 0x45ee | 0x1cdb | 0xca65 | 0x9350 | 0x780f | 0x213a |
| 0x8c46 | 0xd573 | 0x3e2c | 0x6719 | 0xb1a7 | 0xe892 | 0x03cd | 0x5af8 |
| 0xb63d | 0xef08 | 0x0457 | 0x5d62 | 0x8bdc | 0xd2e9 | 0x39b6 | 0x6083 |
| 0xcdff | 0x94ca | 0x7f95 | 0x26a0 | 0xf01e | 0xa92b | 0x4274 | 0x1b41 |
| 0x41b9 | 0x188c | 0xf3d3 | 0xaae6 | 0x7c58 | 0x256d | 0xce32 | 0x9707 |
| 0x3a7b | 0x634e | 0x8811 | 0xd124 | 0x079a | 0x5eaf | 0xb5f0 | 0xecc5 |
| 0x354f | 0x6c7a | 0x8725 | 0xde10 | 0x08ae | 0x519b | 0xbac4 | 0xe3f1 |
| 0x4e8d | 0x17b8 | 0xfce7 | 0xa5d2 | 0x736c | 0x2a59 | 0xc106 | 0x9833 |
| 0xc2cb | 0x9bfe | 0x70a1 | 0x2994 | 0xff2a | 0xa61f | 0x4d40 | 0x1475 |
| 0xb909 | 0xe03c | 0x0b63 | 0x5256 | 0x84e8 | 0xdddd | 0x3682 | 0x6fb7 |
| 0x8372 | 0xda47 | 0x3118 | 0x682d | 0xbe93 | 0xe7a6 | 0x0cf9 | 0x55cc |
| 0xf8b0 | 0xa185 | 0x4ada | 0x13ef | 0xc551 | 0x9c64 | 0x773b | 0x2e0e |
| 0x74f6 | 0x2dc3 | 0xc69c | 0x9fa9 | 0x4917 | 0x1022 | 0xfb7d | 0xa248 |
| 0x0f34 | 0x5601 | 0xbd5e | 0xe46b | 0x32d5 | 0x6be0 | 0x80bf | 0xd98a |
| 0x6a9e | 0x33ab | 0xd8f4 | 0x81c1 | 0x577f | 0x0e4a | 0xe515 | 0xbc20 |
| 0x115c | 0x4869 | 0xa336 | 0xfa03 | 0x2cbd | 0x7588 | 0x9ed7 | 0xc7e2 |
| 0x9d1a | 0xc42f | 0x2f70 | 0x7645 | 0xa0fb | 0xf9ce | 0x1291 | 0x4ba4 |
| 0xe6d8 | 0xbfed | 0x54b2 | 0x0d87 | 0xdb39 | 0x820c | 0x6953 | 0x3066 |
| 0xdca3 | 0x8596 | 0x6ec9 | 0x37fc | 0xe142 | 0xb877 | 0x5328 | 0x0a1d |
| 0xa761 | 0xfe54 | 0x150b | 0x4c3e | 0x9a80 | 0xc3b5 | 0x28ea | 0x71df |
| 0x2b27 | 0x7212 | 0x994d | 0xc078 | 0x16c6 | 0x4ff3 | 0xa4ac | 0xfd99 |
| 0x50e5 | 0x09d0 | 0xe28f | 0xbbba | 0x6d04 | 0x3431 | 0xdf6e | 0x865b |
| 0x5fd1 | 0x06e4 | 0xedbb | 0xb48e | 0x6230 | 0x3b05 | 0xd05a | 0x896f |
| 0x2413 | 0x7d26 | 0x9679 | 0xcf4c | 0x19f2 | 0x40c7 | 0xab98 | 0xf2ad |
| 0xa855 | 0xf160 | 0x1a3f | 0x430a | 0x95b4 | 0xcc81 | 0x27de | 0x7eeb |
| 0xd397 | 0x8aa2 | 0x61fd | 0x38c8 | 0xee76 | 0xb743 | 0x5c1c | 0x0529 |
| 0xe9ec | 0xb0d9 | 0x5b86 | 0x02b3 | 0xd40d | 0x8d38 | 0x6667 | 0x3f52 |
| 0x922e | 0xcb1b | 0x2044 | 0x7971 | 0xafcf | 0xf6fa | 0x1da5 | 0x4490 |
| 0x1e68 | 0x475d | 0xac02 | 0xf537 | 0x2389 | 0x7abc | 0x91e3 | 0xc8d6 |
| 0x65aa | 0x3c9f | 0xd7c0 | 0x8ef5 | 0x584b | 0x017e | 0xea21 | 0xb314 |

C. Constancia de participación en congreso SOMI 2013



CLADET
CENTRO DE CIENCIAS APLICADAS
Y DESARROLLO TECNOLÓGICO



1^{er} CONGRESO IBEROAMERICANO DE INSTRUMENTACIÓN Y CIENCIAS APLICADAS

SOMI XXVIII CONGRESO DE INSTRUMENTACIÓN

28 al 31 de octubre de 2013

El Centro de Ciencias Aplicadas y Desarrollo Tecnológico de la Universidad Nacional Autónoma de México
y la Universidad Autónoma de Campeche

Otorgan la presente

CONSTANCIA

a: Lauro Santiago Cruz, Gustavo Adolfo Medina Tanco, Adrián De la Cruz Martínez,
César Alan López López, Javier Rojas Garcés, Sócrates Alberto Silvarán Guerrero

por haber presentado su trabajo

SISTEMA DE MONITOREO Y ADQUISICIÓN DE DATOS,
HOUSEKEEPING, PARA EL FOTODETECTOR A BORDO DE LA MISIÓN EUSO-BALLOON

Sn. Francisco de Campeche, Campeche, México, 31 de octubre de 2013

Dr. José Manuel Saniger Blesa

PRESIDENTE DEL COMITÉ ORGANIZADOR



Bibliografía y referencias

Bibliografía

Bertoa Manuel F., Troya José M., Vallecillo Antonio, “*Atributos de Calidad para Componentes COTS: Una valoración de la información ofrecida por los vendedores*”, Dpto. Lenguajes y Ciencias de la Computación. Universidad de Málaga.

Coral Calero Muñoz, Moraga de la Rubia María Ángeles, Piattini Velthuis Mario G., “*Calidad de producto y proceso software*”, Editorial Ra-Ma, España, 2010.

Pallas Areny Ramón, “*Sensores y acondicionadores de señales*”, cuarta edición, MARCOMBO S.A., España, 2003.

Robert F. Coughlin, Frederick F. Driscoll, “*Amplificadores operacionales y circuitos integrados lineales*”, Quinta edición, Prentice Hall, México, 1999.

Tocci, Ronal J. y Widmer, Neal S., “*Sistemas digitales*”, Octava edición, Pearson Education, México, 2003.

Universidad de las Américas, Puebla, *Apéndice B: Estándar RS-232*.

Wakerly John F., “*Diseño digital*”, Tercera edición, Pearson Education, México, 2011.

Referencias

- [1] Y. Takahashi, “*The JEM-EUSO Mission*”, [en línea], <<http://arxiv.org/ftp/arxiv/papers/0910/0910.4187.pdf>> [Consulta: 18 Octubre 2013].
- [2] JEM-EUSO Collaboration, “*High Technologies are Supporting JEM-EUSO*”, [en línea], <http://jemeuso.riken.jp/JEM-EUSO_pamphlet_en.pdf> [Consulta: 18 Octubre 2013].
- [3] Moretto, “*Global Description of EUSO-Balloon Instrument*”, [en línea], <<http://hal-univ-diderot.archives-ouvertes.fr/docs/00/84/78/61/PDF/Moretto.pdf>> [Consulta: 18 Octubre 2013].

- [4] Giuseppe Osteria, “*Euso Balloon a pathfinder mission for the JEM-EUSO experiment*”, [en línea], <<http://indico.cern.ch/getFile.py/access?contribId=188&sessionId=7&resId=3&materialId=slides&confId=186337>> [Consulta: 18 Octubre 2013].
- [5] Peter von Ballmoos, “*EUSO-BALLOON Mission Specifications*”, [en línea], <http://euso-balloon.lal.in2p3.fr/IMG/pdf/EUSO-MS-INST-402-IRAP_V1_0.pdf> [Consulta: 18 Octubre 2013].
- [6] A. Z. Marchi, Y. Takahashi, “*The JEM-EUSO optics design*”, [en línea], <<http://galprop.stanford.edu/elibrary/icrc/2011/papers/HE1.4/icrc0852.pdf>> [Consulta: 18 Octubre 2013].
- [7] Y. Takizawa, Y. Takahashi, “*JEM-EUSO optics design and its performance*”, [en línea], <<http://www.srl.utu.fi/AuxDOC/kocharov/ICRC2009/pdf/icrc0792.pdf>> [Consulta: 18 Octubre 2013].
- [8] JEM-EUSO Collaboration, “*Report on the Phase A Study*”, [en línea], <http://www1.na.infn.it/wsubnucl/cosm/JemEuso/pdf/PurpleBook_2010_v.5_MCM.pdf> [Consulta: 18 Octubre 2013].
- [9] Texas instruments, “*Comparing Bus Solutions*” [en línea], Rev. A, Febrero 2004, SLLA067A <http://www.vlsi.ee.upatras.gr/~karagian/comparing_bus.pdf> [Consulta: 10 Octubre 2013].
- [10] Texas instruments, “*Interface circuits for TIA/EIA-232-F*” [en línea], Rev. A, Septiembre 2002, SLLA037A, <<http://www.ti.com/lit/an/slla037a/slla037a.pdf>> [Consulta: 10 Octubre 2013].
- [11] Texas instruments, “*AN-1031 TIA/EIA-422-B Overview*” [en línea], Rev. B, Abril 2013, SNLA044B, <<http://www.ti.com/lit/an/snla044b/snla044b.pdf>> [Consulta: 10 Octubre 2013].
- [12] Texas instruments, “*RS-422 and RS-485 Standards Overview and System Configurations*” [en línea], Rev. A, Mayo 2010, SLLA070A, <<http://www.ti.com/lit/an/slla070a/slla070a.pdf>> [Consulta: 10 Octubre 2013].
- [13] B & B Electronics manufacturing company, “*RS-422 and RS-485 Application Note*” [en línea], Rev. 2506 <<http://www.cpii.com/docs/library/4/485appnote.pdf>> [Consulta: 10 Octubre 2013].

- [14] Texas instruments, “AN-214 Transmission Line Drivers and Receivers for TIA/EIA Standards RS-422 and RS-423” [en línea], Rev. A, Abril 2013, SNLA137A, <<http://www.ti.com/lit/an/snla137a/snla137a.pdf>> [Consulta: 10 Octubre 2013].
- [15] National Semiconductor, “LVDS Owner’s Manual”, 4th edition, 2008, SNLA187, <<http://www.ti.com/lit/ml/snla187/snla187.pdf>> [Consulta: 10 Octubre 2013].
- [16] Texas instruments, “Interface circuits for TIA/EIA-644 (LVDS)” [en línea], Rev. B, Septiembre 2002, SLLA038B, <<http://www.ti.com/lit/an/slla038b/slla038b.pdf>> [Consulta: 10 Octubre 2013].
- [17] Analog Devices, “SPI Interface” [en línea], Rev. 0, AN-1248, <http://www.analog.com/static/imported-files/application_notes/AN-1248.pdf> [Consulta: 10 Octubre 2013].
- [18] Peter von Ballmoos, “Instrument/Gondola Design”, [en línea], <http://euso-balloon.lal.in2p3.fr/IMG/pdf/EUSO-NT-INST-401-IRAP_V1_0.pdf> [Consulta: 18 Octubre 2013].
- [19] A.Ebersoldt, A.Jung, “The Photodetector Module of the JEM-EUSO mission”, [en línea], <<http://euso-balloon.lal.in2p3.fr/IMG/pdf/JEM-EUSO-EA-PDM-2.pdf>> [Consulta: 18 Octubre 2013].
- [20] V. Scotti, G. Osteria, “The Data Processor of the EUSO-Balloon Experiment”, [en línea], <<http://www.bo.infn.it/sminiato/sm13/abstract/posters/v-scotti.pdf>> [Consulta: 18 Octubre 2013].
- [21] Jörg Bayer, “Second level trigger and Cluster Control Board for the JEM-EUSO mission”, [en línea], <<http://www.cbpf.br/~icrc2013/papers/icrc2013-0432.pdf>> [Consulta: 18 Octubre 2013].
- [22] V. Scotti, G. Osteria, “The JEM-EUSO time synchronization system and EUSO BALLOON Data Processor”, [en línea], <http://iopscience.iop.org/1742-6596/409/1/012106/pdf/1742-6596_409_1_012106.pdf> [Consulta: 18 Octubre 2013].
- [23] Jacek Karczmarczyk, “High Voltage System for JEM-EUSO Photomultipliers”, [en línea], <<http://galprop.stanford.edu/elibrary/icrc/2011/papers/HE1.4/icrc0216.pdf>> [Consulta: 18 Octubre 2013].
- [24] Yousuke Hachisu, Hitoshi Ohomori, “Manufacturing of the TA-EUSO and the EUSO-Balloon lenses”, [en línea], <<http://www.cbpf.br/~icrc2013/papers/icrc2013-1040.pdf>> [Consulta: 18 Octubre 2013].

- [25] Joakim Fröberg, “*Software Components and COTS in Software System Development*”.
- [26] ATMEL, “*AVR100: Accessing the EEPROM*” [en línea], Rev. 0932C, Septiembre 2005, <<http://www.atmel.com/Images/doc0932.pdf>> [Consulta: 10 Octubre 2013].
- [27] ATMEL, “*AVR103: Using the EEPROM Programming Modes*” [en línea], Rev. 2578A, Mayo 2005, <<http://www.atmel.com/Images/doc2578.pdf>> [Consulta: 10 Octubre 2013].
- [28] Schurter Electronic Components, “*Introduction to EMC*”, [en línea] <http://www.interferencetechnology.com/wp-content/uploads/2012/06/EMC-for-Dummies_e-News-Final.pdf> [Consulta: 10 Octubre 2013].
- [29] Electromagnetic Service Test, “*EMC Standards*”, [en línea] <<http://www.etsmc.co.uk/services/standards/>> [Consulta: 10 Octubre 2013]
- [30] Internatinal Electrotechnical Commission, “*CISPR22*”, [en línea], 5th edition, 2006, <http://rra.go.kr/pic/download/TCVN%207189-2009_CISPR%2022-2006.pdf> [Consulta: 10 Octubre 2013].
- [31] Electromagnetic Testing Services LTD, “*INFORMATION TECHNOLOGY EQUIPMENT EN55022:2006 EN55024:1998*” [en línea], <<http://www.etsmc.co.uk/wp-content/uploads/EN55022-55024.pdf>> [Consulta: 10 Octubre 2013].
- [32] SANTIAGO CRUZ Lauro, MEDINA TANCO Gustavo Adolfo, DE LA CRUZ MARTÍNEZ Adrián, LÓPEZ LÓPEZ César Alán, ROJAS GARCÉS Javier, SILVARÁN GUERRERO Sócrates Alberto. “*Sistema de monitoreo y adquisición de datos, Housekeeping, para el fotodetector a bordo de la misión EUSO-BALLOON*”. En: 1er. Congreso iberoamericano de instrumentación y ciencias aplicadas. San Francisco de Campeche, Campeche, México 31 de octubre de 2013. SOMI XXVIII Congreso de instrumentación.
- [33] Moreno Iris, “*UNAM traza su ruta la espacio*”, Gaceta Digital FI [en línea], No. 12, Septiembre 2013. pp. 15-17. <http://www.ingenieria.unam.mx/paginas/gaceta/2013/gaceta12_2013.pdf>
- [34] Moreno Iris, “*EUSO-BALLOON inicia pruebas para JEM-EUSO*”, Gaceta Digital FI [en línea], No. 14, Octubre 2013. pp. 8-10. <http://www.ingenieria.unam.mx/paginas/gaceta/2013/gaceta14_2013_print.pdf>