

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica



"Proyecto DAEDALUS"
Prototipo del Sistema de Computación Embebido para Investigación a
Gran Altura
(ACAE)

Informe de Proyecto de Graduación para optar por el título de
Ingeniería Electrónica con el grado académico de Licenciatura

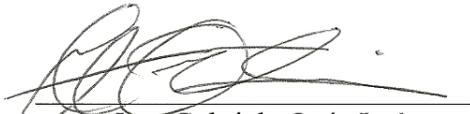
Yoel Wigoda Kaver

I semestre, 2011

INSTITUTO TECNOLOGICO DE COSTA RICA
ESCUELA DE INGENIERIA ELECTRONICA
PROYECTO DE GRADUACIÓN
TRIBUNAL EVALUADOR

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal



Ing. Gabriela Ortiz León

Profesor lector



Ing. Miguel Hernandez Rivera

Profesor Asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica

Cartago, 02 de Mayo del 2011

Declaratoria de Autenticidad

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, 3 de Mayo de 2011.

A handwritten signature in cursive script, appearing to read 'Yoel Wigoda', is written above a horizontal line.

Firma del autor
Yoel Wigoda Kaver
Cédula: 1-1129-20133

Resumen

La Asociación Centroamericana de Aeronáutica y el Espacio (ACAÉ) pretende captar la atención de diferentes entidades desarrollando proyectos de innovación basados en tecnología de punta, con el fin de lograr en el futuro la formación de la Cámara de Industrias Aeroespaciales de Centroamérica.

ACAÉ propuso entonces un primer proyecto, cuyos objetivos giran en torno al cumplimiento de dicha meta, llamado "DAEDALUS", que consiste en el desarrollo de una plataforma de investigación modular de uso científico capaz de realizar mediciones en grandes altitudes.

La tarea de desarrollar el proyecto fue dividida y asignada a varios equipos de ingeniería según el énfasis técnico requerido. Las funciones informáticas realizadas por la plataforma, son efectuadas por el sistema electrónico SCEIGA, cuyo diseño e implementación son abarcados en su totalidad en el presente documento.

El SCEIGA es un sistema de computación móvil de bajo consumo energético especializado en la adquisición de datos en ambientes hostiles. Su uso en la plataforma DAEDALUS permitirá capturar videos y fotografías de la curvatura terrestre, así como la medición de variables atmosféricas y físicas. Tales mediciones podrán ser observadas en tiempo real desde cualquier parte del mundo accediendo a Internet, ya que el sistema es capaz de publicarlas automáticamente a través de la red social Twitter.

El diseño del sistema toma en cuenta las limitaciones físicas que amenazan el funcionamiento adecuado de los dispositivos electrónicos en ambientes hostiles, como las bajas temperaturas, la baja presión atmosférica y la interferencia electromagnética.

Con base en el diseño efectuado, se construyó un prototipo funcional del SCEIGA, utilizando tarjetas de desarrollo, el cual se sometió a rigurosas pruebas de verificación en torno al cumplimiento del objetivo general del proyecto.

Por último, se hizo una serie de recomendaciones técnicas que podrían ser tomadas en cuenta en futuras etapas del proyecto en cuestión u otros proyectos de su misma índole.

PALABRAS CLAVES: Plataforma de investigación, ambientes hostiles, bajo consumo energético, Twitter.

Summary

The Central American Association of Aeronautics and Space (ACAE) aims to capture the attention of various entities through innovative projects based on high-end technology, in order to achieve in the future the formation of the Chamber of Aerospace Industries in Central America.

ACAE then proposed a first project, whose objectives revolve around the fulfillment of this goal called "DAEDALUS", which consist in the development of a modular platform design for scientific research that can take physical measurements at high altitudes.

The development tasks of the project were divided and assigned to various engineering teams based on technical emphasis. The computing functions done by platform, are performed by the electronic system aboard called SCEIGA, whose design and implementation are fully covered in this document.

The SCEIGA is a low-power mobile computing system that specializes in data acquisition in hostile environments. Its use in the DAEDALUS platform will allow video and photo capture of the earth curvature and also to capture atmospheric measurements. Such measurements can be observed in real time from anywhere in the world through Internet access, because the system is able to publish its set of measurements automatically through the social network Twitter.

The system's design foresaw the physical limitations that threaten the proper functioning of electronic devices in hostile environments such as low temperatures, low atmospheric pressure and electromagnetic interference.

Based on the design, a functional SCEIGA prototype was built using development boards. The prototype was fully tested in order to verify that the design objectives were successfully accomplished.

Finally, a series of technical recommendations are presented. This recommendations could be taken in account for development in future stages of this project or in similar projects.

KEYWORDS: Research Platform, hostile environments, low power consumption, Twitter

*A mis padres:
gracias por su apoyo incondicional
y constante motivación*

Agradecimiento

Les agradezco profundamente:

_A mis padres, que me ayudaron a culminar mis estudios universitarios y que siempre me han motivado a cumplir mis metas.

_A mis hermanos, cuyas constantes palabras de aliento me llenaron de fuerza en los momentos más difíciles.

_A mis compañeros de trabajo de laboratorio Diego Ramos y José William Sánchez, por tenderme una mano de ayuda en los momentos en que la necesité.

_A mi profesor asesor, Ing. Miguel Hernández Rivera, por su orientación en el desarrollo de este trabajo.

_Por último le agradezco a Stephanie, quien me brindó todo su apoyo y comprensión durante mis estudios y se conformó sólo con mi fortaleza a cambio.



***"Uno no
debe pensar
que hay cosas
inalcanzables."***

Eurípides

ÍNDICE GENERAL

CAPÍTULO 1. Una visión general de la industria aeroespacial en Centroamérica	15
1.1 La industria aeroespacial y los países en vías de desarrollo	16
1.2 La Asociación Centroamericana de Aeronáutica y Espacio	16
1.3 El primer proyecto de ACAE	17
CAPÍTULO 2. Marco Teórico	18
2.1 Open Software y Open Hardware	19
2.2 Tarjetas de evaluación	20
2.3 BeagleBoard	20
2.4 Arduino	21
2.5 Buses de comunicación	22
2.6 Sistema de Posicionamiento Global (GPS)	24
CAPÍTULO 3. El proyecto Daedalus	26
3.1 Aspectos generales	27
3.2 Plan y objetivos para el desarrollo del SCEIGA	28
CAPÍTULO 4. SCEIGA: Sistema de Computación Embebido para Investigación a Gran Altura	33
4.1 Sistema de Adquisición de Datos (SAD)	34
4.1.1 SAD (AN)	35
4.1.1.1 Selección del computador embebido	35
4.1.1.2 Instalación del sistema operativo	36
4.1.1.2.1 Configuración de la terminal de comunicación	36
4.1.1.2.2 Formateo de la memoria SD	38
4.1.1.2.3 Definición de los parámetros de arranque	40
4.1.1.3 Selección e instalación del programa de captura	42
4.1.1.4 Parámetros de captura	43
4.1.1.5 Automatización de las funciones efectuadas	43
4.1.2 SAD (BN)	44
4.1.2.1 Uso del Bus I ² C	45
4.1.2.2 Acelerómetro	47
4.1.2.2.1 Principio de operación	48
4.1.2.2.2 Conexión eléctrica	49
4.1.2.2.3 Programación	50

4.1.2.3	Sensor de luz _____	52
4.1.2.3.1	Principio de operación _____	52
4.1.2.3.2	Conexión eléctrica _____	52
4.1.2.3.3	Programación _____	53
4.1.2.4	Termómetros _____	54
4.1.2.4.1	Principio de operación _____	55
4.1.2.4.2	Conexión eléctrica _____	55
4.1.2.4.3	Programación _____	56
4.1.2.5	Compás _____	58
4.1.2.5.1	Principio de operación _____	58
4.1.2.5.2	Conexión eléctrica _____	58
4.1.2.5.3	Programación _____	59
4.1.2.6	Presión atmosférica _____	61
4.1.2.6.1	Principio de operación _____	61
4.1.2.6.2	Conexión eléctrica _____	62
4.1.2.6.3	Programación _____	63
4.1.2.7	Sensor de humedad _____	67
4.1.2.7.1	Principio de operación _____	68
4.1.2.7.2	Conexión eléctrica _____	68
4.1.2.7.3	Programación _____	68
4.1.2.8	Almacenamiento de los datos en memorias tipo SD _____	71
4.1.2.8.1	Conjunto de instrucciones _____	71
4.1.2.8.2	Conexión eléctrica _____	72
4.1.2.8.3	Programación _____	72
4.2	Sistema de Recuperación y Rastreo (SRR) _____	76
4.2.1	GPS (coordenadas, altitud, velocidad) _____	77
4.2.1.1	Conexión eléctrica _____	78
4.2.1.2	Programación _____	78
4.2.2	Transmisor/Receptor _____	84
4.2.2.1	Principio de operación _____	86
4.2.2.2	Conexión eléctrica _____	87
4.2.2.3	Programación _____	87
4.2.3	Ethernet Shield _____	91
4.2.3.1	Principio de operación _____	92

4.2.3.2	Conexión eléctrica	93
4.2.3.3	Programación	94
4.3	Sistema de Corte de Emergencia (SCE)	101
4.4	Sistema de Alimentación de Potencia (SAP)	103
<i>CAPÍTULO 5. Ensamblaje del prototipo, resultados experimentales y análisis</i>		105
5.1	Interconexión de bloques del SCEIGA	106
5.2	Programación del prototipo	110
5.3	Resultados experimentales y análisis	113
<i>CAPÍTULO 6. Conclusiones y recomendaciones</i>		119
6.1	Conclusiones	120
6.1.1	Conclusiones propias del desarrollo del SCEIGA	84
6.1.2	Conclusiones de ACAE sobre el proyecto DAEDALUS	121
6.2	Recomendaciones	122
<i>Apéndice A.1. Glosario, abreviaturas y simbología</i>		127
<i>Apéndice A.2. Ruedas de prensa</i>		131
<i>Anexo B.1. Hoja de información del proyecto</i>		137

ÍNDICE DE FIGURAS

<i>Figura 2.1 Tarjeta de evaluación Beagleboard Xm</i>	20
<i>Figura 2.2 Tarjeta de evaluación Arduino Duemilanove</i>	22
<i>Figura 2.3 Ejemplo de uso del bus I²C, un micro-controlador maestro y tres dispositivos periféricos</i>	23
<i>Figura 2.4 Ejemplo de la interconexión de dispositivos mediante el uso del bus SPI</i>	23
<i>Figura 3.1 Distribución general del proyecto DAEDALUS</i>	28
<i>Figura 4.1 Diagrama de bloques del Sistema de Computación Embebido de Investigación a Gran Altura</i>	34
<i>Figura 4.2 Diagrama de bloques del Sistema de Adquisición de Datos (SAD)</i>	35
<i>Figura 4.3 Interfaz de usuario del programa Minicom (pag1)</i>	37
<i>Figura 4.4 Interfaz de usuario del programa Minicom (pag2)</i>	37
<i>Figura 4.5 Interfaz de usuario del programa fdisk</i>	38
<i>Figura 4.6 Secuencia de arranque de la arquitectura OMAP3</i>	41
<i>Figura 4.7 Método escribirEn() utilizado en el proceso de escritura del bus I²C</i>	46
<i>Figura 4.8 Método leerDe() utilizado en el proceso de lectura del bus I²C</i>	47
<i>Figura 4.9 Acelerómetro ADXL345</i>	47
<i>Figura 4.10 Conexión eléctrica básica del ADXL345</i>	49
<i>Figura 4.11 Método SetupAccel() utilizado en el proceso de inicialización del ADXL345</i>	50
<i>Figura 4.12 Método LeeAccel() utilizado en el proceso de lectura del ADXL345</i>	51
<i>Figura 4.13 Sensor de luz TEMENT6000</i>	52
<i>Figura 4.14 Conexión eléctrica del TEMENT6000</i>	53
<i>Figura 4.15 Método LeeLuz() utilizado en el proceso de lectura del TEMENT6000</i>	54
<i>Figura 4.16 Sensor de temperatura TMP102</i>	54
<i>Figura 4.17 Método LeeTemp() utilizado en el proceso de lectura del TMP102</i>	57
<i>Figura 4.18 Compás digital HMC6352</i>	58
<i>Figura 4.19 Método utilizado para obtener lecturas del compás HMC6352</i>	60
<i>Figura 4.20 Sensor de presión barométrica SCP1000</i>	61
<i>Figura 4.21 Conexión eléctrica básica del sensor SCP1000</i>	62
<i>Figura 4.22 Algoritmo utilizado para la escritura SPI</i>	64
<i>Figura 4.23 Algoritmo utilizado para la escritura SPI</i>	65
<i>Figura 4.24 Algoritmo utilizado para la lectura del sensor</i>	66
<i>Figura 4.25 Sensor de humedad HIH-4030</i>	67
<i>Figura 4.26 Curva de tensión versus humedad del sensor HIH-3030</i>	67
<i>Figura 4.27 Conexión eléctrica del sensor de humedad HIH-4030</i>	68

<i>Figura 4.28 Método utilizado para la lectura del sensor HIH-4030</i>	70
<i>Figura 4.29 Módulo SD Openlog</i>	71
<i>Figura 4.30 Método utilizado para la creación de archivos con formato .CSV</i>	74
<i>Figura 4.31 Método utilizado para grabar en los archivos creados</i>	75
<i>Figura 4.32 Diagrama de bloques del Sistema de Recuperación y Rastreo (SRR), parte A</i>	76
<i>Figura 4.33 Diagrama de bloques del Sistema de Recuperación y Rastreo (SRR), parte B</i>	77
<i>Figura 4.34 Módulo GPS LS20031</i>	77
<i>Figura 4.35 Método de adquisición y decodificación de las sentencias NMEA</i>	80
<i>Figura 4.36 Método utilizado para el manejo de números con punto flotante</i>	81
<i>Figura 4.37 Algoritmo de impresión de las coordenadas del sistema</i>	82
<i>Figura 4.38 Métodos utilizados para la impresión de la velocidad y altitud</i>	82
<i>Figura 4.39 Método utilizado para la impresión de la hora</i>	83
<i>Figura 4.40 Método utilizado para la impresión de la fecha</i>	84
<i>Figura 4.41 Módulo Xtend 900</i>	84
<i>Figura 4.42 Antena de monopolo acoplada a los transmisores</i>	85
<i>Figura 4.43 Topología de red punto a punto</i>	86
<i>Figura 4.44 Topología de red puerto a puerto</i>	86
<i>Figura 4.45 Esquema de conexión de los módulos Xtend, según la topología de red punto a punto</i>	87
<i>Figura 4.46 Software de configuración de los módulos Xtend (X-CTU)</i>	88
<i>Figura 4.47 Terminal de configuración (X-CTU)</i>	89
<i>Figura 4.48 Secuencia de configuración de los módulos Xtend</i>	90
<i>Figura 4.49 Controlador Ethernet W5100 empotrado en un escudo para Arduino</i>	91
<i>Figura 4.50 Flujo de la transmisión de datos</i>	91
<i>Figura 4.51 Diagrama de bloques del controlador W5100</i>	93
<i>Figura 4.52 Diagrama de flujo general para el controlador de Ethernet</i>	95
<i>Figura 4.53 Diagrama de flujo de la rutina de configuración inicial del controlador Ethernet</i>	97
<i>Figura 4.54 Diagrama de flujo de la rutina cicloPrincipal() para el controlador de Ethernet</i>	99
<i>Figura 4.55 Diagrama de flujo de la rutina Post() para el controlador de Ethernet</i>	100
<i>Figura 4.56 Diagrama de flujo de la rutina Deco() para el controlador de Ethernet</i>	101
<i>Figura 4.57 Diagrama de bloques del Sistema de Corte de Emergencia (SCE)</i>	102
<i>Figura 4.58 Diagrama de bloques del Sistema de Corte de Emergencia (SCE)</i>	103
<i>Figura 4.59 Curva de descarga del arreglo de baterías</i>	104
<i>Figura 5.1 Entorno de programación Arduino SDK</i>	107
<i>Figura 5.2 Diagrama de conexión de bloques del SCEIGA-PLATAFORMA</i>	107

<i>Figura 5.3 Diagrama de conexión de bloques del SCEIGA-ESTACIÓN TERRESTRE</i>	<i>109</i>
<i>Figura 5.4 Diagrama de flujo de la rutina de configuración inicial del bloque principal</i>	<i>111</i>
<i>Figura 5.5 Diagrama de flujo del ciclo principal que corre en el SCEIGA</i>	<i>112</i>
<i>Figura 5.6 Fotografía capturada con el prototipo SCEIGA-PLATAFORMA</i>	<i>113</i>
<i>Figura 5.7 Perfil de Twitter del prototipo SCEIGA</i>	<i>115</i>
<i>Figura 5.8 Formato de presentación de la ubicación de la sonda</i>	<i>116</i>
<i>Figura 6.1 Imagen de Guanacaste captada a 20 km de altitud</i>	<i>122</i>

ÍNDICE DE TABLAS

<i>Tabla 1 Lista de requerimientos técnicos de los sensores en el SCEIGA</i>	<i>30</i>
<i>Tabla 2 Parámetros de operación del acelerómetro ADXL345</i>	<i>48</i>
<i>Tabla 3 Parámetros de operación del sensor de luz TEMT6000</i>	<i>52</i>
<i>Tabla 4 Parámetros de operación de los sensores de temperatura TMP102</i>	<i>55</i>
<i>Tabla 5 Direcciones de dispositivo posibles para el TMP102</i>	<i>55</i>
<i>Tabla 6 Parámetros de operación del compás HMC6352</i>	<i>58</i>
<i>Tabla 7 Parámetros de operación del sensor de humedad HIH-4030</i>	<i>67</i>
<i>Tabla 8 Propiedades físicas del alambre de nicromo</i>	<i>102</i>
<i>Tabla 9 Formato de presentación de los datos capturados mediante el prototipo SCEIGA-PLATAFORMA</i>	<i>114</i>
<i>Tabla 10 Cálculo de la incertidumbre de los sensores en el SCEIGA-PLATAFORMA</i>	<i>115</i>
<i>Tabla 11 Medición experimental del tiempo de corte de una cuerda de fibra sintética</i>	<i>117</i>
<i>Tabla 12 Tiempo máximo de operación de los módulos pertenecientes al SCEIGA</i>	<i>118</i>

CAPÍTULO 1

*Una visión general de la industria aeroespacial
en Centroamérica*

1.1 La industria aeroespacial y los países en vías de desarrollo

A nivel mundial, se considera que solo los países altamente desarrollados tecnológicamente cuentan con la capacidad de tener una industria aeroespacial, particularmente debido a la creencia de que solo es posible incursionar en este mercado lanzando satélites y estaciones espaciales que requieren billones de dólares de inversión y que están fuera del alcance de los países en vías de desarrollo, incluyendo a las naciones centroamericanas. Sin embargo, frecuentemente se olvidan los hechos históricos que facultaron a los países que actualmente cuentan con una industria aeroespacial próspera para alcanzar dicho desarrollo.

Rusia (anteriormente Unión Soviética) fue el primer país en incursionar en la industria aeroespacial con su satélite Sputnik I en el año 1950, cuya única función era transmitir pitidos que podían ser escuchados en la Tierra con el fin de revisar si el satélite no había colisionado con algún meteorito. Aunque la información otorgada por el satélite no tenía ningún uso útil, el hecho de que fuera posible colocar un satélite en el espacio desató una competencia tecnológica entre varias potencias económicas de la época. Esta carrera eventualmente evolucionó en una industria multimillonaria que ha aportado avances tecnológicos de gran importancia para los seres humanos.

1.2 La Asociación Centroamericana de Aeronáutica y Espacio

La primera asociación que trataba temas aeroespaciales en Centroamérica fue conformada hace ya más de 20 años por un grupo de estudiantes universitarios provenientes de las Escuelas de Ingeniería y Física de la Universidad de Costa Rica, bajo el nombre de la Asociación Costarricense para la Investigación y Difusión Espacial (ACIDE). Esta asociación pretendía difundir e incentivar el avance tecnológico del campo espacial en Costa Rica, sin embargo, dicha entidad nunca prosperó, en parte por su estricta dependencia del gobierno y de las políticas de desarrollo en Ciencia y Tecnología vigentes en la época. Aunque ACIDE nunca alcanzó su objetivo, sirvió de motivación para que un grupo de profesionales relacionados con el paradigma de desarrollo tecnológico que se está gestando actualmente en la región, retomara la idea de asociarse y elevara la propuesta para abarcar a todo Centroamérica. Lo anterior dio paso a la formación de la Asociación Centroamericana

de Aeronáutica y el Espacio (ACAE) en el año 2009, bajo el liderazgo del Ing. Ronald Chang Díaz, director ejecutivo de Ad Astra Rocket Company, empresa precursora del desarrollo de la industria aeroespacial en Centroamérica. ACAE presenta un rango de acción extendido, a diferencia de ACIDE, pues involucra también a diferentes autores estratégicos de la sociedad a parte de los gobiernos (empresas públicas y privadas, universidades, etc.).

ACAE pretende difundir, a través de proyectos de innovación que capten el interés de las diferentes entidades gubernamentales y privadas, la importancia de la industria aeroespacial para el desarrollo de los países y que lo anterior lleve a la formación de alianzas estratégicas y conglomerados empresariales que eventualmente forjen la Cámara de Industrias Aeroespaciales de la región.

1.3 El primer proyecto de ACAE

ACAE tiene una misión a largo plazo que consiste en la congregación de representantes de la industria tecnológica en Centroamérica y de los respectivos gobiernos con el fin común de generar tecnología aeroespacial de punta y también generar el respectivo mercado para comercializar dicha tecnología.

Para captar la atención de las diferentes entidades, ACAE propuso desarrollar un primer proyecto aeroespacial en conjunto con el Ministerio de Relaciones Exteriores de Costa Rica y el SICA (Sistema de Integración Centroamericana). Este primer proyecto debía cumplir con los siguientes requerimientos:

- Debía ser realizable en corto plazo.
- Debía despertar el interés general de las empresas y profesionales por la industria aeroespacial y por ACAE.
- Debía ser fácil de promocionar y tener amplia cobertura mediática.
- Debía servir para motivar a los estudiantes a colaborar con ACAE y a especializarse en carreras del área de la ingeniería.

Con base en estos requerimientos, se planteó el primer proyecto de ACAE: el "proyecto Daedalus".

CAPÍTULO 2

MARCO TEÓRICO

A continuación se presentan algunos conceptos, herramientas e instrumentos estrechamente vinculados con el desarrollo del proyecto, que deberán ser entendidos por el lector con anterioridad con el fin de facilitar su comprensión posterior del resto del documento.

2.1 Open Software y Open Hardware

El *Open Software* representa una de las historias de éxito más significativas del siglo 20 y 21. Está basado en la idea de que cualquier persona, de manera gratuita, puede utilizar, modificar, distribuir y mejorar el software.

El concepto de “*software libre*” fue propuesto en 1976 por Jim Warren, quien dijo: "Cuando el *software* es gratis o muy barato es más fácil pagar por él que duplicarlo, de esa manera no será robado".

Luego, en 1985, el término fue acuñado por Richard Stallman con el lanzamiento de la licencia *GNU*, donde hacía un llamado para la creación de “la comunidad de fuente abierta” (*Open Source*). La idea consiste en que, si el código para un programa determinado es abierto y modificable por cualquier persona, las mejoras al código serán constantes pues cualquier programador podrá aportar su experiencia y conocimiento para mejorar el funcionamiento del programa existente, en vez de tener que escribir su propio código para crear un programa nuevo y similar, como es el caso del *software* comercial.

Actualmente, la idea propuesta por Stallman es altamente aceptada a nivel mundial y, gracias a ella, se han creado algunos de los programas y sistemas operativos más utilizados a nivel mundial, por ejemplo, el sistema operativo Linux y las herramientas de programación *GNU tools*.

Desde ya hace algunos años, se está introduciendo lentamente un concepto novedoso conocido como *Open Hardware*, cuya filosofía está estrechamente vinculada con el *Open Software*. Consiste en aplicar los conceptos *Open Source* para el diseño de *hardware* electrónico, es decir, se debe hacer pública toda la información referente a su diseño, información como los esquemáticos, los materiales, los costos y las plantillas del circuito impreso.

2.2 Tarjetas de evaluación

Una tarjeta de evaluación es una herramienta que se utiliza para el desarrollo de aplicaciones con sistemas empujados. Consiste en un circuito impreso con uno o varios microprocesadores e integrados periféricos que permiten al usuario desarrollar aplicaciones rápidamente sin tener que dedicar mucho tiempo a la elección y ensamblaje del *hardware*. Cuando se termina el diseño de la aplicación y se verifica que funcione adecuadamente en la tarjeta de evaluación, es fácil llevar el diseño a la fase de manufactura, ya que generalmente los fabricantes de dichas tarjetas proporcionan los esquemáticos del *hardware* y las plantillas del circuito impreso correspondiente para ser utilizados en tales fines.

2.3 BeagleBoard

La BeagleBoard es una tarjeta de evaluación de bajo costo y tamaño reducido basada en la tecnología de SoC (*System on Chip*) OMAP3 de Texas Instruments. Incluye un procesador ARM Cortex A8, así como un procesador especializado en el procesamiento digital de señales, lo que hace a esta tarjeta apta para el desarrollo de aplicaciones que requieran alto poder computacional como, por ejemplo, el procesamiento de video y audio.

La BeagleBoard utiliza el mismo sistema de procesamiento que muchos teléfonos móviles inteligentes y computadores portátiles, tiene suficiente capacidad para correr una distribución completa del sistema operativo Linux y para manejar video de alta definición.

La BeagleBoard pertenece a la denominación de *Open Hardware*, es una tarjeta mantenida por una amplia comunidad de usuarios y todos sus documentos de diseño pueden ser descargados de manera gratuita.



Figura 2.1 Tarjeta de evaluación BeagleBoard Xm

2.3.1 Angstrom

Angstrom es una distribución del sistema operativo Linux para dispositivos embebidos. Incluye el núcleo (*Kernel*), un sistema básico de archivos (*File System*), herramientas básicas y un manejador de paquetes de fuente abierta para instalar programas desde repositorios de código. Está optimizado para procesadores de bajo consumo energético como lo es el procesador Cortex A8 contenido en la BeagleBoard. Esta distribución proporciona un sistema base para los desarrolladores que es fácilmente adaptable para aplicaciones particulares.

Esta distribución de Linux, fue portada para la BeagleBoard por sus desarrolladores, Graeme 'Xora' Gregory, Koen Kooi y Philip 'Crofton' Balister. Se puede descargar libremente desde el sitio <http://www.angstrom-distribution.org/>.

2.4 Arduino

Arduino es una plataforma de computación orientada al desarrollo de aplicaciones con un alto grado de interacción con el medio físico. Es una plataforma *Open Source* basada en micro-controladores que operan bajo el *firmware* Arduino. Los fabricantes suplen un entorno de programación de alto nivel para escribir y programar las tarjetas basado en el lenguaje *Processing multimedia programming environment* que facilita el desarrollo de aplicaciones basadas en sus productos.

2.4.1 Hardware Arduino

Las tarjetas de evaluación Arduino están compuestas por micro-controladores Atmel AVR de 8 *bits* y algunos componentes complementarios que facilitan su programación y comunicación con otros bloques de *hardware*. Un detalle importante de dichas tarjetas es que cuentan con conectores para todas las entradas y salidas de sus micro-controladores, lo que permite expandir fácilmente las capacidades de la tarjeta mediante módulos intercambiables, ya sea para manejo de *Ethernet*, *Bluetooth*, motores, etc.



Figura 2.2 Tarjeta de evaluación Arduino Duemilanove

Las tarjetas cuentan con un chip FTDI FT232 que facilita la comunicación serial con los computadores modernos y también permite alimentar la tarjeta, a través de la conexión, con un puerto USB.

Los micro-controladores son pre-programados con un *bootloader* que simplifica el proceso de grabado en la memoria *Flash* del chip. El *bootloader* permite subir, por el puerto serial, directamente nuevos códigos al micro-controlador, esto evita la necesidad de utilizar programadores externos.

2.5 Buses de comunicación

2.5.1 Bus I²C

I²C (*Inter-integrated Circuit*) es un bus estándar de comunicaciones diseñado por Philips cuyo objetivo principal es facilitar la comunicación entre los componentes dentro de un mismo circuito integrado. La versión 1.0 fue lanzada en el año 1992 y ha sido mejorada en múltiples ocasiones. La última versión, la 2.1, fue lanzada en el año 2000.

Este bus tiene prestaciones que promueven su uso en aplicaciones industriales con micro-controladores y periféricos, principalmente porque solo requiere el uso de dos líneas para transmitir información: la línea de datos SDA (*Serial Data*) y la línea de reloj SCL (*Serial Clock*). El protocolo está basado en un arreglo maestro-esclavo: el maestro inicia la comunicación y proporciona la señal de reloj. Cada dispositivo conectado en el bus tiene una dirección única que permite su selección por parte del maestro. También es importante destacar que este protocolo es multi-maestro, es decir, un dispositivo conectado en el bus I²C, puede funcionar tanto como esclavo o maestro dependiendo de los requerimientos.

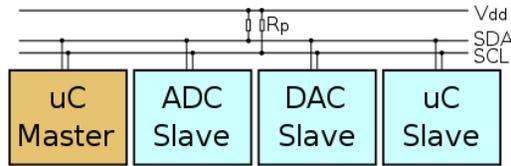


Figura 2.3 Ejemplo de uso del bus I²C , un micro-controlador maestro y tres dispositivos periféricos

2.5.2 Bus SPI

SPI (*Serial Peripheral Interface*) es un bus estándar de comunicaciones utilizado principalmente para transmitir información entre componentes dentro de un mismo circuito impreso.

El bus requiere 4 señales para operar: MOSI (datos entrantes), MISO (datos salientes), SCLK (señal de reloj) y la señal CS (selección de chip) que permite seleccionar entre varios periféricos en el mismo bus.

SPI es un protocolo de comunicación sincrónica, o sea, existe un dispositivo maestro que gobierna la comunicación con el esclavo a través de la señal de reloj.

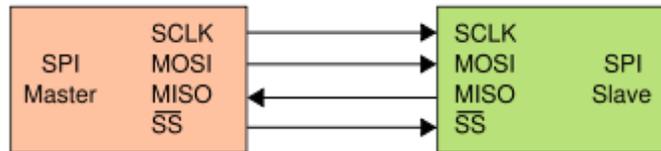


Figura 2.4 Ejemplo de la interconexión de dispositivos mediante el uso del bus SPI

Como la señal de reloj es compartida entre el dispositivo maestro y los esclavos, las tramas de datos no están limitadas a bloques de 8 *bits* como en otros protocolos, sino que depende del dato transmitido y de la configuración de los componentes.

El bus SPI presenta algunas ventajas sobre otros protocolos de comunicación. Es más rápido y más eficiente en el consumo de energía que el bus I²C, permite la comunicación *Full Duplex* y los circuitos son relativamente sencillos. Sin embargo, el uso de este protocolo se dificulta cuando aumenta el número de componentes por comunicar, ya que la única lógica de selección de componentes es a través de la señal CS.

2.5.3 Bus Universal de Transmisión Serial (UART)

UART es el nombre que recibe el módulo de *hardware* encargado de controlar la transmisión y recepción asincrónica de datos en los micro-controladores modernos. Se

conoce como un bus asincrónico porque no hay una señal de reloj compartida entre el emisor y el receptor. El proceso de sincronización se realiza para cada dato transmitido mediante el uso de *bits* con funciones especiales conocidos como "*bits* de inicio y parada".

El receptor no puede predecir con exactitud cuándo recibirá un mensaje, depende completamente de los *bits* de señalización. El receptor y el emisor deben ser configurados para operar en la misma tasa de transmisión de datos. Entre más rápida sea la tasa de transmisión, más alta se vuelve la tasa de errores debido a fallas en el proceso de sincronización. Por esta razón, se suele incluir un *bit* de paridad en la trama de datos, para poder comprobar la integridad de la información transmitida y, de ser necesario, solicitar un reenvío de la trama de datos.

El bus de transmisión asincrónica es ampliamente utilizado en la comunicación de elementos de *hardware* debido a su fácil implementación y bajo costo: solo son necesarias dos conexiones (Tx/Rx).

2.6 Sistema de Posicionamiento Global (GPS)

El Sistema de Posicionamiento Global (GPS) es una herramienta desarrollada en Estados Unidos que brinda a los usuarios la posibilidad de conocer su posición y de tener navegación asistida, servicios de cronometraje y tiempo (PNT). El sistema consiste en 3 segmentos:

- el segmento espacial: consiste en una constelación de 24 satélites en órbita que transmiten constantemente sus coordenadas e información sobre sus horas para efectos de sincronización.
- el segmento de control: consiste en un monitor mundial y estaciones de control que mantienen los satélites en sus debidas órbitas y ajustan los relojes internos de los satélites. Además rastrea los satélites, corrige los datos de navegación y mantiene la integridad de la constelación.
- el segmento de usuario: consiste en receptores de señal GPS, que reciben información de los satélites y usan dicha información para calcular la posición del usuario y la hora respectiva en tres dimensiones.

El segmento espacial y el de control son administrados, operados y mantenidos por la Fuerza Aérea de los Estados Unidos (*U.S Air Force*).

2.6.1 Servicios GPS

Los satélites del GPS prestan servicios para uso civil y militar. El servicio civil se encuentra disponible para todos los usuarios a nivel mundial de manera continua y gratuita. El servicio militar está disponible para las fuerzas armadas de Estados Unidos, así como para fuerzas armadas aliadas y agencias gubernamentales aprobadas.

Existe una variedad de técnicas utilizadas por los fabricantes de unidades GPS para mejorar el rendimiento de los localizadores, de tal forma que las necesidades de los usuarios sean satisfechas. Estas técnicas mejoran la disponibilidad, la precisión y la integridad del servicio disponible para uso civil.

Según el estatuto 31 USC – 9701, es difícil cuantificar en qué medida los usuarios utilizan los servicios GPS y, por lo tanto, resulta imposible establecer un cobro directo por los servicios ofrecidos, por esa razón el servicio GPS es libre de costo directo.

2.6.2 Restricciones del servicio GPS civil

Todos los receptores GPS capaces de operar en altitudes superiores a los 18 km y velocidades superiores a los 515 m/s son considerados potencialmente peligrosos por el ejército estadounidense y, por lo tanto, deben contar con límites artificiales de funcionamiento para poder ser comercializados. Lo anterior es un intento de evitar que se utilice el sistema GPS para guiar misiles balísticos, sin embargo, al ser esta una restricción ambigua, los fabricantes de receptores GPS se dan la libertad de interpretar dicha restricción de diferente manera, lo que hace posible que algunos receptores sean apropiados para ser utilizados en proyectos de gran altitud.

CAPÍTULO 3

El proyecto DAEDALUS

3.1 Aspectos generales

Con el fin de cumplir con los requerimientos establecidos en el apartado 1.3, se formuló el proyecto "Globo Centroamericano de Investigación Espacial" bajo el nombre clave DAEDALUS: una plataforma de investigación científica optimizada para operar en grandes altitudes. La plataforma consiste en una sonda acoplada a un globo meteorológico relleno de helio capaz de elevarse hasta la estratósfera. La sonda transporta un computador embebido que permite capturar fotografías y realizar mediciones de las condiciones ambientales en grandes altitudes, durante un recorrido que dura aproximadamente 6 horas. Un enlace de comunicación inalámbrico permite a cualquier usuario del mundo seguir el recorrido y visualizar en tiempo real las mediciones efectuadas a través de Internet.

La plataforma cuenta con un sistema de soporte estructural que permite balancear la carga y eliminar oscilaciones mecánicas perjudiciales durante el vuelo. Para el descenso, se utiliza un paracaídas que permite disminuir considerablemente la velocidad de caída de la sonda, de manera que la energía del golpe en el aterrizaje sea absorbido por la estructura y el computador se pueda recuperar de manera íntegra en conjunto con la información recopilada.

Para fines prácticos, el diseño de la plataforma se dividió estratégicamente en 3 sub-bloques y equipos de trabajo correspondientes. Primero se tiene el bloque del Sistema de Elevación y Descenso (SED), encargado del sistema de soporte estructural de la sonda y paracaídas. Luego se tiene el bloque del Sistema de Aislamiento Térmico y Electrostático (SATE), encargado del diseño y fabricación de la sonda que protege al computador embebido contra el estrés térmico y la radiación electromagnética presentes en la estratósfera. Por último, se tiene el Sistema de Computación Embebido de Investigación a Gran Altura (SCEIGA¹), que realiza todas las funciones electrónicas de la plataforma y cuyo desarrollo se explica a lo largo del presente documento.

¹ El alcance del proyecto de graduación del estudiante Yoel Wigoda Kaver está limitado al diseño e implementación del SCEIGA únicamente. Se asume que, tanto el SED como el SATE, cumplen con sus objetivos de diseño y cumplen con las especificaciones técnicas requeridas para la operación correcta del SCEIGA.

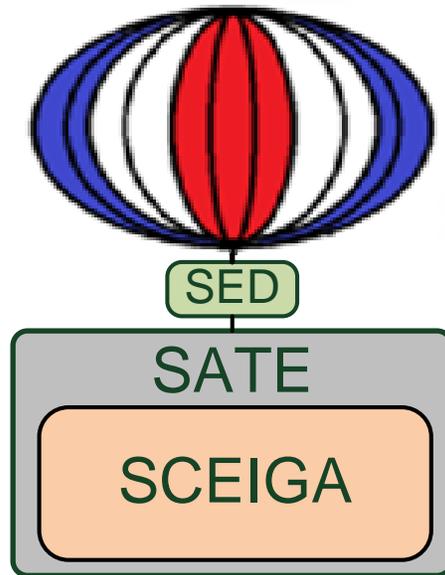


Figura 3.1 Distribución general del proyecto DAEDALUS

3.2 Plan y objetivos para el desarrollo del SCEIGA

Como se mencionó en el apartado anterior, la plataforma DAEDALUS se dividió en 3 sub-bloques de diseño. El presente documento comprende únicamente el desarrollo del bloque "Sistema de Computación Embebido de Investigación a Gran Altura (SCEIGA)", que es un sistema electrónico capaz de realizar las siguientes funciones:

- Captura de videos y fotografías durante el recorrido
- Medición de variables climáticas (temperatura exterior, humedad, presión atmosférica, cantidad de luz)
- Medición de variables físicas propias de la plataforma (aceleración axial, orientación magnética, temperatura interna, velocidad de ascenso, altitud)
- Rastreo satelital por GPS
- Enlace de comunicación inalámbrico entre la plataforma y una estación terrestre
- Acceso público y en tiempo real a la información medida
- Sistema de corte de emergencia
- Funcionamiento continuo por al menos 10 horas

El objetivo principal del proyecto es el desarrollo de un prototipo del SCEIGA capaz de efectuar las funciones descritas anteriormente según las premisas de diseño, manteniendo un bajo costo de implementación, un uso eficiente de la energía y promoviendo la escalabilidad.

El sistema pertenece a una plataforma de investigación de uso general y, por lo tanto, debe ser posible y sencillo modificar sus funciones para que se adecuen a las situaciones específicas presentes en las investigaciones por realizar.

Para permitir la escalabilidad y la adición futura de funciones, se siguió una estrategia de diseño de tipo "divide y vencerás", en la cual se subdivide un problema complejo en sub-problemas simples que resultan más sencillos de resolver. En diseño de *hardware*, esta estrategia se conoce como "diseño modular", que consiste en el desarrollo de módulos simples e independientes que posteriormente se interconectan entre sí y llegan a constituir un sistema íntegro que es capaz de realizar operaciones informáticas mucho más complejas. El diseño modular permite escalar el sistema o intercambiar sus funciones en un futuro dependiendo de la aplicación específica para la cual se requiera la plataforma, sin tener que recurrir a cambios significativos en el *hardware*. Los módulos se asignaron según su finalidad y requerimientos de hardware. El desarrollo de cada módulo corresponde a un objetivo específico.

Primer módulo: Sistema de Adquisición de Datos (SAD)

Este módulo es el encargado de las funciones de adquisición y almacenamiento de información en la plataforma.

El SAD se divide en dos subsistemas electrónicos según su aplicación:

- un computador embebido utilizado para capturar, codificar y almacenar los videos y las fotografías
- un micro-controlador usado para administrar los sensores que capturan las mediciones climáticas y físicas

La información es almacenada en memorias extraíbles de tipo microSD.

Se utilizó una especificación de requerimientos técnicos establecida por ACAE para seleccionar los componentes electrónicos utilizados en el prototipo. Dicha especificación denota el estrés climático que, en el peor de los casos, debe ser soportado por cada componente con el fin de evitar fallas en la operación.

Tabla 1 Lista de requerimientos técnicos de los sensores en el SCEIGA

Variable física	Precisión	Unidad	Temperatura de operación
Temperatura Interna	1	°C	“-40°C a 100°C”
Temperatura Externa	1	°C	“-40°C a 100°C”
Presión atmosférica	1.5	Pa	“-20°C a 70°C”
Humedad	3.5	%	“-40°C a 85°C”
Aceleración axial	8	g(9.8m/s ²)	“-40°C a 85°C”
Altitud(GPS)	10	m	-30°C a 85°C
Velocidad de ascenso(GPS)	1	km/s	-30°C a 85°C
Hora(GPS)	20	ms	-30°C a 85°C
Coordenadas(GPS)	10	m	-30°C a 85°C
Rumbo	2.5	°(grados)	“-20°C a 70°C”
Cantidad de Luz	1	%	“-40°C a 85°C”
Peso Total máximo	1	KG	N/A
Tiempo de operación continuo	10	horas	N/A

Segundo módulo: Sistema de Recuperación y Rastreo (SRR)

Este módulo se utiliza para rastrear la plataforma durante el recorrido y facilitar la recuperación posterior de la sonda.

El módulo está separado físicamente en dos secciones, una contenida en la plataforma y la otra en una estación terrestre. Las secciones del módulo se comunican entre sí a través de un enlace de radiofrecuencia.

En la sección contenida en la plataforma, el módulo cumple la función de localizador geográfico. Utiliza un receptor que opera con el Servicio de Posicionamiento Global (GPS) para extraer sus coordenadas con alta precisión.

La sección del módulo contenida en la estación terrestre publica automáticamente la información recopilada en la plataforma a través de un enlace con la Internet, de manera que cualquier usuario a nivel mundial pueda seguir el recorrido y revisar los datos medidos en tiempo real.

Tercer módulo: Sistema de Corte de Emergencia (SCE)

La plataforma no cuenta con un mecanismo de dirección que permita seleccionar su rumbo, es completamente dependiente de las condiciones climáticas que experimente durante el recorrido. Las corrientes de viento determinan la velocidad horizontal de cruce y el rumbo de la plataforma, la diferencia en la trayectoria total recorrida en un día con poco viento y uno ventoso podría ser hasta de 200 kilómetros.

Por la topología costarricense, existe un gran riesgo de que la plataforma viaje en dirección al océano y que, dependiendo de la velocidad del viento, la sonda eventualmente caiga en él y se pierda de esta forma la información almacenada.

Este módulo entonces permite cortar la cuerda que une la sonda con el globo. La instrucción de corte se emite desde la estación terrestre, de forma que un operador pueda interrumpir el vuelo si es necesario.

Este módulo también permite que el proyecto cumpla con las regulaciones de Aviación Civil para el lanzamiento de globos aerostáticos, ya que reduce las posibilidades de que la plataforma experimente una colisión con algún vehículo aéreo.

Cuarto módulo: Sistema de Alimentación de Potencia (SAP)

Este módulo cumple la función de regulación, aislamiento y suministro de energía del sistema electrónico abordo de la plataforma.

El módulo utiliza baterías de alto rendimiento para permitir la operación del sistema electrónico, sin interrupción, por al menos 10 horas, que es el tiempo total estimado necesario para cubrir la totalidad del viaje, incluyendo la fase de recuperación de la sonda. También se encarga de acondicionar los niveles de tensión en los buses de comunicación,

de forma que se puedan conectar en el mismo bus componentes con distintas tecnologías de fabricación. Por último permite acoplar las etapas de potencia y de control del sistema electrónico lo que permite reducir el ruido eléctrico que podría causar fallas durante la operación.

CAPÍTULO 4

*SCEIGA: Sistema de Computación Embebido
para Investigación a Gran Altura*

El SCEIGA es un sistema de computación móvil especializado en la adquisición de datos en ambientes hostiles. Su uso en la plataforma DAEDALUS permitirá capturar videos y fotografías de la curvatura terrestre, así como la medición de variables atmosféricas y su visualización en tiempo real desde una estación terrestre.

Para facilitar el diseño, el sistema se subdividió en los 4 bloques conceptuales que se plantean en los objetivos. Estos se explican en los apartados posteriores del presente documento.

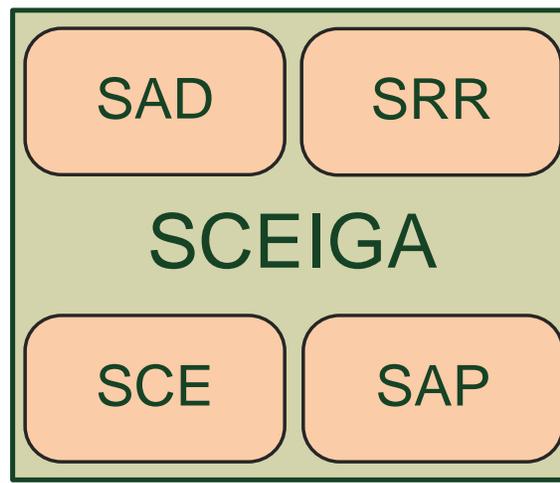


Figura 4.1 Diagrama de bloques del Sistema de Computación Embebido de Investigación a Gran Altura

4.1 Sistema de Adquisición de Datos (SAD)

Como se mencionó en los objetivos, este bloque funcional se encarga de realizar las funciones de adquisición de datos, esto incluye la captura de imágenes y la medición de las variables físicas seleccionadas. Por la naturaleza de las tareas computacionales efectuadas, fue necesario subdividir el bloque en dos etapas, cada una con un enfoque de diseño electrónico particular.

Las funciones de alto nivel de procesamiento, como la captura y compresión de imágenes, fueron asignadas al bloque SAD(AN).

Las funciones de medición y almacenamiento fueron asignadas al bloque SAD(BN). Los bloques están interconectados y son capaces de comunicarse entre sí.

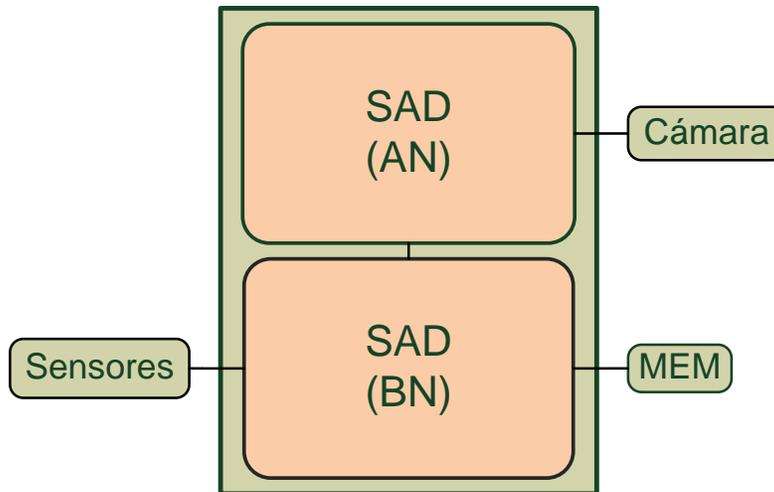


Figura 4.2 Diagrama de bloques del Sistema de Adquisición de Datos (SAD)

4.1.1 SAD (AN)

La etapa del Sistema de Adquisición de Datos de Alto Nivel se encarga de realizar las operaciones de captura, compresión y almacenamiento de imágenes en la plataforma. Este tipo de operaciones, generalmente, son efectuadas por sistemas electrónicos cerrados que solo están disponibles para los usuarios a través de dispositivos finales como las cámaras digitales y, por lo tanto, su integración con otros sistemas electrónicos es difícil.

Hasta hace algunos años, el desarrollo de aplicaciones de visión informática sólo era posible mediante el uso de ordenadores de escritorio que, aunque cuentan con los recursos de *hardware* necesarios, su uso en aplicaciones móviles no es viable. Este nicho del mercado es cubierto actualmente por una de las ramas de la Ingeniería Electrónica de mayor crecimiento en los últimos años: los sistemas embebidos. Estos sistemas se caracterizan por ofrecer un desempeño similar al de una computadora portátil pero con una eficiencia energética considerablemente superior. Es por este motivo que se eligió un sistema embebido para la implementación de las funciones de alto nivel que se realizan en la plataforma.

4.1.1.1 Selección del computador embebido

Existen actualmente en el mercado múltiples compañías que fabrican componentes para sistemas embebidos. Cada una ofrece una amplia gama de productos destinados a

cubrir las necesidades de los desarrolladores que buscan integrarlos en sus aplicaciones. Las compañías forman alianzas estratégicas para promocionar sus productos y crean tarjetas de evaluación que integran componentes de los múltiples fabricantes participantes en las alianzas, de forma que todos se benefician.

Para el desarrollo del módulo SAD(AN), se eligió la tarjeta de evaluación BeagleBoard XM debido a sus características: bajo consumo energético, tamaño compacto, flexibilidad y, principalmente, porque es de fuente abierta (*Open Source*) y cuenta con el soporte de una gran comunidad de desarrolladores a nivel mundial.

4.1.1.2 Instalación del sistema operativo

Al igual que los computadores de escritorio comunes, la BeagleBoard requiere un sistema operativo que administre sus recursos de *hardware* y proporcione los servicios necesarios para ejecutar el *software* eficientemente. Se eligió una distribución compacta de Linux para computadores embebidos llamada Angstrom, altamente recomendada por la comunidad para ser utilizada en la BeagleBoard.

El proceso de instalación de un sistema operativo es particular para cada sistema y obedece a la naturaleza del *hardware* disponible. En el caso de la BeagleBoard XM, el proceso es complejo ya que la tarjeta no posee una memoria de tipo no volátil y, por esta razón, el sistema operativo se debe instalar directamente en una memoria SD que también se utiliza para almacenar la información propia de la aplicación.

A continuación se presentan los pasos del proceso de instalación en el mismo orden en el cual fueron efectuados.

4.1.1.2.1 Configuración de la terminal de comunicación

Linux cuenta con una interfaz característica basada en una línea de comandos que les permite a los usuarios manipular sus funciones básicas. Esta interfaz también es el método principal de interacción entre el usuario y el sistema operativo en sus distribuciones compactas, sin embargo, el acceso a la línea de comandos de Angstrom se realiza de manera indirecta a través de un ordenador conectado al puerto RS232 de la tarjeta de desarrollo. Del lado del ordenador se debe instalar un programa que permita el acceso

remoto a sus puertos. Se seleccionó el paquete universal Minicom distribuido libremente para la plataforma Linux.

Para comunicarse con la tarjeta de desarrollo, el puerto serie utilizado en el ordenador debe ser configurado con los mismos parámetros del puerto serie de la tarjeta de desarrollo. La configuración de estos parámetros se realiza al entrar en el modo de configuración del programa a través de la instrucción `$sudo minicom -s`.

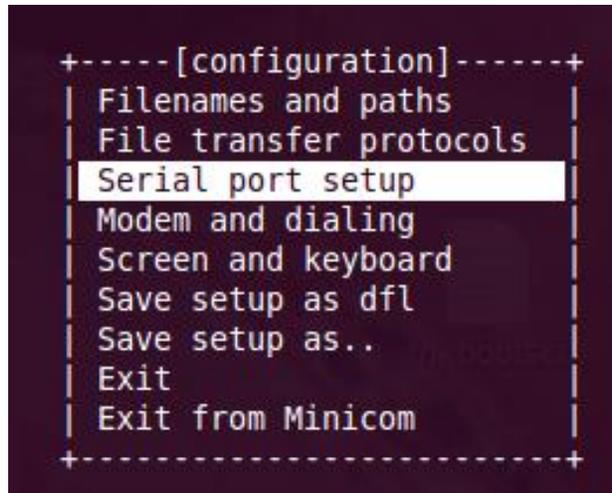


Figura 4.3 Interfaz de usuario del programa Minicom (pag1)

En la primera página del modo configuración se selecciona la opción “*Serial port setup*” con las flechas del teclado.

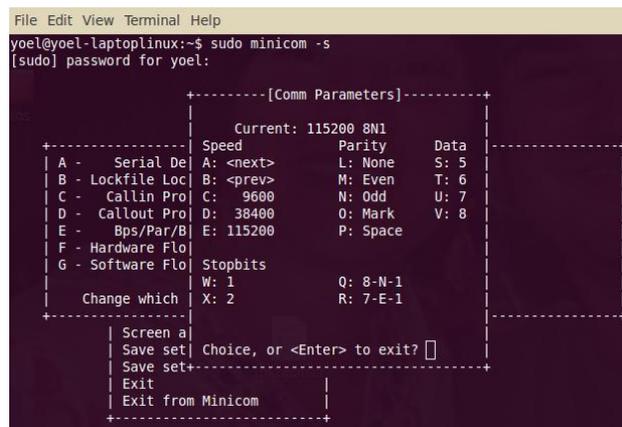


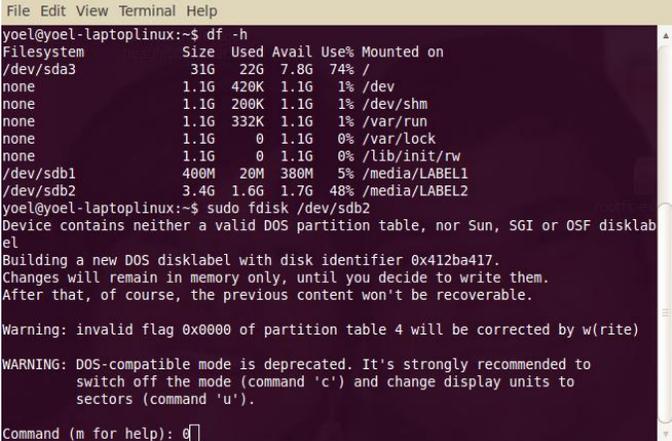
Figura 4.4 Interfaz de usuario del programa Minicom (pag2)

Luego, con la opción “A”, se selecciona el puerto `/dev/ttyUSB0` que corresponde al puerto USB0 con emulación de puerto serie. Posteriormente, con la opción “E”, se

configura el puerto para operar a 115200 baudios con 8 bits de datos, sin paridad y con un bit de parada. Por último, se guardan los cambios y la configuración para que dicho puerto sea utilizado por defecto cuando se ejecute el programa.

4.1.1.2.2 Formateo de la memoria SD

Para que el sistema operativo pueda instalarse en una memoria microSD, se debe definir una geometría particular en su tabla de particiones, de modo que su formato de archivos sea compatible con la arquitectura OMAP3. Esta acción se realiza manualmente desde el ordenador a través de la línea de comandos de Linux, mediante el paquete incluido fdisk.



```
File Edit View Terminal Help
yoel@yoel-laptoplinux:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda3        31G   22G   7.8G   74% /
none            1.1G 420K  1.1G   1% /dev
none            1.1G 200K   1.1G   1% /dev/shm
none            1.1G 332K   1.1G   1% /var/run
none            1.1G   0     1.1G   0% /var/lock
none            1.1G   0     1.1G   0% /lib/init/rw
/dev/sdb1        400M   20M  380M   5% /media/LABEL1
/dev/sdb2        3.4G  1.6G  1.7G  48% /media/LABEL2
yoel@yoel-laptoplinux:~$ sudo fdisk /dev/sdb2
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0x412ba417.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').

Command (m for help): d
```

Figura 4.5 Interfaz de usuario del programa fdisk

Los pasos del proceso se presentan a continuación:

1. Se carga el programa y se selecciona el punto de montaje de la memoria SD a través de la instrucción `fdisk /dev/sdX`. La "X" se debe substituir por la letra correspondiente que es asignada por el sistema operativo en el ordenador. Si el programa logra acceder a la partición seleccionada, su interfaz es cargada para permitirle al usuario efectuar las operaciones requeridas. Los cambios se almacenan en la memoria del ordenador y solamente después de ejecutar la instrucción de escritura se aplican en la memoria SD.
2. Se alista la partición que será utilizada durante el arranque, la cual debe tener el formato FAT32 de DOS, según lo establece el fabricante. Para esto en primer lugar,

se ejecuta el comando "o" para crear una nueva tabla de particiones en formato DOS.

3. Se extrae el tamaño de la memoria SD en *bytes* mediante el comando "p". Este dato es requerido en pasos posteriores.
4. Se entra en el modo experto del programa mediante el comando "x" para establecer la geometría requerida de la tabla de particiones.
5. Mediante el comando, se establece el número de cabezas (*Heads*) en 255.
6. Mediante el comando "s", se establece el número de sectores (*Sectors*) en 63.
7. Se establece el número de cilindros (*Cilinders*) con base en el tamaño de la memoria adquirida en el paso 3. Se calcula con la fórmula:

$c = (t/h/s/b)$ en donde:

_ "t" representa el tamaño de la memoria en *bytes*,

_ "h" el número de cabezas,

_ "s" el número de sectores y

_ "b" el número de bytes por sector.

Para definir el número de cilindros se utiliza el comando "c".

8. Una vez definida la geometría se procede con la creación de las particiones. Se regresa al modo normal con el comando "r".
9. Se crea, en primer lugar, una partición para el arranque mediante el comando "n".
10. Se define como partición primaria con el comando "p".
11. Se establece el tamaño de esta partición en 50 megabytes y se define el cilindro de inicio en 1 y el de terminación en [50+].
12. Se agrega la bandera de arranque a la partición con el comando "a", con el comando "t" se cambia su tipo y con el comando "c" se elige el código de tipo correspondiente a FAT32.

13. Se crea otra partición mediante el comando "n" que se utiliza para almacenar el sistema de archivos del sistema operativo. Su tamaño se define a través de los cilindros sobrantes, se empieza por el 52 y se termina en el último cilindro, según el cálculo efectuado en el paso 7.
14. Se aplican y guardan los cambios efectuados con el comando "w" y se cierra el programa.
15. Por último se definen los formatos y los nombres de cada partición directamente desde la línea de comandos de Linux. La instrucción `$ sudo mkfs.msdos -F 32 /dev/sdc1 -n P1` se utiliza para asignar el formato FAT32 a la primera partición y otorgarle el nombre P1. En forma similar, la instrucción `$ sudo mkfs.ext3 -L P2 /dev/sdc2` se utiliza para asignar el formato ext3 a la segunda partición y otorgarle el nombre P2.

Después de concretar el formateo de la memoria, se copian los parámetros de arranque en la partición P1 y la imagen de Angstrom en la partición P2.

4.1.1.2.3 Definición de los parámetros de arranque

Los procesadores OMAP3 utilizan un sistema de arranque multi-etapa (*Multiple Stage boot*) para cargar el sistema operativo, lo que permite flexibilizar las condiciones de arranque y adecuarlas a la aplicación. Este sistema de arranque consiste en subdividir el procedimiento en etapas que representan niveles de abstracción. Las etapas se ejecutan secuencialmente empezando por la de más bajo nivel. Solo la etapa inicial está implementada en *hardware* (memoria ROM), el resto está implementado en rutinas de *software*, lo que le otorga cierta libertad al desarrollador para modificar algunos parámetros de operación del sistema. La secuencia de ejecución se presenta a continuación:

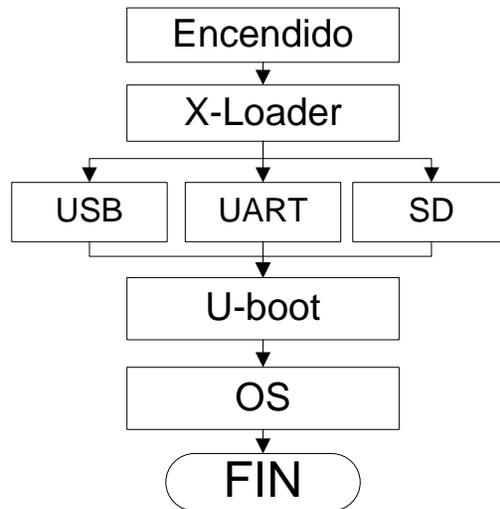


Figura 4.6 Secuencia de arranque de la arquitectura OMAP3

Tras el encendido, el código almacenado en la ROM carga, en la memoria estática SRAM, la primera etapa llamada X-Loader. Como la SRAM es pequeña, en esta etapa solo se inicia la memoria principal y los dispositivos periféricos necesarios para cargar el siguiente programa en la memoria principal.

La siguiente etapa consiste en ejecutar un *bootloader* universal denominado U-boot, el cual permite cargar el núcleo (*Kernel*) del sistema operativo con argumentos de arranque definidos por el desarrollador. La Beagleboard XM no cuenta con memoria NAND, que es donde generalmente se almacenan los argumentos de arranque, y por lo anterior, dichos parámetros se almacenan directamente en la memoria SD en un archivo que cargado por el U-boot mediante un *Script*.

El archivo que contiene los argumentos de arranque se nombró `pinicio.cmd` y su contenido es el siguiente:

```

setenv mpurate 800 # Limitar la frecuencia del reloj del procesador a 800 GHZ
setenv dvmode 1280x720MR-16@60 #Establecer parámetros del puerto DVI
setenv vram 12MB #Cantidad de memoria RAM destinada para el Video
setenv bootcmd 'mmc init; fatload mmc 0:1 0x80300000 uImage; fatload mmc 0:1
0x81600000 uInitrd; bootm 0x80300000 0x81600000' #Inicialización de la memoria
setenv bootargs console=ttyO2,115200n8 console=tty0 root=/dev/mmcblk0p2 rootwait ro
vram=${vram} omapfb.mode=dvi:${dvmode} fixrtc buddy=${buddy} # Parámetros
# de la consola
mpurate=${mpurate}
boot # Arranque
  
```

Para que el U-boot pase estos parámetros al núcleo, se creó el *Script* correspondiente utilizando la herramienta *mkimage* de Linux:

```
mkimage -A arm -O linux -T script -C none -a 0 -e 0 -n "Ubuntu" -d ./pinicio.cmd  
./piniciot.scr
```

Este *Script* se copia en la partición P1 de la memoria SD junto con el archivo X-Loader y U-boot para proceder a encender la tarjeta.

4.1.1.3 Selección e instalación del programa de captura

Para cumplir el objetivo del proyecto referente a la captura de videos y fotografías en la plataforma, se instaló el paquete Motion directamente del servidor mantenido por los desarrolladores de Angstrom a través de la instrucción de línea de comandos *opkg install*.

Motion es un programa de captura de video e imágenes escrito para Linux basado en su *API* video4linux. El programa está escrito en el lenguaje de programación C y su distribución, uso y modificación es libre bajo los términos de la licencia *GPL*. Originalmente fue concebido para ejecutar funciones de vigilancia y detección de movimiento, no obstante posee además las prestaciones necesarias para ejecutar otro tipo de aplicaciones que también involucran el procesamiento de videos o imágenes, por ejemplo:

- Captura de fotografías automáticamente al detectar movimiento
- Soporte para múltiples cámaras simultáneas
- Distribución de video por internet
- Creación, en tiempo real, de videos con el formato de compresión Mpeg
- Compresión de imagen en múltiples formatos
- Captura automática de fotografías en intervalos de tiempo predefinidos
- Envío automático de mensajes de texto y e-mail cuando se detecta movimiento
- Personalización de la interface de presentación de imágenes y videos

Estas posibilidades hacen que su uso sea altamente viable para cumplir con los objetivos planteados referentes a este bloque del sistema.

Los parámetros de operación se definen mediante un archivo de configuración que se presenta a continuación.

4.1.1.4 Parámetros de captura

En el archivo motion.conf se definen las condiciones de operación del programa mediante opciones predefinidas descritas por código. El programa permite personalizar sus funciones, el formato de datos, la resolución de imagen, el tiempo de muestreo, las entradas de texto, etc. El contenido del archivo de configuración empleado para la aplicación en desarrollo se presenta a continuación:

```
videodevice /dev/video0 #Puerto de webcam utilizado  
v4l2_palette 8 #Paleta de colores YUV  
input 8 # Cámara USB  
norm 1 # NTSC  
rotate 0 #No rotación  
width 352 #Ancho de imagen  
height 288 #Largo de imagen  
framerate 100 # Máximo número de cuadros capturados por segundo  
minimum_frame_time 0 # Periodo mínimo de captura de cuadros  
auto_brightness off #Desactivar brillo automático  
brightness 0 #Brillo inicial  
contrast 0 # Contraste automático  
saturation 0 # Saturación de color automático  
hue 0# Matiz automático  
output_all on #Guardar imágenes aunque no haya movimiento  
output_motion off# Desactivar detección de movimiento  
quality 100# Calidad en porcentaje de la compresión JPEG para las fotos  
ffmpeg_cap_new on #Codificar videos en Mpeg en tiempo real  
ffmpeg_bps 400000# Bitrate utilizado en la codificación  
ffmpeg_variable_bitrate 0 #Bitrate fijo  
ffmpeg_video_codec mpeg4 # Códec utilizado por Ffmpeg para la compresión de video  
snapshot_interval 10 # Capturar una fotografía cada 10 segundos  
text_right %Y-%m-%d\n%T-%q # Texto incluido en las fotos Año/mes/día/hora/número  
#de cuadro  
text_left P.Daedalus\n SCEIGA\n ACAE#Texto incluido en las fotos referente al proyecto  
text_event %Y%m%d%H%M%S# Hora y fecha--- Año/mes/día/hora/minuto/segundo  
target_dir /usr/bin/fotos# Directorio donde se almacenan los videos y fotos  
snapshot_filename %v-%Y%m%d%H%M%S-snapshot # Formato de nombre de archivos
```

4.1.1.5 Automatización de las funciones efectuadas

Para que el programa se ejecute automáticamente tras el encendido del prototipo, se deben realizar algunos ajustes preliminares en la configuración del sistema operativo. En primer lugar, se deben eliminar las cuentas de usuario y las claves de acceso, esto mediante la herramienta del sistema operativo *gdmsetup*. Luego se debe crear un *Script* que cargue el

programa automáticamente tras el inicio de la sesión. El *Script* se guarda en la dirección etc/rc2.d con el nombre "S65motion_client". Su contenido es el siguiente:

```
#!/bin/sh
/usr/bin/motion -c /etc/motion/motion.conf
```

Los videos e imágenes se almacenan constantemente en la dirección establecida anteriormente.

4.1.2 SAD (BN)

El Sistema de Adquisición de Datos de Bajo Nivel se encarga de realizar las operaciones de captura y almacenamiento de mediciones climáticas efectuadas desde la plataforma.

En aplicaciones que requieren bajo nivel de procesamiento, es conveniente utilizar micro-controladores, ya que generalmente son económicos y efectivos cuando se utilizan para interactuar con el medio físico, sin embargo, presentan algunas limitaciones técnicas importantes que deben ser consideradas previamente al diseño. El número de periféricos que pueden manejar directamente es relativamente pequeño, lo que representa un factor crítico para la aplicación desarrollada en este proyecto debido a la cantidad importante de variables físicas que deben ser medidas y almacenadas mediante los periféricos.

Cuando en la aplicación se deben efectuar mediciones sensibles al tiempo, es importante también considerar factores tales como la velocidad de procesamiento y la capacidad de efectuar interrupciones por *hardware*.

Una vez consideradas las anteriores limitaciones, se formuló una lista de requisitos técnicos y prioridades para seleccionar los componentes electrónicos que forman parte de este bloque.

Requisitos técnicos

- Como punto base, se minimizó la cantidad de entradas y salidas utilizadas seleccionando componentes digitales capaces de comunicarse bajo el estándar I²C que, al ser un protocolo de bus compartido, permite interconectar una gran cantidad de elementos requiriendo tan solo 2 pines.
- Para las mediciones de alta resolución, se eligieron sensores con interface SPI debido a la elevada tasa de transferencia de datos particular de este protocolo.

- Para realizar las funciones críticas, se seleccionaron componentes con interface UART.
- Se eligieron los sensores analógicos en función de la disponibilidad y los recursos del proyecto.
- Con respecto al entorno de programación, se eligió el SDK Arduino y su respectivo lenguaje de programación, por lo tanto, todos los algoritmos desarrollados para este bloque, se basan en el grupo de instrucciones que define dicho lenguaje.

4.1.2.1 Uso del Bus I²C

Como se ha mencionado, el bus I²C permite comunicar múltiples componentes con tan solo 2 líneas de transmisión. Cada dispositivo conectado cuenta con una dirección lógica única que permite su selección, de esta misma forma los registros de trabajo de cada integrado cuentan con una dirección característica que le permite al maestro configurar y controlar la operación del sistema.

Pueden existir varios dispositivos conectados con las mismas direcciones de registros, no obstante, no pueden conectarse en el mismo bus dos componentes con la misma dirección de dispositivo porque ocurrirían colisiones de datos en el bus y se perdería la información. Las direcciones de los registros son de 8 *bits*, el *bit* menos significativo se utiliza para denotar escritura o lectura.

Para escribir en un registro de un dispositivo I²C , se deben seguir una serie de pasos para establecer la comunicación:

1. El maestro debe mandar la dirección del dispositivo por comunicar a través del bus.
2. Todos los integrados conectados reciben la petición, pero solo hay un dispositivo en la red con la dirección requerida, que valida la operación y responde con un acuse de recibo (*acknowledge*), que da paso al inicio de la comunicación.
3. El maestro manda la dirección del registro por escribir y espera el acuse de recibo.
4. El maestro manda el valor del dato por escribir y finaliza la comunicación.

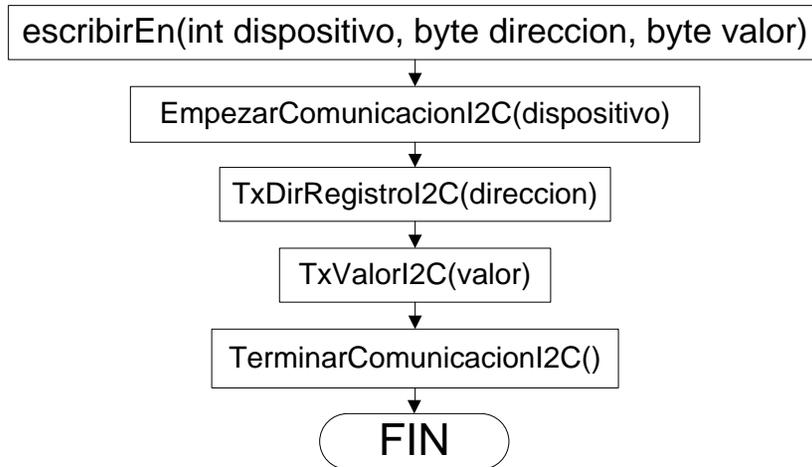


Figura 4.7 Método escribirEn() utilizado en el proceso de escritura del bus I²C

Para leer de un registro I²C , se debe seguir un procedimiento similar:

1. El maestro inicia la comunicación, selecciona el dispositivo y el registro por leer.
2. Luego envía una solicitud para dar comienzo a la lectura.
3. Cada vez que se levanta la bandera de dato disponible, se almacena dicho dato en un *buffer* de memoria y se incrementa el puntero índice del *buffer*.
4. Cuando ya no hay más datos disponibles para leer, el maestro termina la comunicación.

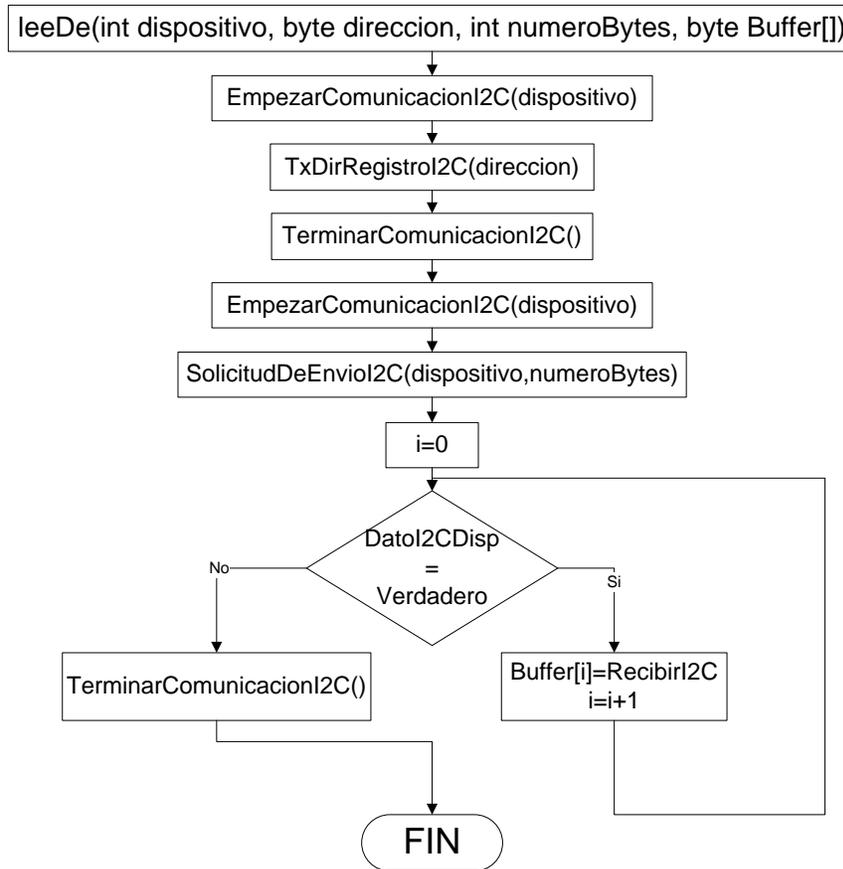


Figura 4.8 Método leerDe() utilizado en el proceso de lectura del bus I²C

4.1.2.2 Acelerómetro

El ADXL345 es un acelerómetro compacto, de muy bajo consumo de energía, capaz de medir la aceleración axial en tres ejes con una resolución de 13 *bits*. Permite medir hasta 16 veces la aceleración de la gravedad (16 g).



Figura 4.9 ADXL345

La información medida sigue un formato de 16 *bits* con complemento a 2 y es accesible a través del bus I²C .

El fabricante recomienda su uso en aplicaciones móviles. Es capaz de medir la aceleración estática de la gravedad, lo que permite cuantificar la inclinación con respecto a un plano de un dispositivo móvil y también es capaz de medir la aceleración dinámica resultante de un movimiento súbito, como un choque o la caída libre. Su alta resolución (3.9mg/LSB) permite detectar cambios menores a 1° en la inclinación.

El integrado también permite realizar funciones complementarias de propósito específico. Cuenta con un modo de operación por medio del cual se detecta un movimiento determinado mediante la comparación de la aceleración medida con un umbral establecido por el usuario. Hace uso de un modo de operación que detecta el desplazamiento en alguna dirección. Por último también posee un modo de operación que permite detectar la caída libre. Estos modos de operación pueden ser configurados para generar interrupciones en cualquiera de los pines del integrado destinado para tal fin.

El dispositivo cuenta con una pila FIFO ('primero en entrar, primero en salir') de 32 niveles utilizada para almacenar la información y minimizar la carga del procesador maestro. También tiene modos de operación de bajo consumo de potencia basados en la detección de movimiento, los cuales son especialmente útiles cuando el sistema es alimentado por baterías.

El sensor es apropiado para ser utilizado en temperaturas de hasta -40°C.

Tabla 2 Parámetros de operación del acelerómetro ADXL345 (Extraído de la hoja de datos)

Temperature Range	
Powered	-40°C to +105°C
Storage	-40°C to +105°C

4.1.2.2.1 Principio de operación

Para cuantificar la aceleración, se mide la deflexión de la estructura del sensor mediante "capacitores diferenciales" que consisten en un conjunto de placas fijas y placas sujetas a una masa de prueba que puede moverse libremente.

Cuando se presenta aceleración, se mueve la masa de prueba correspondientemente y se descargan los “capacitores diferenciales”, produciendo una tensión proporcional a la aceleración. Posteriormente, se usa demodulación de fase para determinar la magnitud y la dirección física de la aceleración.

4.1.2.2.2 Conexión eléctrica

Para comunicar el sensor con el controlador maestro, se utilizó el bus I²C que, como ha sido mencionado anteriormente, solo requiere la conexión de 2 cables para la comunicación. El bus I²C del ADXL345 permite la transmisión de múltiples *bytes* en una sola trama de datos, tanto en modo rápido (400 kHz) como en modo normal (100 kHz).

El pin ALT ADDRESS permite seleccionar entre dos direcciones de dispositivo:

- ✓ Si el pin se conecta a nivel lógico alto, el dispositivo adquiere la dirección de 7 *bits* 0x1D. Para completar la dirección, se concatena un *bit* más en la posición menos significativa que denota escritura o lectura, entonces por ejemplo, la dirección 0x1D se convierte en 0x3A para escritura y en 0x3B para lectura.
- ✓ Si se conecta el pin ALT ADDRESS al nivel lógico bajo, el dispositivo adquiere la dirección 0x53, lo que se convierte en 0xA6 para escritura y en 0xA7 para lectura.

$$Dir. Acel = 0x53 < 1|(R/W)$$

El pin CS se conecta a alto lógico para utilizar el bus I²C . Los pines SDA y SCL se conectan al bus I²C y a resistencias de *pull-up*.

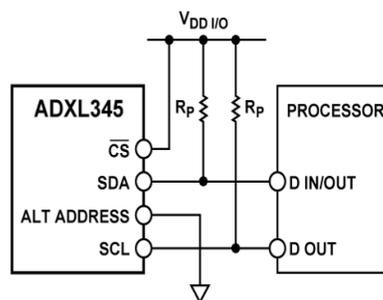


Figura 4.10 Conexión eléctrica básica del ADXL345

4.1.2.2.3 Programación

Para programar el sensor, se utilizan los métodos de manejo del bus I²C LeerDe() y EscribirEn() descritos anteriormente. En la fase de configuración se limpia el registro de control y se coloca la bandera de medición (*measure*) en nivel alto para iniciar el proceso de medición.

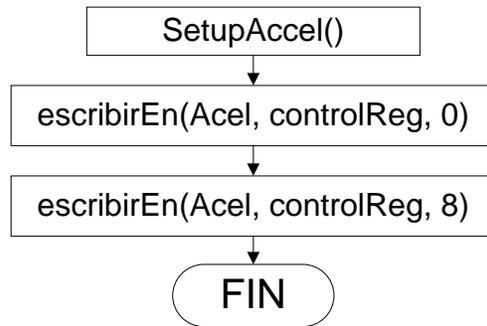


Figura 4.11 Método SetupAccel() utilizado en el proceso de inicialización del ADXL345

Para obtener lecturas de aceleración, se accede a los registros de datos del sensor. Las mediciones en el eje X se almacenan en los registros con las direcciones 0x32 (LSB) y 0x33 (MSB). Las mediciones en el eje Y se almacenan en los registros con las direcciones 0x34 (LSB) y 0x35 (MSB) y, finalmente, las mediciones en el eje Z se almacenan en los registros con las direcciones 0x36 (LSB) y 0x37 (MSB).

Los 6 registros se leen de manera consecutiva en una sola operación de lectura I²C y se almacenan en un *buffer* de memoria, esto permite maximizar la velocidad de transmisión del bus. Posteriormente se lee el *buffer* y se almacena la medición de cada eje en un espacio de memoria con el nombre correspondiente. Se implementó un filtro digital de ruido que consiste en promediar las mediciones anteriores con las nuevas con el fin de suavizar la respuesta del sensor.

Por último, se revisa la bandera de datos nuevos provenientes del GPS para evitar la pérdida de información.

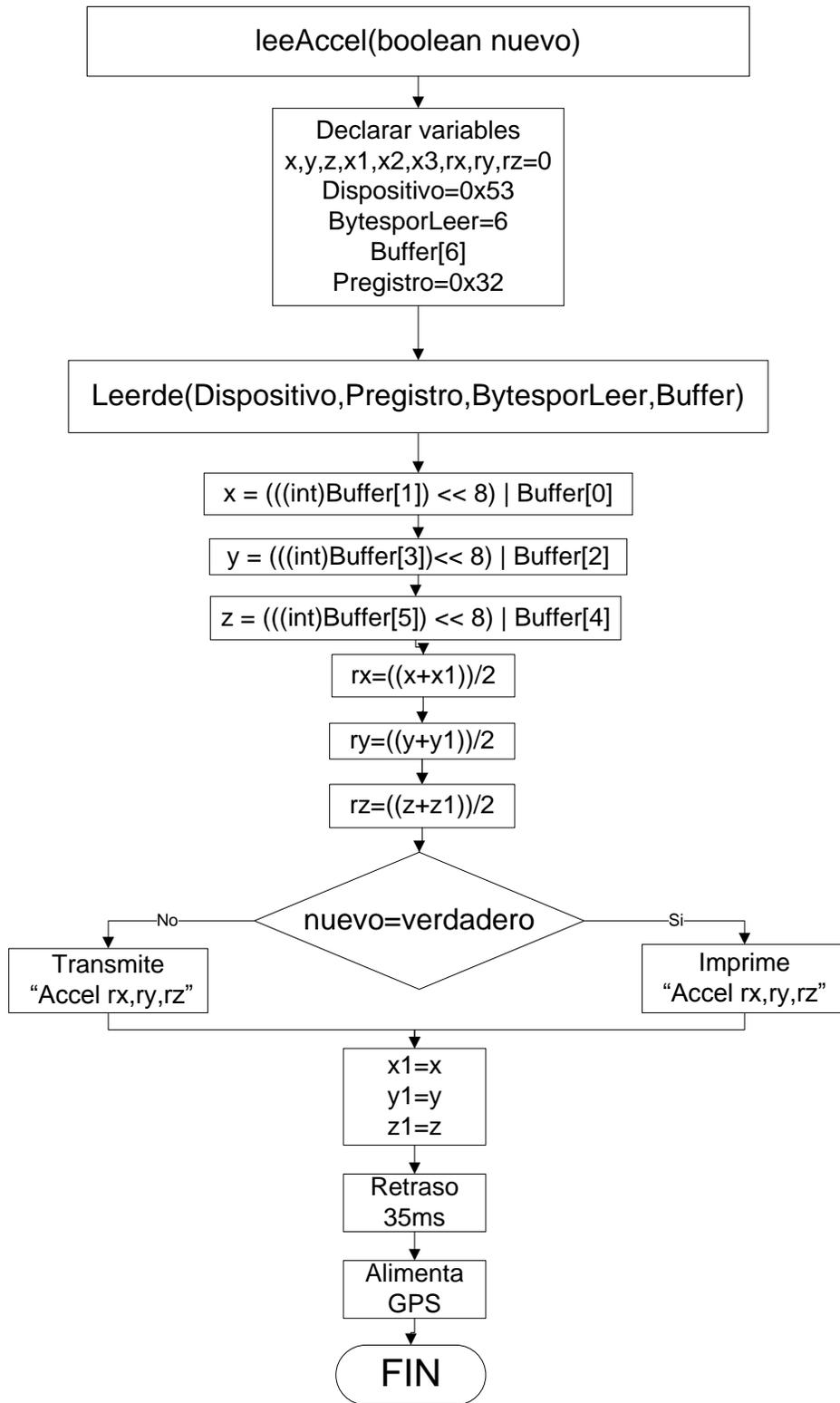


Figura 4.12 Método LeeAccel() utilizado en el proceso de lectura del ADXL345

4.1.2.3 Sensor de luz

El TEMT6000 es un sensor que permite obtener, de manera cuantitativa, la cantidad de luz disponible con base en los límites del ojo humano.



Figura 4.13 TEMT6000

El sensor consiste en un fototransistor NPN configurado como emisor común, de tal forma que la tensión en la resistencia de emisor varía linealmente con la luz disponible. La tensión es medida e interpretada por el micro-controlador maestro a través de un convertidor analógico-digital (ADC). Este sensor puede operar hasta en temperaturas de -40 °C.

Tabla 3 Parámetros de operación del sensor de luz TEMT6000(Extraído de la hoja de datos)

Operating Temperature Range		T_{amb}	- 40 to + 85	°C
Storage Temperature Range		T_{stg}	- 40 to + 85	°C

4.1.2.3.1 Principio de operación

El integrado utiliza un fototransistor NPN que opera de manera similar a un transistor bipolar típico, sin embargo, el encapsulado del fototransistor es transparente y, por lo tanto, la luz llega a la unión colector-base. Los electrones generados por los fotones en la unión base-colector son inyectados en la base y la corriente resultante es multiplicada por la ganancia de corriente del transistor (B), lo que resulta en una operación similar a la de un transistor bipolar NPN.

4.1.2.3.2 Conexión eléctrica

El fototransistor se conecta según la configuración de emisor común, se conecta una resistencia del emisor a tierra, de tal forma que la tensión que cae en la resistencia es proporcional a la luz presente. El nodo correspondiente al emisor se conecta directamente al ADC(0) del maestro.

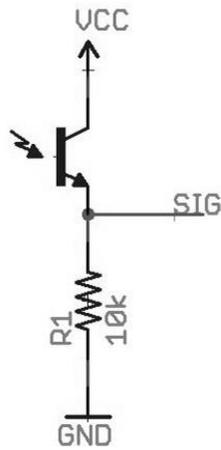


Figura 4.14 Conexión eléctrica del TEMA6000

4.1.2.3.3 Programación

Como el sensor es completamente analógico, no es necesaria su configuración previa para que opere, pero sí se debe configurar e inicializar la operación del ADC(0) del microcontrolador maestro.

Para adquirir mediciones de la cantidad de luz, se utiliza el método leeLuz(). En cada llamada al método, se realiza un proceso de lectura del puerto analógico 0. Los resultados pueden variar entre 1023, para el valor máximo de tensión, y 0, para el valor mínimo. Debido a lo anterior se realiza un ajuste de la curva para obtener una variación porcentual, es decir de 0% a 100%, con el fin de poder interpretar las mediciones efectuadas. La variable lógica "nuevo", permite variar el flujo del programa con el propósito de escoger si la medición es para uso interno o se debe transmitir.

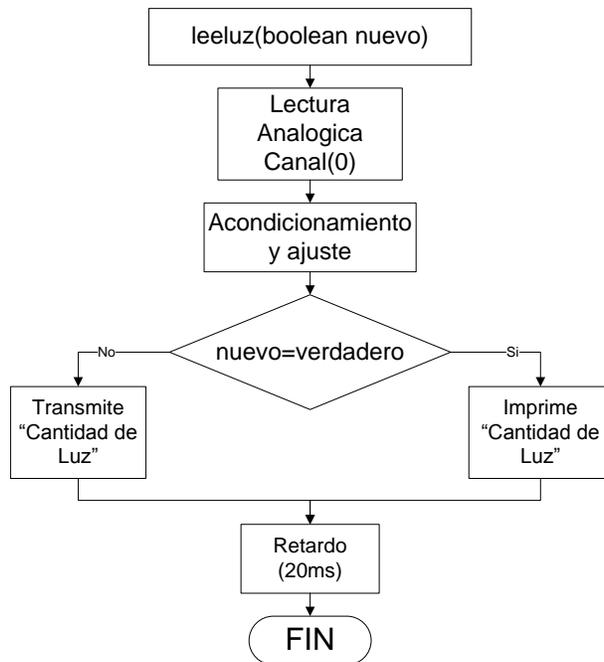


Figura 4.15 Método LeeLuz() utilizado en el proceso de lectura del TEMT6000

4.1.2.4 Termómetros

El TMP102 es un sensor de temperatura que permite obtener mediciones con una precisión de 0.5° C.



Figura 4.16 TMP102

Su bajo consumo de potencia y de recursos de *hardware* hace que sea apropiado para aplicaciones móviles y con condiciones ambientales extremas. El sensor se comunica con el micro-controlador maestro a través del bus I²C .

Este sensor permite medir temperaturas extremadamente bajas con exactitud hasta los -40° C, lo que lo hace apropiado para medir y registrar el cambio de temperatura ambiental durante la elevación y descenso de la plataforma. Es capaz de detectar cambios de temperatura de hasta 0.0625 °C .

Tabla 4 Parámetros de operación de los sensores de temperatura TMP102 (Hoja de datos)

TEMPERATURE RANGE					
Specified Range		-40		+125	°C
Operating Range		-55		+150	°C
Thermal Resistance, SOT563	θ_{JA}		260		°C/W

Se utilizará un sensor interno a la sonda para cuantificar la eficiencia del aislamiento térmico de la carcasa y un sensor externo para medir la temperatura atmosférica.

La precisión del sensor es independiente de la tensión de alimentación, sin embargo empeora gradualmente conforme baja la temperatura más allá de los -40 °C.

4.1.2.4.1 Principio de operación

El transductor de temperatura en el TMP102 es el integrado por sí mismo, es decir dentro del encapsulado hay un circuito térmico metálico que se utiliza para medir indirectamente la temperatura. Como el metal presenta una resistencia térmica menor que el plástico del encapsulado, la mayor parte del calor circula a través del metal, lo que permite obtener una variación de tensión proporcional a la temperatura bajo el principio físico del termopar.

4.1.2.4.2 Conexión eléctrica

Se utilizó el bus I²C para comunicar los sensores con el micro-controlador maestro. Los pines SCL y SDA se conectan al bus compartido existente. El TMP102 puede operar en modo rápido (400 kHz) y modo de alta velocidad (3.4 MHz).

El TMP102 cuenta con un pin denominado A0 que permite variar la dirección del dispositivo, de tal forma que se pueden conectar hasta 4 dispositivos iguales en el mismo bus I²C .

Tabla 5 Direcciones de dispositivo posibles para el TMP102 (Hoja de datos)

DEVICE TWO-WIRE ADDRESS	A0 PIN CONNECTION
1001000	Ground
1001001	V+
1001010	SDA
1001011	SCL

Las direcciones de dispositivo son de 7 *bits*. Al igual que en los otros dispositivos I²C, se debe agregar un *bit* para denotar escritura o lectura.

$$Dir.temp1 = 0x48 < 1|(R/W)$$
$$Dir.temp2 = 0x49 < 1|(R/W)$$

4.1.2.4.3 Programación

El TMP102 cuenta con una serie de registros de lectura y escritura que permiten gobernar la operación del dispositivo. Los registros principales son: el registro de configuración y el registro de temperatura. El acceso a dichos registros se debe realizar a través de un registro apuntador (*Pointer Register*) que sirve para realizar direccionamientos indirectos en la memoria del dispositivo. Cuando el registro puntero apunta a la dirección 0x00, se ingresa al registro de temperatura, donde se almacenan las mediciones efectuadas. Cuando el registro puntero apunta a la dirección 0x01, se ingresa al registro de configuración, donde se pueden realizar ajustes a la operación del sensor.

Si no se modifica el registro de configuración, el sensor opera por defecto, con una resolución de 12 *bits* y con una frecuencia de muestreo de 0.25 Hz.

Se utilizan los métodos de manejo del bus I²C LeerDe() y EscribirEn() para operar el sensor. El método leeTemp() debe recibir como parámetro la dirección del sensor de temperatura por leer para distinguir entre los dos sensores iguales conectados en el bus. El procedimiento es el siguiente:

1. Tras la llamada al método, se inicia la comunicación con el sensor requerido, apuntando hacia el registro 0x00 con el fin de entrar al registro de temperatura.
2. Se solicitan 2 *bytes* a través de los parámetros del método de lectura y se almacenan en un *buffer* de memoria.
3. Posteriormente se lee el *buffer* y se almacena cada *byte* en variables separadas según su peso llamadas MSB o LSB.

4. Mediante una operación de manejo de *bits*, se combinan ambos *bytes* en una variable de 16 *bits* y el resultado se multiplica por 0.065, que es el peso del LSB , con lo que se obtiene la temperatura real en °C.
5. Por último, dependiendo de la dirección de dispositivo del sensor, se imprime la temperatura externa o interna de la sonda.

La variable lógica "nuevo" permite seleccionar si la medición es para uso interno o se debe transmitir.

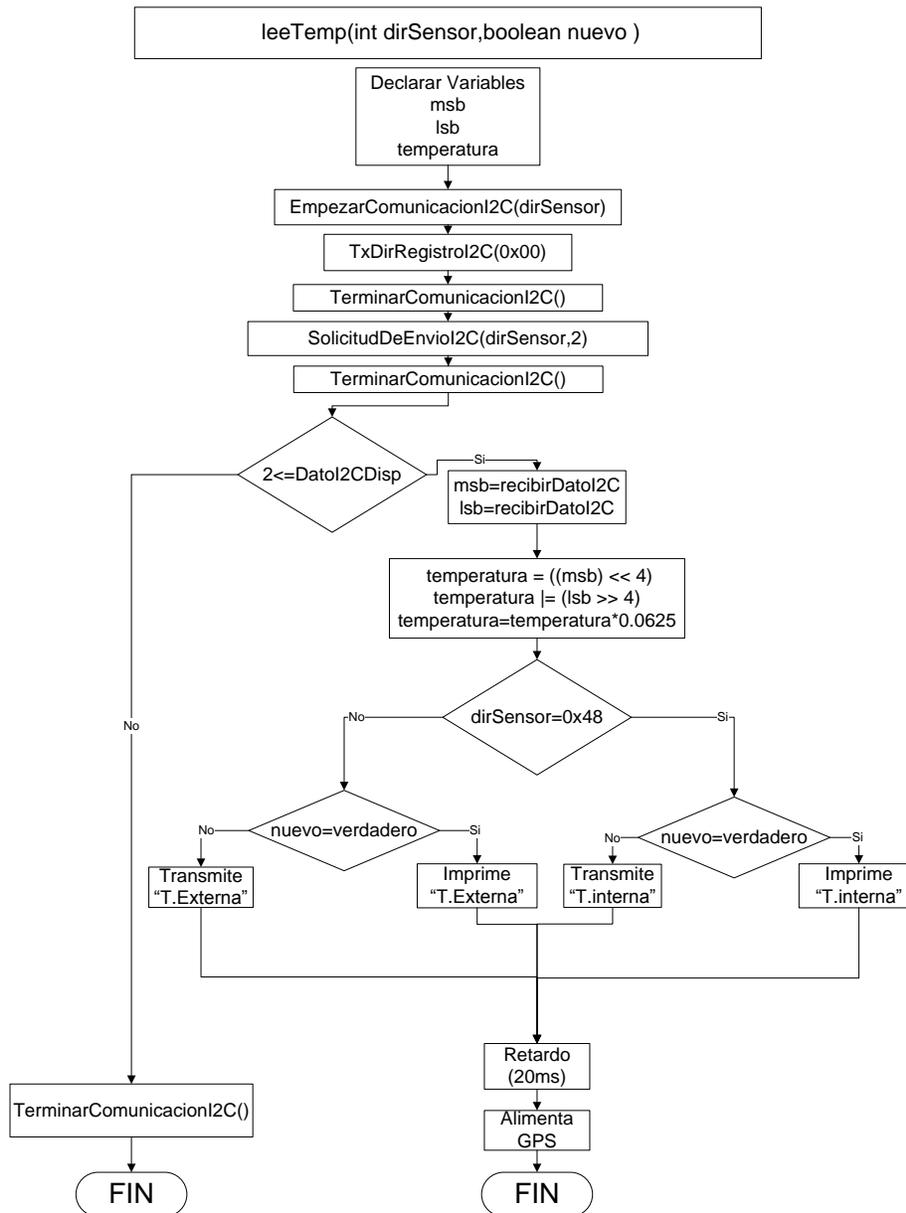


Figura 4.17 Método LeeTemp() utilizado en el proceso de lectura del TMP102

4.1.2.5 Compás

Se seleccionó el integrado HMC6352 de la marca Honeywell para medir la orientación magnética de la sonda. Este integrado es un compás digital que opera en 2 ejes con una precisión de 2.5 grados y que cuenta con una interface digital que habilita la comunicación mediante el bus I²C . El uso de este sensor permite hacer análisis de estabilidad mecánica de la sonda y del efecto del viento en el recorrido.



Figura 4.18 HMC6352

El bajo consumo de energía del HMC6352 (1mA) y su tamaño compacto lo hacen apropiado para aplicaciones móviles que operan con baterías. El sensor es capaz de detectar cambios en el rumbo de hasta 0.5 grados. Puede operar hasta en temperaturas de -20 °C, por lo tanto debe ser colocado por dentro de la carcasa aislante para evitar que se dañe debido a la baja temperatura presente en la estratósfera.

Tabla 6 Parámetros de operación del compás HMC6352 (Hoja de datos)

Operating Temperature	Ambient	-20	70	°C
Storage Temperature	Ambient	-55	125	°C

4.1.2.5.1 Principio de operación

El sensor utiliza magneto-resistencias colocadas en un arreglo tipo puente para medir el campo magnético. Cuando el campo magnético varía, las magneto-resistencias cambian su resistencia de manera proporcional, lo que provoca un desbalance en el puente que se puede cuantificar como una tensión eléctrica.

4.1.2.5.2 Conexión eléctrica

El sensor se conecta al controlador maestro mediante el bus I²C . Los pines SDA y SCL del integrado se conectan directamente al bus I²C existente del sistema. El integrado utiliza la velocidad normal de transmisión I²C de 100 KHz.

La dirección de dispositivo es 0x21, al igual que en los otros dispositivos I²C se debe agregar un *bit* para denotar escritura o lectura. Esta dirección es asignada por el fabricante y no es intercambiable.

$$Dir. Compás = 0x21 < 1|(R/W)$$

4.1.2.5.3 Programación

El HMC6352 contiene varios modos de operación que permiten adecuar su uso para aplicaciones específicas, dependiendo de la eficiencia en el consumo de potencia requerida y de la rapidez en la cual se necesiten las mediciones del sensor.

- a. Primer modo de operación: el sensor está configurado para operar en el modo de “espera” (*standby*): el sensor espera los comandos del controlador maestro para efectuar las mediciones. Mientras que no reciba ninguna instrucción, el sensor se mantiene en un estado inactivo de bajo consumo de potencia hasta recibir una solicitud de lectura I²C del registro 0x41, que conlleva a que el sensor realice una medición y responda con dos *bytes* que representan el valor medido.

El anterior modo de operación explicado es apropiado cuando la latencia en el tiempo de medición no es crítica. El hecho de que esté inactivo mientras no se utilice, hace que sea muy eficiente energéticamente y apropiado para la aplicación desarrollada, ya que el sistema es completamente alimentado por baterías.

- b. Segundo modo de operación: el sensor también cuenta con el modo de operación de cola (*query*) que empieza una nueva medición tras una lectura, lo que reduce la latencia en el tiempo de medición, aunque aumenta el consumo de potencia del dispositivo.
- c. Tercer modo de operación: este modo de operación es el continuo (*continuous*), en el cual el sensor toma constantes mediciones a una frecuencia

de 10 Hz, de forma que se reduce considerablemente el tiempo de acceso a la información medida, sin embargo el consumo de potencia es máximo.

En la aplicación desarrollada se utilizó el primer modo, ya que esta no presenta sensibilidad con respecto al tiempo de adquisición de las mediciones de orientación magnética y aparte como es totalmente alimentada por baterías, es necesario un uso eficiente de la energía.

Con respecto al algoritmo utilizado para obtener los datos, se desarrolló el método `leeCompas()` que permite obtener y almacenar en un *buffer* de memoria mediciones de orientación magnética. Se utilizó el método de manejo I²C `Leerde()` para acceder a los registros de datos del sensor, particularmente al registro con la dirección `0x41`, que contiene el dato medido. Se utilizó también la variable lógica global "nuevo" para definir si la medición es para uso interno o se debe transmitir y se alimentó el GPS para evitar la pérdida de información de las coordenadas de la sonda.

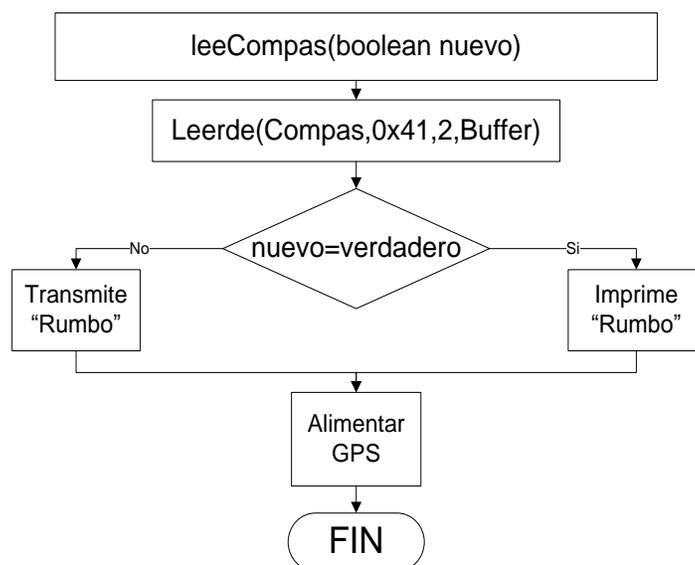


Figura 4.19 Método utilizado para obtener lecturas del compás HMC6352

4.1.2.6 Presión atmosférica

Para medir la presión atmosférica, se seleccionó el integrado SCP1000 del fabricante VTI, que cuenta con una resolución de 17 *bits*, lo que permite detectar cambios sutiles de presión barométrica.

A diferencia de los sensores convencionales de presión, el SCP1000 es capaz de medir la presión absoluta en el ambiente de forma independiente de la altitud.

El sensor cuenta con un procesador interno que se encarga de procesar la información y de realizar la calibración inicial. El procesador interno también administra la interface de comunicación digital del sensor que, a diferencia de los otros elementos digitales de medición del sistema, utiliza el protocolo SPI para intercambiar información con el controlador maestro.



Figura 4.20 SCP1000

El SCP1000 puede operar en un rango de presión de 30kPa a 120kPa, bajo una temperatura de hasta -20°C y con una resolución de 2 Pa en el LSB.

4.1.2.6.1 Principio de operación

El SCP1000 posee un transductor capacitivo para medir la presión barométrica. Mediante el proceso patentado de fabricación VTI, 3D-MEMS (*Three Dimensional Micro Electro Mechanical Systems*) se fabrica una oblea de silicio lo suficientemente delgada como para que funcione como un diafragma sensible a la presión barométrica. El diafragma actúa como la placa móvil del capacitor y una película metálica empotrada en el paquete actúa como la placa fija. Las dos placas se encapsulan mediante soldadura anódica, de forma que queda un vacío hermético entre las placas. El diafragma se contrae por la diferencia de presión entre el medio externo y el vacío interno utilizado de referencia. Dicha contracción resulta en una variación de la capacitancia del transductor medible eléctricamente y que es directamente proporcional a la presión barométrica del ambiente.

4.1.2.6.2 Conexión eléctrica

El SCP1000 utiliza una interface SPI para comunicarse con el micro-controlador maestro. La interface SPI representa un tipo de comunicación sincrónica y por esta razón, aparte de los pines tradicionales para comunicación serial (*SerialIn*, *SerialOut*), entre el controlador maestro y el sensor se debe conectar una línea de reloj para sincronizar la operación. La señal de reloj es siempre puesta por el micro-controlador maestro.

La interface SPI permite realizar operaciones de lectura y escritura a una frecuencia de operación de 500 KHz, lo que hace posible utilizar la resolución máxima disponible de 17 *bits* sin causar latencia en el programa principal del controlador maestro.

La interface de comunicación del sensor también contiene señales para el manejo de interrupciones en el programa principal del sistema, tal es el caso de la señal DRDY (*Data Ready*) que se utiliza para denotar el fin de una medición y activar la interrupción correspondiente. También se conecta la señal CSB (*Chip Select*) de forma que el maestro pueda seleccionar el sensor, en caso de que se agreguen más sensores al bus SPI en una etapa futura del proyecto.

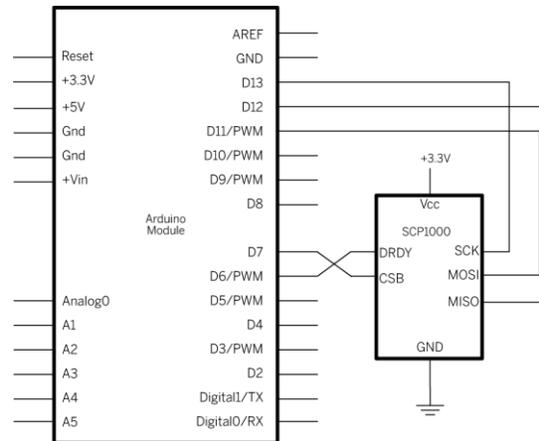


Figura 4.21 Conexión eléctrica básica del sensor SCP1000

4.1.2.6.3 Programación

Como se ha mencionado, el SCP1000 se comunica a través de una interface tipo SPI y, por lo tanto, fue necesaria la utilización de métodos para el manejo de las operaciones de lectura y escritura según el protocolo SPI.

Tanto en la operación de escritura como de lectura, la dirección del registro por acceder en la operación debe ser combinada con un comando característico del sensor para denotar escritura o lectura. Para el caso particular del SCP1000, el comando de escritura es 0x02 y el de lectura es 0xFC. El comando se combina con el registro por ingresar a través de la siguiente operación lógica:

$$datoPorMandar = ((RegPorAccesar \ll 2)|(C.EscLec))$$

El registro por acceder se debe desplazar 2 *bits* hacia la izquierda porque el sensor toma los 6 *bits* más significativos del *byte* como la dirección del registro.

El SDK Arduino proporciona una librería llamada SPI.h que contiene los *drivers* para manejar el bus SPI en el micro-controlador maestro. La instrucción SPI.transfer() permite mandar datos por el bus.

Para el caso de la operación de escritura, se escribió el método escribirRegSPI(). Este método recibe como parámetro la dirección del registro por escribir y el valor del dato que se escribirá. Sigue el siguiente proceso:

1. Tras la llamada al método, se calcula la dirección efectiva del registro por escribir.
2. Se selecciona el sensor poniendo la señal CSB en estado lógico bajo.
3. Se transmite por el bus la dirección efectiva del registro seguida por el valor del dato por guardar.
4. Se desactiva el sensor al poner la señal CSB en estado lógico alto.

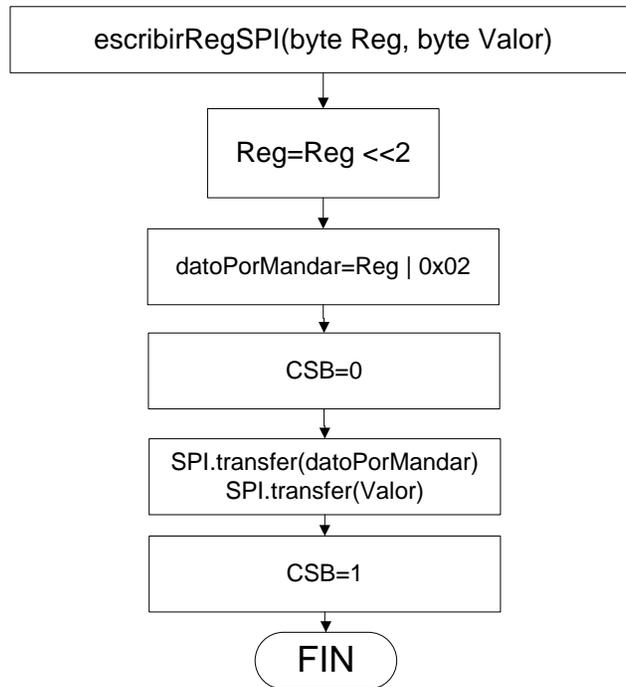


Figura 4.22 Algoritmo utilizado para la escritura SPI

Para el caso de la operación de lectura SPI, se escribió el método leerRegSPI(). Este método recibe como parámetro la dirección del registro y la cantidad de bytes por leer. Sigue el siguiente procedimiento:

1. Tras la llamada al método, se calcula la dirección efectiva del registro por leer y se procede a seleccionar el sensor con la señal CSB.
2. Se manda por el bus la dirección efectiva del sensor y el valor 0x00 para solicitar la lectura de un *byte* cuyo valor se almacena en la variable Resultado.
3. Como ya se extrajo un *byte*, se disminuye en 1 la cantidad de *bytes* por leer.
4. Si todavía existen *bytes* por leer, se desplaza 8 *bits* a la izquierda el resultado parcial y se manda 0x00 por el bus nuevamente para solicitar la lectura de otro *byte*, cuyo valor es combinado con el resultado parcial en una sola variable de 16 *bits*.
5. Finalmente se desactiva el sensor mediante la señal CSB.

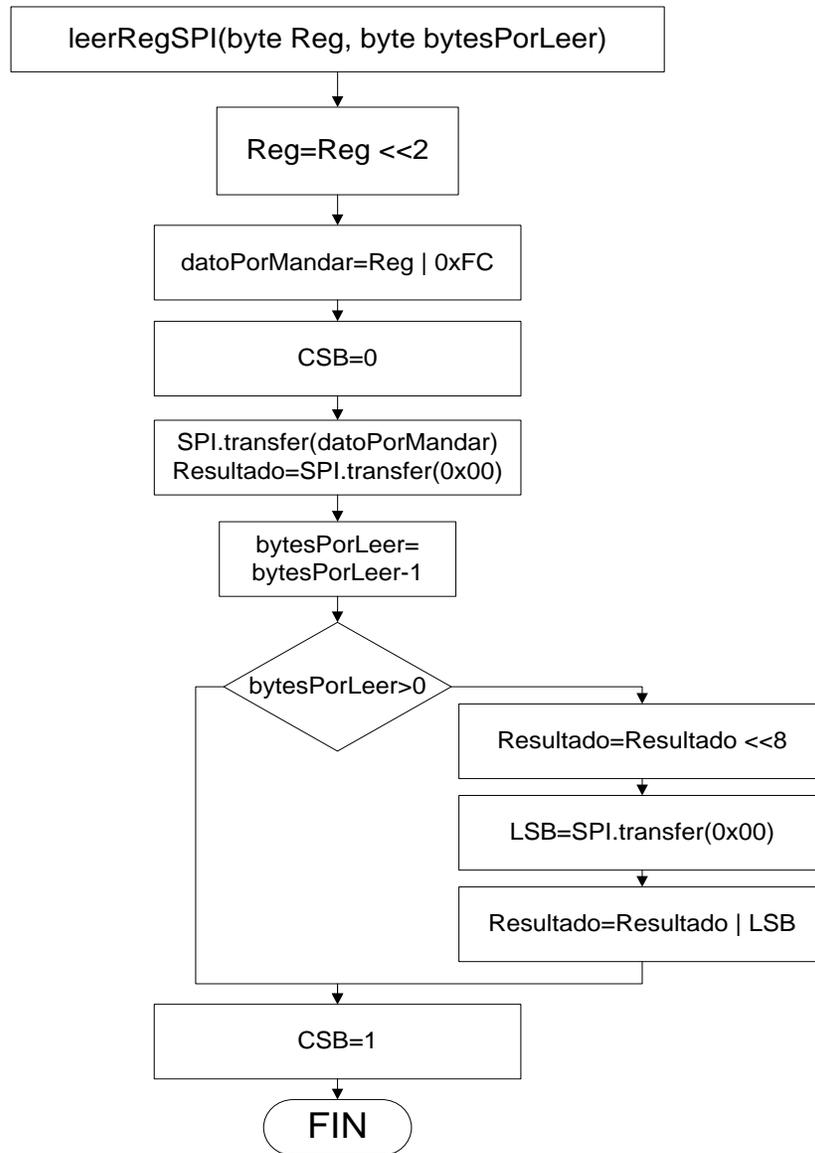


Figura 4.23 Algoritmo utilizado para la escritura SPI

Para la lectura de la presión barométrica se escribió el método Presión() haciendo uso de los métodos descritos anteriormente.

El SCP1000 es calibrado directamente en la fábrica. Los parámetros de calibración se encuentran almacenados en una memoria no volátil y son extraídos automáticamente por el procesador interno del sensor durante la etapa de inicialización.

El sensor posee múltiples modos de operación, según los requerimientos de la aplicación. Como para el caso particular el tiempo de reacción no es crítico, se seleccionó el modo de alta resolución que permite obtener las mediciones disponibles de mayor

precisión. El modo de operación deseado (alta resolución) se selecciona escribiendo el valor 0x0A en el registro de operación que tiene la dirección lógica 0x03.

Los valores medidos de presión barométrica se almacenan automáticamente en los registros de salida. Para obtener el resultado, se lee primero el registro 0x1F, de donde se obtienen los *bits* más significativos del resultado real. El valor leído se almacena en la variable PresionH. El resto del *byte* no se ocupa y, por lo tanto, se pone en cero lógico a través de una operación AND con 0x07. Los otros 2 *bytes* del resultado se obtienen del registro con la dirección 0x20 y se almacenan en la variable PresionL. El resultado real se obtiene desplazando la variable PresionH 16 *bits* hacia la izquierda y combinándola con la variable PresionL. Por último, se divide el resultado entre 4 por requisito del fabricante y se almacena el resultado final en la variable Presión. Al igual que en los otros sensores, se utiliza la condición lógica "nuevo" para definir si la medición realizada se ocupa para uso interno o se debe transmitir.

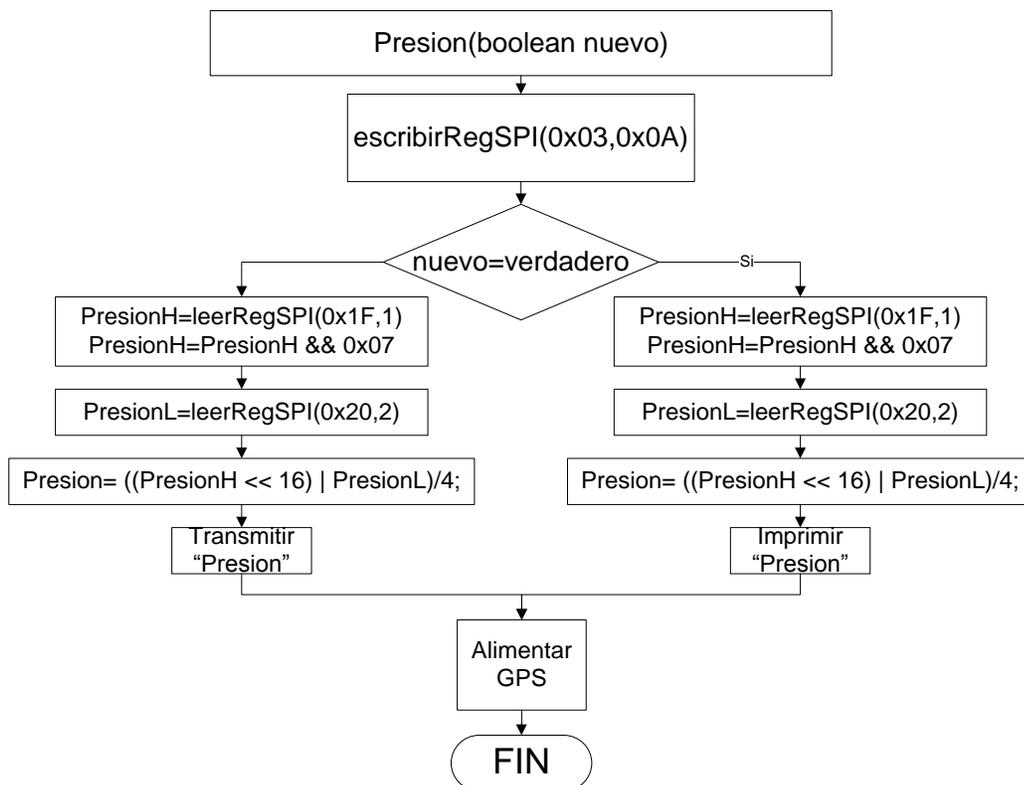


Figura 4.24 Algoritmo utilizado para la lectura del sensor

4.1.2.7 Sensor de humedad

El HIH-4030 es un sensor analógico compacto que permite medir la humedad relativa del ambiente con alta precisión.



Figura 4.25 HIH-4030

El sensor representa la humedad relativa a través de una salida analógica de tensión con una curva altamente lineal, lo que hace posible que dicha salida pueda alimentar directamente a uno de los puertos analógicos del micro-controlador maestro que interpreta las mediciones.

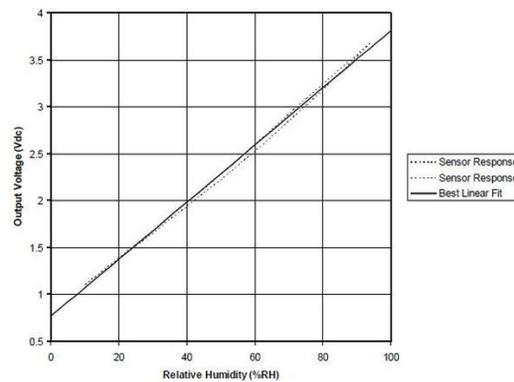


Figura 4.26 Curva de tensión versus humedad del sensor HIH-3030 (extraída de la hoja de datos)

El sensor requiere muy poca corriente para operar, tan solo 200 μ A, lo que lo hace apto para aplicaciones móviles alimentadas con baterías. El sensor puede operar hasta en temperaturas de -40 °C.

Tabla 7 Parámetros del HIH-4030 (Hoja de datos)

Operating temperature	-40[-40]	See Figure 1.	85[185]	°C[°F]	-
Operating humidity (HIH-4030)	0	See Figure 1.	100	% RH	5

4.1.2.7.1 Principio de operación

El sensor utiliza un polímero capacitivo termo-estable como transductor. La permeabilidad del dieléctrico en el capacitor varía linealmente con la humedad del ambiente, resultando en una variación de la capacitancia del transductor. Dicha variación se mide con un circuito acondicionador y el resultado es presentado como una tensión eléctrica, que es la salida del sensor. La construcción multi-capa del transductor permite la operación del sensor en ambientes de trabajo hostiles.

4.1.2.7.2 Conexión eléctrica

El sensor únicamente requiere para operar de una conexión con el sistema de alimentación de potencia. La tensión de salida se conecta directamente al puerto 1 del controlador maestro (ADC(1)). Se coloca un condensador adicional para filtrar el ruido eléctrico y una resistencia de carga, en paralelo con la salida del sensor, para evitar desacoples de impedancia.

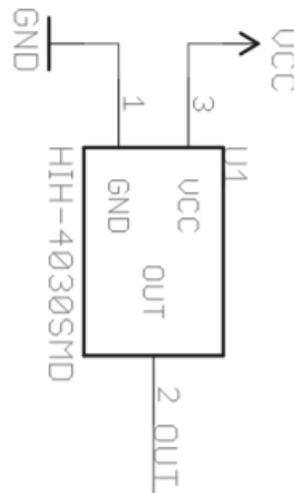


Figura 4.27 Conexión eléctrica del sensor de humedad HIH-4030

4.1.2.7.3 Programación

Para obtener mediciones de humedad relativa mediante el sensor HIH-4030, se utiliza el puerto analógico 1 del micro-controlador maestro (ADC(1)). El sensor no requiere una configuración inicial para operar debido a su naturaleza analógica, sin embargo, en el micro-controlador se debe iniciar la operación del puerto analógico correspondiente.

El puerto ADC(1) del micro-controlador maestro cuenta con una resolución de 10 *bits* y permite una tensión máxima de 5 V que se mapea en el número 1023 por lo tanto, se debe realizar un acondicionamiento matemático para poder hacer cálculos correctos e interpretar correctamente la medición. El valor ajustado de la medición se almacena en la variable *Valormap*:

$$Valormap = \frac{sHumedad * 5}{1023}$$

Las mediciones del sensor utilizado se ven afectadas por la tensión de alimentación y por la temperatura de operación. Estos parámetros deben ser incluidos en el cálculo de la humedad relativa con el fin de obtener mediciones con un bajo porcentaje de error.

El fabricante brinda una fórmula para incluir el efecto de la tensión de alimentación en la medición. Con base en mediciones experimentales de la tensión de alimentación proporcionada por el Sistema de Alimentación de Potencia, se utiliza como parámetro *Valim*=4.95 V. Se despeja de la fórmula la variable *SensorRH*, que representa el porcentaje de humedad relativa en el ambiente a una temperatura de 25°C.

$$Valormap = (Valim)(0.0062(SensorRH) + 0.16)$$

$$SensorRH = \frac{Valormap - 0.958}{0.0307} \text{ con } Valim = 4.95 \text{ V}$$

Posteriormente, se debe compensar el efecto de la temperatura ambiental en la medición del sensor, esto se realiza mediante un ajuste matemático. Se recurre al sensor de temperatura exterior y al método *LeeTemp()* para obtener el parámetro "T", que representa la temperatura ambiental en °C.

$$RHreal = SensorRH(1.0546 - 0.00216T) \text{ con } T = \text{Temperatura en } ^\circ\text{C}$$

Por último se utiliza la variable lógica "nuevo" para determinar si la medición de humedad relativa se requiere para uso interno ó debe ser transmitida y se alimenta el GPS.

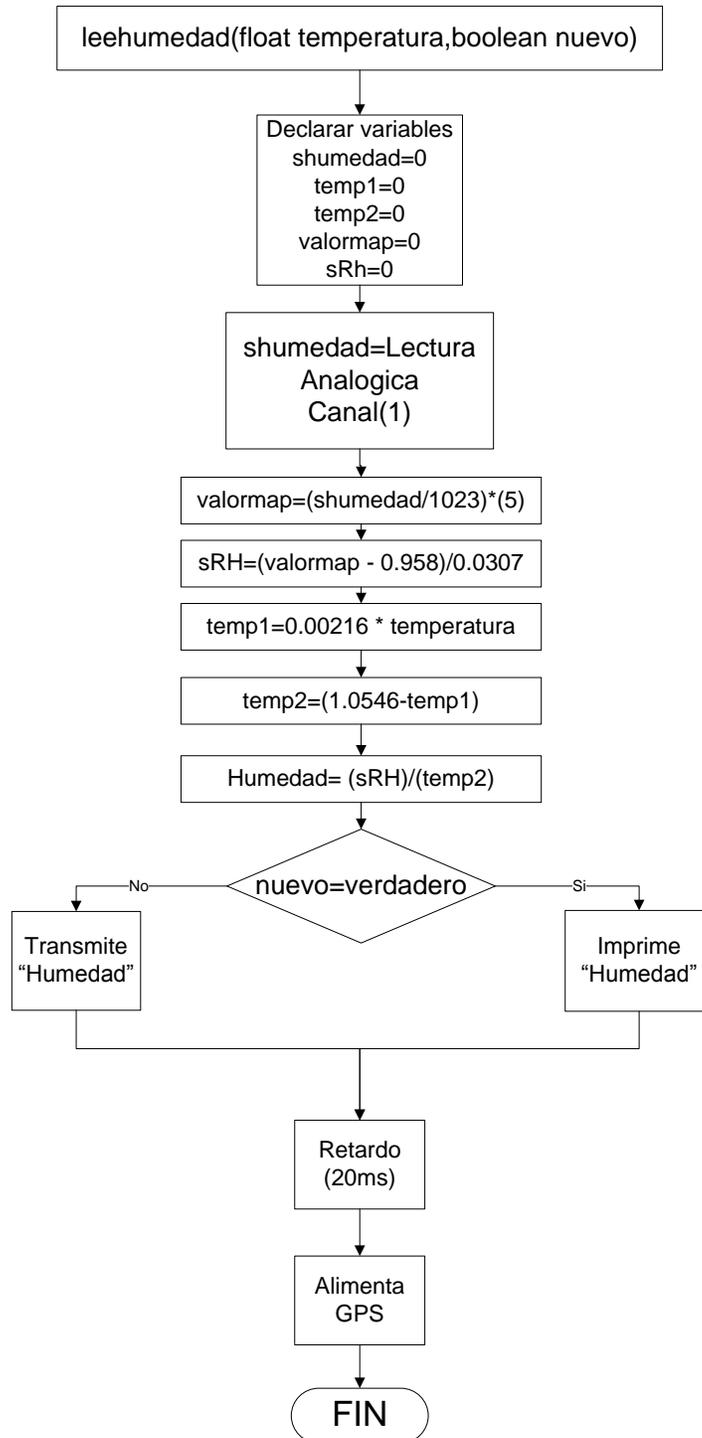


Figura 4.28 Método utilizado para la lectura del sensor HIH-4030

4.1.2.8 Almacenamiento de los datos en memorias tipo SD

Para el almacenamiento de las mediciones, se seleccionó el integrado Openlog del fabricante Sparkfun, que permite almacenar en una memoria extraíble SD, archivos de tipo variable según sea la necesidad del usuario.



Figura 4.29 Openlog

El integrado OpenLog es un *datalogger* basado en un micro-controlador Atmel ATmega328 operando a 16MHz. Cuenta con una interface serial que permite enviar y recibir datos provenientes de otro bloque electrónico. El OpenLog permite el manejo de memorias SD de alta densidad (mayores a 2GB), lo que lo hace apropiado para ser utilizado cuando se requiere almacenar una gran cantidad de datos. En operación normal consume 2 mA y puede operar hasta en temperaturas de -30° C. Para su programación, cuenta con un *bootloader* serial estándar (stk500v1).

4.1.2.8.1 Conjunto de instrucciones

El OpenLog cuenta con un conjunto de instrucciones que permiten gobernar la operación de almacenamiento de datos. Estas instrucciones flexibilizan el funcionamiento del componente de tal forma que se pueda integrar fácilmente en una aplicación concreta. Para utilizar estas instrucciones, el OpenLog debe estar en modo de configuración (*Command Mode*). El conjunto de instrucciones reducido se presenta a continuación:

- ✓ **New "Nombre" ".ext"**: Esta instrucción permite la creación de un nuevo archivo. El campo "Nombre" se reemplaza por el nombre del archivo y el campo ".ext" se reemplaza por la extensión del archivo que da el formato. Por ejemplo: la instrucción "new Texto.txt" crea un nuevo archivo de texto en la memoria SD con el nombre Texto.txt.
- ✓ **Append "Nombre" ".ext"**: Esta instrucción permite abrir un archivo existente y escribir en él de manera secuencial. Si el archivo ya ha sido escrito con anterioridad, la escritura empieza al final del archivo existente. El campo "Nombre" y ".ext" deben coincidir con el archivo creado mediante la instrucción

"New()". Por ejemplo: la instrucción "append Texto.txt" abre el archivo de texto almacenado en la memoria SD y prepara el "buffer" interno de memoria para recibir y escribir en la memoria SD los datos.

- ✓ **Modo de configuración ("Command Promt"):** En este modo, el Openlog entra en modo de configuración, donde es posible utilizar los comandos para manipular archivos, configurar la transmisión serial de datos y establecer los códigos que permiten entrar en este modo. Por defecto, la tecla Ctrl+z se utiliza para entrar y salir del modo de configuración, sin embargo se puede configurar el dispositivo para que utilice una nueva tecla para este fin. Para entrar en modo de configuración, el OpenLog debe recibir el comando de escape (Ctrl+z por defecto) 3 veces consecutivas. Para salir sólo es necesaria una vez.

4.1.2.8.2 Conexión eléctrica

El OpenLog cuenta con una interface serial asincrónica. Los pines RXI(Receptor) y TXO (Transmisor) se conectan directamente a la UART(3) del micro-controlador maestro. La patilla GRN se conecta al RESET del micro-controlador, con el fin de sincronizar la operación del dispositivo con el flujo del programa principal del sistema.

4.1.2.8.3 Programación

Para almacenar la información medida en el computador de vuelo, se decidió utilizar el formato *.CSV (Comma Separated Values)*. Este formato permite crear de manera simple bases de datos basadas en texto. Este tipo de formato es compatible con el paquete de *software* EXCEL de Microsoft, que cuenta con una serie de herramientas para realizar estudios matemáticos y gráficos, de tal forma que se pueda aprovechar al máximo la información medida. En dicho formato, se utiliza el carácter ";" como separador de columnas y la tecla CR (*Enter*) como separador de filas. Por ejemplo, un archivo *.CSV* que contenga el texto...

Hora; Temperatura

10:50; 25°C

...es interpretado en EXCEL como una tabla de esta forma:

Hora	Temperatura
10:50	25°C

Entonces, se puede presentar la información medida por los sensores en una manera conveniente, utilizando la hora como variable indirecta en las tablas.

Para crear un archivo .CSV en la memoria SD administrada por el OpenLog, se utilizan las instrucciones "New" y "append". Para asignar los nombres de los archivos se utiliza la función `random()` y la función `randomSeed()` de la librería `math.c`, para que cada vez que se cree un archivo nuevo, tenga un nombre diferente y no sobrescriba uno ya existente. Se utiliza el ruido eléctrico del canal analógico ADC(3) que está desconectado como parámetro de la función `randomSeed()`.

Luego de crear el archivo, se escriben en él los títulos de las columnas y se entra en modo secuencial para empezar con el proceso de captura y almacenamiento de las mediciones, que es administrado por el micro-controlador maestro utilizando los distintos métodos para el manejo de los sensores.

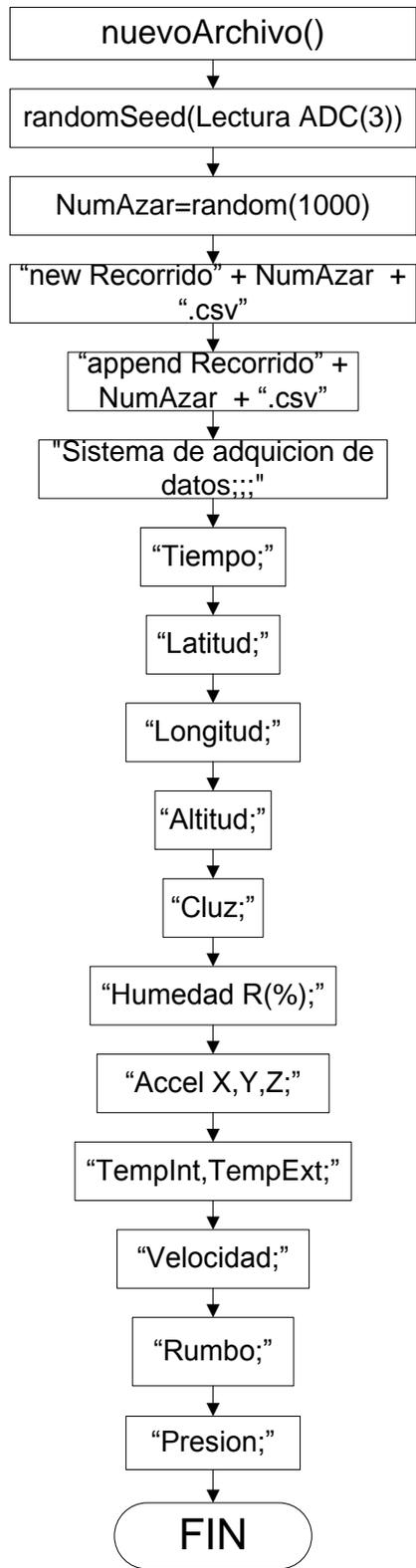


Figura 4.30 Método utilizado para la creación de archivos con formato .CSV

Para grabar en los archivos se escribió el método Grabar(), que simplemente invoca de manera secuencial los algoritmos encargados de manejar los sensor. En cada cambio de paso en el algoritmo, se cambia de columna y, en cada iteración total, se cambia de fila.

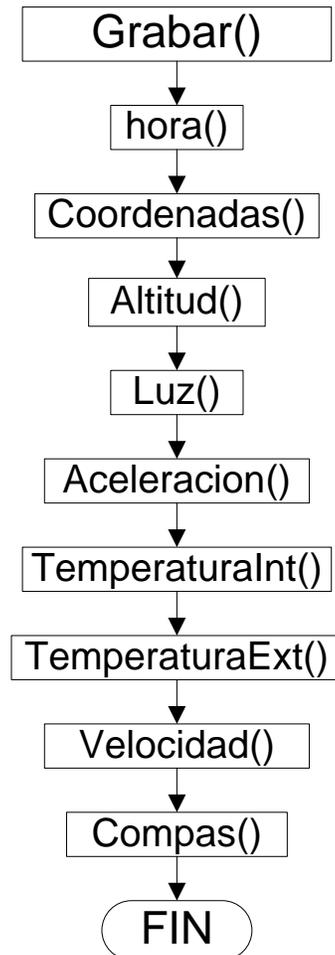


Figura 4.31 Método utilizado para grabar en los archivos creados

4.2 Sistema de Recuperación y Rastreo (SRR)

El globo meteorológico que será utilizado para transportar el SCEIGA no incluirá un mecanismo de control de recorrido y es altamente dependiente de las condiciones atmosféricas presentes el día del lanzamiento, especialmente de la velocidad y de la dirección del viento. Este factor se traduce en la imposibilidad de predecir con exactitud la trayectoria y la posición de aterrizaje de la sonda.

El problema descrito se solucionó con la incorporación del Sistema de Recuperación y Rastreo (SRR), que es un bloque funcional del SCEIGA capaz de exponer la información adquirida por la sonda en un medio de comunicación masivo (Internet) para que así cualquier usuario en el mundo pueda seguir el recorrido en tiempo real.

El SRR consiste en un arreglo de dos subsistemas divididos físicamente. El bloque A está ubicado en la plataforma y tiene la función principal de adquirir las coordenadas de la sonda y transmitir las, sin embargo sus recursos también son empleados para adquirir y transmitir otra serie de variables físicas como la hora, la velocidad de ascenso y la altitud de la sonda. El bloque consiste en un módulo GPS, que recibe constantemente sus coordenadas, y un transmisor de radio FM, que opera a 900Mhz, con la potencia necesaria para establecer un enlace de comunicación con una estación terrestre ubicada en una distancia inferior a 30 km de la sonda.

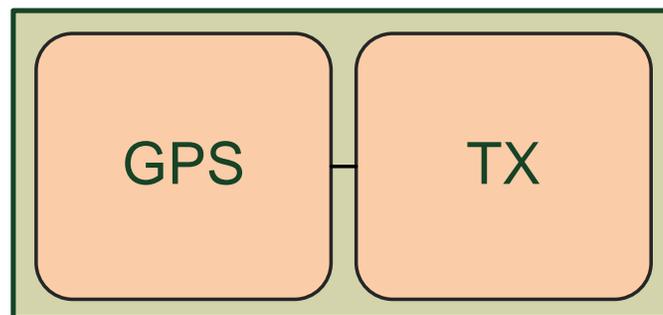


Figura 4.32 Diagrama de bloques del Sistema de Recuperación y Rastreo (SRR), parte A

En la estación terrestre se encuentra el bloque B del SRR que se encarga de recibir la información transmitida por el bloque A, procesarla y hacerla pública a través de la Internet. El bloque está compuesto por una tarjeta de evaluación Arduino Diemilanove, un transmisor de radio FM similar al utilizado en el bloque A y un módulo de expansión

Ethernet (*WebShield*) que dota a la tarjeta con acceso a Internet. Se utiliza la red social Twitter como plataforma para publicar la información medida por la sonda y Google Maps para ubicar las coordenadas de la sonda en el mapa mundial. El micro-controlador embebido en la tarjeta de evaluación administra la operación de este subsistema.

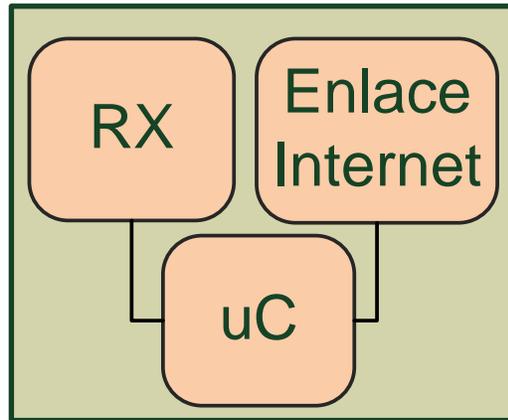


Figura 4.33 Diagrama de bloques del Sistema de Recuperación y Rastreo (SRR), parte B

El SRR permite observar la información recopilada por la sonda desde cualquier dispositivo con acceso a Internet, por ejemplo, un teléfono móvil o un computador portátil.

4.2.1 GPS (coordenadas, altitud, velocidad)

El GPS es la base de la función crítica del computador de vuelo, es decir, la recuperación de la sonda y de la información adquirida. Por esta razón, se seleccionó un módulo Locosys LS20031 que puede recibir información de 66 satélites simultáneos, a una frecuencia de 5 Hz.



Figura 4.34 GPS LS20031

Este módulo cuenta con una antena de cerámica empotrada, de manera que no requiere *hardware* adicional para operar. El LS20031 puede operar hasta en temperaturas de -30 ° C.

El módulo utiliza el protocolo NMEA para presentar la información adquirida a través de una interface de comunicación serial asincrónica. Este dispositivo brinda las coordenadas, altitud y la velocidad de la sonda.

4.2.1.1 Conexión eléctrica

El LS20031 presenta una interface serial asincrónica y solo requiere la conexión de los pines TX y RX para operar, no obstante el dispositivo se alimenta con 3.3 V, por lo que requiere un convertidor de nivel para comunicarse con el micro-controlador maestro que opera a 5 V, de otra manera podría averiarse y comprometer la función del sistema. Las señales acondicionadas por el convertidor de nivel se conectan a la UART(1) del micro-controlador maestro. Inmediatamente después de encender el dispositivo, este empieza a recibir información del servicio GPS y a triangular las coordenadas iniciales.

El LS20031 posee una memoria interna no volátil para almacenar información de la última localización conocida del dispositivo, con el fin de que el período inicial de adquisición de las coordenadas se reduzca. Este proceso se denomina arranque caliente (*Hot Start*). Si el dispositivo se utiliza en una localización lejana a la última almacenada en la memoria, se deben adquirir las coordenadas iniciales nuevamente, lo que causa una demora considerable antes de que se pueda extraer información válida del sensor. Esto se conoce como arranque frío (*Cold Start*). Cuando el receptor está operando y la información suministrada es reciente y válida, enciende un LED rojo intermitente cada minuto.

4.2.1.2 Programación

Para incorporar el receptor en el SRRA, fue necesario considerar varios aspectos técnicos del receptor GPS.

La interface serial asincrónica del LS20031 opera a una velocidad de transmisión de 57600 baudios, a diferencia de los receptores GPS convencionales que transmiten a 9600 baudios, por esta razón se debe modificar la configuración de la UART1 del controlador maestro para adecuarse a esta condición.

Otro aspecto importante es que el receptor utilizado se actualiza con una frecuencia de 1 Hz, que es mucho más lenta que la frecuencia de operación del micro-controlador maestro y, por lo tanto, extraer los datos con un régimen de lecturas periódicas (*Polling*)

sería un desperdicio de los recursos del controlador maestro. Para solucionar este problema se utilizó la interrupción del *buffer* lleno de la UART(1), de modo que, cada vez que se recibe información nueva, se activa una bandera correspondiente que le permite al programa principal procesar la información sin necesidad de bloquear el flujo del programa.

El LS20031 codifica la información del servicio GPS mediante el protocolo estándar NMEA. Dicho protocolo está basado en sentencias encabezadas por una palabra abreviada que indica el tipo de información. El inicio de una nueva sentencia se denota con el signo "\$". Por ejemplo en la sentencia:

```
$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,*47
```

\$GPGGA representa que se trata de información reciente, 123519 es la hora en la cual se recibió el dato, 4807.038,N es la latitud, 01131.000,E es la longitud, el 1 es la exactitud de la información, 08 es la cantidad de satélites vistos, 0.9 representa la dilución de la precisión horizontal, 545.4,M es la altitud en metros y *47 se utiliza para verificar la integridad de la sentencia.

Para decodificar las sentencias en el controlador maestro se utilizó la librería de fuente abierta TinyGPS.h, que permite extraer la información más pertinente del receptor sin comprometer los recursos de *hardware* del micro-controlador.

Para administrar la adquisición de datos GPS, se escribió el método `alimentaGPS()`, que permite verificar si hay datos recientes y decodificar, de ser necesario, las sentencias NMEA recibidas. Cuando la pila de memoria de la UART(1) se llena, se procede a decodificar las sentencias a través de la instrucción `GPS.Encode()` de la librería TinyGPS.h. y se activa una bandera lógica indicando la disponibilidad de información reciente.

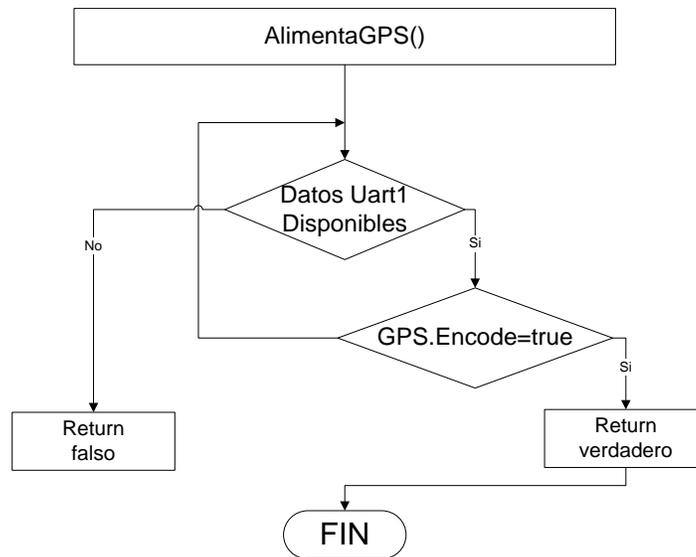


Figura 4.35 Método de adquisición y decodificación de las sentencias NMEA

La información decodificada se almacena en variables dinámicas y se lee a través de punteros para minimizar la cantidad de memoria requerida.

La librería utilizada no maneja el formato de los números con punto flotante, por lo tanto se escribió el método `imprimeFloat()` para insertar la coma y realizar el redondeo de un número que así lo requiera. El método recibe como parámetro el número que requiere formato y la cantidad de decimales deseados. Inicialmente se verifica si es un número positivo o negativo y se imprime el signo "-" si se necesita. Se define el umbral para el redondeo en 0.5. Seguidamente se calcula el restante que se debe sumar para redondear la parte entera del número y se adiciona al número. Luego se extrae el residuo restando el número en formato decimal con el mismo número pero en formato entero. El número se convierte en entero mediante la operación `atoi()` y se almacena en la variable `P.entera`. Posteriormente se imprime el valor de dicha variable y la "," si el parámetro "dígitos" es diferente de cero. Por último se redondea el residuo y se imprime cada dígito, según la cantidad deseada de decimales que fue establecida como parámetro.

variable lógica "nuevo" para determinar si el dato se requiere para uso interno o se debe transmitir.

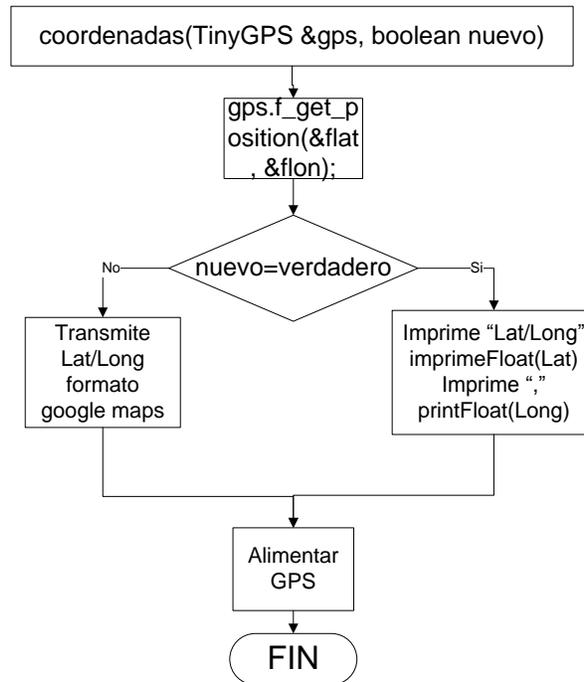


Figura 4.37 Algoritmo de impresión de las coordenadas del sistema

Los métodos Altitud() y Velocidad() utilizan las instrucciones gps.f_altitude() y gps.f_speed_kmph() de la librería TinyGPS.h. Aquí también se utiliza la variable "nuevo" para determinar si los datos se requieren para uso interno o se deben transmitir.

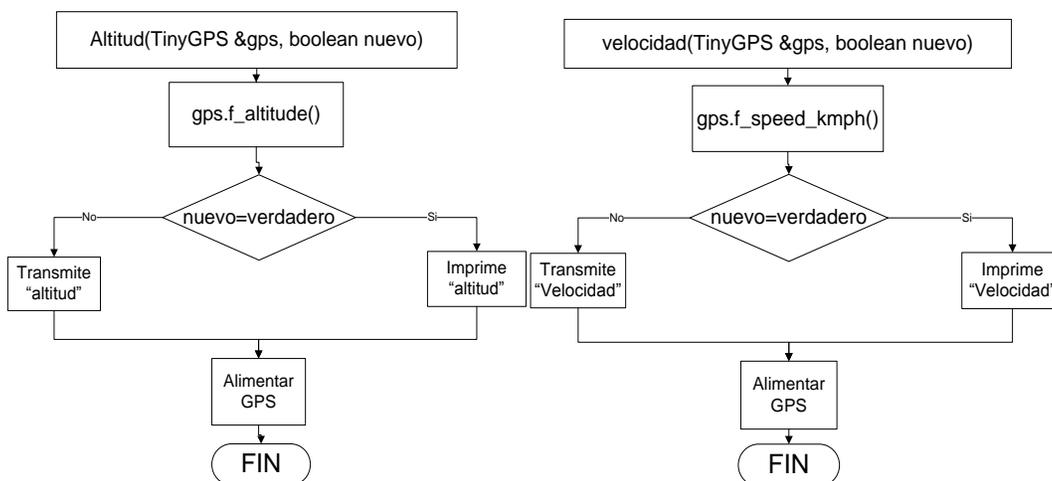


Figura 4.38 Métodos utilizados para la impresión de la velocidad y altitud

Para agregar un reloj de tiempo real al sistema (*RTC*), se escribió el método *Hora()*, que utiliza los recursos del sistema GPS para obtener la hora mundial con una precisión de milisegundos. El método utiliza la instrucción `gps.crack_datetime()` de la librería *TinyGPS.h*. Para calcular la hora en la zona horaria de Costa Rica, se suman o restan 6 horas dependiendo de si ya ha pasado o no la media noche en el reloj mundial (*GMT*). La variable "nuevo" se utiliza para determinar si el dato se requiere para uso interno o se debe transmitir.

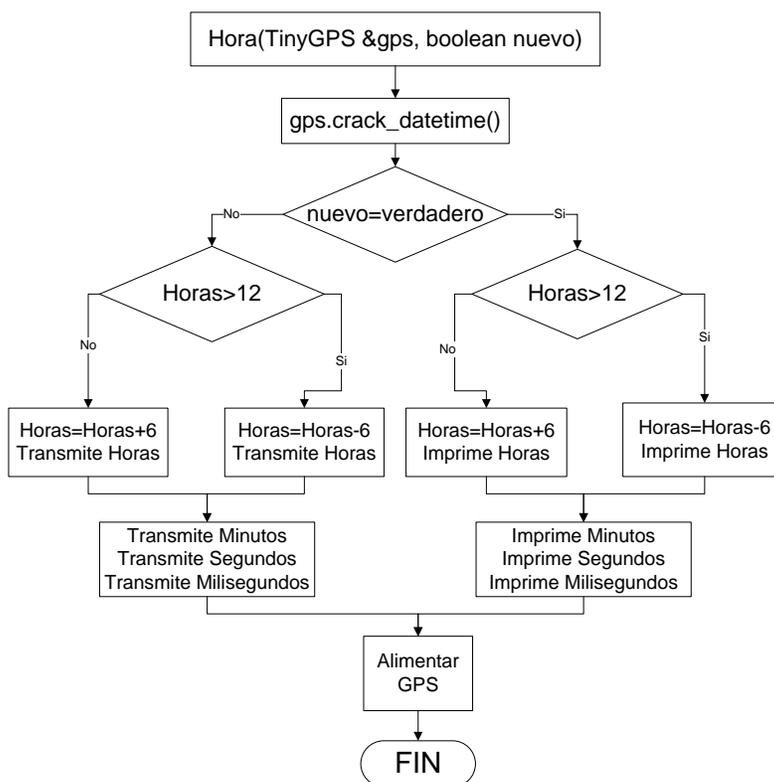


Figura 4.39 Método utilizado para la impresión de la hora

Por último, se escribió el método *Fecha()* que, de manera similar, utiliza los recursos del sistema GPS para obtener la fecha. El método utiliza la instrucción `gps.crack_datetime()` de la librería *TinyGPS.h* para decodificar la información correspondiente al día, el mes y el año. En concordancia con el resto de los algoritmos, la variable "nuevo" se utiliza para determinar si el dato se requiere para uso interno o se debe transmitir.

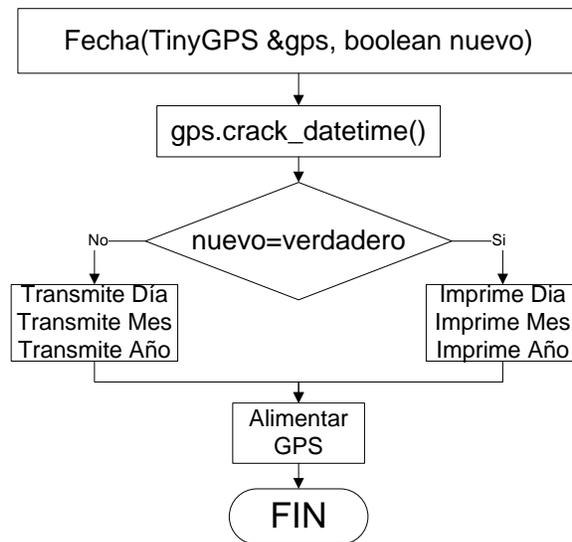


Figura 4.40 Método utilizado para la impresión de la fecha

4.2.2 Transmisor/Receptor

Para comunicar la plataforma con la estación terrestre, se utiliza un enlace por radiofrecuencia. Se seleccionó el transceiver Xtend 900 del fabricante Digi para implementar el enlace. Se requiere uno en cada sección del SRR.



Figura 4.41 Xtend 900

Los módulos Xtend operan en la banda de frecuencia libre ISM 900 Mhz ("banda industrial, científica y médica") cuyo uso es libre de licencia en Costa Rica, por lo tanto no es necesario solicitar permisos para utilizarla. Los módulos se comunican con los microcontroladores a través de una interface serial asincrónica capaz de transmitir a una velocidad máxima de 115.2 kbps. La velocidad de transmisión es variable y puede ser elegida por el usuario mediante *software*. Es importante mencionar que, entre mayor sea la velocidad de transmisión utilizada, mayor es la tasa de errores de comunicación y, por ende, menor el alcance máximo del enlace. Por ejemplo, para una velocidad de 115.2 kbps, el alcance máximo con línea de vista es de 11 km con una antena de dipolo. Con una

velocidad de 9600 kbps, el alcance máximo aumenta a 22 km con el mismo tipo de antena. El prototipo implementado únicamente transmite mediciones, por lo tanto el ancho de banda requerido es corto y se puede utilizar una velocidad de transmisión relativamente lenta sin comprometer el desempeño del bloque.

Los módulos Xtend utilizan algoritmos de compresión y modulación digital para obtener el alcance propuesto, sin requerir más potencia que la máxima permitida para un transmisor que opera en la banda ISM 900 Mhz (1 W).

También puede operar hasta en temperaturas de -40°C , de forma que podría ubicarse, de ser necesario, fuera de la sonda protectora para garantizar así una señal fuerte.

En la sección del SRR ubicada en la plataforma, se le acopla al módulo Xtend una antena de monopolo para 900 Mhz. Esta antena cuenta con una ganancia de antena de 2 dbi y tiene la ventaja de ser muy liviana y compacta, de forma que no se compromete la carga útil de la sonda.



Figura 4.42 Antena de monopolo acoplada a los transmisores

En la sección del SRR ubicada en la estación terrestre, se acopla una antena similar a la anterior para el prototipo, sin embargo se recomienda utilizar una antena de alta ganancia para incrementar el alcance total del enlace. Por ejemplo, con una antena Yagi-uda acoplada al módulo de la estación terrestre, el alcance en línea de vista se extiende a 64 km.

Los módulos contienen *buffers* de memoria para la entrada y salida de datos. Los datos por transmitir que ingresan al módulo por el bus se almacenan en el *buffer* de entrada hasta que puedan ser procesados. El tamaño del *buffer* se define por *software*. Cuando el *buffer* se llena, se inicia la comunicación por radiofrecuencia. Los datos recibidos por radiofrecuencia entran en el *buffer* de salida. Cuando este se llena, la información se transmite al micro-controlador por el bus serial y cualquier dato adicional se pierde. Ambos *buffers* pueden almacenar al menos 2.1 kbytes de datos.

4.2.2.1 Principio de operación

El módulo Xtend 900 opera en la banda de frecuencia de 900 Mhz, espacio libre de licencia reservado para aplicaciones industriales, médicas y científicas. Como la banda es libre, se corre el riesgo de experimentar interferencia electromagnética producida por otro emisor de ondas de radio moduladas en 900 Mhz. Para reducir la interferencia, el módulo cuenta con la tecnología FHSS (*Frequency Hopping Spread Spectrum*), que consiste en variar la frecuencia de la onda portadora en una secuencia establecida que es conocida únicamente por el transmisor y el receptor de un enlace particular. Esta tecnología causa que la frecuencia de la onda portadora varíe entre 902 Mhz y 928 Mhz. La secuencia de variación de la portadora se define mediante programación, durante la configuración inicial de los módulos pertenecientes al enlace.

El radio utiliza modulación digital por frecuencia (*FSK*) para centrar la frecuencia de la portadora en 900 Mhz.

El sistema Xtend soporta varias topologías de red para acomodar el enlace a la aplicación en desarrollo. Con el propósito de implementar el prototipo, se utiliza la topología básica punto a punto, ya que solo se dispone de dos módulos en la fase actual del proyecto. Para cada módulo se programa una dirección fuente y una dirección destino. La dirección destino de un módulo es la dirección fuente del otro y viceversa.



Figura 4.43 Topología de red punto a punto

Conforme se requiera un enlace más robusto, se pueden agregar más módulos configurados como repetidores o implementar una topología puerto a puerto (*P2P*).

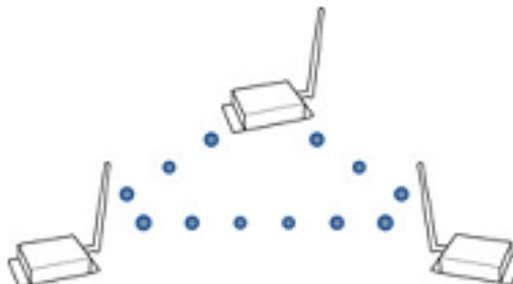


Figura 4.44 Topología de red puerto a puerto

Para asegurar la información, el sistema Xtend dispone de encriptación de datos AES de 128 bits y 256 bits.

4.2.2.2 Conexión eléctrica

Los módulos Xtend 900 utilizan una interface serial asincrónica para intercambiar información con un componente maestro que gobierna la comunicación. Para el caso propio del prototipo implementado, se utilizan micro-controladores como dispositivos maestros.

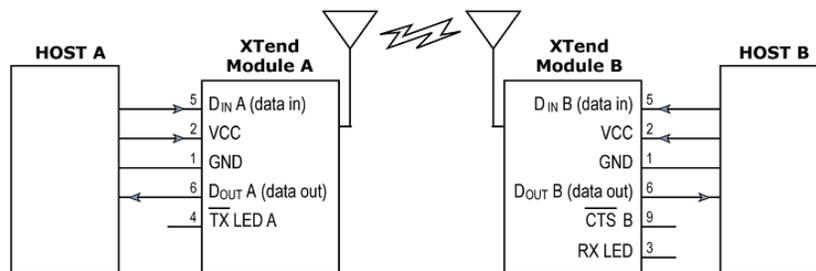


Figura 4.45 Esquema de conexión de los módulos Xtend, según la topología de red punto a punto

En el SRRA, el módulo Xtend se conecta directamente con el puerto UART(3) del micro-controlador maestro. El bus se configura para operar a 9600 baudios mediante programación. En el SRRB, el módulo se conecta con el puerto UART(0) del maestro y se configura el bus para operación a 9600 baudios.

El encendido y apagado de los módulos se puede efectuar a nivel lógico mediante la señal de entrada SHDN(*Shut down*). Dicha entrada permite apagar los módulos si su uso no es requerido con el fin de reducir el consumo de potencia total del sistema. Se utiliza una salida digital de las tarjetas de evaluación para administrar dicha señal.

Para configurar los módulos, se deben conectar directamente a un ordenador. Se utiliza un integrado convertidor USB a serie (FTDI) para efectuar el acople entre los buses de comunicación.

4.2.2.3 Programación

Los módulos Xtend utilizados en el enlace se deben configurar antes de conectarlos con el sistema electrónico al que pertenecen. Los parámetros de operación se almacenan en

memorias no volátiles, por lo que solo se requiere configurar los módulos una vez, a menos que se desee cambiar la función de algún módulo. El fabricante provee el paquete de *software* X-CTU, que permite reescribir la memoria de configuración según sea necesario.

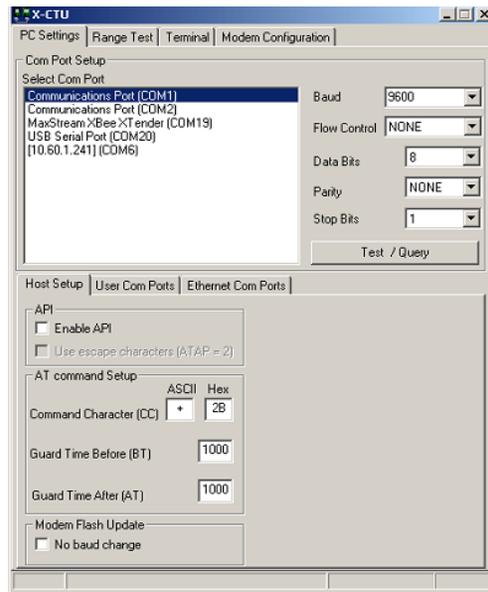


Figura 4.46 Software de configuración de los módulos Xtend (X-CTU)

El programa cuenta con una interfaz gráfica para facilitar la configuración de los puertos del ordenador. Los parámetros del puerto deben coincidir con los parámetros de las UART de los módulos. Por defecto, dichos parámetros son los siguientes:

- 9600 baudios
- Sin control de flujo
- 8 bits de datos
- Sin paridad
- 1 bit de parada

El paquete X-CTU cuenta con una terminal que permite modificar los parámetros de operación de los módulos. Los cambios requeridos se escriben en la terminal con base en una tabla de comandos proporcionada por el fabricante.

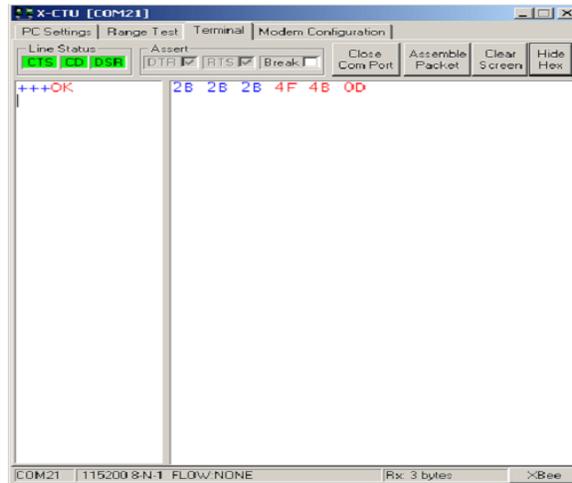


Figura 4.47 Terminal de configuración (X-CTU)

El módulo compila las instrucciones recibidas y responde de acuerdo con la validez y el tipo de comando.

Con respecto a la configuración para los módulos utilizados en el prototipo, se establecieron los parámetros de operación recomendados por el fabricante para la aplicación desarrollada. El proceso es como sigue:

1. Se entra en modo configuración enviando el comando "+++", lo que detiene el funcionamiento del módulo.
2. Se establece la potencia de transmisión de los módulos en el valor máximo permitido (1W).
3. Se asignan las direcciones destino y fuente para cada módulo. Las direcciones son de 64 *bits* y se separan en parte alta y parte baja, cada una de 32 *bits*. Las direcciones se eligen en forma aleatoria, sin embargo la dirección destino de un módulo debe ser la misma que la dirección fuente del otro módulo y viceversa.

$$\text{DirecciónDSRRA} = (0x0000000000000001)$$

$$\text{DirecciónFSRRA} = (0x0013A20000000000)$$

$$\text{DirecciónDSRRB} = (0x0013A20000000000)$$

$$\text{DirecciónFSRRB} = (0x0000000000000001)$$

La dirección de red también se debe definir y debe ser la misma para ambos módulos. Se utilizó la dirección 0x3332.

4. Se selecciona la secuencia de brinco "0" en ambos módulos para configurar la operación del sistema FHSS.
5. Se configuran los parámetros de la interface serial asincrónica.
6. Se programa la velocidad de transmisión en 9600 baudios, 8 *bits* de datos, sin paridad y 1 *bit* de parada.
7. Se guardan los cambios en la memoria no volátil y se sale del modo de configuración, reanudando la operación de los nódulos en modo normal.

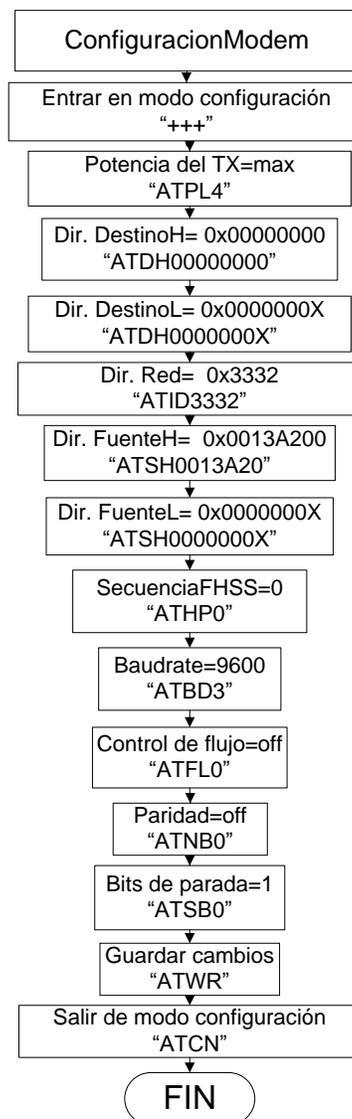


Figura 4.48 Secuencia de configuración de los módulos Xtend

4.2.3 Ethernet Shield

Para hacer pública la información capturada, se le agregó una tarjeta de expansión *Ethernet Shield* al micro-controlador maestro del SRRB. Dicha tarjeta incluye un controlador de Ethernet que dota al micro-controlador con la capacidad de conectarse con la Internet y de utilizar sus facilidades para permitir la comunicación masiva de información y que así cualquier usuario a nivel mundial obtenga en tiempo real las mediciones efectuadas en la plataforma.



Figura 4.49 Controlador Ethernet W5100 empotrado en un escudo de Internet para Arduino

El proceso de comunicación de la información medida en la plataforma está dividido en varias etapas, según la naturaleza de los canales que se usan para transmitirla. La información capturada en la plataforma es transmitida al SRRB a través del enlace de radiofrecuencia. El micro-controlador se encarga de almacenar la información en una pila de memoria y de darle formato para luego pasarla al siguiente bloque y publicarla en la red con el controlador de Ethernet. El flujo de la transmisión de datos es controlado por el micro-controlador en el SRRB.

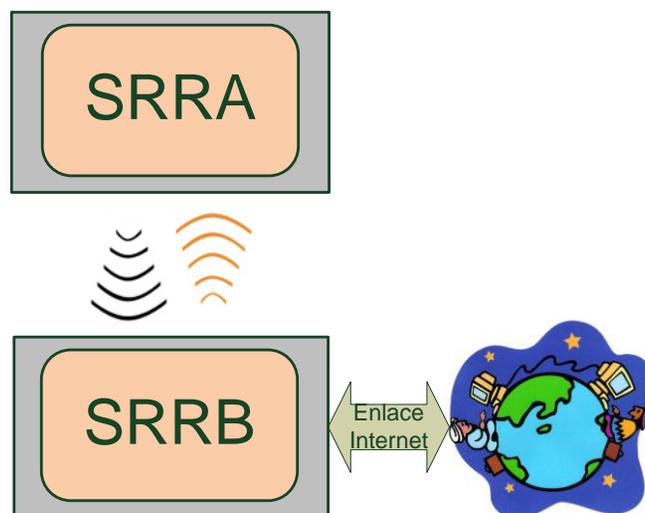


Figura 4.50 Flujo de la transmisión de datos

Para publicar la información en internet, se utiliza la red social Twitter, a la cual pertenecen millones de personas a nivel mundial. La red social funciona con base en perfiles y seguidores. Para publicar mensajes en la red solo es necesario crear un perfil que describa al usuario y que puede ser representado por una persona real, un grupo o empresa, un sistema electrónico, etc. Cada usuario puede elegir a quien "seguir". Los mensajes publicados por cada usuario seguido se agrupan en orden cronológico en la ventana de inicio de la red social, lo que permite conocer el estado de todos los usuarios en un solo sitio.

Para la aplicación en desarrollo, se creó un perfil del SCEIGA. El SRRB administra automáticamente la publicación de mensajes sin necesidad de usar un computador personal, sin embargo el controlador no se conecta a Twitter directamente, se conecta con un servidor *web* que maneja la encriptación del nombre de usuario y la contraseña para entrar al perfil respectivo. Esto permite reservar la memoria del micro-controlador para uso propio de la aplicación.

El uso de una red social de acceso masivo va a permitir que miles de personas sigan el viaje de la plataforma y revisen las mediciones efectuadas en ella, con lo que se espera generar gran publicidad para la ACAE.

4.2.3.1 Principio de operación

El controlador utilizado en el módulo es el W5100 del fabricante Wiznet, que se basa en el modelo de capas TCP/IP. Este modelo integra una implementación en *hardware* de la capa de transporte, Internet y red en el mismo chip, facilitando el desarrollo de aplicaciones embebidas en donde la estabilidad, facilidad de integración y el bajo consumo de potencia son requerimientos esenciales. La implementación en *hardware* del controlador es especialmente útil en sistemas electrónicos que no cuentan con un sistema operativo (*OS*).

En el núcleo del chip, se ha realizado la implementación de una pila TCP/IP, que soporta el uso de los protocolos estándar TCP, UDP, IPv4, ICMP, ARP, IGMP y PPPoE. Para la transmisión y recepción de datos, la pila cuenta con un *buffer* de memoria interno de 16 Kbytes. Dentro del núcleo también se encuentran implementados varios elementos propios de la capa de red: un controlador de direcciones *MAC* de Ethernet empotrado y una

interface eléctrica para acoplar una conexión tipo 10 BASE-T con el conector RJ45. El W5100 también incluye, en el mismo chip, una interface de comunicación síncrona tipo SPI para enlazarse con un micro-controlador, donde se ubica lógicamente la capa de aplicación según el modelo TCP/IP.

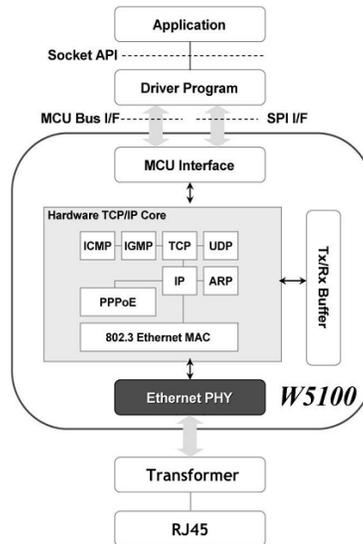


Figura 4.51 Diagrama de bloques del controlador W5100(Extraído de su hoja de datos)

4.2.3.2 Conexión eléctrica

El W5100 se conecta con el micro-controlador maestro a través del bus SPI que, como se ha mencionado anteriormente, requiere una conexión mínima de 3 señales ("SerialIn, SerialOut, Reloj") para la comunicación bilateral entre 2 dos dispositivos. Sin embargo, el chip está empotrado en la tarjeta de expansión que incluye un lector de memoria SD conectado en el mismo bus y, por lo tanto, se agregan dos señal más de control para que el maestro pueda seleccionar entre los periféricos. Solo un periférico SPI puede ser utilizado simultáneamente.

También se conecta la señal RESET del maestro con la patilla correspondiente en la tarjeta de expansión para evitar problemas de sincronización entre los bloques durante la ejecución del programa principal.

Para el acceso a Internet, se utiliza la interface de capa física del W5100 que, como ya se dijo, cumple con las especificaciones eléctricas 10 BASE-T. La tarjeta de expansión incluye un conector RJ45 empotrado para facilitar la conexión del módulo con otros

dispositivos de red a través de un cable UTP. Según la topología estándar de red, el módulo utilizado se considera un dispositivo final (*End Device*), por lo tanto se debe utilizar un cable cruzado para conectarlo con un ordenador o un *Router* que no cuente con detección automática MDI/MDI-X.

La tarjeta de expansión posee una serie de diodos emisores de luz (*LED*) que permiten al desarrollador conocer aspectos importantes de la operación del controlador y detectar posibles fallas. Los indicadores disponibles son:

- PWR: Indica que el controlador de *Ethernet* y el micro-controlador maestro están encendidos.
- LINK: Indica que existe conectividad con la red y parpadea cuando se envía o recibe información.
- FULLD: Indica que la conexión establecida es "*Full Duplex*".
- 100M: Indica una conexión a 100 Mbps.
- RX: Parpadea cuando se reciben datos.
- TX: Parpadea cuando se envían datos.
- COLL: Parpadea cuando se detecta una colisión en la red.

4.2.3.3 Programación

La programación del W5100 se realiza directamente desde la capa de aplicación, representada en este caso por el micro-controlador maestro del bloque SRRB.

El fabricante de las tarjetas de desarrollo Arduino proporciona los *drivers* del controlador de *Ethernet* y del bus SPI mediante librerías incluidas en el SDK. También se utilizan librerías de fuente abierta que permiten la utilización de los protocolos de red DHCP y DNS. Por encima de la jerarquía, se utiliza la librería pública *Twitter.h*, que facilita el acceso a dicha red social directamente desde el micro-controlador.

El hecho que el *driver* del controlador de *Ethernet* sea manejado a través de una librería de software otorga ciertas libertades no convencionales en implementaciones de *hardware* de capa física, por ejemplo, permite que el programador defina la dirección *MAC*

del controlador de *Ethernet*, lo que podría ocasionar conflictos con otro dispositivo conectado en la misma red.

La red social Twitter utiliza el protocolo de autenticación "OAuth", el cual les permite a los usuarios compartir su información sin que sea necesario revelar su identidad física. Los protocolos de encriptación requieren gran uso de recursos computacionales y podrían ocupar una parte importante de la memoria del microprocesador, por esta razón, la autenticación de usuario se maneja a través de un servidor dedicado que, además de administrar la encriptación, también controla la conexión y la publicación de los mensajes en Twitter. El micro-controlador se identifica en el servidor con una ficha (*Token*) estática que representa una codificación del usuario y la contraseña necesaria para ingresar en el perfil, de forma que el procesamiento pesado se realiza en el servidor y se deja libre la memoria del micro-controlador para que sea utilizada específicamente en la aplicación. La conexión del bloque SRRB con la red social se considera indirecta, pues el micro-controlador envía los datos hacia el servidor y este se encarga de publicarlos en Twitter.

En el programa principal se utiliza un *buffer* de memoria de 141 caracteres, ya que este es el tamaño máximo de mensaje permitido en Twitter.

En la fase de encendido del sistema, se corre el programa de configuración inicial del módulo *Ethernet* y luego se entra en el ciclo principal que se repite indefinidamente, a menos que se produzca un Reset en el bloque.

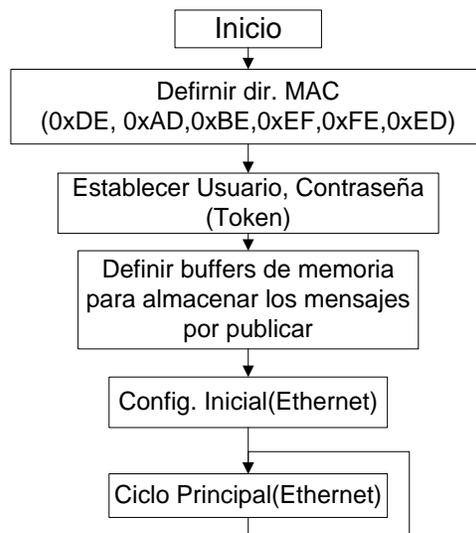


Figura 4.52 Diagrama de flujo general para el controlador de Ethernet

En la fase de configuración inicial, primero se habilita el protocolo DHCP mediante la utilización de la función `EthernetDHCP.begin()` de la librería `EthernetDHCP.h`, esta función utiliza como parámetro la dirección *MAC* definida con anterioridad. Este protocolo le asigna automáticamente una dirección IP dinámica al micro-controlador para facilitar la conectividad con la red. También se inicia la interface asincrónica del micro-controlador para operar a 9600 baudios. Esta interface se utiliza en conjunto con el sistema de transmisión de radio, que se utiliza para recibir la información medida en la plataforma. Una vez configurado el acceso a la red, se procede a publicar un mensaje de inicio en el perfil del sistema.

Las políticas de funcionamiento de Twitter no permiten la publicación de 2 mensajes iguales concurrentes y por eso se utiliza una lógica de eliminación de opciones y códigos de error con el fin de garantizar que el mensaje publicado sea diferente al anterior y que no se detenga inesperadamente la ejecución del algoritmo. El código de error de mensajes en Twitter es 200 y se adquiere a través de la función `Status()` de la librería `Twitter.h`.

Luego de publicar el mensaje inicial, se establece la comunicación con el SRRA. Tras la respuesta del bloque SRRA, inicia la ejecución del ciclo principal.

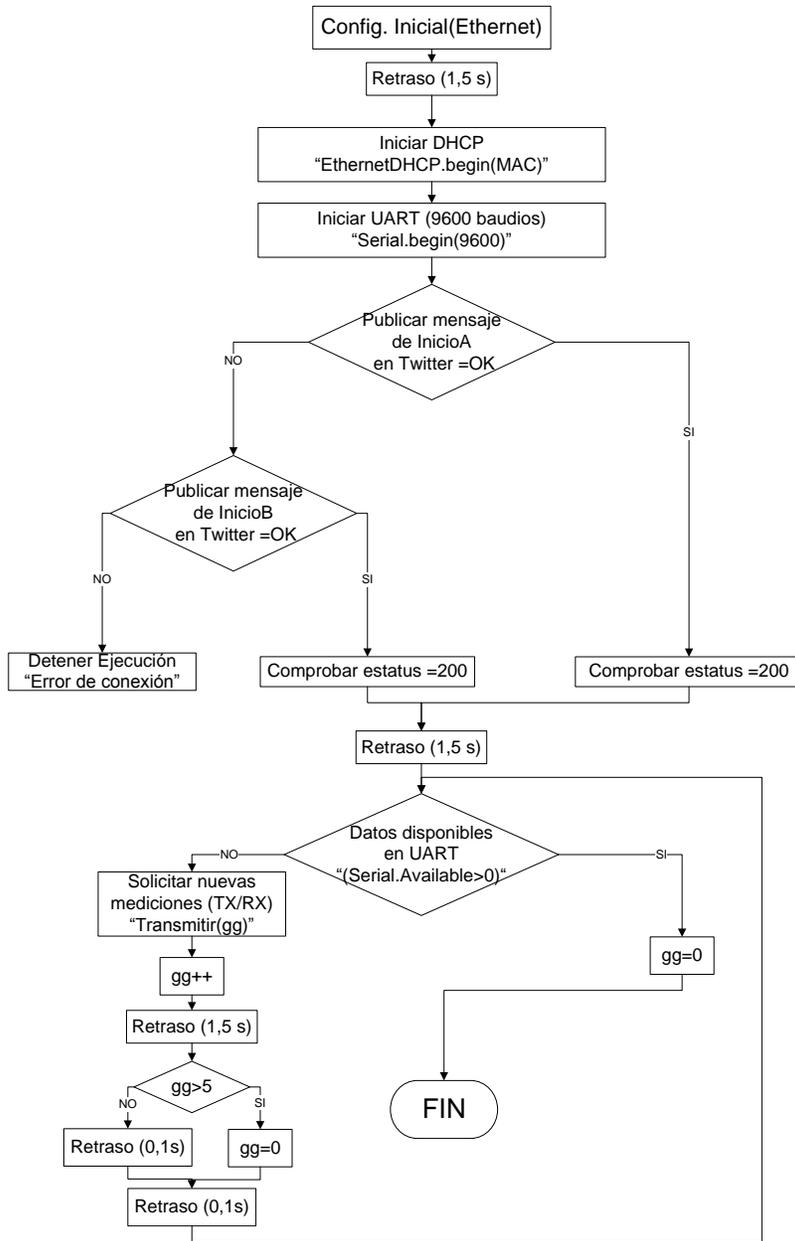


Figura 4.53 Diagrama de flujo de la rutina de configuración inicial del controlador Ethernet

El ciclo principal se ejecuta en forma repetida indefinidamente, siempre y cuando el *hardware* esté funcionando adecuadamente. El algoritmo utilizado para el intercambio de información entre los bloques del SRR opera según el modelo de comunicación de apretón de manos (*Hand Shake*), en el cual la sincronización entre los bloques se establece en forma preliminar a cada mensaje enviado. El SRRB funciona como bloque maestro con control de

flujo. Tiene la capacidad de regular el flujo de la transmisión de mensajes según sea necesario para adecuarse al nivel de saturación de Twitter.

Con respecto al algoritmo utilizado, el SRRB solicita el envío de un nuevo mensaje al transmitir un código específico para cada medición disponible, lo que permite configurar el formato de la publicación de los mensajes para la aplicación desarrollada. Tras recibir el código de mensaje, el SRRA decodifica la petición y responde con un mensaje con la información solicitada. El mensaje entrante es procesado en el SRRB, carácter por carácter, y es almacenado en el *buffer* de memoria creado para ese fin. Cada carácter es analizado en búsqueda de un comando de borrado sencillo (*Delete*) o un cambio de línea (*Enter*).

Si se detecta un comando de borrado, el puntero índice del *buffer* se devuelve dos lugares hacia la izquierda, de forma que la siguiente letra remplace a la anterior. El comando de cambio de línea se interpreta como el final del mensaje y, por ello, acciona el procedimiento para publicar el mensaje a través del método `Post()`. Se estableció un tope para la cantidad de caracteres que podrían ser almacenados en un mismo mensaje y se limitó el valor máximo del puntero índice del *buffer* a 140.

También se implementó un temporizador de perro guardián (*Watchdog Timer*) que permite reanudar la comunicación entre los bloques del SRR en caso de que se pierda la sincronía. Si se activa el temporizador, se solicita el reenvío de los mensajes mediante un código de error que es transmitido hacia el SRRA.

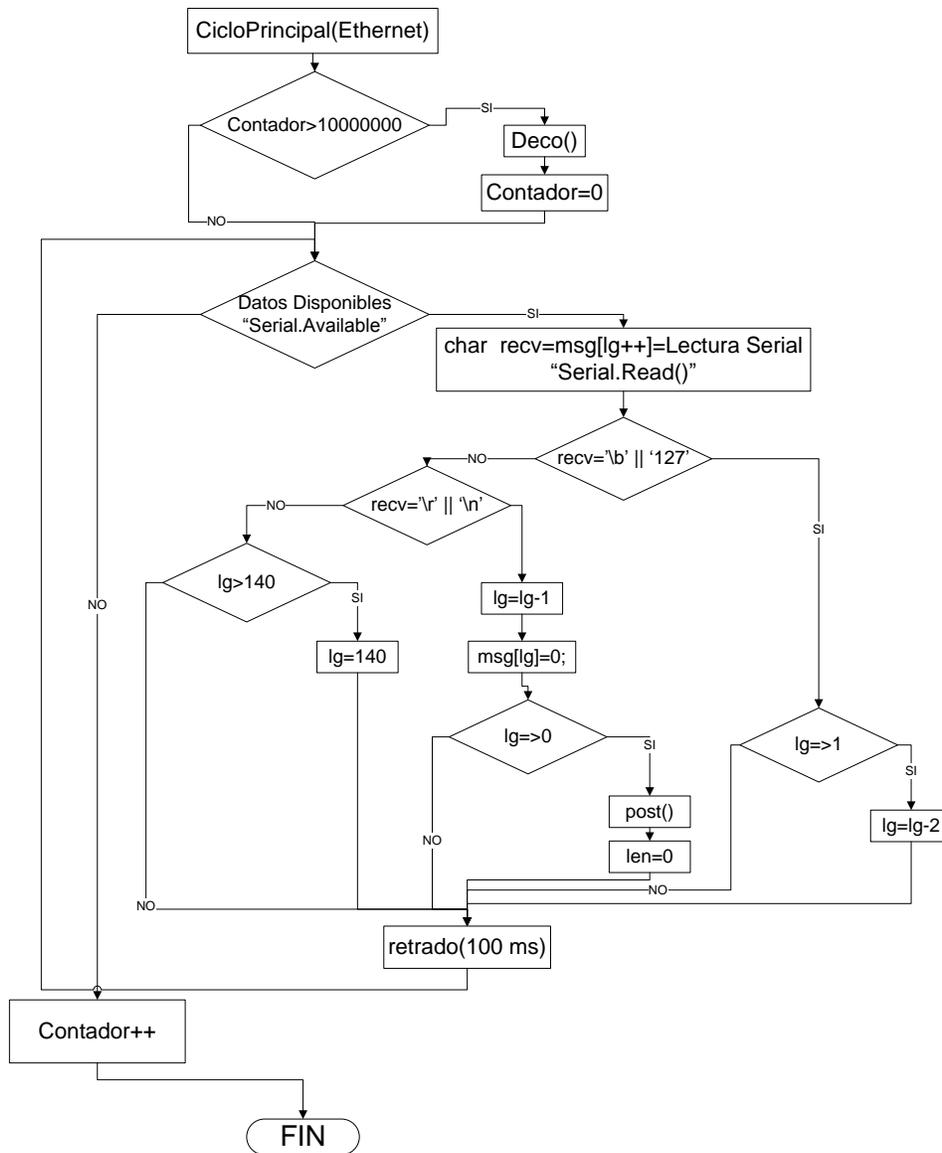


Figura 4.54 Diagrama de flujo de la rutina cicloPrincipal() para el controlador de Ethernet

El método Post() se utiliza para publicar los mensajes en Twitter. Se usan funciones de la librería Twitter.h para darle formato al mensaje, publicarlo y adquirir e interpretar los códigos de operación de la red social. El mensaje almacenado en el *buffer* de memoria se pasa como un parámetro al método. Mediante la instrucción Twitter.post(msg), se publica el mensaje en el perfil. La operación devuelve una variable “booleana” indicando si fue posible publicar el mensaje, luego se adquiere el código de error correspondiente a la operación realizada mediante la instrucción Twitter.wait() y se almacena en la variable Estatus. Si el código de error es 200, la publicación se efectuó correctamente y se procede a

solicitar un mensaje nuevo, después de esperar los 45 segundos requeridos según las condiciones de usuario de Twitter. La solicitud de un nuevo mensaje se efectúa mediante una llamada al método Deco(). Si el código de error es otro, se solicita un mensaje distinto con el fin de evitar un ciclo indeseado que podría ocurrir si el sistema trata de publicar un mensaje igual al anterior. Si la instrucción Twitter.post(msg) devuelve un resultado “booleano” falso, significa que se detectó una falla en la conexión a la red y se debe reintentar el procedimiento.

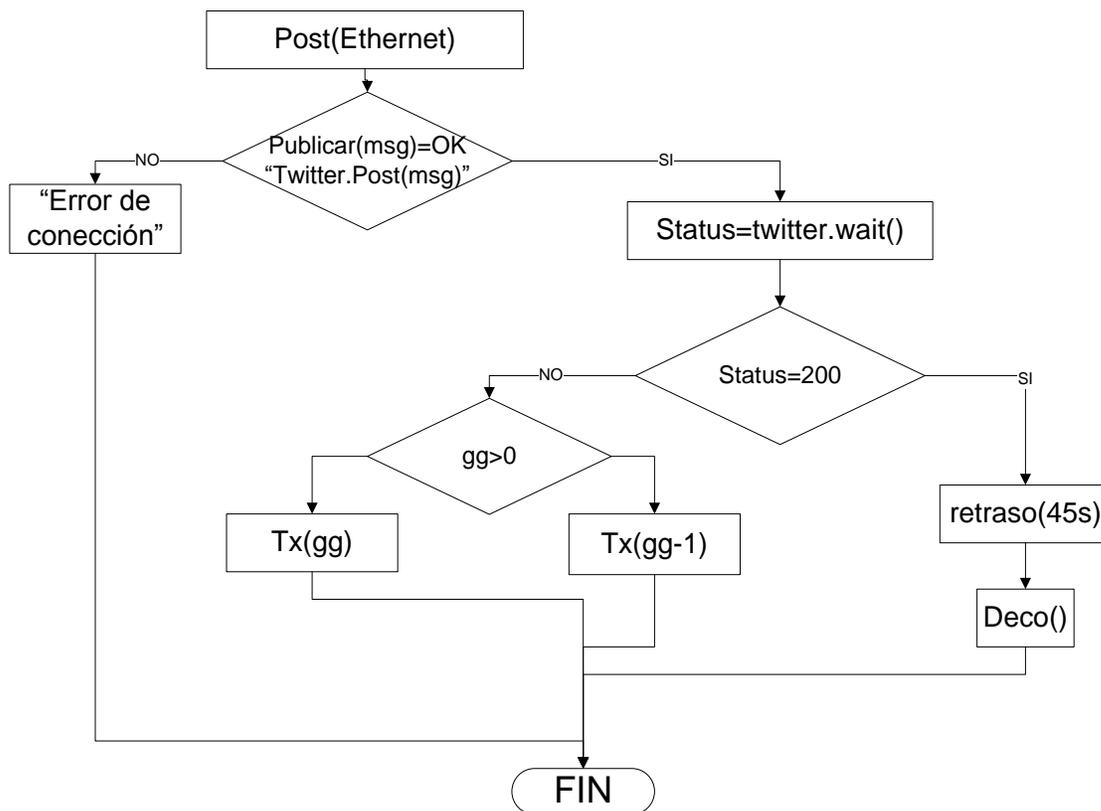


Figura 4.55 Diagrama de flujo de la rutina Post() para el controlador de Ethernet

El método Deco() se utiliza para enviar los códigos de solicitud de mensaje nuevo. Los códigos se almacenan en un arreglo llamado "gg". En cada llamada al método, se incrementa el índice del arreglo y se solicita un mensaje diferente. Hay 10 códigos de mensaje correspondientes a las mediciones que pueden ser realizadas en la plataforma:

- Latitud/longitud=0
- Hora, fecha=1

- Temperatura interna=2
- Temperatura externa=3
- Presión atmosférica=4
- Humedad relativa=5
- Altitud, velocidad de ascenso=6
- Aceleración axial=7
- Rumbo=8
- Cantidad de luz ambiental =9

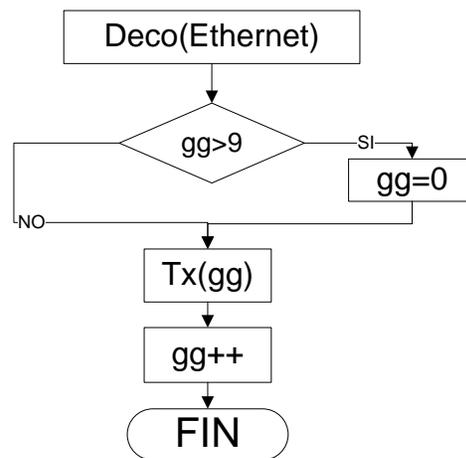


Figura 4.56 Diagrama de flujo de la rutina Deco() para el controlador de Ethernet

4.3 Sistema de Corte de Emergencia (SCE)

La Dirección de Aviación Civil en Costa Rica sugiere que los objetos no tripulados que utilicen el espacio aéreo costarricense cuenten con al menos 2 métodos de terminación de vuelo, los globos aerostáticos por naturaleza cuentan con uno. Conforme la plataforma asciende, la presión atmosférica ejercida sobre el helio que rellena el globo disminuye, causando que el helio se vuelva menos denso y que el volumen del globo se incremente proporcionalmente hasta que las fibras del látex se rompan y estalle el globo.

Como segundo método de terminación de vuelo para el presente prototipo, se incluyó el Sistema de Corte de Emergencia que se encarga de cortar la cuerda que sostiene la sonda, en caso de que se reporte una situación de peligro cercana, por ejemplo, el aterrizaje

forzoso de un vehículo aéreo que vuele cerca de la plataforma. De ser necesario, se enviará una señal desde la estación terrestre solicitando la interrupción inmediata del viaje utilizando los recursos del SRR, entonces el SCE accionará un actuador que reviente la cuerda. El SCE también permite, hasta cierto punto, controlar cual será la zona de aterrizaje de la sonda, que es sumamente importante para este proyecto debido a la topología de Costa Rica, ya que las corrientes de viento podrían ocasionar que el globo se desplace hasta el océano, lo que resultaría en un inconveniente mayor en el proceso de recuperación de la sonda.

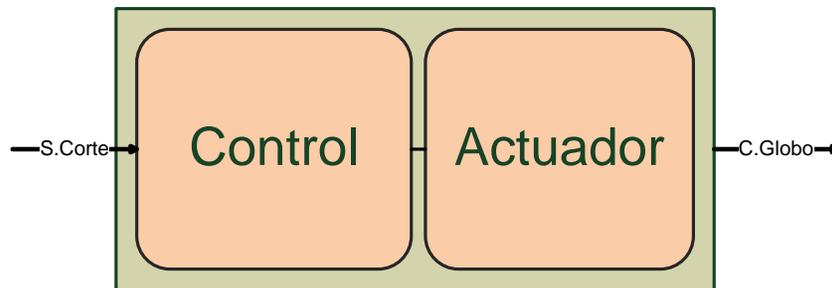


Figura 4.57 Diagrama de bloques del Sistema de Corte de Emergencia (SCE)

4.3.1 Conexión eléctrica (actuador)

El circuito utilizado como actuador para quemar la cuerda que sostiene la sonda, consiste en un interruptor digital basado en un transistor de potencia NPN que, cuando se polariza, permite el paso de la corriente eléctrica a través de una resistencia térmica. La resistencia se construyó con alambre de nicromo, cuyas propiedades físicas se presentan a continuación:

Tabla 8 Propiedades físicas del alambre de nicromo

Propiedad	Valor	Unidad
Resistividad eléctrica	1.0×10^{-6}	Ωm
Conductividad térmica	11.3	$\text{W}/(\text{m}^\circ\text{C})$
Atracción magnética	-	-
Coefficiente de expansión térmica	13.4×10^{-6}	$1/^\circ\text{C}$
Punto de fusión	1.400	$^\circ\text{C}$
Calor específico	450	$\text{J}/(\text{kg}^\circ\text{C})$

El alambre se enrolla alrededor de la cuerda uniformemente, guardando una separación de 5 mm entre cada vuelta para evitar cortos circuitos que podrían ocasionar una concentración indebida del calor. Cuando se acciona el actuador, el alambre se calienta rápidamente y quema las fibras del cordón que sostiene la sonda. Para comunicar el microcontrolador maestro con el SCE se utiliza un acople óptico, de manera que se tiene una separación galvánica que permite aislar las etapas.

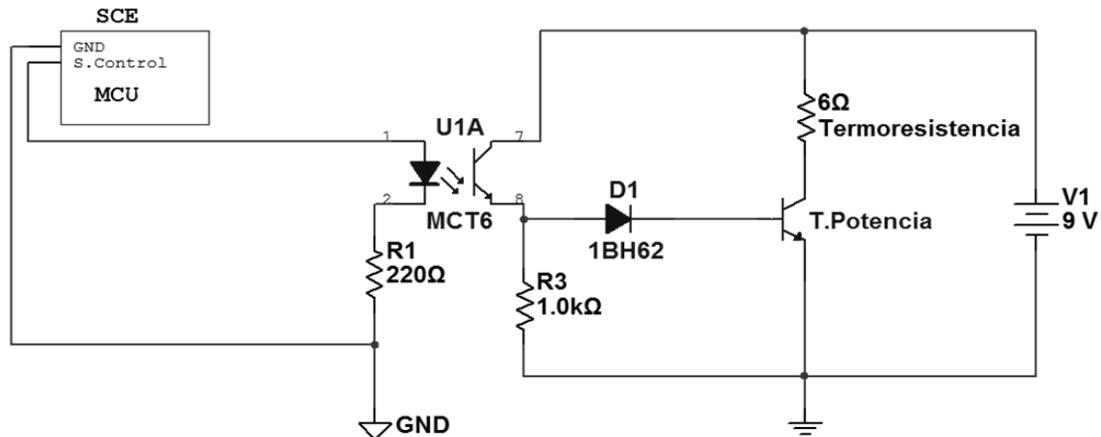


Figura 4.58 Diagrama de bloques del Sistema de Corte de Emergencia (SCE)

4.4 Sistema de Alimentación de Potencia (SAP)

El Sistema de Alimentación de Potencia se encarga de suplir y acondicionar la energía eléctrica a los distintos subsistemas del SCEIGA durante el viaje. El SAP se alimenta con un arreglo de baterías con la carga necesaria para permitir el funcionamiento del sistema por al menos 10 horas, tiempo suficiente pues se espera que el recorrido total dure 5 horas y la fase de recuperación al menos 2 horas. El SCEIGA transmite constantemente información hasta que se gasta la carga de las baterías.

Para suministrar las distintas tensiones requeridas por los componentes del sistema, se utilizan reguladores de tensión de 5V y de 3.3V, que mantienen efectivamente la tensión regulada siempre y cuando la tensión de alimentación sea mayor que la tensión nominal. Por esta razón, el SAP debe mantener la tensión por encima del valor nominal durante las 10 horas propuestas. Para cumplir este requisito, se utilizaron 4 baterías de polímero de iones de litio de 3.7 V, capaces de suministrar 2000mAh cada una. Las baterías se conectan

de forma que la tensión de salida es 7.4 V con carga máxima. La descarga de las baterías se aproxima a través de una curva logarítmica.

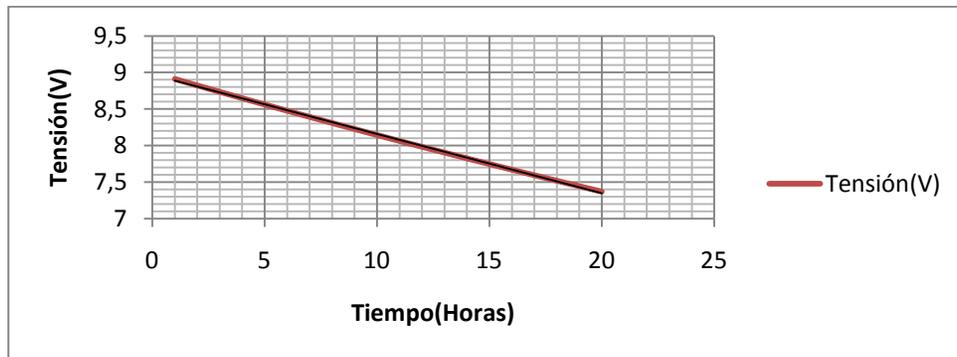


Figura 4.59 Curva de descarga del arreglo de baterías de polímero de iones de Litio

Las baterías de polímero de iones de litio cuentan con un limitador de corriente con un umbral de 2A y por esta razón, no se pueden utilizar para alimentar el actuador del SCE. Esta etapa, puede ser alimentada con baterías alcalinas comunes, ya que la potencia requerida por el actuador es relativamente baja.

CAPÍTULO 5

*Ensamblaje del prototipo, resultados
experimentales y análisis*

5.1 Interconexión de bloques del SCEIGA

Siguiendo la estrategia de diseño formulada en el capítulo 3, se procedió a interconectar los bloques tratados en la sección anterior con el fin de cumplir el objetivo general del proyecto.

Como bien se ha expuesto, la herramienta de división por bloques funcionales se usa en forma conceptual para ordenar el diseño del prototipo y facilitar su implementación, sin embargo en la práctica, esta división no resultó ser tan evidente.

Las tarjetas de desarrollo utilizadas cuentan con el *hardware* y la memoria suficiente para acomodar programas extensos, por lo tanto, con el fin de reducir el *hardware* requerido al mínimo, se obvió la división de bloques y se agruparon los componentes según sus interfaces de comunicación con el fin de interconectarlos. Aún después de aplicar este criterio, fue necesario tomar en cuenta otro factor de peso para sintetizar el prototipo: la separación espacial especificada en el diseño del bloque SRR cuyo objetivo es comunicar la información adquirida en la plataforma en tiempo real. Esto resultó en la implementación de 2 sistemas electrónicos aislados físicamente: uno para la plataforma y otro para la estación terrestre.

5.1.1 SCEIGA-PLATAFORMA

Al prototipo construido para la sonda se le nombró SCEIGA-PLATAFORMA. Su implementación se llevó a cabo con base en una tarjeta de evaluación Arduino Mega. Esta tarjeta cuenta con un micro-controlador Atmel ATmega1280, cargado con el *bootloader* Arduino y, por esta razón, ofrece compatibilidad directa con el SDK Arduino utilizado y puede entonces ejecutar los programas escritos sin necesidad de efectuar modificaciones.

El SDK permite escribir el código en la memoria *Flash* del micro-controlador en forma directa, a través de una función disponible en el entorno. Solo es necesario un cable USB conectado entre la tarjeta y el computador que hospeda el SDK para transferir y evaluar los programas escritos gracias al chip FTDI integrado. Lo anterior agiliza el proceso de desarrollo y evaluación de las aplicaciones al no tener que recurrir a un programador universal de micro-controladores cada vez que se modifica el código.

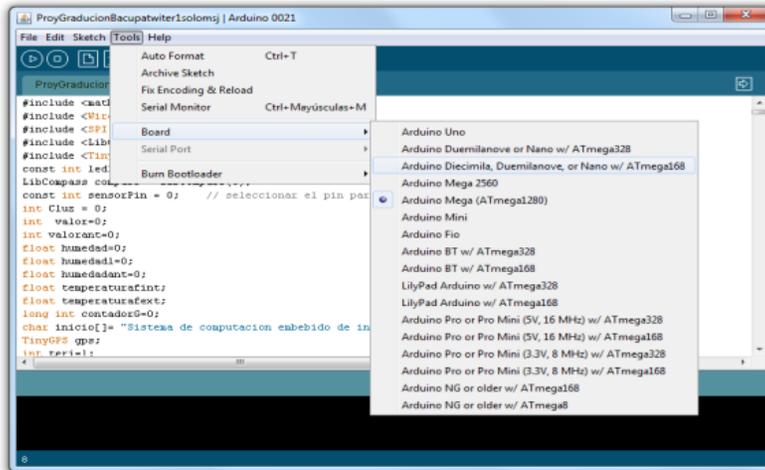


Figura 5.1 Entorno de programación Arduino SDK

En la pestaña "*Tools>Board*" del entorno Arduino, se debe seleccionar la tarjeta utilizada para que el código se compile correctamente y pueda ser transferido posteriormente a la memoria.

Con respecto a la interconexión de módulos, se planteó la distribución abajo descrita y que sigue los esquemas de conexión eléctrica especificados en el capítulo anterior para cada componente:

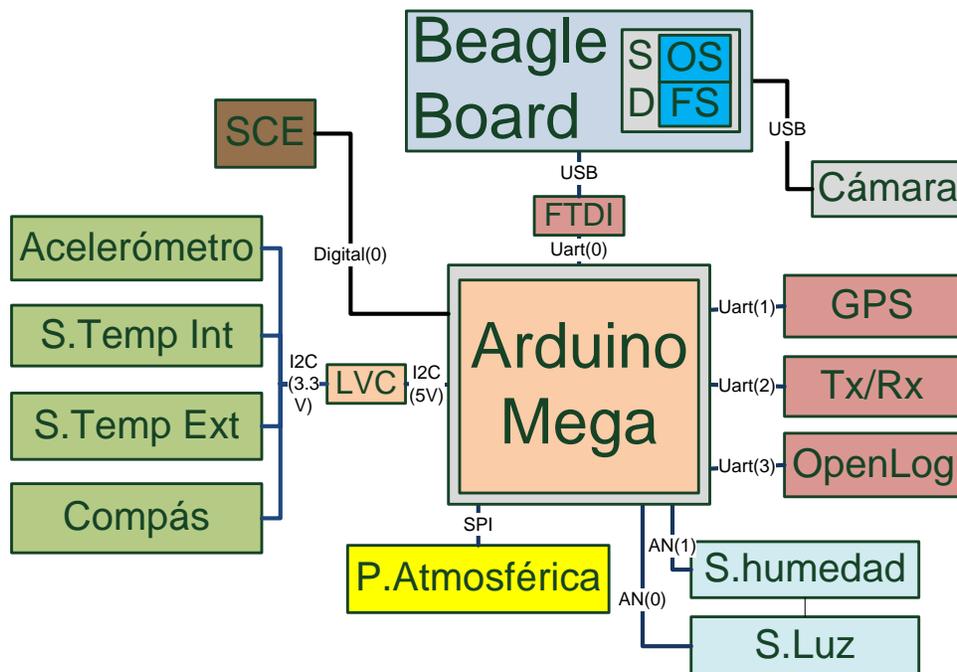


Figura 5.2 Diagrama de conexión de bloques del SCEIGA-PLATAFORMA

En primer lugar se aprecia en la figura que la integración de componentes se centró alrededor de la tarjeta Arduino Mega. El diseño de esta tarjeta es muy práctico para el desarrollo de prototipos, ya que los pines de su micro-controlador integrado están expuestos y organizados por interfaces de comunicación, lo que le permite al desarrollador ahorrar tiempo y estructurar el cableado de elementos externos a la tarjeta.

En segundo lugar se aprecia en la figura el agrupamiento de integrados según sus interfaces y divididos por colores:

- El color verde representa a los componentes que se conectan en el bus I²C : un convertidor de tensión se utiliza para acoplar ambos lados del bus que, por la diferencia en sus procesos de fabricación, requieren distintos niveles de tensión para denotar los mismos estados lógicos.
- El color amarillo representa al único componente conectado al bus SPI. Como solo es uno, no es necesario agregar *hardware* adicional para administrar su selección.
- El color rojo representa a los elementos conectados en los puertos UART. Estos componentes realizan funciones cruciales en el prototipo y por lo tanto se tratan con alta prioridad. Cada UART cuenta con un *buffer* de memoria y con control de flujo por *hardware*, esto permite recibir o transmitir datos de los 4 elementos simultáneamente sin incurrir en errores en la ejecución del programa principal.
- El color celeste representa a los elementos conectados en los puertos analógicos.
- El color café representa a los componentes conectados en los puertos digitales.

5.1.2 SCEIGA-ESTACIÓN TERRESTRE

El prototipo del sistema electrónico ubicado en la base se nombró SCEIGA-ESTACIÓN TERRESTRE. Su implementación se basó en una tarjeta de desarrollo Arduino Duemilanove que incluye un micro-controlador Atmel ATmega328, cargado con el bootloader Arduino, por lo tanto, igual que el caso anterior, es compatible con el SDK

utilizado para la escritura del código. Esta tarjeta también cuenta con el chip FTDI que permite escribir la memoria Flash del ATmega328 directamente desde el computador.

En la pestaña "*Tools>Board*" del entorno Arduino, se debe seleccionar la tarjeta utilizada tal como se expuso en el caso anterior.

Con respecto a la interconexión de módulos para este sistema, se utilizó el siguiente esquema de conexión:

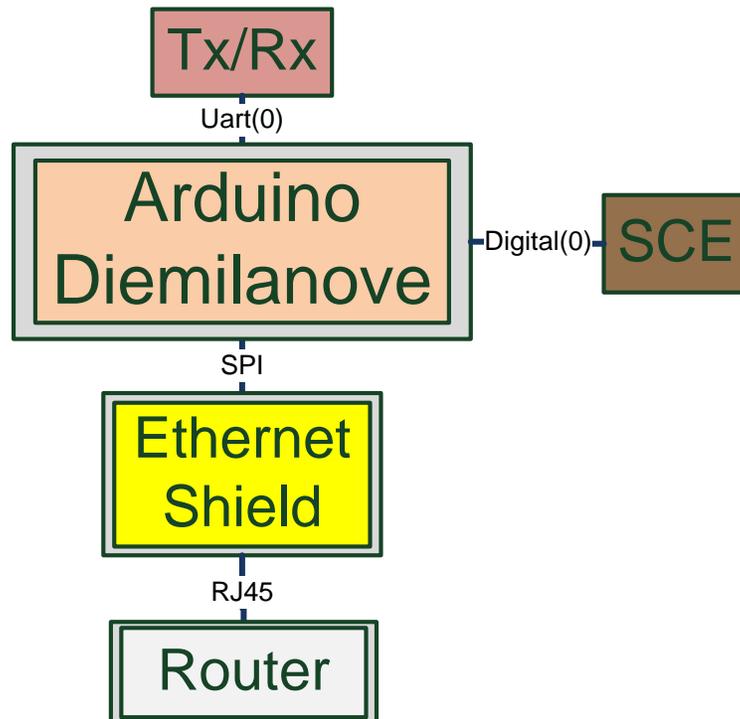


Figura 5.3 Diagrama de conexión de bloques del SCEIGA-ESTACIÓN TERRESTRE

Se siguió el formato de representación por colores expuesto anteriormente:

- El color rojo representa al único componente conectado en la UART
- El color amarillo representa a la tarjeta de expansión *Ethernet Shield* que se conecta directamente al puerto SPI del micro-controlador y a un *Router* con acceso a Internet.
- El color café representa a un interruptor de tipo *Push-Button* conectado al puerto digital que se utiliza para accionar el SCE.

5.2 Programación del prototipo

Con base en los métodos de *software* descritos en el capítulo 4, se procedió a escribir el programa principal del prototipo, el cual pertenece al estrato más alto en la jerarquía de niveles de abstracción de *software*. El programa se divide 2 partes:

- I. La etapa de configuración inicial que se ejecuta una sola vez tras el encendido.
- II. El ciclo principal que, como su nombre lo establece, se repite indefinidamente siempre y cuando el *hardware* lo permita.

5.2.1 Configuración inicial

En la etapa de configuración inicial se establecen los parámetros de operación del prototipo y se habilita el *hardware* necesario según la conexión realizada. Esta rutina se carga automáticamente luego del encendido o tras un RESET.

El algoritmo se ejecuta secuencialmente empezando por las operaciones de mayor prioridad:

1. Se inician las interfaces de comunicación y se define la velocidad de transmisión de datos para los buses que así lo requieren.
2. Se definen los pines digitales utilizados como entradas o salidas en función de la conexión eléctrica. En la BeagleBoard se carga el programa de captura en forma automática mediante el *Script* escrito.
3. Se escriben los registros de configuración de los sensores del SAD que definen sus parámetros de operación y se crea un archivo en la memoria SD para almacenar las mediciones.
4. Se entra en un ciclo de espera hasta que el receptor GPS adquiera sus coordenadas iniciales.
5. Se inicia el sistema de transmisión de radio.
6. Se define el período de muestreo para las mediciones.

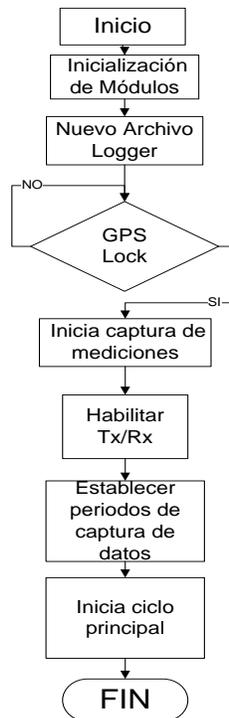


Figura 5.4 Diagrama de flujo de la rutina de configuración inicial del bloque principal

5.2.2 Ciclo principal

Este proceso controla las funciones generales del prototipo. Permite que los diferentes componentes del sistema interactúen entre sí, a pesar de las diferencias en sus protocolos de comunicación.

Para procesar la recepción de datos provenientes de los módulos principales, se utilizó la técnica de consulta constante (*Polling*) de atención de dispositivos externos, en combinación con el uso de banderas lógicas. Lo anterior permite mantener constante el flujo del ciclo, aunque se desconecte alguno de estos módulos o se presente una falla de *hardware*. Cuando se reciben nuevos datos en algún puerto UART, se levanta una bandera lógica indicándolo y mediante instrucciones condicionales se atiende la petición.

Cada iteración del algoritmo se inicia con la revisión de la bandera de datos disponibles en la UART(0), correspondiente a la BeagleBoard. De estar en alto la bandera, se almacena el dato recibido en la variable "Teri" y se pasa como parámetro al método Deco(), el cual permite decodificar el dato y responder con la última medición del sensor seleccionado. Seguidamente se revisa la bandera de datos disponibles en la UART(2), que

corresponde al transmisor de radio. De forma similar, si existen datos disponibles, se almacenan en la variable "Teri" y esta se envía como parámetro al método Deco() para su decodificación y respuesta correspondiente. El método Deco() requiere como segundo parámetro una asignación "booleana" para distinguir entre la UART(0) y la UART(2).

La sección posterior del código se utiliza para la administración del almacenamiento de datos en la memoria externa. A través de la combinación de una bandera lógica y un contador, se escoge el número de mediciones que se almacenan en cada archivo. Cuando se terminan de capturar las muestras, se graba la fecha del lanzamiento y se cierra el archivo. Por último, se agrega un pequeño retraso de tiempo para evitar errores por sobre-flujo de datos en los buses de comunicación.

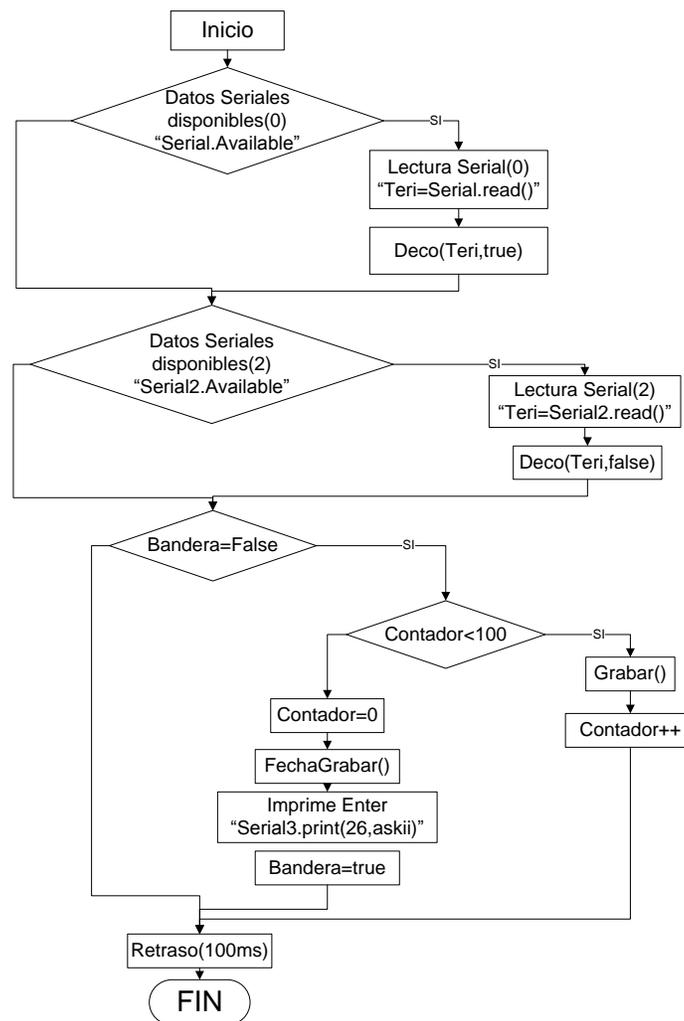


Figura 5.5 Diagrama de flujo del ciclo principal que corre en el SCEIGA

5.3 Resultados experimentales y análisis

A continuación se presentan los resultados generales de las pruebas efectuadas con el prototipo. Dichas pruebas se realizaron en un establecimiento con temperatura controlada, guardando una distancia aproximada de 100 metros entre los dos prototipos electrónicos contruidos.

El prototipo del sistema SCEIGA-PLATAFORMA se mantuvo en una posición fija por razones de control y verificación. El SCEIGA-ESTACIÓN TERRESTRE se conectó a la red del establecimiento con un cable cruzado.

Los resultados se presentan por bloques en función de los objetivos específicos planteados en los lineamientos de diseño y siguen el orden en el cual fueron implementados.

5.3.1 SAD

En primer lugar se presentan los resultados obtenidos con el módulo SAD(AN) del SCEIGA-PLATAFORMA.



Figura 5.6 Fotografía capturada con el prototipo SCEIGA-PLATAFORMA

En la figura anterior se aprecia una fotografía del espacio de prueba captada automáticamente por el módulo. Para esta prueba se utilizó una cámara de 1.3 Megapíxeles de tipo *Webcam*, conectada a un puerto USB de la BeagleBoard. Se configuró la resolución de las imágenes en 320X240 cuadros con compresión JPEG2, lo cual permite reducir el tamaño de cada archivo considerablemente, aunque es notable la reducción en la calidad de

la imagen que esto ocasiona. Este compromiso se puede modificar fácilmente de ser necesario y en caso de que las necesidades de ACAE varíen en el futuro, sin embargo se tendría que adquirir una cámara con mejores prestaciones. Las fotografías se nombraron automáticamente según la hora de su captura, lo que es conveniente para mantener la organización de los archivos y contextualizar las imágenes con su respectivo conjunto de mediciones físicas.

A continuación se presentan los resultados del módulo SAD(BN) del SCEIGA-PLATAFORMA.

Tabla 9 Formato de presentación de los datos capturados mediante el prototipo SCEIGA-PLATAFORMA

Tiempo	Latitud	Longitud	Alt(m)	C.Luz	Humedad R(%)	Accel X	Accel Y	Accel Z	Temp Int. C	Temp Ext. C	Vel(k mph)	Rumbo (g)
6:48:43.40	9.955.401	-84.197.113	848,5	112	82,98%	0	248	-17	25,53	25,52	1,09	261,7
6:48:43.80	9.955.420	-84.197.128	846,7	113	82,83%	1	247	-16	25,57	25,57	1,09	257,1
6:48:44.20	9.955.430	-84.197.143	846,2	112	83,30%	2	248	-15	25,57	25,57	1,56	258,5
6:48:44.60	9.955.420	-84.197.143	847,9	112	82,98%	0	249	-16	25,58	25,58	1,7	258,5
6:48:45.0	9.955.430	-84.197.143	847,8	112	83,14%	0	248	-15	25,58	25,58	1,83	258,5

En la tabla anterior se aprecia el formato en el que se presentan las mediciones efectuadas por el módulo. Esta tabla se extrajo directamente del archivo Recorrido0001.CSV creado y modificado automáticamente por el sistema. Para esta prueba se estableció un período de muestreo de 40 milisegundos para agilizar la depuración, pero para el viaje real el período puede modificarse en función del estudio realizado.

Para aproximar la precisión de los sensores, se calculó la desviación estándar en una población de 5 mediciones consecutivas para cada variable. Por la naturaleza física de las variables atmosféricas medidas, esta herramienta estadística es un buen criterio de análisis, siempre y cuando se utilice un período de muestreo mucho más corto que el tiempo físico natural de cambio de la variable en estudio. Por otra parte, como el prototipo se mantuvo en una posición fija, este criterio también sirve para cuantificar la precisión de los transductores que miden variables relacionadas con el movimiento de la plataforma. La precisión experimental de los sensores utilizados se presenta en la siguiente tabla.

Tabla 10 Cálculo de la incertidumbre de los sensores en el SCEIGA-PLATAFORMA

Tiempo	Latitud	Longitud	Alt(m)	C.Luz	Humedad R(%)	Accel X	Accel Y	Accel Z	Temp Int. (C)	Temp Ext. (C)	Vel(kmph)	Rumbo(g)
Media	9955420,2	-84197134	847,42	112,2	0,83046	0,6	248	-15,8	25,566	25,564	1,454	258,86
D.estandar	12	13	0,9418	0,4472	0,0018	0,8944	0,7071	0,8367	0,0207	0,0251	0,3457	1,6994
Incr.(+-)	24	27	1,8836	0,8944	0,0036	1,7889	1,4142	1,6733	0,0415	0,0502	0,6915	3,3988

5.3.2 SRR

A continuación se presentan los resultados de las pruebas efectuadas con el módulo SRR que forma parte tanto del SCEIGA-PLATAFORMA, como del SCEIGA-ESTACION TERRESTRE.

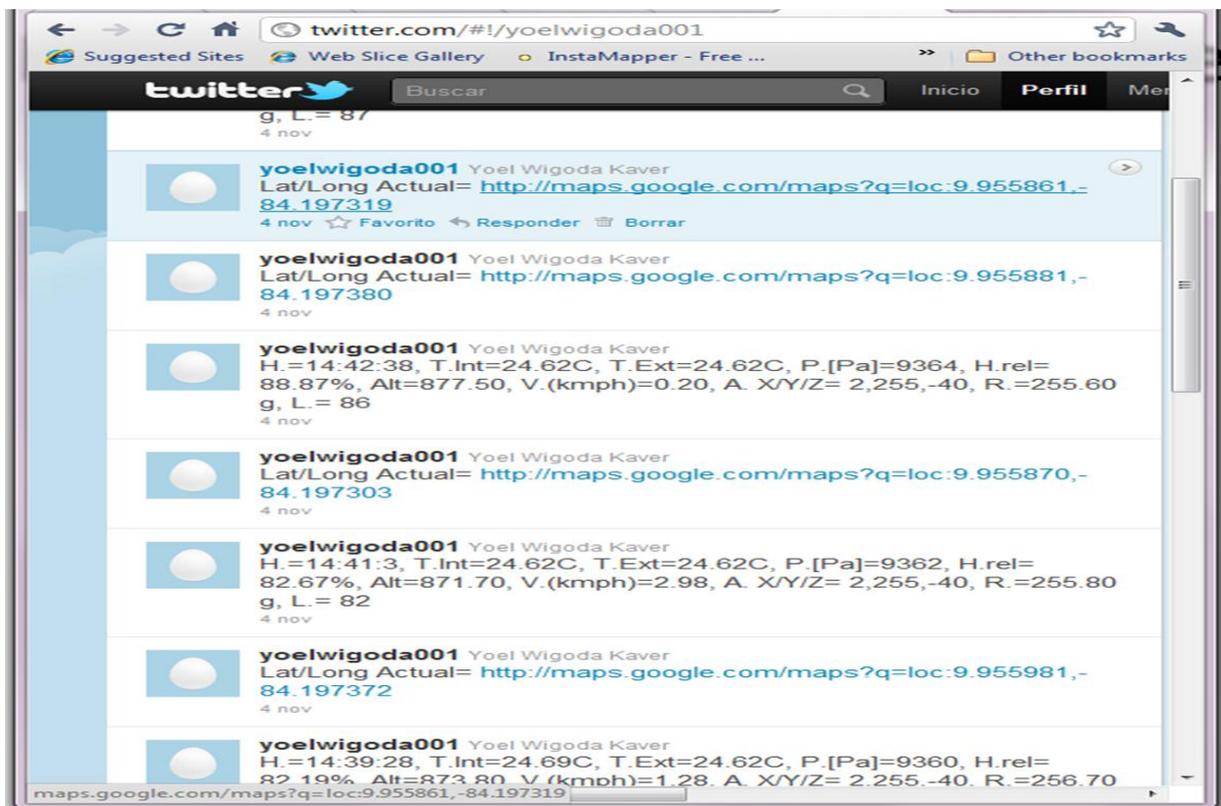


Figura 5.7 Perfil de Twitter del prototipo SCEIGA

En la figura anterior se expone el método utilizado para habilitar el acceso público a la información adquirida con el prototipo SCEIGA-PLATAFORMA. Como ha sido plasmado con anterioridad, se utilizan las prestaciones de la red social Twitter para este fin.

El perfil del SCEIGA se accede con un explorador Web a través del dominio www.twitter.com/yoelwigoda001. Los datos se publican a través de 2 tipos de mensajes:

- el primero contiene las coordenadas de la sonda, con un hipervínculo hacia el servicio gratuito “Google maps”, que permite a los usuarios ubicar gráficamente la plataforma en un mapa satelital.
- el segundo tipo de mensaje reúne el resto de mediciones efectuadas por el SAD. Cada 5 minutos se publica un nuevo mensaje con información más reciente.

Si se presiona el *link* del hipervínculo asociado al primer tipo de mensaje, automáticamente el explorador redirige al usuario al sitio de Google Maps, con una etiqueta plantada sobre la última posición conocida de la sonda, como se observa en la siguiente figura.

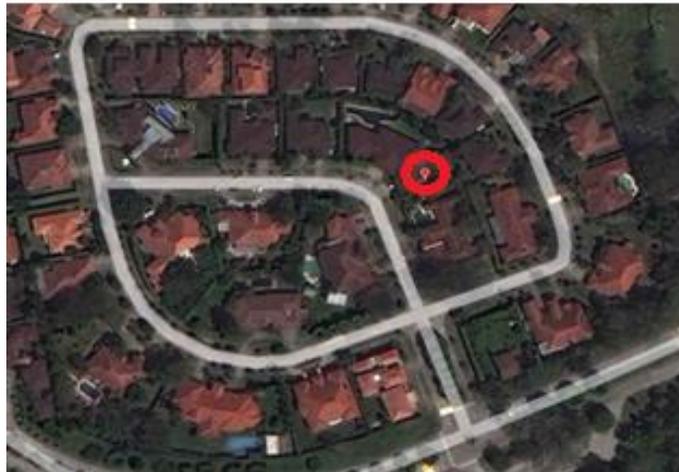


Figura 5.8 Formato de presentación de la ubicación de la sonda

Este formato de la información capturada permitirá que personas alrededor del mundo interesadas en el proyecto sigan, de manera intuitiva, el recorrido y el estado de la plataforma. Se garantiza así que el evento de lanzamiento sea un éxito y se cumpla el objetivo principal del primer proyecto de ACAE.

5.3.3 SCE

Para probar el SCE, se diseñó un experimento basado en la medición del tiempo necesitado por el sistema para cortar una cuerda de fibra sintética desde el momento en el

cual se presiona el botón de emergencia en el prototipo SCEIGA-ESTACION TERRESTRE. Este experimento es destructivo y por lo tanto su replicación es limitada.

Tabla 11 Medición experimental del tiempo de corte de una cuerda de fibra sintética

N.medición	Tiempo (s)
1	9,5
2	8
3	8
4	7,4
5	8,1
Promedio	8

De la tabla anterior se extrae el tiempo promedio necesitado por el sistema para cortar una cuerda similar a la utilizada en la plataforma. Este resultado es aceptable, si se considera el tiempo de reacción humana característico de una situación de posible riesgo aéreo. Durante el lanzamiento real, será posible comprobar la ruptura de la cuerda a través de las lecturas de aceleración y velocidad percibidas por los sensores del SAD.

5.3.4 SAP

Se comprobó el cumplimiento del objetivo específico referente al SAP con la implementación de un circuito de medición básico. Este circuito consiste en la conexión en serie de un amperímetro capaz de medir corriente media con cada batería utilizada. El sistema se puso a operar a plena carga durante la medición. Después se utilizó la siguiente fórmula para calcular indirectamente el tiempo máximo de operación en cada caso.

$$T_{max} = \frac{C. bat(mA * h)}{I_m(mA)} = \text{horas de operación}$$

Por la naturaleza de las funciones electrónicas efectuadas, se consideró el tiempo de operación del SAD(BN) como determinante del tiempo de operación global debido a las funciones críticas que realiza en el sistema.

Tabla 12 Tiempo máximo de operación de los módulos pertenecientes al SCEIGA

Bateria	Im(mA)	Carga(mAh)	Tmax(h)
SADAN	703,125	4500	6,4
SADBN	770	8000	10,4
SCE	1.500	250	0,17

De la tabla anterior se extrae que el prototipo SCEIGA-PLATAFORMA puede operar en plena carga por 10 horas consecutivas, tiempo necesario para que el equipo de recuperación cumpla con su función. Las funciones de captura fotográfica se pueden efectuar constantemente por 6,4 horas, tiempo suficiente para que la plataforma alcance la altitud esperada. Como última conclusión extraída de la tabla se tiene que, en caso de ser necesaria la utilización del SCE, este puede operar por 17 minutos, que es más que suficiente si se considera el tiempo promedio obtenido del corte experimental (8 segundos).

CAPÍTULO 6

Conclusiones y recomendaciones

6.1 Conclusiones

6.1.1 Conclusiones propias del desarrollo del SCEIGA

- El prototipo del SCEIGA construido, es capaz de operar bajo las condiciones climáticas características de la estratósfera, ya que todos los componentes utilizados para su implementación, cumplen con las especificaciones técnicas correspondientes presentadas en la tabla 1.
- La versión implementada del SAD(AN), es capaz de capturar, comprimir y almacenar videos y fotografías en forma autónoma.
- El sistema operativo Angstrom, utilizado para administrar los recursos de la BeagleBoard, incluye el *kernel* 2.6.32 de Linux, que contiene todos los *drivers* necesarios para el manejo de cámaras VGA conectadas al puerto USB y para módulos de expansión de captura de video compatibles.
- El paquete de *software* libre *Motion*, permite la captura directa de videos en formato MPEG4 y de fotografías en formato JPEG2. Estos formatos cuentan con un índice de compresión elevado, de modo que el espacio de memoria destinado para el almacenamiento de imágenes se aprovecha al máximo.
- Mediante la modificación del *Script* de configuración inicial *Motion.conf*, es posible automatizar la función de captura fotográfica al definir la tasa de captura como un parámetro de operación. Para el prototipo implementado, se definió dicha tasa en una foto por minuto.
- El SAD(BN) implementado, es capaz de capturar mediciones de 11 variables físicas diferentes y de archivarlas con formato .CSV, en una memoria extraíble tipo microSD.
- La propiedad de bus compartido del estándar I²C, permite que su uso sea altamente recomendado en desarrollo de sistemas embebidos.
- El SRR implementado es capaz de adquirir las coordenadas del sistema, 1 vez por minuto, mediante el servicio GPS.
- El SRR implementado permite la comunicación electrónica bilateral de datos entre la plataforma y una estación terrestre.

- El SRR implementado permite que el sistema acceda a Internet y publique, en forma autónoma, la información obtenida en un perfil de Twitter.
- El SRR implementado utiliza una conexión con un servidor dedicado que maneja la encriptación (OAuth) requerida para acceder a Twitter y publicar mensajes.
- El SCE implementado, sirve como método alternativo de terminación de viaje, y permite que la plataforma cumpla con las recomendaciones de la Dirección de Aviación Civil para el lanzamiento de globos aerostáticos.
- El SAP implementado, es capaz de alimentar todos los sub-sistemas electrónicos a bordo de la plataforma por un período de 10 horas sin interrupción.
- El SCE no se puede alimentar con las baterías de litio utilizadas debido a sus elevados requerimientos de corriente.

6.1.2 Conclusiones de ACAE en relación al proyecto DAEDALUS

El desarrollo del proyecto DAEDALUS resultó ser una herramienta poderosa para captar la atención de los medios de comunicación y del posible mercado para ACAE.

Tras el planteamiento del proyecto, múltiples entidades gubernamentales y privadas mostraron interés por la Asociación y su potencial para acelerar el desarrollo aeroespacial en Centroamérica.

La iniciativa permitió que ACAE fuese invitada a formar parte del Consejo Nacional de Investigación y Desarrollo Aeroespacial (CONIDA) junto con el Ministerio de Ciencia y Tecnología; la Asociación Estrategia Siglo XXI; los ministerios de Comercio Exterior, Educación Pública y Relaciones Exteriores; el Instituto Nacional de Aprendizaje; la Coalición Costarricense de Iniciativas de Desarrollo; el Sistema Bancario Nacional y la Promotora de Comercio Exterior. El consejo fue creado el 25 de julio del 2010 a través de un decreto firmado por la presidenta Laura Chinchilla.

La creación del CONIDA colocó a ACAE en una posición estratégica para el desarrollo de la industria en cuestión, pues le fue asignada la tarea de liderar la Cámara de Industrias de Aeronáutica en Costa Rica. La presentación del equipo DAEDALUS en el evento de lanzamiento del CONIDA promovió el interés del SICA (Secretaría de Integración Centroamericana) en proyectos de índole científico con sello centroamericano y

específicamente en DAEDALUS. El SICA financió en su totalidad el proyecto desarrollado.

El 20 de noviembre de 2010 se lanzó el primer prototipo de un globo experimental gracias a un esfuerzo conjunto de ACAE y Ad Astra Rockets. Aunque no se pudo recuperar la sonda, el evento sirvió para promocionar la misión de ACAE de forma masiva, con la ayuda de los medios de comunicación de mayor influencia en Latinoamérica, y llevó a la apertura de una sede de ACAE en Guatemala.

El siguiente lanzamiento se ejecutó el 18 de diciembre y, en esta ocasión, el objetivo consistió en evaluar el trabajo del equipo de ingeniería mecánica. Mediante una cámara especializada se logró captar imágenes de Guanacaste a 20 kilómetros de altura.



Figura 6.1 Imagen de Guanacaste captada a 20 km de altitud

El siguiente paso del equipo DAEDALUS será probar el prototipo construido del SCEIGA en un lanzamiento previsto para junio del 2011.

6.2 Recomendaciones

Como resultado del trabajo efectuado, se plantean varias recomendaciones para desarrollar en futuras etapas del proyecto, de forma que se incrementen las capacidades globales de la plataforma. Dichas recomendaciones se presentan a continuación:

- Transmisión de imágenes en tiempo real: estadísticamente, la mayoría de las fallas en proyectos que involucran sondas meteorológicas ocurren durante la fase de recuperación, principalmente por la incertidumbre inherente a la trayectoria seguida por los globos aerostáticos y a su posición final. Si se complementa la transmisión de mediciones actualmente implementada con la transmisión de imágenes en tiempo real, se lograría eliminar por completo la necesidad de recuperar la sonda para obtener la información recopilada por la plataforma durante el recorrido.
- Reforzar el SRR: el prototipo implementado cuenta con un solo método de localización geográfica que es completamente dependiente del servicio GPS. Aunque el servicio es altamente confiable, los receptores de uso civil se apegan a limitaciones técnicas impuestas artificialmente que restringen su funcionalidad en grandes alturas y, por lo tanto, existe el riesgo de perder información referente a la ubicación de la sonda, lo que compromete su recuperación posterior. Para corregir este problema se recomienda agregar un método alternativo de rastreo que se base en otra tecnología, como por ejemplo, las etiquetas RTLS o localizadores por radio (*Radio Beacon*).
- Estabilización de la plataforma por control automático: la estructura utilizada para sujetar la sonda no cuenta con un sistema de estabilización que elimine las oscilaciones mecánicas indeseadas que aparecen producto de las condiciones atmosféricas en grandes altitudes. Las oscilaciones afectan negativamente la función de captura de video y fotografía de la plataforma y reducen la cantidad de información útil captada. Si se utiliza el acelerómetro incluido y se agrega un actuador electromecánico, como por ejemplo, un juego de alerones, es factible implementar un sistema de control automático que estabilice la plataforma sobre dos ejes y permita captar imágenes de mejor calidad.
- Control de trayectoria asistida: Como fue mencionado en secciones anteriores, la plataforma no contiene mecanismos de elección de trayectoria y, debido a esto, es

completamente dependiente de las condiciones ambientales experimentadas durante el vuelo. Si se utilizan los recursos del SRR y se agrega un actuador electromecánico, como por ejemplo, un juego de alerones, se podría implementar un mecanismo de control automático que permita que la plataforma mantenga el rumbo deseado.

- Medidores de contaminación ambiental: en los últimos años se ha incrementado el interés colectivo por temas relacionados con el medio ambiente tales como el calentamiento global y el debilitamiento de la capa de ozono. En altitudes estratosféricas, existen indicadores químicos que permiten cuantificar el daño ambiental. Como ya fueron implementados los algoritmos de manejo de los buses PC ,SPI,UART y de los puertos analógicos, agregar nuevos sensores al SAD es relativamente sencillo, solo es necesario realizar pequeñas modificaciones en el programa principal del sistema.
- Cámara multi-espectro (*multispectral camera*): la cámara utilizada en el prototipo solo permite la captura de imágenes y videos en el espectro electromagnético visible (450-690 nm), que son de poca utilidad para el análisis por computadora. A diferencia de las cámaras convencionales, las multi-espectro captan el espectro visible, infrarrojo y de radar en una misma imagen, lo que podría utilizarse para detectar incendios, para medir la profundidad de los océanos y la erosión de los suelos desde el espacio aéreo.

BIBLIOGRAFÍA

Libros

- [1]. Evans, brian (2007). "*Arduino Programming Notebook*". Versión 1.0. USA
- [2]. Guochang, Xu. (2003). GPS: theory, algorithms, and applications. 2da edición, Alemania, Postdam: GFZ
- [3]. Gutiérrez, M. y Calvo, N. (2010). Normas de presentación de los informes de prácticas de especialidad, tesis, seminarios y otros del ITCR en formato digital. Cartago: Biblioteca José Figueres Ferrer, Instituto Tecnológico de Costa Rica.
- [4]. Letham, Lawrence. (2001). GPS fácil: uso del Sistema de Posicionamiento Global. Barcelona: Paidotribu
- [5]. Lombardo, John (2002). Embedded Linux. USA: New Riders publishing.
- [6]. Long, LN. (2004). "*Computing, Information and Communication: The Fifth pillar of Aerospace Engineering*". Journal of Aerospace Computing, Information and Communication, Vol 1, No 1.
- [7]. Pons, Nicolas (2009). Linux: Principios básicos del uso del sistema. Barcelona: ediciones-edi.
- [8]. Tanwer, Ashish et al. (2009). "Deployable low cost outdoor aerial surveillance system". Electronics and Communication Engineering Department, Thapar University, India.
- [9]. Xiaoyong, Ye et al. (2008). "*Design of intelligent mobile vehicle and its global optimal path planning*". USA: IEEE international conference on industrial technology

Páginas web

- [10]. Analog Devices Inc©. ADXL345 Data Sheet. Descargado el 14 marzo 2011 de la World Wide Web: <http://www.analog.com/static/imported-files/data_sheets/ADXL345.pdf>
- [11]. Atmel Corporation Inc©. ATmega Family Data Sheet. Descargado el 14 marzo 2011 de la World Wide Web: <http://www.atmel.com/dyn/products/product_docs.asp?category_id=163&family_id=607&subfamily_id=760&part_id=4198>.
- [12]. Arduiana. "TinyGPS". [En línea]. (Última modificación 15 enero de 2011). Consultado el 14 marzo de 2011 de la World Wide Web: <http://arduiniana.org/libraries/tinygps>
- [13]. Arduino. "Arduino Language Reference". [En línea]. (Última modificación 11 marzo 2011). Consultado el día 14 marzo de 2011 de la World Wide Web: <http://arduino.cc/en/Reference/HomePage>

- [14]. Arduino. "Twitter Library for Arduino". [En línea]. Última modificación 5 febrero 2011. Consultado el 14 marzo de 2011 de la World Wide Web: <http://www.arduino.cc/playground/Code/TwitterLibrary>
- [15]. Beagleboard.org. Beagleboard Xm System Reference Manual. Descargado el 14 marzo 2011 de la World Wide Web:< <http://beagleboard.org/hardware-xM>>
- [16]. Digi International Inc©. Xtend 900 Data Sheet. Descargado el 14 marzo 2011 de la World Wide Web:< http://www.digi.com/pdf/ds_xtendmodule.pdf>
- [17]. Embedded Linux Wiki. "BeagleBoard Beginners". [En línea]. (Última modificación 8 marzo 2011). Consultado el día 14 marzo de 2011 de la World Wide Web: <http://elinux.org/BeagleBoardBeginners>
- [18]. Honeywell Inc©. HIH-4030 Data Sheet. Descargado el 14 marzo 2011 de la World Wide Web:
<http://sensing.honeywell.com/index.cfm/ci_id/142958/la_id/1/document/1/re_id/0>
- [19]. The Angstrom Distribution Wiki."Angstrom Manual". [En línea]. (Última modificación 18 febrero 2011). Consultado el día 14 marzo de 2011 de la World Wide Web: <http://www.linuxtogo.org/gowiki/AngstromManual#GettingStarted>
- [20]. Locosys Inc©. GPS module Data Sheet. Descargado el 14 marzo 2011 de la World Wide Web:
<http://www.sparkfun.com/datasheets/GPS/Modules/LS20030~3_datasheet_v1.2.pdf>
- [21]. Sparfun Electronics Inc©. Openlog Data Sheet. Descargado el 14 marzo 2011 de la World Wide Web:
<<https://github.com/nseidle/OpenLog/wiki/Datasheet>>
- [22]. Texas Instruments Inc©. TMP102 Data Sheet. Descargado el 14 marzo 2011 de la World Wide Web:< <http://focus.ti.com/lit/ds/symlink/tmp102.pdf>>
- [23]. Vishay Semiconductor Inc©. TEMT6000 Data Sheet. Descargado el 14 marzo 2011 de la World Wide Web: <<http://www.vishay.com/docs/81579/temt6000.pdf>>
- [24]. VTI Technologies Inc©. SCP1000 Data Sheet. Descargado el 14 marzo 2011 de la World Wide Web: <http://www.vti.fi/midcom-serveattachmentguid-5ea0f5e98491572da5ae79b70f9f0a41/AN33_SCP1000_Pressure_Sensor_as_barometer_and_altimeter.pdf>
- [25]. Wiretronic Inc©. Nichrome 80 & Other Resistance Alloys - Technical Data & Properties. Descargado el 19 abril 2011 de la World Wide Web: <<http://www.wiretron.com/nicrdat.html>>
- [26]. Wiznet Inc©. W5100 Data Sheet. Descargado el 14 marzo 2011 de la World Wide Web: < http://www.wiznet.co.kr/UpLoad_Files/ReferenceFiles/W5100_Datasheet_v1.2.3.pdf>

Apéndice A.1. Glosario, abreviaturas y simbología

OAuth: Es un protocolo abierto de encriptación, cuyos servicios están disponibles para los desarrolladores a través de una API.

Es el protocolo utilizado actualmente por Twitter para encriptar las cuentas de usuario.

ADC:"Analog-to-digital converter". Circuito electrónico capaz de convertir señales continuas en números digitales discretos.

AES:"Advanced Encryption Standard". Protocolo de encriptación de datos basado en un esquema de cifrado por bloques. Fue adoptado como un estándar por el gobierno de Estados Unidos.

API: "Application Programming Interface". Métodos ó funciones que ofrecen algunas librerías para ser utilizados por otro "software" como una capa de abstracción .

video4linux: es una API de captura de video para Linux. Muchas webcams USB, sintonizadoras de tv, y otros periféricos son soportados. Video4Linux está integrado con el núcleo Linux.

Bootloader: Programa básico que carga las funciones necesarias para cargar un sistema operativo. Generalmente utilizados en arranques multi-etapa.

Buffer: Espacio de memoria que se utiliza para almacenar datos temporariamente durante una sesión de trabajo.

Cortex A8: Es un procesador fabricado por ARM cuyo diseño está optimizado para el bajo consumo energético.

DHCP: "Dynamic Host Configuration Protocol". Protocolo que permite asignar dinámicamente las direcciones IP de los clientes que se conectan a una red.

Driver: Programa que administra la interacción entre un dispositivo de "hardware" y el sistema operativo.

DVI: Interfaz de video digital que permite obtener la mayor calidad de imagen posible.

Ethernet: es un estándar de redes de computadoras de área local perteneciente a la capa de enlace de datos y de red en el modelo OSI.

EXT3: "Third Extended Filesystem". Sistema de archivos comúnmente utilizado en el sistema operativo Linux.

FHSS: "Frequency Hopping Spread Spectrum". Técnica de modulación digital que consiste en variar la frecuencia de la portadora con base en una secuencia predefinida que solo el transmisor y el receptor conocen, de modo que se reduce en gran parte la interferencia en el proceso de comunicación.

FSK: "Frequency-shift keying". Tipo de modulación digital basada en variación de frecuencia.

GPS: "Global Positioning System". Sistema de localización geográfica que permite obtener las coordenadas de un objeto mediante la emisión de señales desde una constelación de satélites artificiales.

I²C : Interface de comunicación serie de bus compartido.

IP Address: "Internet Protocol Address". Consiste en una etiqueta numérica que se utiliza para identificar de manera lógica, a una interface de un dispositivo conectado en una red que utilice el protocolo IP. A diferencia de la dirección MAC, la dirección IP puede variar constantemente.

MAC: "Media Access Control". Es una dirección única que se utiliza para identificar a un controlador Ethernet en una red local.

mAh: Es la medida estándar para la capacidad de almacenar energía de una batería. Indica la cantidad de carga eléctrica que fluye por las terminales de una batería.

Memoria Flash: Tipo de memoria no volátil que permite la lectura y escritura de múltiples datos en una sola operación y por lo tanto permite velocidades de operación superiores frente a otras tecnologías de memoria más antiguas.

MDI/MDI-X: "Medium Dependent Interface". Es un puerto Ethernet que es capaz de cambiar lógicamente un cable cruzado por uno directo ó viceversa automáticamente según sea requerido por la aplicación.

MEMs 3D : Proceso de fabricación de sensores que consiste en moldear placas de silicio para formar estructuras tridimensionales, que posteriormente son utilizadas como transductores en sensores de presión.

Minicom: Programa equivalente a hyperterminal para Linux. Permite establecer la comunicación serial a través de los puertos del ordenador.

Motion: Programa libre y de código abierto para capturar video y fotografías.

MMC:"Multimedia card". Es un estándar de tarjetas de memoria basado en memoria "Flash" que es prácticamente igual a la memoria SD con la salvedad de que carece del pin de seguridad que evita rescribirla.

NMEA 0183: Es un protocolo de comunicación utilizado mundialmente por los sistemas GPS para transmitir información.

Open Source: "Software" ó "hardware" de fuente abierta y desarrollado libremente.

P2P: Red de computadoras en la que todos los elementos pueden operar como servidores o clientes, de modo que se forman nodos en la red capaces de intercambiar información entre sí.

PNT: "Positioning, Navigation, Timing". Servicios ofrecidos por el sistema GPS de manera gratuita.

Processing multimedia programming environment: Es un lenguaje de programación de fuente abierta utilizado para el desarrollo de interfaces usuario-maquina.

RTLS: "Real Time Locating System". Es un sistema de posicionamiento geográfico basado en la colocación de etiquetas electrónicas en los objetos rastreados, de forma que mediante triangulación se puede averiguar su ubicación geográfica.

Script: Archivo de órdenes escritas en un archivo que permite automatizar funciones, combinar componentes e interactuar con el usuario ó con el sistema operativo.

SDK: entorno de desarrollo que le permite a los programadores crear aplicaciones para un sistema concreto.

SD: "Secure Digital". es un tipo de memoria "Flash" que es mayormente utilizado en dispositivos portátiles tales como cámaras, PDAs, Nintendos, etc.

SoC "System on a Chip": Consiste en un "Chip" en el cual se integran todos los elementos de una computadora.

Twitter: Red social de acceso masivo que permite que miles de personas a nivel mundial se contacten entre sí a través de la publicación de mensajes llamados "Tweets".

U.S Air Force: Fuerza aérea del ejército de los Estados Unidos de Norte América.

USB: "Universal Serial Bus". Puerto de comunicación serial que permite conectar periféricos en un ordenador.

Watchdog Timer: Es un temporizador automático utilizado en el desarrollo de "hardware" que permite evitar que sus programas caigan en ciclos infinitos de los cuales no se podrían recuperar normalmente .

YUV: Espacio de color típicamente usado como parte de un conducto de imagen en color.

Apéndice A.2. Ruedas de prensa

Anexo B.1. Hoja de información del proyecto

Información del estudiante:

Nombre: Yoel Wigoda Kaver

Cédula: 1-12920-133. **Carné ITCR:** 200579932.

Email: awtlana@gmail.com

Información del proyecto:

Nombre del Proyecto: Sistema de Computación Embebido para Investigación en Grandes Alturas

Área del Proyecto: Diseño de Sistemas Digitales, Análisis y diseño de algoritmos, Sistemas Embebidos.

Información de la Empresa:

Nombre: Asociación Centroamericana de Aeronáutica y el espacio **Zona:** Los Ángeles, Cartago.

Dirección: 300 metros al este de la Escuela Sony.

Actividad Principal: Gerencia de proyectos.

Información del encargado en la Institución:

Nombre: Miguel Hernández Rivera.

Puesto que ocupa: Profesor Ing. Electrónica **Departamento:** Escuela Electrónica

Profesión: Ingeniero Electrónico **Grado académico:** Licenciatura

Teléfono: 2550-9237 **Email:** mhernandez@itcr.ac.cr

Información del asesor en la empresa:

Nombre: Juan Manuel Ortega Castillo

Puesto que ocupa: Gerente del proyecto **Departamento:** Ingeniería

Profesión: Ingeniero Electromecánico **Grado académico:** Licenciatura

Teléfono: 88196197 **Email:** ocjmcr@gmail.com