



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**SISTEMA DE ADQUISICIÓN, REGISTRO
Y MONITOREO DE PARÁMETROS
VEHICULARES VÍA GSM-GPRS**

TESIS

Que para obtener el título de

Ingeniero Eléctrico Electrónico

P R E S E N T A

Omar Muñoz Villarruel

DIRECTOR DE TESIS

M.I. Lauro Santiago Cruz



Ciudad Universitaria, Cd. Mx., 2017



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

DEDICATORIA

*A mi mamá, mi hermano y mi abuelo Ramón,
por darme su cariño, inspirarme a seguir mis
sueños y su apoyarme incondicionalmente.*

AGRADECIMIENTOS

A mi Alma mater, por la educación y las experiencias de vida recibidas

Al maestro Lauro, por sus enseñanzas, su paciencia y su amistad.

A mis profesores, por compartir su pasión por la ingeniería.

A Fausto, Pamela, Rosalba, Darío, Rodrigo, Sergio, Raquel, Víctor, Jesús, Itzel y Diana, por su amistad y las lecciones que compartieron conmigo.

A Paola, por revolucionar mi vida..

Investigación realizada gracias al Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica (PAPIIT) de la UNAM IG100914-2, denominado: “Integración de un Sistema para la Obtención y Monitoreo de Datos de Vehículos Automotores, Basado en los Protocolos CAN-OBDII.

Agradezco a la DGAPA-UNAM la beca recibida

ÍNDICE GENERAL

INDICE DE FIGURAS.....	xi
ÍNDICE DE TABLAS Y GRÁFICAS	xiii
PRÓLOGO	xv
INTRODUCCIÓN.....	1
CAPÍTULO 1 GENERALIDADES	3
1.1. Sistemas de adquisición de datos.....	3
1.2. Ciclos de manejo	4
1.2.1. Importancia de los ciclos de manejo en México.....	4
1.2.2. Metodología para el desarrollo de ciclos de manejo representativos	4
1.3. Protocolos de comunicación.....	5
1.3.1. Comunicación serie SPI	6
1.3.2. Protocolo de comunicación I²C	9
1.3.3. Comunicación serie asíncrona.....	11
1.3.3.1 Protocolo RS-232	13
1.4. Sistema de Posicionamiento Global	14
1.4.1. Protocolo NMEA 0183	16
1.4.2 Fuentes de error.....	17
1.4.3. Descripción general del receptor GPS a utilizar.....	18
1.5. Reloj de tiempo real.....	19
1.5.1. Descripción general del reloj de tiempo real implementado	19
1.5.2. Confiabilidad de los datos.....	21
1.6. Acelerómetro.....	22
1.6.1. Medición de aceleración por efectos capacitivos	23
1.6.2. Características generales del acelerómetro a utilizar	27
1.6.3. Confiabilidad de los datos.....	34

1.7. Pantallas de cristal líquido.....	34
1.7.1. Pantallas táctiles de membrana resistiva.....	37
1.7.2. Descripción general de la pantalla táctil a usar	38
1.8. Tarjetas de memoria SD	39
1.8.1. Tabla de asignación de archivos.....	43
1.8.2. Sistemas de archivos FAT12, FAT16 y FAT32.....	45
1.8.3. Características del dispositivo de almacenamiento a usar	46
1.9. Sistema Global para las Comunicaciones Móviles	46
1.9.1. Sistema de señalización SS7.....	50
1.9.2. Técnicas de acceso múltiple	52
1.9.3. Técnica de acceso múltiple por división de tiempo y de frecuencia	52
1.9.4. Servicio General de Paquetes Vía Radio	54
1.9.5. Descripción de la estación móvil a usar	56
1.10. Analizador de gases	57
1.10.1. Sensores electroquímicos de gases	58
1.10.2. Sensores ópticos no dispersivos	59
1.10.3. Descripción del analizador de gases a utilizar	60
1.11. Microcontroladores	61
1.11.1. Características del microcontrolador a implementar	63
1.11.2. Lenguajes de programación	63
1.11.3. Descripción del software de desarrollo a usar	64
CAPÍTULO 2 DESCRIPCIÓN DEL PROYECTO	65
2.1. Descripción general del sistema.....	65
2.2. Desarrollo del sistema	66
2.2.1. Microcontrolador.....	66
2.2.2. Receptor GPS.....	67
2.2.3. Acelerómetro.....	69
2.2.4. Analizador de gases	70
2.2.5. Reloj de tiempo real.....	72

2.2.6. Pantalla táctil	77
2.2.7. Medio de almacenamiento	85
2.2.8. Transceptor GSM/GPRS	87
2.2.9. Microcontrolador.....	92
2.2.10.Placa de circuito impreso	96
CAPÍTULO 3 PUESTA A PUNTO DEL SISTEMA	107
3.1. Primeras pruebas.....	107
3.2. Puesta a punto de la estación móvil celular.....	111
CAPÍTULO 4 RESULTADOS Y CONCLUSIONES	119
4. Resultados	119
4.1. Medición de Ángulos	119
4.2. Batería	122
4.3. Pruebas de funcionamiento.....	123
4.4. Conclusiones.....	127
GLOSARIO DE TÉRMINOS Y ABREVIATURAS	131
BIBLIOGRAFÍA.....	137

INDICE DE FIGURAS

Figura I.1.	Diagrama de bloques del sistema Cycle-DAQ.....	1
Figura 1.1.	Representación esquemática de SPI.....	6
Figura 1.2.	Diagrama de tiempo de I ² C.....	11
Figura 1.3.	Esquema simplificado de un UART.....	12
Figura 1.4.	Diagrama de muestro en comunicación serie asíncrona.....	13
Figura 1.5.	Planos orbitales de la constelación satelital GPS.....	14
Figura 1.6.	Segmentos GPS.....	15
Figura 1.7.	Diagrama de bloques del SiRF Star IV.....	18
Figura 1.8.	Esquema de conexión del RTC.....	20
Figura 1.9.	Esquema general de un oscilador Pierce dentro de un RTC.....	21
Figura 1.10.	Desviación de frecuencia contra temperatura.....	22
Figura 1.11.	Esquema simplificado de un acelerómetro MEMS.....	24
Figura 1.12.	Demodulación sensible a la fase en acelerómetros MEMS.....	26
Figura 1.13.	Estructura general de los cristales líquidos nemáticos.....	35
Figura 1.14.	Estructura de una celda de un LCD.....	35
Figura 1.15.	Respuesta de una celda de LC a la tensión.....	36
Figura 1.16.	Sensor táctil resistivo de cuatro hilos.....	37
Figura 1.17.	Diagrama de bloques de la pantalla táctil.....	38
Figura 1.18.	Tarjetas de memoria SD.....	39
Figura 1.19.	Estructura del formato FAT.....	45
Figura 1.20.	Red celular simplificada.....	50
Figura 1.21.	Red celular con la red de señalización indicada.....	51
Figura 1.22.	División de espectro en FDMA.....	53
Figura 1.23.	Estructura y organización de la técnica TDMA.....	54
Figura 1.24.	Tarjeta de desarrollo con módulo GSM-GPRS.....	57
Figura 1.25.	Estructura de un sensor no dispersivo.....	59
Figura 2.1.	Diagrama de bloques del Cycle-DAQ.....	66
Figura 2.2.	Diagrama de flujo de la primera rutina para el receptor GPS.....	67
Figura 2.3.	Diagrama de flujo de la rutina completa para el receptor GPS.....	68
Figura 2.4.	Diagrama de flujo de prueba de acelerómetro.....	70
Figura 2.5.	Diagrama de flujo del analizador de gases.....	71
Figura. 2.6.	Rutinas de reloj de tiempo real.....	75

Figura 2.7.	Diagrama funcionamiento de la pantalla táctil (Parte 1).	80
Figura 2.8.	Diagrama funcionamiento de la pantalla táctil (Parte 2).	82
Figura 2.9.	Diagrama de flujo de rutina para mostrar imagen.	83
Figura 2.10.	Imágenes desplegadas en pantalla.	84
Figura 2.11.	Diagramas de flujo de rutinas para almacenar información.	86
Figura 2.12.	Diagramas de flujo de rutinas para almacenar información.	89
Figura 2.13.	Diagramas de conexiones del sistema.	94
Figura 2.14.	Mejoras al diseño de PCB.	100
Figura 2.15.	Circuito integrado de encapsulado cuadrado plano de 100 terminales.	105
Figura 4.1.	Arreglo para la medición de ángulos.	120
Figura 4.2.	Archivo de texto generado por el equipo.	124
Figura 4.3.	Grafica velocidad contra tiempo mostrada en el servidor web.	125
Figura 4.4.	Trayectoria de vehículo bajo prueba.	126

ÍNDICE DE TABLAS Y GRÁFICAS

TABLAS

Tabla 1.1. Mapa de los módulos de memoria de SPI.	7
Tabla 1.2. Modos de operación de SPI	8
Tabla 1.3. Explicación de los términos de un enunciado GGA.	16
Tabla 1.4. Comparación entre los receptores GPS.	19
Tabla 1.5. Registros de memoria.....	21
Tabla 1.6. Registros del acelerómetro.	27
Tabla 1.7. Comparación entre los sistemas de visualización.....	39
Tabla 1.8. Terminales de tarjeta de memoria microSD.....	41
Tabla 2.1. Comparación de relojes de tiempo real.....	72

GRÁFICAS

Grafica 4.1. Caracterización de la medición de ángulos.....	121
---	-----

PRÓLOGO

El presente trabajo describe el funcionamiento y desarrollo del sistema de adquisición, registro y monitoreo de parámetros vehiculares desarrollado en la Coordinación de Instrumentación (CI) del Instituto de Ingeniería (II), y en conjunto con el Laboratorio de Control de Emisiones (LCE) de la Facultad de Ingeniería (FI), ambos de la Universidad Nacional Autónoma de México (UNAM). Dicho desarrollo se llevó a cabo en el marco del proyecto institucional Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica PAPIIT IG100914-2, denominado: “*Integración de un Sistema para la Obtención y Monitoreo de Datos de Vehículos Automotores, Basado en los Protocolos CAN-OBDII*”.

Durante los años 2012 y 2013 la CI del II y el LCE desarrollaron un sistema de registro de datos vehiculares denominado *Cycle-DAQ*. El sistema *Cycle-DAQ* fue diseñado para obtener la altitud, longitud, latitud, velocidad, distancia recorrida, inclinación del vehículo y emisiones contaminantes, con el fin de poder realizar ciclos de manejo. Desde el año 2013 el *Cycle-DAQ* ha sido utilizado en diversas pruebas dinámicas, una de ellas la caracterización de un vehículo eléctrico.

Aunque el *Cycle-DAQ* es un sistema funcional, se implementaron algunas mejoras que derivaron en un sistema más versátil. Todas las acciones llevadas a cabo para mejorar al sistema *Cycle-DAQ*, entre ellas la elección de sistemas de posicionamiento global (GPS) más precisos, la incorporación de una pantalla táctil, de un reloj de tiempo real y de un sistema de comunicación mediante protocolos GSM-GPRS, se presentan en este trabajo de tesis.

El trabajo está organizado de la siguiente manera: una introducción, cuatro capítulos, la bibliografía y los apéndices. Los temas que se tratan en cada apartado son:

Introducción. Se describen los antecedentes del proyecto.

Capítulo 1. Se abordan los principios de funcionamiento de cada una de las partes que componen el sistema desarrollado. Estos temas permitirán comprender el funcionamiento del mismo.

Capítulo 2. Se presenta la información correspondiente al desarrollo del sistema. Se describe tanto el hardware como el software del equipo.

Capítulo 3. Se muestran las acciones que se llevaron a cabo para corregir los problemas detectados en el sistema.

Capítulo 4. Se exponen los resultados obtenidos con el sistema completamente funcional y se dan las conclusiones del trabajo realizado.

Es importante mencionar que, a lo largo del presente trabajo se utilizan siglas y acrónimos. El significado de las siglas y acrónimos cuyo significado se encuentra en el glosario de términos.

INTRODUCCIÓN

Durante los años 2012 y 2013 y con el fin de desarrollar ciclos de manejo, personal del Laboratorio de Control de Emisiones de la Facultad de Ingeniería y de la Coordinación de Instrumentación del Instituto de Ingeniería, desarrolló un sistema no invasivo que permite realizar la adquisición y el registro de los siguientes parámetros vehiculares: posición (longitud, latitud, altitud e inclinación), velocidad, distancia recorrida y medición de las emisiones contaminantes. Cabe mencionar que, la finalidad de la información recolectada por el dispositivo es la de integrar ciclos de manejo.

El sistema desarrollado recibió el nombre de *Cycle-DAQ*, figura I.1, y está constituido por: un GPS para obtener la longitud, latitud y altitud; un acelerómetro para obtener la inclinación; una interfaz RS-232 para un analizador de gases; una memoria microSD para almacenar la información obtenida; una pantalla para desplegar algunos de los parámetros adquiridos y un microcontrolador para calcular la distancia y la velocidad, procesar la información y controlar el funcionamiento de los demás dispositivos.

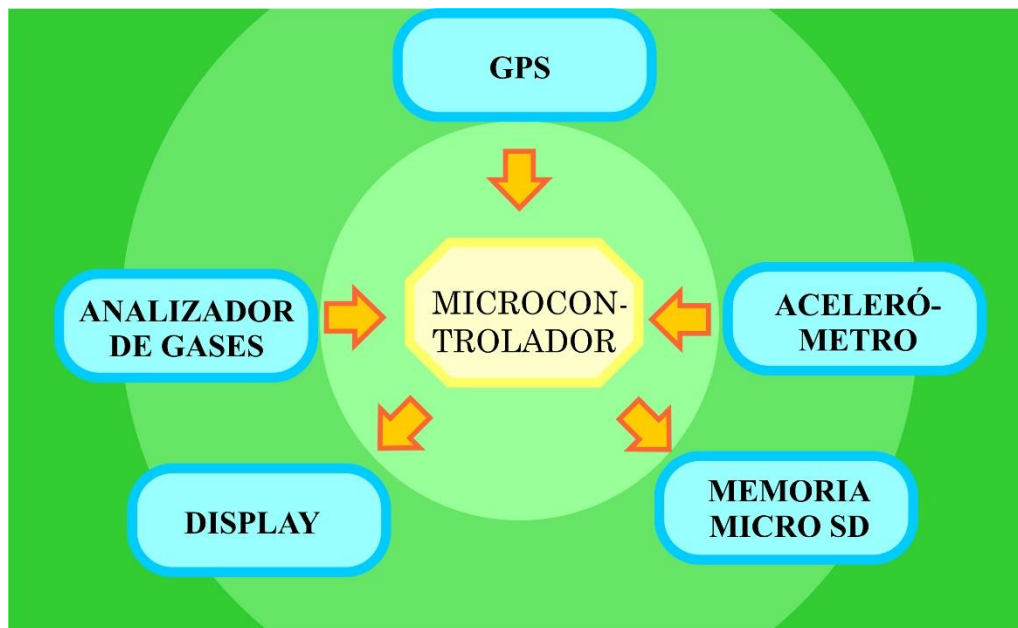


Figura I.1. Diagrama de bloques del sistema Cycle-DAQ.

A inicios del año 2014 se propusieron diversas mejoras al sistema, entre ellas la de integrar una pantalla táctil, un reloj de tiempo real y un sistema que permitiera la comunicación mediante los protocolos GSM-GPRS. Como parte de las actividades de los alumnos de servicio social, a cargo del M.I. Lauro Santiago Cruz, se comenzó a trabajar sobre dichas mejoras. Paralelamente, el sistema con el que se contaba fue sometido a diferentes pruebas para poder conocer las fallas del mismo y con dicha información desarrollar un sistema fiable. Durante dicho periodo se lograron hacer diversos avances en el proyecto, sin embargo no se logró que el sistema fuera estable. Para dar conclusión al proyecto y las dificultades de éste, se propuso el presente tema de tesis, cuyo principal objetivo es el de desarrollar un sistema más versátil que su antecesor, que esté construido en una sola placa de circuito impreso, que mantenga las prestaciones del sistema anterior y que permita el monitoreo de los parámetros vehiculares en un puesto central.

GENERALIDADES

En este capítulo se desarrollan los temas básicos para comprender el funcionamiento del sistema. En él se explica qué son los ciclos de manejo y las características de los mismos. Así mismo se presentan los protocolos de comunicación, los principios de funcionamiento y las características de los dispositivos que conforman el sistema.

1.1. Sistemas de adquisición de datos

Los sistemas de adquisición de datos son instrumentos utilizados para recolectar información para documentar o analizar un fenómeno. Dependiendo de la naturaleza del fenómeno y las características de las variables de interés se puede utilizar dos tipos de sistemas de adquisición de datos: basados en una computadora y sistemas autónomos.

Los sistemas basados en computadora se conforman por dos partes, las cuales son:

- a) **Computadora personal (PC, *personal computer*):** Se usa para programar equipo, manipular o almacenar información, generar reportes o realizar gráficos. No es necesario que la PC esté conectada o controlando todo el tiempo al equipo de adquisición de datos.
- b) **Equipo de pruebas:** El equipo de pruebas puede realizar una o más medidas o procesos de control usando diferentes combinaciones de entradas y salidas analógicas o digitales. Generalmente está constituido por sensores, acondicionadores de señal, procesadores, dispositivos de salida y unidades de almacenamiento.

Por otro lado, los sistemas autónomos están diseñados para obtener información del fenómeno sin estar conectados a una computadora. Es decir, cuentan con una alimentación propia y la medición, registro, almacenamiento y despliegue de información son llevados a cabo por el propio sistema. Dichas características permiten que estos dispositivos puedan ser portátiles. Aunque estos sistemas no dependen de una computadora pueden establecer comunicación con una.

1.2. Ciclos de manejo

Un ciclo de manejo es un perfil de velocidades trazado en una gráfica velocidad-tiempo, que presenta una forma típica de conducir por los distintos tipos de calles en ciudades y por autopistas, tomando en cuenta factores como la tecnología del vehículo, las características del tráfico, condiciones climáticas (velocidad y dirección de viento, temperatura y presión atmosférica) y geográficas (latitud, longitud, altitud, velocidad), así como las características de conducción de los vehículos por parte de los usuarios.

Los ciclos de manejo se utilizan para conocer el rendimiento de combustible y las emisiones contaminantes de los vehículos, y también para modelar el desempeño de los motores de combustión interna, las transmisiones y las baterías, entre otros. Para lograr conocer todo esto, se han establecido ciclos de manejo estándar, los cuales han sido desarrollados por diversos países y compañías automotrices. Los ciclos de manejo estándar se llevan a cabo en dinamómetros y consisten en pruebas que simulan el recorrido de un automóvil a lo largo de una distancia definida; el recorrido está integrado por etapas de aceleración y frenado, las cuales tienen tiempos definidos y velocidades máximas.

Uno de los ciclos de manejo estándar más conocidos a nivel mundial es el *Urban Dynamometer Driving Schedule (UDDS)*; el cual, desde la década de los 60's, ha sido modificado a lo largo de los años para adecuarse a las nuevas tecnologías y condiciones de manejo.

1.2.1. Importancia de los ciclos de manejo en México

Así como en otros países, en México se han instaurado normas para reducir las emisiones contaminantes de los vehículos automotores. En nuestro país, la NOM (Norma Oficial Mexicana)-044-SEMARNAT (Secretaría de Medio Ambiente y Recursos Naturales)-2014 y la NOM-076-SEMARNAT-2012, son ejemplos de normas que regulan los niveles máximos permisibles de emisiones contaminantes producidas por vehículos automotores. Entre otras cosas, estas normas establecen que para determinar las emisiones contaminantes se debe utilizar un ciclo de manejo conocido como FTP (*Federal Test Procedure*)-75. El ciclo de manejo FTP-75 es una variación del ciclo de manejo UDDS.

1.2.2. Metodología para el desarrollo de ciclos de manejo representativos

Un ciclo de manejo es representativo cuando todos los factores que intervienen en el proceso de desarrollo son lo más cercanos posible a las condiciones reales de operación de los vehículos. Dichos factores son:

- Características del conductor

- Características del vehículo
- Características del tráfico

El proceso para desarrollar ciclos de manejo representativos comienza con la adquisición y registro de los datos de las condiciones reales en que el automóvil se encuentra. La información adquirida se procesará en cuatro etapas: a) segmentación de la información, b) clasificación de la información, c) síntesis del ciclo de manejo y d) evaluación del ciclo de manejo y análisis comparativo. Tras finalizar este procesamiento se obtendrá el ciclo de manejo.

1.3. Protocolos de comunicación

Los protocolos de comunicación son un conjunto de reglas que definen el formato y el significado de los paquetes de información intercambiados entre los sistemas digitales. La comunicación para adquirir la información de los paquetes digitales, se puede hacer mediante comunicación paralela o comunicación serie.

La comunicación paralela consiste en utilizar varios hilos para enviar y recibir información en múltiples transmisiones realizadas simultáneamente. La ventaja de utilizar este tipo de comunicación es que la misma cantidad de información puede ser enviada y recibida rápidamente, siendo la velocidad proporcional a la cantidad de conexiones físicas. Por otro lado, la desventaja de utilizar este tipo de comunicación es que tiene un costo elevado, ya que ocupa mucho espacio debido a las conexiones físicas necesarias para realizarla; así mismo, la programación requerida para controlar el flujo de datos debe ser robusta.

La comunicación serie es un método que permite ocupar menos conexiones físicas para realizar la transferencia de datos. En la comunicación serie existen dos posibilidades para transferir datos: comunicación síncrona (mantiene la sincronización entre el emisor y receptor por medio de una señal de reloj) y comunicación asíncrona (no requiere de señales de reloj, en cambio se agregan bits de inicio y de finalización a la información que se envía). Independientemente de que la comunicación sea síncrona o asíncrona, la información se manda por menos canales que en la comunicación paralela. Al reducir la cantidad de canales, logramos reducir el espacio y los costos del sistema; sin embargo, se reduce la velocidad de transferencia y en algunos casos se requiere establecer protocolos de comunicación para poder lograr una transferencia sin errores. A lo largo de los años se han creado diferentes métodos de comunicación y establecido varios protocolos; no obstante lo anterior, en el presente trabajo sólo se describirán tres tipos de comunicación serie: SPI (*Serial Peripheral Interface*), I²C (*Inter Integrated Circuits*) y asíncrona, los cuales corresponden a los protocolos implementados en el sistema desarrollado.

1.3.1. Comunicación serie SPI

El protocolo SPI es un estándar de comunicación serie síncrona desarrollado por Motorola que permite la comunicación entre dos o más dispositivos, usando sólo cuatro hilos, llamados: *Master Output Slave Input* (MOSI), *Master Input Slave Output* (MISO), *Serial Clock* (SCK) y *Slave Select* (SS).

Los dispositivos conectados mediante SPI son clasificados como maestros (*Masters*) o esclavos (*Slaves*). Los maestros son los dispositivos que pueden proporcionar una señal de reloj en la línea SCK y pueden seleccionar a otros dispositivos mediante la línea SS, los demás son considerados esclavos.

En el protocolo SPI las líneas del mismo nombre se deben conectar entre sí, tal y como se muestra en la figura 1.1.¹ En esta figura, también se puede notar que ambos dispositivos cuentan con un registro de corrimiento de 8 bits, los cuales están conectados mediante los hilos MOSI y MISO formando un registro de 16-bits distribuido.

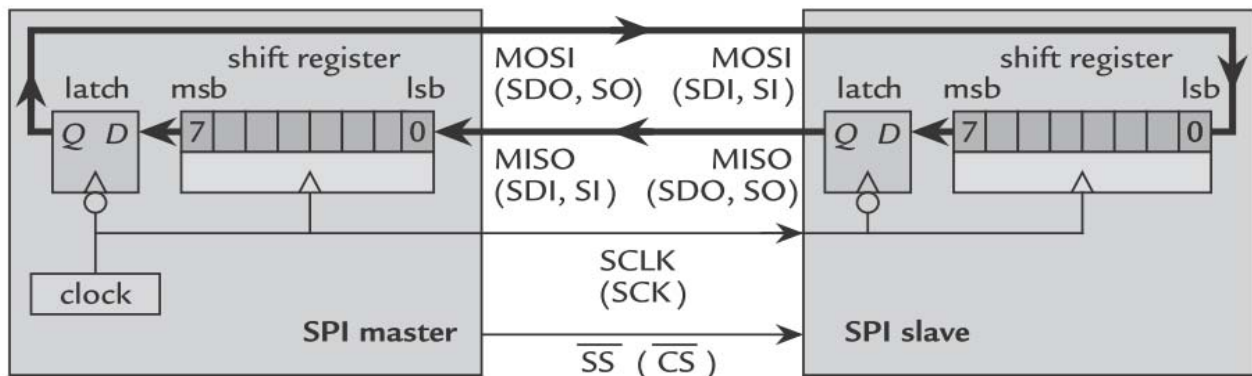


Figura 1.1. Representación esquemática de SPI.

Aunque no se muestra en la figura 1.1 los dispositivos conectados mediante SPI (o al menos el módulo maestro) cuentan con ocho secciones de memoria, los cuales se muestran en la Tabla 1.1. Cabe mencionar que las direcciones presentadas en dicha tabla son sólo con fines ilustrativos.

Las funciones de los registros que pueden ser modificados por el usuario son las siguientes: los registros de control y de velocidad de transferencia, entre otras funciones, son los que determinan la velocidad, el modo y la forma en que se intercambiará la información; el registro de estado, permite conocer si se está transfiriendo información; el registro de información contiene la información que se envía y se recibe (corresponde al registro de corrimiento mostrado en la figura 1.1).

¹ Davies, J. (2008). *Serial peripheral interface between a master and a single slave. Conceptually the whole system is a shift register. I show a selection of the more common names used for the signals and choose a particular set of option for the protocol.* [Figura]. En *MSP430 Microcontroller Basics* (p.498). Estados Unidos: Elsevier.

Dirección	Función	Permisos
\$_0	Registro de control 1 de SPI (SPICR1)	Lectura/Escritura
\$_1	Registro de control 2 de SPI (SPICR2)	Lectura/Escritura
\$_2	Registro de velocidad de transferencia SPI (SPIBR)	Lectura/Escritura
\$_3	Registro de estado de SPI (SPISR)	Lectura
\$_4	Reservado	Sin permisos
\$_5	Registro de información de SPI (SPIDR)	Lectura/Escritura
\$_6	Reservado	Sin permisos
\$_7	Reservado	Sin permisos

Tabla 1.2. Mapa de los módulos de memoria de SPI.

Dentro del registro de control 1 existen dos bits llamados: *Clock Polarity* (CPOL) y *Clock Phase* (CPHA). CPOL define cuando se considera inactiva la señal de reloj y CPHA cuando se muestrean los datos. Estos bits permiten establecer uno de los cuatro posibles formatos de la señal de reloj, lo cual permite comunicar componentes de distintos fabricantes. Los cuatro formatos, también llamados modos de operación se describen en la Tabla 1.2.

El resto de los bits que conforman el SPICR1 indican si se enviará el primero el bit más significativo (MSB) o el bit menos significativo (LSB), cual módulo es el maestro y cual el esclavo e interrupciones para inicio y fin de transmisión. Por otro lado, el SPICR2 está formado por bits encargados de establecer si la comunicación será bidireccional, si hubo fallas en la transmisión y si el sistema SPI está en modo de espera.

El SPIBR contiene bits que indican la velocidad de transferencia del sistema. Los bits del SPISR proporcionan información sobre la condición del registro de información, por ejemplo si el SPIDR está vacío o si está recibiendo información. El SPIDR está dedicado para almacenar la información que se envía o recibe durante una transmisión.

Cuando las líneas están conectadas adecuadamente y los registros estén configurados de acuerdo a las especificaciones de los componentes, es posible realizar la comunicación de datos bajo este protocolo.

De manera general la transferencia de información se lleva a cabo de la siguiente manera:

- El maestro suministra la señal de reloj.
- El maestro selecciona el dispositivo esclavo con el que se comunicará mediante la línea \overline{SS} .
- De acuerdo al modo de operación se transmite un bit del registro de información en las transiciones de la señal de reloj. Dependiendo de la configuración de los registros, se envía primer el MSB o el LSB. MOSI transfiere la información del maestro y MISO la

información del esclavo. La información que se envía fue previamente cargada a los registros de información.

- Después de ocho ciclos de reloj la información de los registros de información se ha intercambiado por completo.
- Se lee la información de los registros de información y se reinicia el ciclo.

CPOL	CPHA	Descripción
0	0	La señal de reloj se considera inactiva cuando está en nivel bajo. La información debe estar disponible antes del flanco de subida y debe mantenerse mientras la señal de reloj esté en alto. La información se adquiere en la transición de bajo a alto y se recorre en la transición de alto a bajo.
0	1	La señal de reloj se considera inactiva cuando está en nivel bajo. La información debe estar disponible antes del flanco de bajada y debe mantenerse mientras la señal de reloj esté en bajo. La información se adquiere en la transición de alto a bajo y se recorre en la transición de bajo a alto.
1	0	La señal de reloj se considera inactiva cuando está en nivel alto. La información debe estar disponible antes del flanco de bajada y debe mantenerse mientras la señal de reloj esté en bajo. La información se adquiere en la transición de alto a bajo y se recorre en la transición de bajo a alto.
1	1	La señal de reloj se considera inactiva cuando está en nivel alto. La información debe estar disponible antes del flanco subida y debe mantenerse mientras la señal de reloj esté en alto. La información se adquiere en la transición de bajo a alto y se recorre en la transición de alto a bajo.

Tabla 1.2. Modos de operación de SPI.

Como ya se ha mencionado, el protocolo SPI está diseñado para funcionar como un registro de corrimiento de 16 bits distribuido. Ya que, cuando el maestro envía un bit por MOSI, los bits que aún no se han enviado se recorren una posición en el registro de información, el espacio desocupado se utiliza para almacenar un bit recibido en MISO (la transferencia y recepción se realiza al mismo tiempo). Esto mismo sucede en el esclavo y se repite hasta intercambiar los bits de los registros de información. Sin embargo, no todos los dispositivos pueden transmitir información al mismo tiempo que la reciben, e inclusive hay dispositivos que sólo reciben información. Para mantener el concepto de registro de corrimiento el sistema mantiene su operación, es decir se transmite bit por bit hasta completar una transmisión de ocho bits y cada vez que se trasmite un bit, los bits restantes se recorren una posición. Sin embargo, los espacios vacíos (por el proceso de corrimiento) son ocupados por bits basura y con ello se conserva el principio de operación. Por ejemplo, en el caso de transmisión half dúplex (la transferencia y recepción de información se realiza en procesos por separado) el maestro envía información por

MOSI y el registro del maestro se llena con información basura, ya que MISO no está transmitiendo; en la siguiente proceso, MISO está transmitiendo bits y como MOSI está desactivado el registro de información del esclavo es ocupado por información basura. Es importante hacer notar que sólo cuando se ha completado el envío o recepción de información se pueden leer o escribir los registros de información.

Para obtener mayor información sobre el protocolo SPI o los registros SPI se recomienda consultar el documento *SPI Block Guide V03.06* de Motorola.

1.3.2. Protocolo de comunicación I²C

El protocolo de comunicación de circuitos inter-integrados (*I²C*) fue creado por Philips Semiconductor. Desde que la patente expiró en 2006, este protocolo se ha convertido en un protocolo ampliamente usado. Su principal característica es que se comunica sólo mediante dos líneas bidireccionales, llamadas *Serial data (SDA)* y *Serial Clock (SCL)*. Debido a esta característica el protocolo también es conocido como interfaz de dos hilos (*TWT*). El costo de proporcionar una comunicación usando menos líneas que la comunicación SPI se refleja en la pérdida de velocidad, en el envío de bits adicionales a la información y en la necesidad de implementar hardware más robusto, como una máquina de estados, en vez de un simple registro de corrimientos.

Con el fin de realizar una transferencia de información en este protocolo, la cual se lleva a cabo entre un maestro y un esclavo, se requiere que el maestro envíe una secuencia de bytes con el MSB enviado primero.

La transferencia de información sigue las siguientes condiciones y reglas:

- La información en la línea en SDA debe mantener estable mientras SCL está en alto.
- El estado de SDA debe cambiar sólo cuando SCL esté en bajo.
- La información debe ser leída tras los flancos de subida y escrita después de los flancos de bajada.
- Cuando no se esté realizando comunicación las líneas deben estar en un nivel alto.
- La condición de inicio corresponde a un flanco de bajada en SDA mientras la señal de SCL está en alto. Esta condición es enviada por el maestro.
- Una vez que se haya dado la condición de inicio, el maestro debe iniciar los pulsos del reloj, así mismo debe colocar el primer bit de la dirección en SDA mientras SCL está en nivel bajo.
- El valor en SDA es válido después de que SCL ha cambiado a un nivel alto. Este valor es leído por todos los esclavos conectados a las líneas.

- Los dos pasos anteriores se repiten hasta que se envían 7 bits, los cuales corresponden a la dirección de los dispositivos conectados a la línea. Cabe mencionar que cada dispositivo cuenta con una dirección única.
- El siguiente bit indica si se escribirá o se leerá el dispositivo de la dirección indicada. Con este bit se termina la transferencia del primer byte.
- Como respuesta al primer byte, el esclavo debe enviar un bit de reconocimiento. El componente que reconozca la dirección responderá con un nivel bajo, con ello indica que ha reconocido la dirección.
- El maestro debe de comprobar el bit de reconocimiento, si no ha obtenido respuesta se aborta la transferencia de información.
- Los siguientes ocho ciclos de reloj se utilizarán para enviar un byte de información, ya sea del maestro al esclavo, cuando se escribe, o del esclavo al maestro cuando se lee. El maestro continúa proveyendo la señal de reloj.
- Se envía un bit de reconocimiento. Tanto el maestro como el esclavo tienen que enviar el bit de reconocimiento cuando reciben información. Si el bit de reconocimiento no es enviado el maestro puede enviar la condición de paro o mandar la condición de inicio nuevamente.
- Después del bit de reconocimiento es posible volver a recibir o enviar un byte. Es decir, que se enviará un byte y el dispositivo receptor de la información deberá enviar el bit de reconocimiento. Este proceso se puede repetir indefinidamente.
- Es posible que antes de enviar la condición de paro se pueda escribir a otro registro, cambiar la acción de escritura a lectura y viceversa, o bien cambiar de dispositivo esclavo. Para realizar estas acciones no debe haber transferencia de información, se vuelve a dar la condición de inicio y la dirección del dispositivo esclavo, tras lo cual se repiten los procesos descritos anteriormente hasta que se dé la condición de paro.
- Para finalizar la comunicación el maestro pone a la línea de SDA en un nivel bajo después de un flanco de bajada de la señal de reloj. La línea SDA regresa a un nivel alto después del último flanco de subida del reloj, con ello se da un flanco de subida en SDA mientras SCL está en alto, con lo que se culmina la transmisión.

En la figura 1.2² se presenta el diagrama de tiempo correspondiente a la transmisión de información mediante I²C. Para ampliar la información sobre este protocolo se recomienda leer el documento *AN10216-01 I²C Manual* de NXP Semiconductors.

² Davies, J. (2008). *Simple transfer over I²C. The master writes an address, which is acknowledge by the slave, and read a single byte from the slave.* [Figura]. En *MSP430 Microcontroller Basics* (p.537). Estados Unidos: Elsevier.

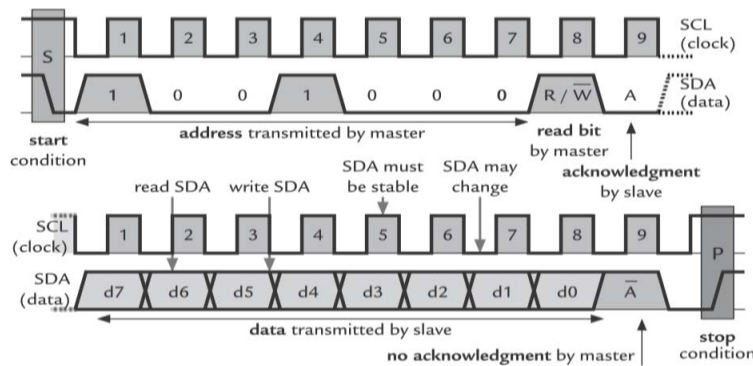


Figura 1.2. Diagrama de tiempo de I²C.

1.3.3. Comunicación serie asíncrona

La comunicación serie asíncrona se caracteriza por utilizar únicamente dos hilos de comunicación llamados Rx y Tx. Al igual que en el protocolo I²C, la utilización de sólo dos hilos para la comunicación implica la utilización de un hardware robusto y de un protocolo de comunicación bien definido. Cabe mencionar que la comunicación serie asíncrona, en diversos textos y medios de consulta, recibe el nombre de comunicación serie.

La comunicación serie asíncrona está definida por los siguientes parámetros:

- **Velocidad de transferencia:** La velocidad de transferencia o *baud rate*, es el parámetro que define la cantidad de bits que se envían por segundo en la línea.
- **Bloques de información:** En la comunicación asíncrona no hay una señal de reloj que determine los tiempos de transferencia de información. Por ello, para poder indicar el inicio y finalización de una transferencia se deben agregar dos bits (uno de inicio y otro de finalización) a la información que se desea enviar. Es muy común que los bloques de información se conformen de por un bit de inicio, ocho bits de información y un bit de parada. Dependiendo de las características de los sistemas de comunicación se pueden conformar bloques con más de ocho bits de información e incluir un bit de paridad. Los parámetros mencionados deben de ser iguales en los sistemas que se comunican, de no ser así se presentará información basura.

Una de las características más importantes de este protocolo es que no cuenta con una línea de reloj, de ahí el apelativo asíncrono. No obstante, cada uno de los dispositivos que soporta esta comunicación cuenta con un reloj local. Las líneas Rx (recepción) y Tx (transmisión) se utilizan únicamente para enviar información.

Ya que la comunicación no está coordinada por un señal de reloj, una parte fundamental de la comunicación serie asíncrona es la implementación de circuito conocido como UART (*Universal*

Asynchronous Receiver/Transmitter). El UART es el dispositivo responsable de transmitir y recibir la información mediante comunicación serie asíncrona. Considerando las funciones del UART, éste se puede dividir en dos bloques, transmisión y recepción. El bloque de transmisión se encarga de crear el paquete de información, éste añade los bits de sincronización (inicio y paro) y los bits de paridad a la información que se enviará en la línea Tx (la información se envía con la velocidad de transmisión programada en el UART). El bloque de recepción se encarga de muestrear la línea de Rx a una velocidad superior y proporcional a la velocidad de transmisión, además se encarga de quitar los bits de sincronización y paridad, para obtener la información. Como se puede observar en la figura 1.3, el módulo UART tiene una sección de comunicación paralela y la otra la sección de comunicación serie asíncrona, la función del lado paralelo es la de comunicarse con un procesador.

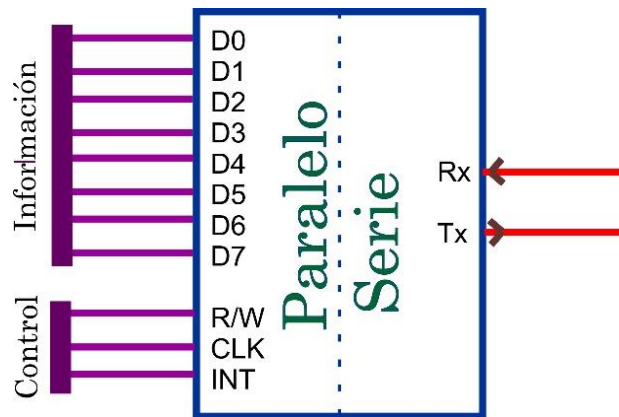


Figura 1.3. Esquema simplificado de un UART.

Debido a que los bloques de información se pueden construir de diferentes maneras (más bits de información, más de un bit de parada o la existencia de un bit de paridad) la comunicación serie asíncrona no está definida para un solo protocolo, por lo cual, es necesario consultar la guía de referencia del protocolo a usar. Algunos ejemplos de los protocolos de comunicación serie asíncrona son: RS-232, RS-422 y RS-485.

No obstante de la variedad de protocolos, éstos siguen algunos principios como los que se enlistan a continuación:

- Los sistemas que se comunican deben estar configurados para la misma velocidad de transferencia. Los receptores deben correr a velocidades superiores a la velocidad de transferencia.
- El tamaño de bloque de información debe ser igual para ambos dispositivos.
- Deben de seguir el mismo protocolo de comunicación.
- La comunicación comienza con una transición del nivel de señal en espera a un cambio de estado. Este cambio está determinado por el bit de inicio.

- Una vez que se ha mandado el bit de inicio se muestrea la entrada. Ésta es muestreada durante la mitad del tiempo que dura un bit. Esto se realiza para confirmar que la condición de inicio no sea un disparo en falso. Esto se ilustra en la figura 1.4.
- Una vez que se ha confirmado la condición de inicio, se comienza a muestrear la entrada. En esta etapa el tiempo de muestreo dura lo que un bit y con ello se consigue leer el primer bit (enviar primero el bit menos significativo es la tendencia más seguida actualmente).
- El paso anterior se repite hasta recibir todos los bits de información.
- Se espera el tiempo que dura un bit y se corrobora que la entrada se encuentra en el nivel de línea en espera como corresponde al bit de paro.

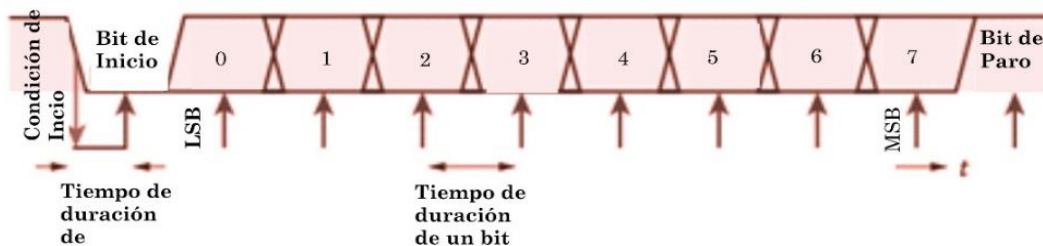


Figura 1.4. Diagrama de muestreo en comunicación serie asíncrona.

1.3.3.1 Protocolo RS-232

El protocolo de comunicación RS (*Recommended Standard*)-232, es un protocolo de comunicación serie asíncrono desarrollado en la década de 1960, creado para conectar equipos como los teletipos a un modem. El estándar RS-232 establece que:

- El uno lógico se representa por niveles de tensión entre -15 [V] y -3[V]. El uno lógico también se conoce como marca.
- El cero lógico se presenta por niveles de tensión entre +3 [V] y + 15[V]. El cero lógico también es llamado espacio.
- Los niveles de tensión entre -3 y +3 [V], no están definidos para información válida.

Este estándar sigue los principios de comunicación serie asíncrona antes definidos. Por otro lado, aunque este estándar tiene varios años de existencia, no ha perdido vigencia y aún se utiliza para comunicar diversos equipos a una computadora. Es importante hacer notar, que la mayoría de los dispositivos actualmente fabricados trabajan con lógica de 0-5 [V] y de 0-3 [V]; para poder comunicar este tipo dispositivos con los que utilizan el protocolo RS-232 es necesaria una interfaz.

Cabe mencionar que la conexión más completa de este protocolo se implementa con un conector de 25 terminales, al utilizar la conexión más completa se tiene acceso a dos canales de comunicación asíncrona. No obstante lo anterior, el protocolo es más ampliamente utilizado con un conector de nueve terminales, con lo cual sólo se tiene acceso a un canal de comunicación. Es posible utilizar el conector de nueve canales sin habilitar todas sus terminales, en otras palabras se puede implementar el protocolo RS-232 únicamente con las terminales de transmisión, recepción y tierra. Esta estructura es la que se utilizará en el desarrollo del presente trabajo de tesis. Para conocer más sobre el protocolo y sus conexiones se recomienda consultar la norma TIA (*Telecommunications Industry Alliance*)/EIA (*Electronic Industries Alliance*)-232-F.

1.4. Sistema de Posicionamiento Global

El GPS (*Global Positioning System*), es un sistema de navegación satelital que fue desarrollado por el Departamento de Defensa de los Estados Unidos a inicios de la década de 1970. Aunque inicialmente fue pensado únicamente para fines militares, el gobierno de los Estados Unidos decidió que el sistema estuviera disponible para el público en general.

El GPS está formado por tres segmentos: el segmento espacial, el segmento de control y el segmento de los usuarios. Las cuales se describen a continuación:

- a) **Segmento espacial:** El segmento está conformado por una constelación de 24 satélites que se encuentran orbitando la Tierra a aproximadamente 20,200 km sobre el nivel del mar. Para asegurar la cobertura internacional, los satélites están acomodados de tal manera que cuatro satélites se encuentre en cada uno de los seis planos orbitales (figura 1.5³) transmitiendo ininterrumpidamente.

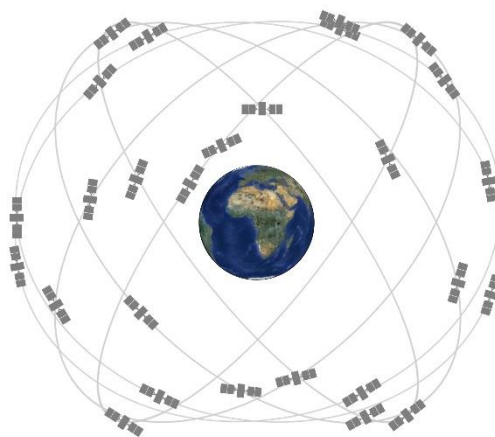


Figura 1.5. Planos orbitales de la constelación satelital GPS.

³ *Expandable 24-Slot satellite constellation, as defined in the SPS Performance Standard* [Imagen]. Recuperado de <http://www.gps.gov/systems/gps/space/>

- b) **Segmento de control:** Consiste en una red internacional de estaciones de monitoreo, todas ellas comandadas por una estación maestra de control (MCS). Esta red se encarga de monitorear a los satélites para determinar y predecir su ubicación, conocer su integridad, el funcionamiento de los relojes atómicos con los que cuenta cada satélite, entre otras. Esta sección también se encarga de modificar la información de los satélites.
- c) **Segmento de usuarios:** Se conforma por todos los usuarios (civiles o militares), a nivel mundial, que cuentan con un receptor GPS.

Cabe mencionar, que la fuerza aérea de Estados Unidos, administra y da mantenimiento a los segmentos espacial y de control. En la figura 1.6, se muestra la interacción entre los segmentos GPS.

La idea básica del GPS es la de proveer a los usuarios, bajo cualquier condición climatológica y en todo momento, información de posición, navegación y tiempo. Para cumplir con esta idea, los satélites GPS transmiten continuamente una señal de radio en la que indican su posición y el tiempo de sus relojes atómicos. El receptor de GPS a través de su antena recibirá la señal de los satélites. La información recibida se procesa con el software interno del receptor, lo que permite obtener la distancia a los satélites y las coordenadas de los satélites. Teóricamente, sólo es necesario conocer la posición y distancias de tres satélites para poder triangular la posición del receptor en la Tierra, sin embargo en la práctica se utiliza un cuarto satélite para poder corregir la base de tiempo del receptor y con ello lograr una medición lo más precisa posible. Es importante mencionar que se han desarrollado técnicas, servicios y tecnología para mejorar la precisión de las mediciones obtenidas mediante GPS; debido a que en el presente trabajo sólo se utiliza el servicio estándar de GPS, no se describirán las opciones para obtener mediciones más precisas.

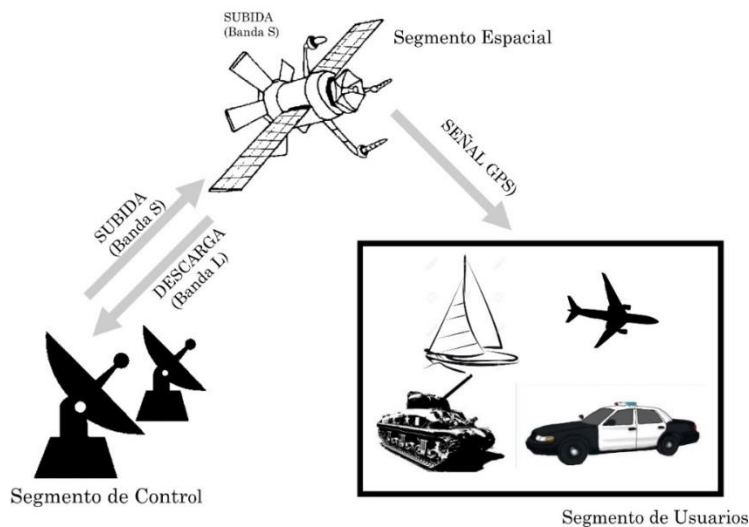


Figura 1.6. Segmentos GPS.

1.4.1. Protocolo NMEA 0183

Al igual que los diversos protocolos de comunicación que se han desarrollado, también se han creado diversos formatos para obtener la información entregada por el receptor de GPS. Uno de los más ampliamente usados es el NMEA (*National Marine Electronics Association*) 0183.

La NMEA fue fundada en 1957 por un grupo de comerciantes, esto con el fin de estrechar relaciones con los fabricantes de equipo electrónico y con ello lograr una interfaz que permitiera el intercambio de información digital entre los diferentes productos electrónicos marítimos. En 1983 la NMEA adopta el formato NMEA 0183 como interfaz para comunicar los dispositivos marítimos. Algunas de las características de este formato son: que debe haber un flujo de datos en formato ASCII, la velocidad de transmisión debe ser de 4800 bps, los datos son enviados en forma de enunciados, cada enunciado inicia con un signo de pesos "\$", después de cada signo de pesos se escribe un campo de dirección de cinco caracteres (el campo de dirección se forma con identificadores, los cuales indican que dispositivo está transfiriendo la información y que tipo de información se enviará), a continuación se envía la información correspondientes y se termina con un retorno de carro y un salto de renglón (<CR><LF>).

En la industria marítima los receptores GPS son ampliamente usados, por ello dentro de la gama de identificadores que maneja el protocolo NMEA 0183, el identificador GGA es uno de los más importantes. El identificador GGA está asociado a la información correspondiente al tiempo y la posición. Para el presente trabajó sólo es necesario la obtención del enunciado GGA, cuya estructura es como de la siguiente manera:

*\$GPGGA,hhmmss.ss,lll.ll,a,yyyy.yy,a,x,xx,x.x,x.x,M,x.x,M,x.x,xxxx*hh<CR><LF>*

El significado de los términos se muestra en la Tabla 1.3.

Término	Descripción
\$	Delimitador de inicio de enunciado
GP	Identificador del dispositivo que envía la información (GPS en este caso)
GGA	Identificador de la información (GGA, <i>Global Positioning System Fix Data</i>)
,	Delimitador de campo de información
hhmmss.ss	Hora de la posición en formato UTC (horasminutossegundos.decimales)
lll.ll	Latitud (gradosminutos.decimales)
a	Norte o Sur (N o S)

Tabla 1.3. Explicación de los términos de un enunciado GGA. (Continúa)

Término	Descripción
yyyy.yy	Longitud (gradosminutos.decimales)
a	Este u Oeste (E o W)
X	Indicador de calidad de GPS. En una escala de uno a cinco, donde 1 corresponde a la calidad de información ofrecida por un receptor estándar.
xx	Número de satélites usados para genera la medición
x.x	HDOP, dilución de precisión horizontal. Indica el efecto sobre la medición debido a la posición geométrica de los satélites
x.x	Altitud sobre el nivel del mar
M	Metros, unidad en la que se mide la altitud
x.x	Altura sobre el nivel del mar considerando a la Tierra como un geoide, considerando el sistema coordenado WGS84
M	Metros, unidad en la que se mide el parámetro anterior.
x.x	Tiempo en segundos desde la última vez que se actualizó la información de un sistema con una técnica diferencial para obtener la información. La técnica diferencial es un método para obtener medidas más precisas.
xxxx	Identificador de la base utilizada en el método diferencial.
*	Carácter delimitador de la suma de error
Hh	Suma de error
<CR><LF>	Finalización del enunciado

1.4.2 Fuentes de error

Así como cualquier otra medición, las mediciones realizadas mediante GPS son susceptibles de errores sistemáticos y aleatorios. Estos errores se pueden clasificar dependiendo de la fuente que los origina.

- **Errores debido a satélites:** Estos errores incluyen los debidos a una trayectoria satelital equivocada y a los errores de los relojes satelitales.
- **Errores debido a los receptores:** Estos son generados por errores en el reloj del receptor, ruido en el receptor, problemas con la posición de la antena del receptor con respecto al satélite o la recepción de una señal que viaja por dos diferentes rutas.

- **Errores debido a la propagación de la señal:** Son debidos al retraso de la señal GPS al atravesar la ionosfera y la estratósfera.

Además de los efectos producidos por estos errores, la exactitud en la medición GPS de la posición también se ve afectada por la posición geométrica de los satélites a los que se enlaza el receptor. Dado que la posición de los satélites es muy importante, existe un número adimensional que indica la separación entre los satélites llamado dilución de la precisión (DOP, *Dilution of Precision*); este número entre más pequeño sea indica una buena distribución satelital, es decir que los satélites se encuentran muy bien distribuidos en el campo de visión del GPS. Para fines prácticos, el número DOP se estudia para la distribución vertical (VDOP, *Vertical Dilution of Precision*) y horizontal (HDOP, *Horizontal Dilution of Precision*) de los satélites; los cuales suelen representarse en valores enteros con un decimal en un rango de 0-15.

1.4.3. Descripción general del receptor GPS a utilizar

El sistema GPS utilizado en esta tesis es un receptor que cuenta con un procesador especializado en aplicaciones de navegación, SiRF Star IV, de la empresa CSR, el cual ofrece una sensibilidad grande (-163 dB) y un desempeño muy bueno en ambientes urbanos o con maleza espesa. El procesador cuenta con una tecnología que le permite conectarse a los satélites en menos de 35 segundos (en la mayoría de los casos), un amplificador de bajo ruido, un tamaño compacto (3 cm x 3 cm x 1.07 cm), permite la comunicación serie asíncrona y soporta el protocolo NMEA 0183 para los identificadores (GGA, GSA, GSV, RMC, VTG, GLL, ZDA). El diagrama de bloques del receptor se presenta en la figura 1.7.⁴

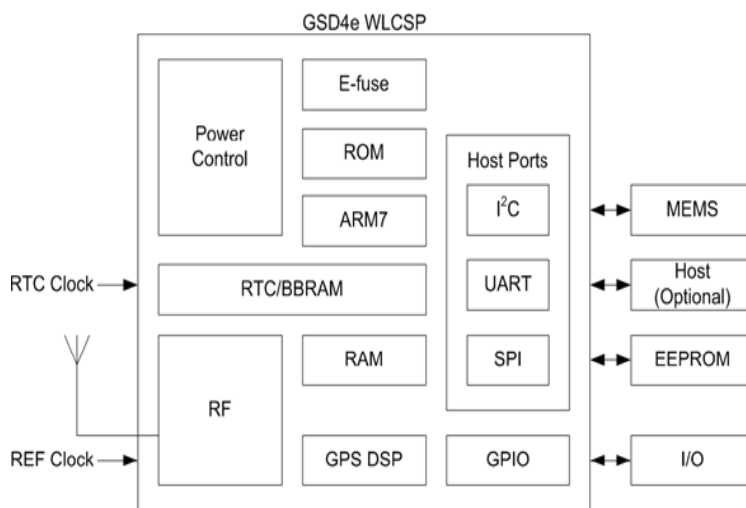


Figura 1.7. Diagrama de bloques del SiRF Star IV.

⁴ SiRFstarIV GSD4e [Imagen]. Recuperado de: <http://www.csr.com/products/35/sirfstariv-gsd4e>

Como se mencionó anteriormente, una de las mejoras instauradas al sistema *Cycle-DAQ* fue la elección de un receptor GPS más preciso. En la tabla 1.4 se hace la comparación entre el receptor usado en las versiones anteriores y el usado en la versión actual.

Parámetros	Receptor anterior	Receptor Actual
Alimentación	4.5-6.5 VDC	4.5-6.5 VDC
Chipset GPS	SiRF Star III	SiRF Star Iv
Frecuencia	1575.42 MHz	1575.42 MHz
Canales	20 canales	48 canales
Sensibilidad	-159 dB	-163 dB
Precisión de posición	5-10 metros	<2.5 metros
Precisión de velocidad	0.1 m/s	<0.01 m/s
Máxima altitud de trabajo	<18,000 m	<18,000 m
Velocidad máxima de trabajo	515 m/s	515 m/s
Aceleración máxima	<4G	<4G

Tabla 1.4. Comparación entre los receptores GPS.

Como se puede ver en la Tabla 1.4, el receptor actual cuenta con mayor número de canales de conexión, una mayor sensibilidad y es más preciso para determinar velocidad y posición. Estas cualidades permitirán que el sistema *Cycle-DAQ* obtenga información más exacta.

1.5. Reloj de tiempo real

Un reloj de tiempo real (RTC) es un circuito integrado que provee la hora y la fecha. Normalmente los RTC pueden ser alimentados con dos fuentes de alimentación, una de ellas corresponde a la alimentación del sistema donde se implementa y la otra es una batería de respaldo. Debido a que los RTC son circuitos de bajo consumo, la batería de respaldo suele ser una batería de baja potencia (por ejemplo, 3 [V] a 240 [mAh]). La batería de respaldo permite que los RTC mantengan su operación aunque la alimentación general del sistema no esté presente.

1.5.1. Descripción general del reloj de tiempo real implementado

El reloj de tiempo real implementado es un circuito capaz de entregar la fecha (año, mes y día); realizar correcciones para año bisiesto (hasta el año 2100); de ajustar, a fin de mes, la fecha para

los meses con menos de 31 días; proporcionar la hora (hora, minutos y segundos); entregar la hora, en formato de 24 y 12 horas; indicar si la hora es AM o PM, cuando el formato de 12 horas está activado; entregar una señal cuadrada y comunicarse mediante el protocolo I²C .

Como se muestra en la figura 1.8, el sistema se puede alimentar con una fuente externa (V_{CC}) y con una batería (V_{BAT}). La fuente de energía V_{CC} debe ser de 5 [V] y suministrar al menos 1.5 [mA], mientras que la batería debe ser de 3 [V] y suministrar como mínimo 500 [nA]. Si la fuente y batería están conectados al mismo tiempo, la fuente proporcionará la energía al circuito y sólo cuando ella no esté presente la batería alimentará al circuito. La batería sirve como fuente de alimentación de respaldo, lo cual permite que el sistema mantenga sus funciones cada vez que V_{CC} deja de alimentar al sistema; si la batería no está conectada y V_{CC} no está presente, el sistema perderá su información.

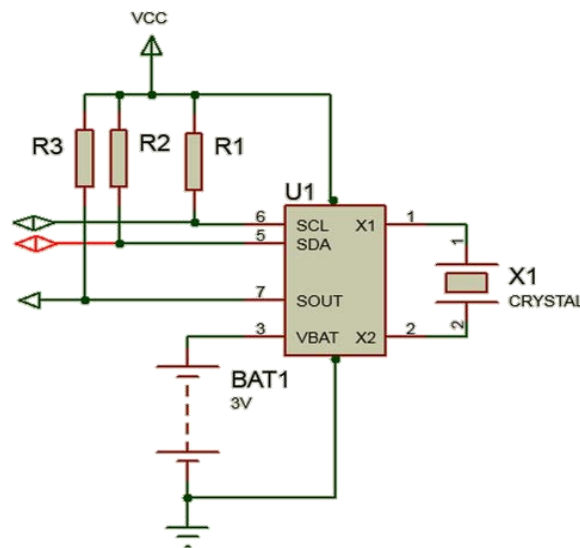


Figura 1.8. Esquema de conexión del RTC.

El RTC contiene siete registros que sirven para almacenar la información de la fecha y la hora, así como para configurar la señal cuadrada que puede entregar el circuito en la terminal llamada SOUT. En la Tabla 2.5 se muestran los registros de memoria y las direcciones de dichos registros:

Los registros de memoria cuya dirección va de 08h a 3Fh, corresponden a una memoria de acceso aleatorio estática (SRAM, *Static Random Access Memory*) disponible para almacenar 56 bytes de información (la SRAM puede ser ocupada para guardar cualquier información).

Como se mencionó con anterioridad, el protocolo I²C se utiliza para comunicarse con el reloj de tiempo real, el cual sólo se desempeñará como un dispositivo esclavo. Mediante este protocolo es posible cargar los registros con nuevos valores o bien leer los valores actuales, los cuales se encuentran en formato BCD (*Binary-Coded Decimal*). Cabe mencionar que si el dispositivo es nuevo, no ha sido energizado correctamente o se intentó programar con información inválida, el

bit de paro (CH) se pondrá en nivel alto. Cuando CH está en nivel alto, el sistema detiene la cuenta de tiempo y la generación de la señal cuadrada. Una vez que CH se pone en un nivel bajo el reloj comenzará a trabajar desde la última fecha y hora en la que se detuvo (sólo cuando se intentó programar con información errónea) o bien el sistema se reiniciará.

Dirección	Función	Permisos
00h	Guardar los segundos y cuenta con un bit de paro de reloj (CH, <i>Clock Halt</i>)	Lectura/Escritura
01h	Almacena los minutos	Lectura/Escritura
02h	Contiene la hora, así mismo maneja el formato de la hora	Lectura/Escritura
03h	Tiene la información del día (de 1 a 31)	Lectura/Escritura
04h	Indica el día de la semana (de lunes a domingo)	Lectura/Escritura
05h	Proporciona el mes	Lectura/Escritura
06h	Dedicado el año	Lectura/Escritura
07h	Establece los parámetros de la señal de cuadrada	Lectura/Escritura
08h-3Fh	Memoria de acceso aleatorio	Lectura/Escritura

Tabla 1.5. Registros de memoria.

1.5.2. Confiabilidad de los datos

El circuito integrado del RTC tiene un oscilador tipo Pierce, mostrado en la figura 1.9⁵, que proporciona la señal que permite llevar la cuenta del tiempo y que da origen a la señal cuadrada. Actualmente, los osciladores tipo Pierce son implementados en diversas aplicaciones; entre las ventajas de usar esta configuración es que es más económica, más pequeña y de menos consumo que otro tipo de circuitos osciladores.

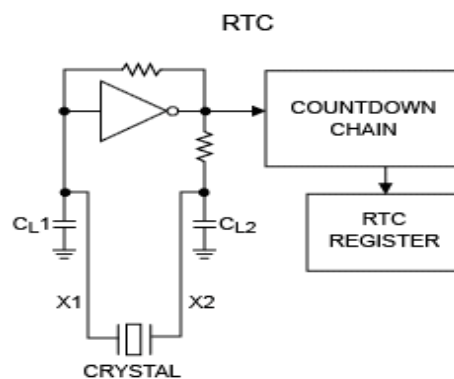


Figura 1.9. Esquema general de un oscilador Pierce dentro de un RTC.

⁵ RTC oscillator with internal load capacitors and bias resistors [Imagen]. Recuperado de: <http://www.maximintegrated.com/en/app-notes/index.mvp/id/58>

El mayor inconveniente al usar este tipo de circuitos es que la frecuencia del oscilador depende de un correcto acoplamiento de capacitancias. En otras palabras, para que el oscilador Pierce trabaje a la frecuencia correcta la capacitancia del cristal debe ser muy cercana o igual a la capacitancia del resto del circuito. Si las capacitancias no están bien acopladas, el oscilador interno del RTC trabajará a frecuencias más altas cuando la capacitancia del cristal sea menor y frecuencias menores cuando la capacitancia del cristal sea más alta. Es importante hacer notar que la capacitancia a la que se hace referencia está indicada en las hojas de datos de los relojes de tiempo real y de los cristales con el nombre de C_L .

Además del error debido al acoplamiento de capacitancias, la exactitud de los RTC es afectada por la variación de frecuencia, debido a la temperatura que sufren los cristales, esto se ilustra en la figura 1.10.⁶

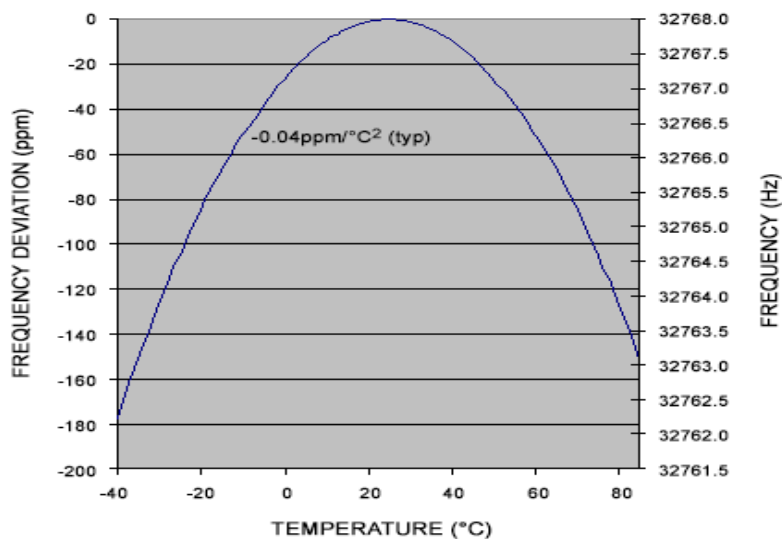


Figura 1.10.Desviación de frecuencia contra temperatura.

1.6. Acelerómetro

Un acelerómetro es un dispositivo electromecánico que mide las fuerzas que experimenta un cuerpo debido a su aceleración, las fuerzas pueden ser estáticas, como la fuerza de gravedad, o dinámicas, como las experimentadas durante un movimiento.

Existe una gran variedad de acelerómetros, muchos de ellos construidos con materiales piezoeléctricos, los cuales tienen la desventaja de ser sensores muy grandes. Dado que en diversas aplicaciones se requería utilizar acelerómetros, se comenzaron a buscar opciones que permitieran reducir el tamaño de los mismos. Uno de los campos en los que se llevó a cabo esta

⁶ *Crystal frequency vs. Temperature* [Imagen]. Recuperado de: <http://www.maximintegrated.com/en/app-notes/index.mvp/id/58>

investigación fue en el de la microelectrónica y en esta área se desarrollaron los acelerómetros implementando sistemas microelectromecánicos (MEMS, *Microelectromechanical Systems*).

El acrónimo MEMS fue acuñado en 1989, para describir un campo de investigación en los que algunos dispositivos mecánicos estaban siendo construidos en procesos más cercanos a la microelectrónica. Muchos de los conocimientos y técnicas utilizadas para construir circuitos microelectrónicos se adaptaron para construir dispositivos MEMS. Una gran diferencia al elaborar circuitos microelectrónicos y MEMS, es que los primeros son estructuras sólidas y compactas, mientras que los segundos suelen tener canales, cavidades, voladizos o membranas.

Existen diferentes formas de construir un acelerómetro usando MEMS, dos de los procesos más empleados son utilizar el efecto piezoeléctrico o medir la variación de capacitancia. Cuando se usa el efecto piezoeléctrico, se construyen sistemas con estructuras cristalinas microscópicas que al sufrir una fuerza de aceleración producen corriente eléctrica. Por otro lado, al medir la variación en la capacitancia se mide el efecto al variar el área entre los electrodos o bien la distancia entre los mismos. En el presente trabajo de tesis únicamente se explicarán los principios de funcionamiento de los acelerómetros que utilizan efectos capacitivos.

1.6.1. Medición de aceleración por efectos capacitivos

Los circuitos integrados que miden la aceleración con base en los efectos capacitivos son los más comunes actualmente, debido a que los procesos de fabricación son sencillos, tienen una sensibilidad muy buena, el mecanismo de transducción hace que el sistema sea independiente de la temperatura y la medición de la capacitancia es independiente del material con el que se construye el sistema, dependiendo únicamente de los cambios entre las placas del capacitor. Estos circuitos integrados son construidos de tal manera que la capacitancia que ocupan sea similar a la de un capacitor de placas paralelas (Ecuación 1.1).

$$C_0 = \epsilon_0 \epsilon \frac{A}{d} = \epsilon_A \frac{1}{d} \quad (1.1.)$$

Donde C_0 es la capacitancia, ϵ_0 es la permitividad eléctrica del vacío, ϵ es la permitividad del eléctrica del material entre las placas, A es el área de las placas y d la distancia entre placas.

Los acelerómetros MEMS en su mayoría son construidos para establecer una relación entre la variación de la capacitancia y los cambios de la distancia entre las placas de los capacitores. Con dicho fin, los circuitos integrados de acelerómetros MEMS cuentan con una masa de prueba que cuenta con varios voladizos y está sujeta por una suspensión mecánica (resortes) dentro de una estructura fija. Los voladizos de la masa de prueba se encuentran entre dos placas externas fijas (figura 1.11).

Como se puede observar en la en la figura 1.11, por cada voladizo se miden dos capacitancias, es decir se conforman dos capacitores C_1 y C_2 . También se puede observar que todos los capacitores C_1 y todos los capacitores C_2 están conectadas entre sí en un arreglo paralelo. Es importante mencionar que la existencia de varios voladizos y la conexión de todos los capacitores

del mismo tipo juntos, es con el fin de poder detectar los cambios en la capacitancia, ya que de otra manera los cambios serían indetectables. C_1 y C_2 son función de los desplazamientos x_1 y x_2 y su comportamiento está definido por:

$$C_1 = \epsilon_A \frac{1}{x_1} = \epsilon_A \frac{1}{d-x} = \frac{\epsilon_A}{d} \frac{1}{1-\frac{x}{d}}, \quad C_2 = \epsilon_A \frac{1}{x_2} = \epsilon_A \frac{1}{d+x} = \frac{\epsilon_A}{d} \frac{1}{1+\frac{x}{d}} \quad (1.2.)$$



Figura 1.11. Esquema simplificado de un acelerómetro MEMS

Si no hay aceleración las capacitancias C_1 y C_2 son iguales, debido a que los desplazamientos x_1 x_2 también son iguales. Por otro lado, si existe aceleración, la masa de prueba sufrirá un desplazamiento, $x \neq 0$, cuando esto sucede la diferencia de las capacitancias estará relacionada con el desplazamiento de la masa de pruebas, como se muestra a continuación:

$$C_1 - C_2 = \Delta C = \epsilon_A \frac{1}{d-x} - \epsilon_A \frac{1}{d+x} = \epsilon_A \left(\frac{d+x-d+x}{d^2-x^2} \right)$$

$$\Delta C = 2\epsilon_A \frac{x}{d^2-x^2} = 2 \frac{\epsilon_A}{d} \frac{\frac{x}{d}}{1-\left(\frac{x}{d}\right)^2} = 2C_0 \frac{\frac{x}{d}}{1-\left(\frac{x}{d}\right)^2} \quad (1.3.)$$

En la ecuación anterior el término $\left(\frac{x}{d}\right)^2$ será lo suficientemente pequeño como para prescindir de él, por lo que la ecuación quedaría como:

$$\Delta C \approx 2C_0 \frac{x}{d} \quad (1.4.)$$

Dos de los problemas que se hacen evidentes al analizar la ecuación 1.4 son: 1) que la ecuación es sólo una aproximación para obtener una relación lineal entre la capacitancia y la distancia y 2) esta ecuación cuenta con varios parámetros geométricos (considerados en C_0). Si únicamente se considerara esto, medir la diferencia de capacitancias para conocer el desplazamiento de la masa de pruebas, se podría asegurar que la fabricación de estos circuitos sería muy complicada y no

conveniente para su comercialización⁷. Sin embargo, para superar estas limitaciones los desarrolladores de acelerómetros MEMS diseñaron un circuito con un esquema de demodulación sensible a la fase, lo cual permite mantener una relación entre el desplazamiento y la diferencia de capacitancias sin medirla directamente.

El principio de funcionamiento de la demodulación sensible a la fase consiste en utilizar una señal alterna de referencia (normalmente una señal cuadrada) que se multiplica con la señal de interés. El resultado de la multiplicación es una señal alterna, la cual se hace pasar por un filtro pasa bajos. La salida del filtro pasa bajos en la mayoría de los casos será una señal nula y sólo cuando la frecuencia de la señal de entrada sea igual a la frecuencia de la señal de referencia se obtendrá una señal de corriente directa, cuya amplitud será proporcional a la amplitud de la señal de entrada.

Para aplicar el principio de demodulación sensible a la fase en los acelerómetros MEMS un oscilador interno produce dos señales alternas cuadradas de la misma amplitud y frecuencia (típicamente de 1 [MHz]), pero desfasadas entre sí 180° (V_0 y $-V_0$). Las señales producidas por el oscilador se conectan a los capacitores C_1 y C_2 , cada uno es alimentado sólo con una señal. La conexión de V_0 y $-V_0$ produce una diferencia de potencia (V_x) en la masa de prueba, la cual se hará pasar por un amplificador seguidor de tensión para evitar que se atenué. La señal de salida del amplificador será la entrada del demodulador sensible a la fase, mientras que una de las señales

(V_0 o $-V_0$) será la referencia del demodulador. En la figura 1.12 se muestra el esquema de conexión descrito.

Como se puede ver en la figura 1.12, la variable de entrada (variable de interés) del demodulador es la tensión V_y , la cual es igual V_x . En el nodo donde se encuentra V_x se puede considerar que el amplificador no proporciona ni drena corriente, por ello en el nodo la carga total de los capacitores debe ser nula. Considerando la relación $Q = CV$ (donde Q es la carga del capacitor, C es la capacitancia y V la tensión entre las placas del capacitor) se obtiene la siguiente expresión:

$$Q_1 + Q_2 = C_1(V_0 - V_x) + C_2(-V_0 - V_x) = 0 \quad (1.5.)$$

Despejando V_x de la ecuación 1.5, y usando las expresiones 1.2, y 1.3, se obtiene:

$$V_x = V_0 \frac{C_1 - C_2}{C_1 + C_2} = V_0 \frac{2C_0 \frac{\frac{x}{d}}{1 - (\frac{x}{d})^2}}{2C_0 \frac{1}{1 - (\frac{x}{d})^2}} = V_0 \frac{x}{d} \quad (1.6.)$$

Si la ecuación 1.6 se expresa en términos de los desplazamientos x_1 y x_2 resulta:

⁷ Andrejasic, M. (2008). *MEMS Accelerometers*. Eslovenia: Facultad de Matemáticas y Física, Departamento de Física, Universidad de Ljubljana.

$$V_x = V_0 \frac{C_1 - C_2}{C_1 + C_2} = V_0 \frac{\epsilon A \frac{x_2 - x_1}{x_1 x_2}}{\epsilon A \frac{x_2 + x_1}{x_1 x_2}} = V_0 \frac{x_2 - x_1}{x_1 + x_2} \quad (1.7.)$$

Se puede ver que la ecuación 1.6, a diferencia de la ecuación 1.4, es completamente lineal, no depende de la capacitancia inicial C_0 , geoméricamente sólo depende de la distancia inicial entre placas d y se puede utilizar x para determinar la aceleración sin recurrir a alguna aproximación; también se puede notar que V_x tendrá una magnitud y signo proporcional al desplazamiento. En la ecuación 1.7 se puede notar mejor la relación entre el desplazamiento y el signo del mismo. No obstante de las ventajas antes mencionadas, con la expresión 1.6 no se puede establecer la amplitud de la aceleración, pues la señal V_0 es una señal alterna. Para poder hacer esto se requiere la acción del demodulador.

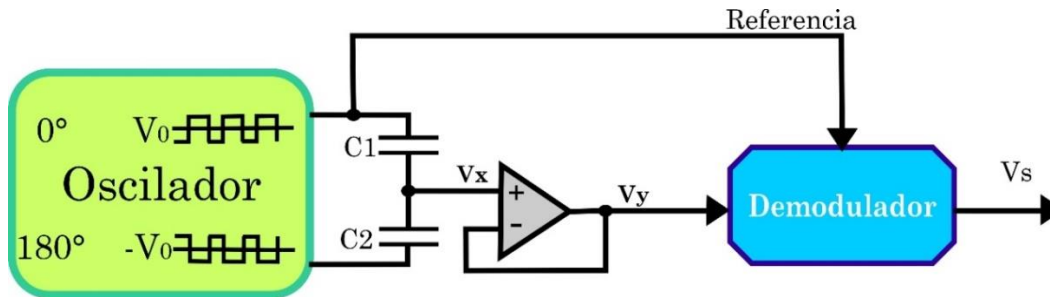


Figura 1.12. Demodulación sensible a la fase en acelerómetros MEMS.

En el demodulador la señal V_x se multiplicará por GV_0 , donde G es la ganancia del multiplicador del demodulador y V_0 es la señal de referencia. El demodulador entregará a la salida una señal (V_s) distinta de cero sólo cuando haya un desplazamiento. La señal V_s será de corriente directa cuya tensión será igual a:

$$V_s = (GV_0)V_0 \frac{x}{d} = GV_0^2 \frac{x}{d} = GV_0^2 \frac{x_1 - x_2}{x_1 + x_2} \quad (1.8.)$$

Dependiendo de si el acelerómetro es digital o analógico, la señal V_s se hará pasar por un convertidor analógico-digital o bien por un amplificador seguidor de tensión, respectivamente.

La relación que existe entre la aceleración y el desplazamiento está dada por la fuerza que aplica la masa sobre los resortes que la mantienen suspendida. El material con el que se construyen los resortes, silicio principalmente, permite que los resortes tengan un comportamiento muy cercano al ideal. Considerando la ley de Hooke ($F_s = k_s x$) y la segunda ley de Newton ($F = ma$), la aceleración con respecto al desplazamiento se expresa como:

$$a = \frac{k_s x}{m} = \frac{k_s d}{m V_0} V_x \quad (1.9.)$$

Donde a es la aceleración, k_s es la constante de elasticidad de los resortes y m es la masa. Es importante hacer notar que se pueden construir acelerómetros MEMS de más de un eje, en ellos la estructura que se ha descrito se repite y se acomoda de acuerdo al eje que se desea medir.

1.6.2. Características generales del acelerómetro a utilizar

El acelerómetro que se eligió para implementarse en el sistema *Cycle DAQ*, es un acelerómetro de 3 ejes: con salida digital de hasta 13 bits, capaz de medir desde 3.9 mg hasta ± 16 g (g, medida de la fuerza que se ejerce); un sistema de manejo de memoria FIFO (*First Input First Output*) para almacenar hasta 32 lecturas; comunicarse mediante los protocolos I²C y SPI; detectar si el sistema está en caída libre; establecer si el sistema está fijo, en movimiento o ha sido pulsado; establecer interrupciones en dos de sus terminales y modular automáticamente su consumo de energía de acuerdo a la velocidad con la que proporciona información. El circuito integrado trabaja con lógica digital de 3.3V, consume como máximo 140 [μA] y cuenta con modos de ahorro de energía en los cuales puede llegar a consumir desde 0.1 [μA].

Como ya se mencionó, el sistema se puede comunicar por los protocolos I²C y SPI, sin embargo sólo se describirá la comunicación mediante el protocolo I²C. Para habilitar la comunicación usando el protocolo I²C la terminal \overline{CS} (\overline{CS} , *Chip Select*) debe estar en un nivel alto y las líneas SDA y SCL se conecten con sus homónimas en el sistema. La dirección que identifica al circuito dentro de I²C puede ser elegida conectado a un nivel alto o a un nivel bajo la terminal ALT ADDRESS del circuito integrado. Si ALT ADDRESS está en alto la dirección del circuito será 0X1D, mientras que en nivel bajo la dirección es 0x53. Con estas conexiones realizadas se puede iniciar la transferencia de información por I²C, dicha transferencia se puede realizar a con una velocidad de la línea de reloj (SCL) de 400 o 100 [kHz]. Con la velocidad de 400 [kHz] la máxima velocidad de transferencia de información es de 800 [Hz], mientras que con la otra velocidad la transferencia de información es de 200 [Hz], para configurar éste y otros parámetros hay que modificar los valores de los registros que posee el circuito integrado. Para acceder a los registros, programarlos o leerlos, se debe seguir la secuencia descrita para el protocolo I²C.

El acelerómetro cuenta con 57 registros de ocho bits cada uno, de los cuales 29 están disponibles para el usuario. La función y dirección de cada registro se muestra en la Tabla 1.6. Cabe mencionar, que los nombres mostrados son sólo con fines ilustrativos.

Dirección	Nombre	Permisos (Lectura L, Escritura E)	Descripción
00h	DEVID	L	En este registro se almacena el valor E5h. Dicho valor fue asignado por el fabricante para identificarlo de otros de sus circuitos integrados. Una manera de corroborar que el circuito está correctamente conectado es tratando de leer el valor contenido en este registro.
01h-1Ch	Reservados		Registros reservados

Tabla 1.6. Registros del acelerómetro (Continúa).

Dirección	Nombre	Permisos (Lectura L, Escritura E)	Descripción
1Dh	THRES_TAP	L/E	Cuando se desea detectar una pulsación sobre el acelerómetro (evento de pulsación) en este registro se debe programar el mínimo valor, leído en alguno de los ejes, para determinar si el acelerómetro ha sido pulsado. Si el registro se programa con un valor cero se podrían generar lecturas erróneas
1Eh	OFSX	L/E	En estos registros se programan los valores, para cada eje, que permiten calibrar las lecturas realizadas por el acelerómetro. La calibración consiste en sumar el valor de estos registros a la lectura del eje correspondiente. El valor de calibración es de 15.6 [mg/LSB] y debe escribirse en complemento a dos.
1Fh	OFSY	L/E	
20h	OFZ	L/E	
21h	DUR	L/E	En este registro se debe escribir el tiempo máximo que una lectura debe sobrepasar el valor del registro 1Dh para que se considere como un evento de pulsación. Para conocer el tiempo que representa el valor escrito en el registro, éste se debe multiplicar por 165 [μ s/LSB].
22h	Latent	L/E	Cuando se requieren medir dos eventos de pulsación consecutivos, este registro debe ser programado para indicar el tiempo de espera que debe transcurrir desde que se detectó el primer evento de pulsación hasta que se inicie la detección del segundo evento. Un valor de 00h en el registro desactiva la función de detección de dos toques. Para conocer el tiempo que representa el valor del registro éste se debe multiplicar por 1.25 [ms/LSB].

Tabla 1.6. Registros del acelerómetro (Continúa).

Dirección	Nombre	Permisos (Lectura L, Escritura E)	Descripción
23h	Window	L/E	Se debe programar este registro cuando se desea detectar dos pulsaciones consecutivas. El valor del registro indica el tiempo durante el cual se puede detectar el segundo toque. La cuenta de tiempo comienza cuando la cuenta del registro 22h ha finalizado. Un valor de 00h en el registro desactiva la función de detección de dos toques. El tiempo que representan los valores de este registro se puede calcular multiplicando el valor del registro por 1.25 [ms/LSB].
24h	THRESH_ACT	L/E	El valor programado en este registro será considerado como el mínimo valor de lectura en alguno de los ejes, que se considerará para establecer que el acelerómetro se ha movido (evento de actividad) y comenzar a realizar mediciones de manera normal. Los eventos de actividad sólo se consideran cuando el circuito integrado trabaja en los modos de bajo consumo.
25h	THRESH_INACT	L/E	En los modos de bajo consumo, este registro se programa con el máximo valor de lectura, en alguno de los ejes, que determina si el circuito está inactivo. Se considera que el sistema está inactivo cuando sus lecturas están debajo del valor de este registro por el tiempo determinado en el registro 26h. Si el circuito está inactivo se activa el modo de bajo consumo.
26h	TIME_INACT	L/E	En este registro se programa el tiempo necesario que las lecturas, en alguno de los ejes, deben estar debajo de los valores del registro 25h para que el acelerómetro se considere como inactivo. El tiempo que representa el valor de este registro se puede conocer multiplicándolo por 1[s/LSB].
27h	ACT_INACT_CTL	L/E	En este registro se pueden elegir los ejes que se involucraran en la detección de eventos de actividad e inactividad.

Tabla 1.6. Registros del acelerómetro (Continúa).

Dirección	Nombre	Permisos (Lectura L, Escritura E)	Descripción
28h	THRESH_FF	L/E	El registro se programa con el valor mínimo de lectura, en alguno de los ejes, para considerar que el circuito está en caída libre.
29h	TIME_FF	L/E	En este registro se escribe la mínima cantidad de tiempo que las lecturas, en alguno de los ejes, debe ser mayor al valor del registro 28h para considerar la caída libre.
2Ah	TAP_AXES	L/E	En este registro se eligen los ejes que se involucraran en la detección de un evento de pulsación.
2Bh	ACT_TAP_STATUS	L	El byte que se almacena en este registro permite conocer cuales ejes, activados en los registros 27h y 2Ah, se involucraron en la detección de un evento de pulsación y cuales en la detección de un evento de actividad.
2Ch	BW_RATE	L/E	En este registro el usuario puede definir la tasa de velocidad a la que el circuito entregará la información y activar uno de los modos de ahorro energía. La velocidad a la que se entrega la información está relacionada con el consumo de energía, es decir, que a velocidades más altas el consumo de energía será mayor. Sin embargo, al habilitar el modo de bajo consumo de energía, disponible en este registro, es posible trabajar con algunas de las velocidades altas consumiendo la mitad de energía que la consumida en el modo normal. Es recomendable que la velocidad a la que el acelerómetro entrega la información sea menor que la velocidad del protocolo de comunicación con el que se esté trabajando.

Tabla 1.6. Registros del acelerómetro (Continúa).

Dirección	Nombre	Permisos (Lectura L, Escritura E)	Descripción
2Dh	POWER_CTL	L/E	En este registro se pueden activar los siguientes modos de bajo consumo energético: “sueño” automático, el sistema cambia entre el modo de “sueño” o el modo normal de acuerdo a las condiciones de eventos de actividad e inactividad, definidas en los registros 24h y 25h; “sueño”, el sistema desactiva la función de la memoria FIFO y disminuye la frecuencia a la que entrega información (el registro cuenta con dos bits para elegir entre 1, 2, 4 y 8 [Hz]) y espera, el sistema desactiva completamente sus funciones de medición.
2Eh	INT_ENABLE	L/E	En este registro el usuario puede elegir que eventos (caída libre, pulsación, doble pulsación, actividad, inactividad o eventos de la memoria FIFO) pueden generar una interrupción.
2Fh	INT_MAP	L/E	En este registro el usuario establece la terminal, de las dos dedicadas a interrupciones, en las que se presentarán las interrupciones de cada evento.
30h	INT_SOURCE	L	El byte que se almacena en este registro informa que eventos han producido una interrupción.

Tabla 1.6. Registros del acelerómetro (Continúa).

Dirección	Nombre	Permisos (Lectura L, Escritura E)	Descripción
31h	DATA_FORMAT	L/E	En este registro el usuario elige si el sistema se calibrará automáticamente, si se trabajará con el protocolo SPI de tres o cuatro hilo, si la información de las lecturas estará justificada a la izquierda o derecha, el rango de medición con el que se trabajará (± 2 , ± 4 , ± 8 o ± 16 g) y si se medirá con alta resolución. Al activar la alta resolución el sistema ocupará los bits necesarios para lograr que las mediciones en cualquier rango tengan una sensibilidad de 3.9 [mg/LSB.] Si no se activa la alta resolución las lecturas en cualquier rango se hacen con 10 bits. Cabe mencionar que la justificación a la derecha consiste en agregar ceros al inicio de la lectura y la justificación a la izquierda en agregar ceros al final de la lectura, esto se hace para ocupar los bits, de los bytes que guardan las lecturas, que no se usan en las lecturas.
32h	DATA_X0	L	En estos registros el acelerómetro almacena la información de las lecturas obtenidas en cada uno de sus ejes. La información guardada está en complemento a dos y los registro con 0 en su nombre (p. ej. DATA_X0) guardan el byte menos significativo y los otros (p. ej. DATA_X1) el byte más significativo, dependiendo de la configuración del registro 31h la información se justificara a la izquierda o a la derecha Se recomienda leer los dos bytes de cada eje en una sola ocasión.
33h	DATA_X1		
34h	DATA_Y0		
35h	DATA_Y1		
36h	DATA_Z0		
37h	DATA_Z1		

Tabla 1.6. Registros del acelerómetro (Continúa).

Dirección	Nombre	Permisos (Lectura L, Escritura E)	Descripción
38h	FIFO_CTL	L/E	<p>En este registro el usuario puede activar la memoria FIFO y elegir el modo de funcionamiento de ésta. La memoria FIFO puede trabajar bajo los siguientes esquemas: a) modo FIFO, en este modo el acelerómetro guardará 32 lecturas de todos los ejes y cuando se han guardado las 32 lecturas el acelerómetro se continúan realizando mediciones, más las nuevas mediciones son descartadas mientras no exista espacio en la memoria; b) modo secuencial, el acelerómetro guardará 32 lecturas de todos los ejes en la memoria FIFO, cuando la memoria ha almacenado las 32 lecturas las más antiguas se reemplazan con lecturas nuevas; c) modo de disparo, en este modo el acelerómetro comienza a trabajar como en el modo secuencial, sin embargo cuando una interrupción se activa en una de las terminales de interrupción (se elige cual), la memoria retiene las últimas n lecturas realizadas (n es menor o igual a 32) y comienza a trabajar en el modo FIFO. Además de lo anterior, en este registro se escribe el número n y se define cual de las dos terminales de interrupciones se considerará para el modo de disparo. Por otro lado, los valores de la memoria FIFO se leen en los registros 32h a 37h, cada vez que se lee un valor de la memoria FIFO se libera una localidad para recibir una nueva lectura. Cabe mencionar que si se activa la memoria FIFO en los modos FIFO o secuencial, se elige en el registro 2Eh activar las interrupciones de la memoria y se escribe un número n (en el presente registro), cada vez que se realicen n lecturas se activará una interrupción. En el caso de la memoria FIFO la interrupción sólo se presentara mientras la memoria no esté llena.</p>
39h	FIFO_STATUS	L	<p>En este registro el acelerómetro permite conocer la cantidad de lecturas que se han guardado en la memoria.</p>

1.6.3. Confiabilidad de los datos

La información proporcionada por el fabricante del acelerómetro asegura que la mínima resolución del circuito integrado es de 3.9 [mg/LSB], las lecturas tendrán una desviación del valor ideal del $\pm 1\%$ y las lecturas se verán afectadas por la temperatura en $\pm 0.01\%$ por cada $[\text{°C}]$. Así mismo, el fabricante asegura que el valor que entregan los ejes X y Y cuando el circuito está estático diferirá de cero en un rango de ± 150 [mg], la desviación del valor ideal es de $\pm 35\%$ y la salida variaría por la temperatura $\pm 0.4\%$ por cada $[\text{°C}]$; mientras que el eje Z , bajo esas mismas condiciones, entregará una lectura diferente de cero dentro de un rango de ± 250 [mg], con una desviación del valor ideal del $\pm 40\%$ y la temperatura afectará la lectura $\pm 1.2\%$ por cada $[\text{°C}]$. La potencia de ruido del sistema es de 2.29 [mg_{rms}]. Cabe mencionar que la información proporcionada por el acelerómetro se utilizará para calcular la inclinación del vehículo, dado que las inclinaciones que se medirán serán menores a los 20° y los cambios de pendiente no son bruscos⁸, el rango de $\pm 2g$ ⁹ es suficiente.

1.7. Pantallas de cristal líquido

Los visualizadores de cristal líquido (LCD, *Liquid Crystal Display*) son un tipo de pantallas que permiten desplegar información (texto, imágenes o videos) mediante el control de fenómenos electro-ópticos en pequeñas celdas (píxeles) llenas de un compuesto químico (cristal líquido).

Los cristales líquidos (LC, *liquid crystal*) son un tipo de sustancias (normalmente orgánicas) que se encuentran en un estado de agregación de la materia entre los sólidos y los líquidos, es decir, sus moléculas se encuentran en posiciones aleatorias y se pueden mover como en un líquido y a la vez tienen una fuerte interacción molecular, como en un sólido, que les permite tener una organización regular. Según la forma en la que se organicen las moléculas, los LC son clasificados en tres categorías. Una de esas categorías es la de los LC nemáticos, los cuales son los más ampliamente usados para la fabricación de pantallas. La representación general de los LC nemáticos se representa en la figura 1.13, la principal característica de éstos es que poseen moléculas que no tienen una posición espacial definida pero que están orientadas en una misma dirección (las moléculas son paralelas entre sí).

Una de las propiedades más importantes de los LC, que se deriva de la organización de sus moléculas, es la birrefringencia. Esta propiedad genera, entre otras cosas, que un haz de luz polarizado linealmente que atraviese el material birrefringente cambie su polarización por una circular o elíptica. Si este último haz se hace pasar a través de un polarizador lineal se logrará obtener un haz con la intensidad disminuida con respecto al haz de entrada.

⁸ Secretaría de Desarrollo Social. *Programa de Asistencia Técnica en Transporte Urbano para las Ciudades Medias Mexicanas. Tomo IV*. Recuperado de: <http://imipmexicali.org.mx/normatividad/4.-%20NORMAS%20TECNICAS/10%20MANUAL%20DISE%3%91O%20GEO%20VIAL.pdf>

⁹ Gad-el-Hak, M. (2006) *The MEMS Handbook: MEMS Applications*. Estados Unidos: Taylor & Francis.

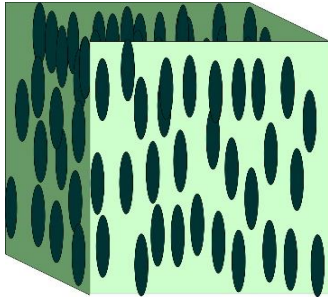


Figura 1.13. Estructura general de los cristales líquidos nemáticos.

Actualmente los LCD de color se conforman por varias celdas como la mostrada en la figura 1.14, cada celda constituye un pixel dentro de la pantalla.



Figura 1.14. Estructura de una celda de un LCD.

Como se puede observar en la figura 1.14, las celdas de cristal líquido son iluminadas con luz blanca sin polarizar, cuando un haz de luz atraviesa la celda el primer polarizador polariza linealmente al haz, dado que los cristales y los electrodos son transparentes el haz no sufrirá cambio al atravesarlos, en el cristal líquido se producirá el fenómeno descrito en párrafos anteriores. El haz a la salida del LC pasará a través de un filtro de color y finalmente pasará por un segundo polarizador, lo cual producirá un haz de luz de color con la intensidad luminosa controlada. Es importante hacer notar que el primer y segundo polarizador tienen ejes de transmisión perpendiculares entre sí.

Por otro lado, la superficie de los electrodos, que hace contacto con el cristal líquido, cuenta con surcos donde se acomodan algunas moléculas del cristal líquido. Debido a la fuerte interacción molecular, las moléculas del LC se orientan en la dirección de las moléculas alojadas en los surcos. Sin embargo, los surcos de los electrodos superior e inferior son perpendiculares entre sí,

lo cual produce que las moléculas del cristal líquido formen una hélice. Así mismo, si se aplica una diferencia de potencial en los electrodos, el campo eléctrico entre los electrodos orientará a las moléculas en su dirección. Dependiendo de la intensidad del campo eléctrico las moléculas se orientarán en mayor o menor grado, esto se muestra en la figura 1.15. En dicha figura se puede observar que, cuando un haz de luz se hace pasar a través de una celda cuyos electrodos no han sido electrificados, éste saldrá de la celda tal y como salió del primer polarizador, pues el LC rota 90° el eje de vibración del haz y hace coincidir los ejes de vibración del haz y el de transmisión del segundo polarizador. Mas, al aplicar la tensión a los electrodos las moléculas del LC modificarán su orientación y con ello la polarización lineal del haz por una polarización circular o elíptica, dependiendo del grado de polarización el haz saldrá de la celda con la intensidad disminuida. Cuando las moléculas del LC son paralelas al eje de transmisión del haz, el LC no modificará al haz de luz y éste no saldrá de la celda porque su eje de vibración y el eje de transmisión del segundo polarizador son perpendiculares.

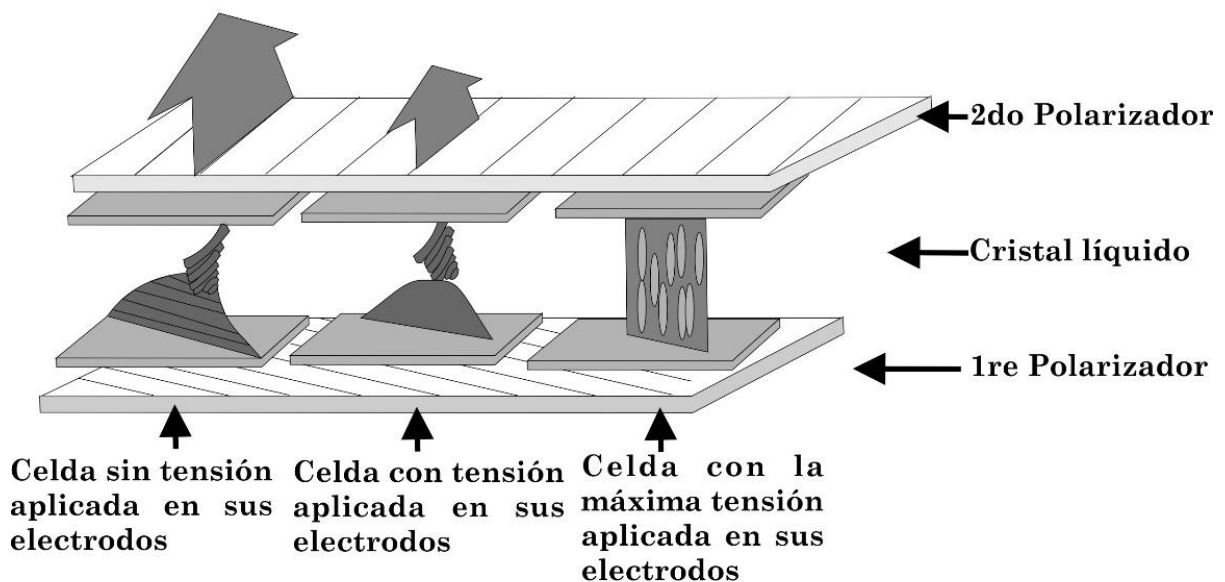


Figura 1.15. Respuesta de una celda de LC a la tensión.

Como se mencionó con anterioridad, en las pantallas de LCD a color, el haz que cruza la celda de LC pasa por un filtro color, el cual está construido por filtros de color azul, rojo y verde. Cada una de las celdas de LC está cubierta con los tres filtros de color (figura 1.14), los filtros de color de cada celda están controlados por una señal PWM (*Pulse Width Modulation*), lo cual permite controlar la intensidad de luz que cruza por el filtro de cada color. La modulación sobre los filtros produce una paleta de colores. Así, con el control de la intensidad luminosa de las celdas de LC (que producen la iluminación de una imagen) y de los filtros de color (que forman los colores de la imagen) logran el funcionamiento de un LCD.

1.7.1. Pantallas táctiles de membrana resistiva

Las pantallas táctiles son un tipo de pantallas que cuentan con un dispositivo para detectar el contacto de un objeto sobre su superficie en una zona determinada. Existen diferentes formas para detectar el contacto sobre la pantalla, una de ellas es basa en la medición de efectos resistivos. Estos últimos son dispositivos que se superponen en la pantalla que trabajarán y consisten en dos placas rectangulares (una flexible, sobre la que se tocará, y la otra rígida) de vidrio o acrílico separadas entre sí. Cada placa tiene una cara cubierta con una membrana transparente de un material resistivo, dichas caras se enfrentan. Existen dos arquitecturas bajo las cuales se construyen las membranas resistivas: arquitectura de cinco hilos y arquitectura de cuatro hilos. La arquitectura de cuatro hilos es la más simple y la más usada, en esta arquitectura, cada una de las membranas cuenta con un par de electrodos en sus extremos y entre ellos existe una resistencia fija. Las membranas son colocadas de tal manera que los electrodos sean perpendiculares entre sí, como se muestra en la figura 1.16. Por otro lado, en la arquitectura de cinco hilos, la placa rígida cuenta con cuatro electrodos, dos colocados a lo largo y dos colocados a lo ancho de la placa (por cada par de electrodos hay una resistencia fija), mientras que la placa flexible contará con un solo electrodo (la membrana es completamente conductiva).

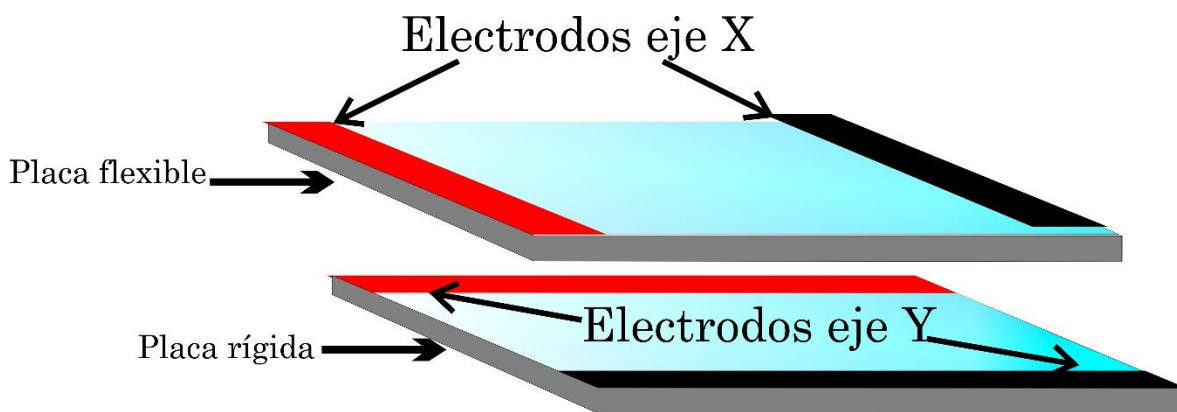


Figura 1.16. Sensor táctil resistivo de cuatro hilos.

Cuando un objeto hace contacto sobre el sensor la placa flexible se deforma y las dos membranas que estaban separadas hacen contacto. En la arquitectura de cuatro hilos la posición del contacto se determina suministrando una tensión, normalmente 5 [V], al electrodo X+ (color rojo en figura 1.16) y colocando el electrodo X- (color negro en figura 1.16) a tierra, esta conexiones forma un divisor de tensión. Uno de los electrodos del eje Y se usa para medir el valor de tensión producido, una vez registrado el valor de tensión para el eje X, se mide la diferencia de potencial producida por el divisor de tensión en el eje Y, es decir se repite el proceso para el eje X en el eje Y. En el caso de la arquitectura de cinco hilos se realiza el proceso de aplicar tensión a los electrodos y medir la diferencia de potencial producida por un divisor de tensión, la diferencia con la arquitectura de cuatro hilos es que la tensión será medida por el electrodo de la placa flexible.

Cabe mencionar que una práctica muy común de los fabricantes de pantallas táctiles es utilizar dos circuitos controladores. Uno de ellos está dedicado a controlar el despliegue de información en la pantalla y el otro a realizar las actividades descritas para la detección de contacto, además de convertir las lecturas de pantalla táctil (valores analógicos) a valores digitales. En ambos casos los controladores tienen líneas para comunicarse con otros circuitos.

1.7.2. Descripción general de la pantalla táctil a usar

La pantalla táctil que se usará cuenta con las siguientes características: pantalla de 2.8 pulgadas, una resolución de 240 por 320 píxeles, la capacidad de generar hasta 262,144 colores (cada filtro de color puede producir 64 tonalidades diferentes de color), tiene un sistema de retroiluminación de tecnología LED (*Light Emitting Diode*) que es la fuente que producirá el haz sin polarizar que entra a la celda de LC), un sensor táctil resistivo, comunicación mediante el protocolo SPI y un consumo de 100 [mA] a 5[V]. En la figura 1.17 se muestra el diagrama de bloques simplificado de la pantalla táctil elegida.

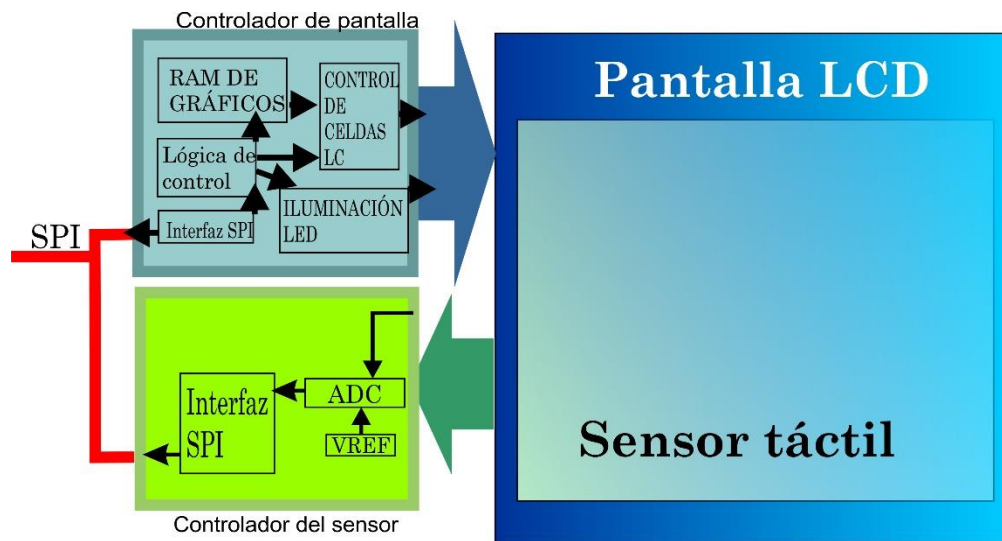


Figura 1.17. Diagrama de bloques de la pantalla táctil

Es importante mencionar que otra de las mejoras implementadas a la primera versión del *Cycle-DAQ* fue cambiar de sistema de visualización. En la Tabla 1.7 se hace una comparación entre los dos sistemas de visualización.

En la Tabla 1.7 podemos observar que al sustituir la pantalla anterior por la pantalla táctil elegida se logra reducir el consumo de energía, se pueden desplegar imágenes a color, se reducen el número de canales necesarios para el control de la pantalla, se obtiene un mayor ángulo de visión (el ángulo máximo al cual una pantalla logra desempeño visual aceptable) mayor, se obtiene una mejor resolución y la membrana resistiva evita la inclusión de otros componentes para interactuar con el sistema.

Parámetros	Pantalla anterior	Pantalla actual
Alimentación	360[mA] a 4.5-5.5 VDC	100 [mA] a 4.5-5.5 VDC
Tipo de pantalla	LCD blanco y negro	LCD color
Resolución (en pixeles)	128 por 64	240 por 320
Sensor táctil	Sin sensor	Membrana resistiva
Tipo de comunicación	Serial paralela de 8 vías	Protocolo SPI de 4 vías
Angulo de visión promedio	30°	66°
Despliega imágenes	Sí, sólo mapas de bits monocromáticos	Sí, mapas de bits de hasta 24 bits de color.

Tabla 1.7. Comparación entre los sistemas de visualización.

1.8. Tarjetas de memoria SD

Una tarjeta de memoria es un dispositivo de almacenamiento de información, construida con electrónica de estado sólido, capaz de retener los datos que guarda aún en la ausencia de una fuente de alimentación.

En el año 2000, las compañías Panasonic, SanDisk y Toshiba lanzaron al mercado las tarjetas de memoria SD (*Secure Data*), la cuales median 24 x 32 x 2.1 [mm] y pesaban 2 [g]. El éxito comercial de las tarjetas y el avance tecnológico, permitió que los fabricantes desarrollaran tarjetas de memoria más ligeras y de menores dimensiones. Así, en el año 2003 se comercializaron las primeras tarjetas de memoria miniSD y en el año 2008 las primeras tarjetas microSD. En la figura 1.18 se muestran los tres tipos de tarjetas SD y sus dimensiones correspondientes.

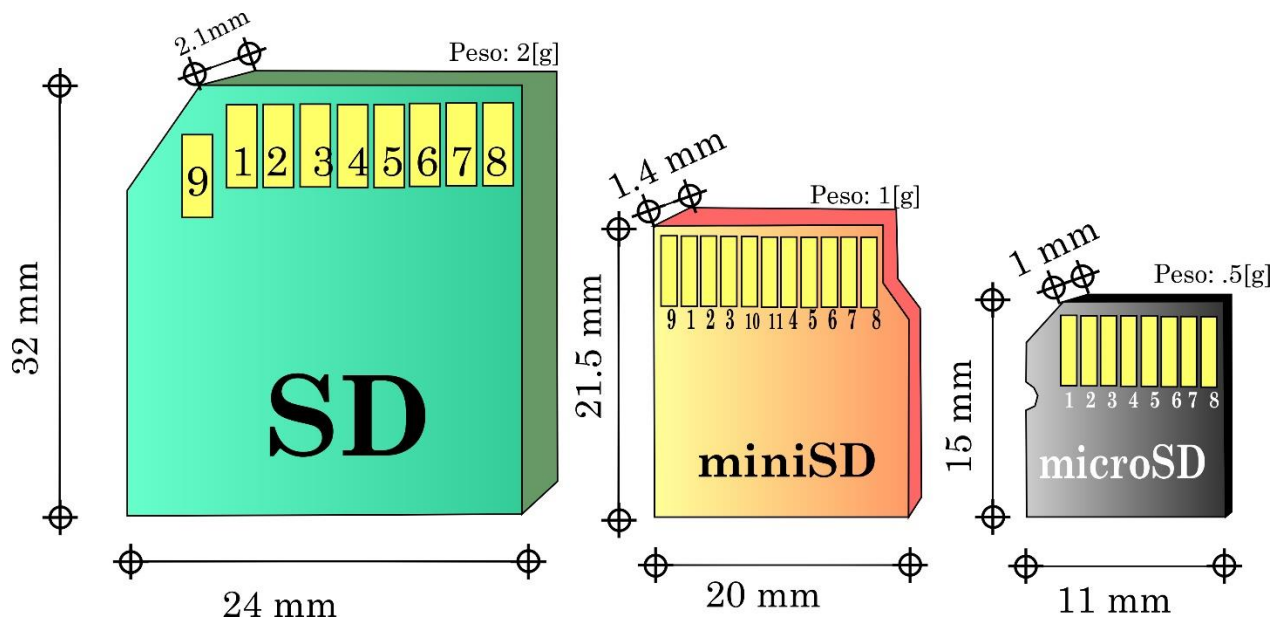


Figura 1.18. Tarjetas de memoria SD.

Así como el aspecto físico de las memorias fue modificado con el paso del tiempo, también se cambiaron las características de almacenamiento, la velocidad de transferencia y la tensión de trabajo. Como resultado de las modificaciones surgieron diferentes clasificaciones de las tarjetas de memoria SD. A continuación se presenta la clasificación de acuerdo a la capacidad de almacenamiento de datos:

- **SDSC® (*SD Standar Capacity*):** Son todas aquellas memorias SD que son fabricadas para almacenar hasta 2 [GB].
- **SDHC® (*SD High Capacity*):** Son las tarjetas de memoria fabricadas cuya capacidad máxima de almacenamiento de información está en un rango de 2 a 32 [GB].
- **SDXC® (*SD Extended Capacity*):** En esta categoría se clasifican a las tarjetas de memoria SD cuya capacidad máxima de almacenamiento está en el rango que va de los 32 [GB] a los 2[TB].

Además de la clasificación anterior, las tarjetas de memoria SD son catalogadas de acuerdo a la mínima velocidad constante a la que un dispositivo de grabación puede almacenar una secuencia de imágenes o videos:

- **Clase 0:** Pertenecen a esta clase las tarjetas de memorias en las que no se especifica la mínima velocidad constante de transferencia de información y que trabajan con una lógica de 3.3 [V].
- **Clase 2:** Se consideran aquellas tarjetas de memorias que pueden recibir y guardar información a una velocidad de al menos 2 [MB/s] y que funcionan con lógica de 3.3 [V]. Las tarjetas SDSC® y las SDHC® entran en esta categoría.
- **Clase 4:** Son el tipo de tarjetas de memoria que pueden guardar y recibir información con una velocidad constante de al menos 4 [MB/s] y operan con lógica de 3.3 [V]. Sólo las tarjetas SDHC® corresponden a esta categoría.
- **Clase 6:** Estas tarjetas trabajan con una lógica de 3.3 [V], pueden guardar y recibir información con una velocidad mínima constante de al menos 6[MB/s]. Únicamente las tarjetas SDHC® entran en esta categoría.
- **Clase 10:** Estas tarjetas de memoria operan con una lógica de 3.3 [V], pueden recibir y guardar información de manera constante con una velocidad mínima de 10[MB/s]. Las tarjetas SDHC® y SDXC® corresponden a esta categoría.
- **UHS (*Ultra High Speed*)-I:** La mínima velocidad de transferencia constante de información en estas tarjetas de memoria es de 10 [MB/s]. Esta clase se diferencia de la clase 10 por las velocidades a la que trabajan los canales por los que se envía la

información y porque trabaja con una lógica de 1.8 [V]. Las tarjetas SDHC® y las SDXC® pueden considerarse en esta clasificación.

- **UHS-III:** Son las tarjetas de memoria con una mínima velocidad de transferencia constante de información de 30 [MB/s]. Sólo las tarjetas SDXC® están en esta clasificación. Cabe mencionar que este tipo de tarjetas de memoria tienen una mayor cantidad de terminales que las mostradas en la figura 1.17 y trabajan con una lógica de 0.4 [V].

Es importante aclarar que no existen las clases 1, 3, 5, 7, 8, 9 y UHS-II para clasificar tarjetas de memoria SD. Por otro lado, las tarjetas de memorias SD están especificadas para una velocidad mínima debido a que las velocidades máximas dependen de cada fabricante, siendo las memorias UHS son las que alcanzan velocidades de transferencias superiores a las demás clases. La clase es relevante para grabar videos o secuencias de imágenes, para otras aplicaciones la clase no suele ser importante¹⁰.

En la figura 1.17 se puede ver que las terminales de cada tarjeta (representadas por rectángulos de color amarillo) cuentan con una numeración. Para nuestra aplicación sólo se explicará la distribución y utilidad de cada terminal de una tarjeta microSD del tipo SDHC, lo cual se muestra en la Tabla 1.8. En dicha tabla los nombres mostrados son con fines ilustrativos y corresponden a los asignados por los creadores de las tarjetas de memoria SD.

No. de terminal	Modo SD		Modo SPI	
	Nombre	Descripción	Nombre	Descripción
1	DAT2	Línea de datos 2	No usada	
2	CD/DAT3	Detección de tarjeta/ Línea de datos 3	CS	Selección de la tarjeta
3	CMD	Envío y respuesta de comandos	DI	Información de entrada (MOSI)
4	VDD	Tensión de alimentación	VDD	Tensión de alimentación
5	CLK	Señal de reloj	SCLK	Señal de reloj
6	VSS	Tierra	VSS	Tierra
7	DAT0	Línea de datos 0	DO	Información de salida (MISO)
8	DAT1	Línea de datos 1	No usada	

Tabla 1.8 Terminales de tarjeta de memoria microSD.

En la Tabla 1.8 se puede observar que la utilidad de cada terminal (salvo las de alimentación) cambia de acuerdo al modo en que se trabaje, cada uno de los modos corresponde a los protocolos de comunicación SD y SPI.

¹⁰ SD Association. (2014). *SD Memory Card Choices*. Fecha de consulta: 25 de mayo 2015. URL: <https://www.sdcard.org/consumers/choices/index.html>

El protocolo de comunicación SD consiste en un arreglo maestro/esclavo donde el flujo de datos se realiza de acuerdo a una señal de reloj (con velocidades desde 25 a 208 [MHz]) proporcionada por el dispositivo maestro. El intercambio de información se controla mediante comandos enviados por el maestro al esclavo y respuestas enviadas del esclavo al maestro. Cada comando cuenta con 48 bits y puede carecer de respuesta, producir una respuesta de 48 bits o una respuesta de 136 bits. El intercambio de información se puede realizar a través de una o varias líneas de comunicación (DAT0, 1, 2 Y 3), éste se realiza típicamente en bloques de 512 bytes. Los comandos, las respuestas y los bloques de información incluyen unos bits para poder realizar una verificación de redundancia cicla (un algoritmo para detectar errores). Este protocolo permite que las líneas de información puedan ser usadas para indicarle al maestro tiempos de espera (útil para indicar que la tarjeta se encuentra realizando otros proceso), para indicar la recepción exitosa de bloques de información y para declarar interrupciones. Es posible conectar varias memorias utilizando las mismas líneas. Al ser el protocolo nativo de las tarjetas de memoria SD, este es el único protocolo que soporta la comunicación a todas las velocidades de transmisión disponibles.

El protocolo de comunicación SPI se diferencia del modo SD por utilizar menos líneas de comunicación, con las líneas MOSI y MISO se deben transferir los comandos, las respuestas y la información. Así mismo, el algoritmo de detección de errores no es tan confiable como en el caso del modo SD. Debido a que el protocolo SPI no trabaja con señales menores a 3.3 [V], funciona con frecuencias de reloj de hasta 100 [MHz], no se pueden trabajar con las clases 10, UHS-I y UHS-III. Para conectar varias tarjetas de memoria se requieren de líneas individuales. La gran ventaja de utilizar el protocolo SPI es que no se requieren de licencias para utilizar las tarjetas SD.

Independientemente del protocolo de comunicación elegido, las memorias deben seguir el mismo procedimiento de inicialización antes de poder transferir información. Una vez que se alimenta la tarjeta de memoria, ésta entra en el modo de comunicación SD y espera por un comando. Si se trabajará con el modo SD *grosso modo* sucede lo siguiente. El maestro envía un comando para reconocer las características eléctricas de la memoria, una vez comprobado esto el maestro enviará un comando para conocer la capacidad de la memoria y otros parámetros sobre su funcionamiento, después de recibir la respuesta de la memoria, el maestro iniciará un proceso para asignar una dirección virtual a la tarjeta de memoria (la identificará de otras tarjetas de memoria conectadas). Al finalizar la asignación de direcciones, la memoria entra en un estado de espera de comandos de lectura o escritura. Cuando el sistema se comunicará mediante el protocolo de comunicación SPI a grandes rasgos se hace lo siguiente, en el estado de espera, que se da inmediatamente después de la conexión a alimentación, se debe escribir el comando de reinicio, definido por la normativa de SD, con la línea CS en un nivel lógico bajo, tras lo cual se comienza el proceso de reconocimiento de la memoria y suceden los procedimientos descritos para el modo SD.

Cabe mencionar que las tarjetas de memoria no son los únicos sistemas de almacenamiento disponibles en el mercado. Sin embargo, la capacidad de trabajar mediante el protocolo de comunicaciones de SPI facilita su implementación en aplicaciones desarrolladas con microcontroladores

1.8.1. Tabla de asignación de archivos

La tabla de asignación de archivos (FAT, *File Allocation System*) es un sistema de archivos introducido en el mercado por la empresa Microsoft® en 1980. Los sistemas de archivos son los métodos y las estructuras de datos que utilizan los sistemas operativos para organizar la información que se guarda en un dispositivo de almacenamiento. Existen diferentes métodos para organizar la información, no obstante todos ellos cuentan con una unidad mínima de almacenamiento llamada *cluster*. Cada *cluster* está conformado por sectores, éstos últimos son un bloque de bytes con un tamaño definido por el fabricante del dispositivo (normalmente 512 bytes). Cabe mencionar que el tamaño de los *clusters* es modificable, esto se debe a que los *clusters* son divisiones virtuales. Más, el tamaño de los sectores no es configurable puesto que son estructuras físicas.

Un dispositivo de almacenamiento con el sistema de archivos FAT se divide en cuatro sectores:

- **Sector de Arranque:** En este sector se describe la capacidad, las particiones y el tamaño de los *clusters* de la unidad de almacenamiento. Así mismo, indica la localización de los otros cuatro sectores. Si la unidad almacena un sistema operativo contiene la información para ingresar a él.
- **Tabla de asignación de archivos:** Es una estructura de datos (conjunto de variables), la cual le da el nombre al sistema de archivos, que está conformada, entre otras cosas, por una serie de direcciones (entradas) asociadas a los *clusters* del sector de información. Los datos almacenados en la tabla permiten conocer el estado de un *cluster* (vacío o dañado), el *cluster* siguiente en una cadena de *clusters* o el fin de una cadena de *clusters*. Debido a la importancia de esta sección, el sistema crea dos tablas de asignación de archivos llamadas FAT1 y FAT2, las cuales son copias la una de la otra.
- **Directorio raíz o carpeta raíz:** Este sector consiste en una tabla (tabla de direcciones) que contiene la información (nombre, tamaño, dirección del primer *cluster*, entre otros) de los archivos o directorios dentro de la carpeta raíz. El sistema FAT es una estructura jerárquica y el directorio raíz es la primera estructura en la que se pueden almacenar otros archivos y directorios. Es importante mencionar que cada directorio es un archivo especial, que se encuentra dentro del dispositivo de almacenamiento y que permite rastrear a otros archivos. Cada directorio consiste en una estructura de datos que forman una tabla de direcciones, ésta contiene información del archivo como nombre, fecha de creación o modificación, tamaño, primer *cluster*, entre otros.
- **Área de información:** Este sector ocupa la mayor parte de la unidad de almacenamiento, en él se encuentran los *clusters* que guardan la información de los archivos y de los directorios.

Para guardar archivos en un dispositivo con formato FAT se llevan a cabo dos pasos principales. El primero paso consiste en asignar un directorio al archivo, es decir que hay que generar una entrada (elemento de una tabla) en la tabla de direcciones del directorio donde se alojará la

información. Cabe aclarar que cualquier archivo será almacenado en un número entero de *clusters* (dependiendo del tamaño del archivo), para conocer en cuales *clusters* está almacenado el archivo, la dirección de los *clusters* se almacena en la tabla de direcciones (el primer clúster) y en la tabla de asignación de archivos (las direcciones restantes hasta el final de la secuencia de *clusters*). Entonces, el segundo paso consiste en escribir las entradas en la tabla de asignación de archivos para saber en cuales *clusters* está almacenado el archivo.

En la figura 1.19 se muestra un dispositivo de almacenamiento con formato FAT. En la imagen se pueden ver los sectores (sector de arranque, tabla de asignación de archivos, directorio raíz y área de información) en los que se divide el dispositivo y la distribución de los mismos. Por otro lado, en la figura se representa una porción de la tabla de asignación de archivos FAT1 y una sección de los *clusters* en el área de información, en ambos casos se ejemplifica como se guarda la información de cuatro archivos (representados por bloques del mismo color y cuyas direcciones sólo son con fines ilustrativos). En el caso de la FAT1 se puede observar que en el primer renglón, de arriba abajo y de izquierda a derecha, los bloques son de color blanco y sólo presentan direcciones con valores nulos, estos representan entradas disponibles. En el segundo renglón, siguiendo el mismo orden, se pueden ver un conjunto de bloques color azul pastel atravesados por flecha negra, este primer conjunto de bloques representa un archivo almacenado en la memoria y la flecha representa el orden en que está almacenado el archivo. Este primer archivo está almacenado en *clusters* contiguos y la cadena de *clusters* termina en la entrada con la leyenda EOC (*End of clusters*). Se recuerda que un archivo es dividido en un número entero de *clusters* para ser almacenado, que en las tablas de asignación de archivos sólo se guardan direcciones, que cada entrada en la tabla está asociada directamente a un *cluster* y que cada entrada de la tabla contiene la dirección del *cluster* siguiente en el que se guardó un archivo. El segundo archivo, bloques de color rosa, está almacenado en *clusters* contiguos tal y como en el primer caso. El tercer y cuarto archivo (bloques morados y rojos respectivamente) no están almacenados en *clusters* contiguos. Por ejemplo, el tercer archivo tiene su primer entrada con la dirección 113 y la segunda entrada con la dirección 134, a diferencia de los archivos anteriores la tercer entrada no se almacena en una entrada contigua. La tercer entrada se almacena en aquella relacionada con el cluster 134 y en ella se puede observar la dirección del siguiente *cluster* (135). Para entender claramente esto no hay que perder de vista que las entradas de la FAT están directamente relacionadas con los *clusters*, esto se puede observar claramente si se compara la posición de las entradas en la FAT y la posición de los *clusters* del área de información.

En la figura 1.19 también se muestra un directorio. Se puede ver que un directorio es almacenado como un archivo, es decir dentro de un clúster. La información de este archivo consiste en entradas de 32 bytes, las cuales en conjunto forman la tabla de direcciones. Se recuerda que por cada directorio se crea un archivo, cada archivo contendrá una tabla de direcciones. En el caso del directorio raíz todos los *clusters* que forman al sector son para almacenar una sola tabla (un solo archivo), la tabla de direcciones del directorio raíz.

El sistema de archivos FAT, desde su primera aparición hasta la fecha ha sido mejorado con el fin de poder manejar una mayor cantidad de información y optimizar la organización de la información. Los sistemas FAT que se han desarrollado son FAT12, FAT16 y FAT32, donde el número indica la cantidad de bits que se utilizan para generar las direcciones de los *clusters*.

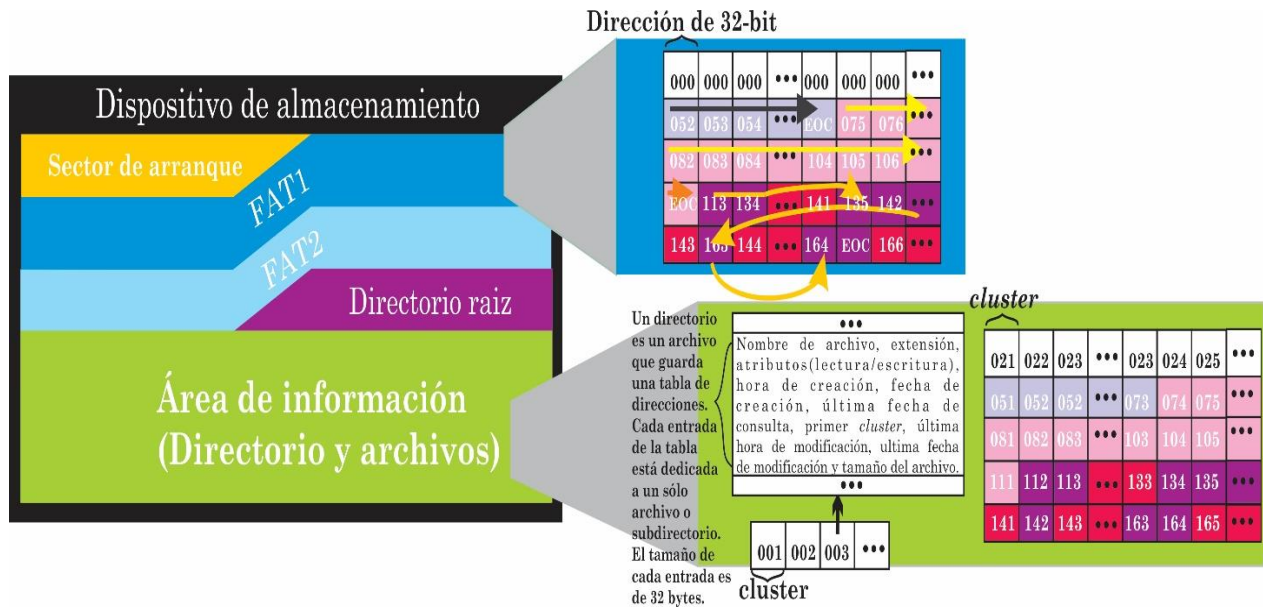


Figura 1.19. Estructura del formato FAT.

1.8.2. Sistemas de archivos FAT12, FAT16 y FAT32

El sistema de archivos FAT12 fue el primer sistema de archivos utilizado con una computadora personal. Fue diseñado para ser usado con dispositivos de almacenamiento con capacidades de hasta 16 [MB] y manejar hasta 4,096 (2^{12}) clusters. Este sistema fue usado con la IBM PC original y usado como formato para discos flexibles. Actualmente el sistema ha quedado en desuso.

El sistema de archivos FAT16 fue implementado por primera vez con los sistemas operativos Windows 95 y Windows95a. El sistema fue diseñado para manejar unidades de almacenamiento con capacidades mayores a 16[MB], trabajar con hasta 65,536 (2^{16}) clusters, limitar el tamaño de la tabla de direcciones y fijar el tamaño de las tablas de asignación de archivos FAT1 y FAT2 a 128 [KB] cada una. Esta última característica es una desventaja pues hace que el número de sectores por clusters aumente conforme al tamaño de la unidad. El aumento de sectores por cada cluster genera que el sistema de archivos se vuelva ineficiente con unidades con una capacidad mayor a 2[GB], debido a que mucho espacio de almacenamiento se desperdiciará. Se debe recordar que la información es alojada en números pares de clusters y cuando la información que aloja en el cluster es menor al del cluster, todos los sectores que no son ocupados quedarán vacíos. Esto es lo que produce el desperdicio de espacio de almacenamiento.

El sistema de archivos FAT32 fue incluido por primera vez con Windows 95B y es el sucesor de FAT16, el cual cubre las deficiencias presentadas por su predecesor. Dentro de las mejoras se pueden considerar las siguientes: manejo de hasta 268, 435, 456 (2^{28}) clusters, organización más eficiente de la información al reducir el número de sectores por cada cluster, respaldo del sector

de arranque y las tablas de asignación de archivos y de direcciones no tienen un tamaño limitado (pudiendo crecer hasta que el dispositivo de almacenamiento estuviese lleno). Pese a estas mejoras, el sistema sólo puede trabajar con unidades con una capacidad de hasta 32 [GB].

Aunque en el presente trabajo de tesis se ha descrito, someramente, a la familia de sistemas de archivos FAT este sistema de archivos no es el único que existe. Sólo se presenta a la familia FAT pues ésta es soportada por la mayoría de los sistemas operativos y es el sistema de archivos más ampliamente usado en tarjetas de memorias¹¹. Para obtener mayor información sobre la familia de sistemas de archivos FAT se recomienda consultar el documento *Microsoft Extensible Firmware Initiative FAT32 File System Specification. FAT: General Overview of On-Disk Format*.

1.8.3. Características del dispositivo de almacenamiento a usar

El dispositivo de almacenamiento que se usará en el sistema *Cycle DAQ* en desarrollo será una tarjeta de memoria microSD® con una capacidad de 4 [GB], la cual funcionará con el protocolo de comunicación SPI y trabajará con el sistema de archivos FAT32.

Debido a la experiencia adquirida al usar el sistema *Cycle DAQ* anterior se puede afirmar que las características de la tarjeta de memoria son suficientes, ya que en la tarjeta de memoria se almacenarán archivos de texto, principalmente, los cuales en general ocupan menos de 512[KB] y no se suelen almacenar más de 50 archivos. Es de mencionar que la capacidad elegida para la memoria es puramente comercial, pues en el mercado actual la menor capacidad más común en tarjetas de memoria es la de 4 [GB]. La tarjeta de memoria junto con el acelerómetro y el analizador de gases son las partes que se respetaron del sistema *Cycle-DAQ* anterior.

1.9. Sistema Global para las Comunicaciones Móviles

El Sistema Global para las Comunicaciones Móviles (GSM, *Global System for Mobile Communication*) es un sistema estándar utilizado mundialmente en redes celulares digitales para transmitir voz e información.

El estándar GSM tiene sus orígenes en Europa en el año de 1980 cuando el organismo de normalización *Groupe Special Mobile* fue creado. En la época en que se creó este organismo la comunicación inalámbrica se hacía a través de sistemas de radiocomunicación analógicos (primera generación o 1G), los cuales trabajaban en espectros de frecuencias limitados de acuerdo a las normas de cada país. Las diferencias en las regulaciones hacían que la comunicación inalámbrica entre países fuera complicada. Todas estas dificultades se buscaron subsanar con la creación del estándar GSM, el cual fue concebido como un sistema digital de

¹¹ Di Jasio, L. (2007). *Programming 16-bit PIC Microcontrollers in C: learning to Fly the PIC 24*. Estados Unidos: Elsevier.

radio comunicación en la banda de los 900 [MHz] y que permitiría la comunicación inalámbrica en toda Europa.

Inicialmente el protocolo GSM fue diseñado para funcionar como la red de teléfonos fijos, es decir que tuviera el siguiente comportamiento: cuando un usuario hace una petición de un enlace, la red debe tratar de establecer una conexión única y directa entre el sistema que hace la petición y el receptor. La conexión entre los usuarios se lleva a cabo en un centro de conmutación, en el que a través del cierre de interruptores se intenta formar un circuito entre el usuario emisor y receptor (este proceso se llama conmutación). Una vez que se ha establecido la ruta correcta (enlace exitoso) la transmisión de información comienza y se mantiene hasta que una de las dos partes finaliza el enlace. Después de esto los interruptores que fueron usados para establecer el enlace quedan disponibles para nuevas comunicaciones. Este principio de funcionamiento se ha mantenido a lo largo de los años, sin embargo el estándar GSM se ha modificado para adaptarse al avance tecnológico y a la creciente demanda por mejores servicios de comunicación inalámbrica. Cabe mencionar que el estándar GSM no es el único sistema de radio comunicaciones digitales. En Estados Unidos también se desarrollaron sistemas digitales como el IS (*Interim Standard*)-95. Algunos de los sistemas digitales estadounidenses han sido adoptados por varios países y ofrecen servicios con la misma calidad que los ofrecidos por el estándar GSM. Sin embargo, para los fines de este trabajo no se tratarán.

En la actualidad, el estándar GSM es el más usado mundialmente para ofrecer servicios de comunicación móvil (transmisión de voz y datos). Estos servicios son ofrecidos por distintas empresas concesionadas para explotar un rango de frecuencias (banda de 800 o 1800 [MHz]) del espectro radioeléctrico de cada país. No obstante, de que pueda haber diferencia entre las empresas en la cobertura, tarifas, banda de operación y cobertura, todas ellas utilizan el mismo modelo de red para ofrecer sus servicios.

El modelo de red usado para el estándar GSM es el de redes celulares. Una red celular es un tipo de red inalámbrica en la que los sistemas que constituyen la red se comunican entre sí mediante señales de radiofrecuencia. La red celular recibe ese nombre porque está formada por zonas geográficas, donde es posible hacer uso de los servicios de comunicación, llamadas celdas o células. El área de una la celda puede variar desde unos cientos de metros a varios kilómetros, cada celda cuenta con una estación transceptora base cuya potencia, entre otros aspectos, define el tamaño de la célula. El conjunto de celdas y el tamaño de las mismas definen la región de cobertura de un operador.

Se hace hincapié en el hecho de que las celdas sólo definen el área geográfica donde la red celular puede operar. Sin embargo, una red celular se conforma por los siguientes elementos:

- a) **Estación móvil (MS, *Mobile Station*):** Está formada por un equipo móvil (terminal) y una tarjeta SIM (*Subscriber Identity Module*). El equipo móvil es una pequeña estación transceptora portátil con la cual el usuario (subscriber) puede usar los servicios de la red celular, mientras que la tarjeta SIM es un circuito integrado que tiene información que permite identificar al usuario dentro de la red celular. La tarjeta SIM se coloca en una ranura especial dentro del equipo móvil y se puede insertar en distintos equipos móviles.

- b) **Estación transceptora base (BTS, *Base Transceiver Station*):** Consiste en dos partes, la primera de ellas es un conjunto de circuitos (amplificadores de radiofrecuencia, transceptores de señales de radio, fuentes de alimentación, entre otros) que permiten la comunicación inalámbrica entre las estaciones móviles dentro de una celda y los demás elementos de la red celular. La segunda parte está integrada por un grupo de antenas colocadas en estructura elevada como un edificio o una torre de radio (torre de transmisión).
- c) **Controlador de estación base (BSC, *Base Station Controller*):** El BSC es un dispositivo que se encarga de controlar los canales y frecuencias de una o varias BTS. Así mismo es responsable de comunicar una estación móvil con el centro conmutador móvil y cambiar la celda con la que se comunican las estaciones móviles. Esto último se conoce como transferencia, la BSC sólo maneja transferencias entre celdas controladas por él. Las transferencias son necesarias para asegurar que una MS cuente con los servicios de la red celular a pesar de que la MS esté en movimiento. El grupo formado por las BTS y los BCS se conoce como subsistema de estación base (BSS, *Base Station Subsystem*).
- d) **Centro conmutador móvil (MSC, *Mobile Switching Center*):** El MSC es el componente de la red que coordina los procesos y los canales necesarios para establecer los enlaces entre dos dispositivos. El MSC es capaz de conmutar dos dispositivos que utilizan la red celular, al igual que es capaz de comunicar un dispositivo de la red celular con otro de la red de telefonía fija. Además de las acciones de conmutación el MSC lleva un registro de los usuarios conectados a la red, la localización de los usuarios, maneja el servicio de mensajes cortos (SMS, *Short Message Service*) y hace transferencias entre celdas controladas por diferentes BCS. Pueden existir distintos MSC correspondientes a un solo proveedor. Para poder llevar a cabo todas sus funciones el MSC se auxilia de otros elementos, alguno de ellos son:
- 1) **Registro de ubicación de base (HLR, *Home Location Register*):** Es la base de datos de la empresa que ofrece los servicios de la red celular en la que se almacenan y administran los datos de los suscriptores, por ejemplo el tipo de servicios que se contrataron, la localización del dispositivo y las actividad que se encuentra realizando el usuario (llamando, enviando mensajes, navegando por Internet, entre otros). Cuando se compra una suscripción a la red celular, adquiriendo una tarjeta SIM, toda la información acerca de esta suscripción se almacena en la HLR del operador. Las HLR son las bases de datos más importantes dentro de la red celular.
 - 2) **Registro de ubicación de visitante (VLR, *Visitor Location Register*):** Es una base de datos integrada al MSC, la cual almacenará temporalmente la información de un suscriptor que se encuentra fuera del área de cobertura de su operador, esto permitirá que el suscriptor continúe usando los servicios de la red celular.

- 3) **Centro de autenticación (AUC, *Authentication Center*):** Es una base de datos en la que se guarda una copia de un código secreto almacenado en la tarjeta SIM. La finalidad de esta base de datos es la de poder comprobar (comparando el código secreto de la AUC y de la tarjeta SIM) que los servicios de la red celular están siendo usados por un suscripto válido.
- 4) **Registro de identificación de equipo (EIR, *Equipment Identity Register*):** Es una base de datos en la que se guarda una lista de todos los equipos móviles que pueden ser usados dentro de la red celular. Cabe mencionar que cada equipo móvil tiene un identificador que lo distingue mundialmente del resto de los equipos móviles, este número se conoce como IMEI (*International Mobile Equipment Identity*). El número IMEI se guarda en el EIR junto con código para indicar si el equipo móvil puede usarse en la red celular o tiene reporte de robo.

El grupo formado por el MSC y los componentes en los que se auxilia se conoce como subsistema de red (NSS, *Network Subsystem*).

- e) **Red de telefonía pública conmutada (PSTN, *Public Switched Telephone Network*):** Es un red con circuitos de conmutación tradicional (la configuración de esta red es en a que se basaron los desarrolladores del estándar GSM), la cual es la que conecta a los sistemas de telefonía fija. Propiamente no es parte de la red celular, sin embargo como algunos enlaces se harán a teléfonos fijos se considera como parte de la red.

En la figura 1.20 se esquematiza la estructura simplificada de una red celular. En dicha figura se pueden ver las relaciones que existen entre los diferentes elementos de la red, es de importancia notar que la comunicación entre los elementos de la red es bidireccional.

Una red celular digital, de manera simplificada, trabaja de la siguiente manera: cuando una MS se enciende realiza una búsqueda de las BTS disponibles. Una vez finalizada la búsqueda, la MS manda su información a la BTS más cercana (la que ofrece una señal de mayor intensidad). La BTS retransmite la información a su BSC y éste se comunica con su MSC. En el MSC se guarda la información de la MS, si la información de la MS no se guarda en el MSC el sistema no se puede utilizar en la red celular. Para saber si la MS se está moviendo o sigue en la misma celda, la MS manda señales periódicamente y se actualiza, cuando es necesario, su localización en el MSC.

Una MS registrada en la red celular puede solicitar un enlace para transmitir voz (llamada) haciendo uso de un número telefónico (número asociado a la SIM de otra MS o bien a una línea telefónica fija). En cualquiera de los casos la MS que solicita el enlace, manda su petición y en el MSC se encargará de procesar la petición. Si el número telefónico corresponde a una MS se realiza una búsqueda en la base de datos y si la MS receptora está activa en la red se realizan conexiones necesarias para producir el enlace. Mas si el número telefónico corresponde al de una línea telefónica fija, el MSC realiza las conexiones necesarias para comunicarse con la PSTN. En

el caso de que el número telefónico no corresponda a alguna MS o alguna línea fija o la MS receptora no está conectada a la red celular se le informa al usuario de este error. Es importante mencionar que los procedimientos descritos en el funcionamiento de la red celular y en general para la transmisión de información no serían posibles sin un sistema de señalización.

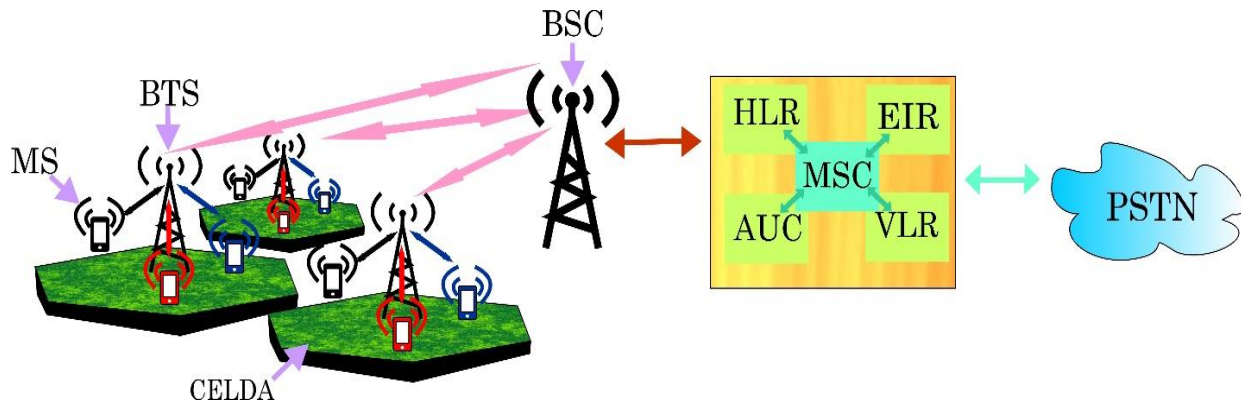


Figura 1.20. Red celular simplificada.

1.9.1. Sistema de señalización SS7

La señalización, en telefonía, es el intercambio de información (señales) entre diferentes elementos de una red para coordinar su funcionamiento y así proveer servicios de comunicación. En el caso de la red celular GSM uno de los sistemas de señalización más importante es el sistema de señalización SS-7 (*Common Channel Signaling System No. 7*).

El SS7 es un conjunto de protocolos de señalización fuera de línea. Es decir que los procesos de señalización se llevan a cabo en canales independientes a los canales usados por los servicios de comunicación, como consecuencia de esta característica los protocolos SS-7 definen y usan una red estrechamente relacionada, pero independiente, de la red telefónica.

La red usada para los protocolos SS-7 se conforma por nodos llamados puntos de señalización, cada punto de señalización cuenta con una dirección formada por 14-bits, normalmente, la cual sirve para identificar a los nodos en la red. Todos los puntos de señalización están conectados entre sí a través de canales donde se mandan los mensajes de coordinación, los cuales viajan a una velocidad de 64 o 56 [kbps]; para proveer una mayor velocidad de transferencia o bien poder generar redundancia dos nodos puede conectarse hasta con 16 canales. Las diferentes conexiones entre puntos de señalización son rutas estáticas, lo cual quiere decir que los nodos se conectan directamente entre sí por rutas definidas. Existen tres diferentes tipos de nodos (puntos de transferencia de señal, punto de conmutación de servicios y punto de control de servicios) y cinco tipos de enlaces (A, B, C, D, E y F) definidos por el sistema de señalización SS7, los cuales se pueden conectar de diferentes maneras llamadas arquitecturas. Una de las arquitecturas más conocida es la red de señalización norteamericana. Aunque las arquitecturas definen la forma de conexión estas no afectan las actividades que desempeña el sistema de señalización, algunas de

ellas son: a) iniciar, mantener y finalizar llamadas b) administración de servicios como votación telefónica, números gratuitos (aquellos que contienen el número 800 en su inicio), identificación de llamadas (usado en líneas de telefonía fija, permite conocer el número y nombre de la persona que llama a otra) y portabilidad numérica (permite conservar el número telefónico aunque se cambie de empresa telefónica) y c) en la red celular GSM permite el envío de mensajes cortos (son de una longitud máxima de 160 caracteres y se envían por los canales de señalización), interconexión entre redes celulares, cobro por servicios de comunicación, manejo de movilidad (permite que se realicen los cambios de zona de cobertura durante una llamada), itinerancia (manejo de servicios en áreas de cobertura que no pertenecen a la compañía celular con la que se contrató la suscripción a la red celular) y el manejo de servicios de datos. En la figura 1.21¹² se muestra esquemáticamente como se implementa el sistema de señalización SS7 en la red celular.

En la figura 1.21 se puede ver que la conexión entre los componentes de la red tiene nombres como A-bis, A y MAP/B, C, D, E, F, G y H. Los nombres de la conexión permiten conocer los tipos de sistema de señalización que se utilizan, en el caso los enlaces U_m y A-bis se utiliza un sistema de señalización basado en unos estándares ISDN (*Integrated Services for Digital Networks*), las demás conexiones se basan en el sistema de señalización SS7. Para conocer más sobre el sistema de señalización SS7 en la red GSM se recomienda consultar la norma *ETSI TS 100 974 Digital Cellular Telecommunications Systems (Phase 2 +); Mobile Application Part (MAP) Specification*.

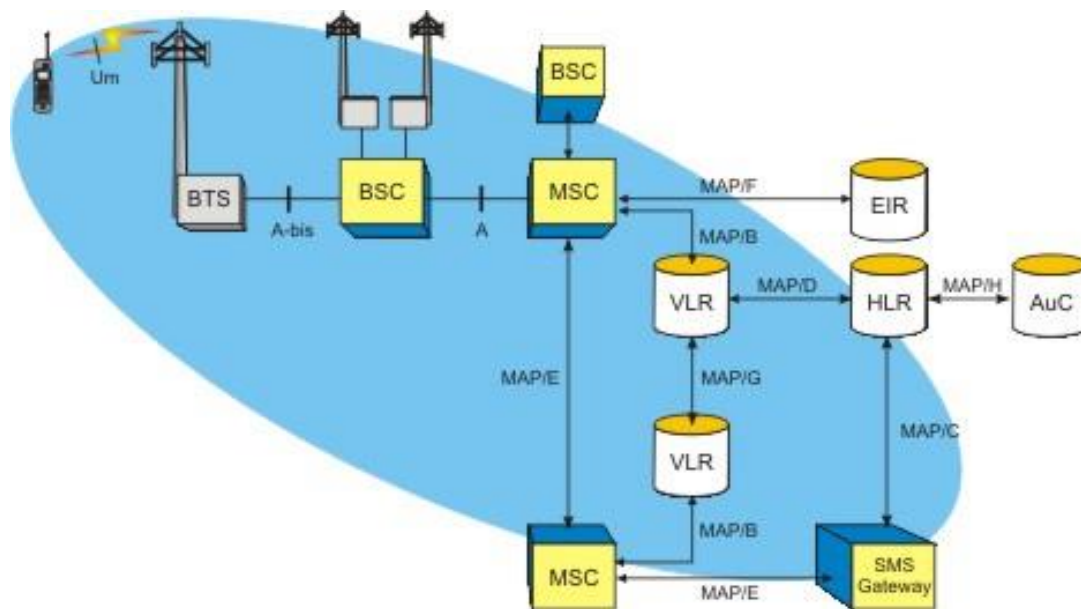


Figura 1.21. Red celular con la red de señalización indicada.

¹² Harte, R., Dreher, R., Bowler, D. & Beninger T. [Imagen sin título de descripción del trabajo]. Recuperado de: <http://www.althosbooks.com/intosisy7sad.html>

1.9.2. Técnicas de acceso múltiple

El recurso más importante, más costoso y limitado en cualquier red celular es el espectro radioeléctrico (banda de frecuencias), es decir los canales disponibles para realizar la comunicación inalámbrica. Debido a las limitaciones del espectro radioeléctrico y con el fin de cubrir la demanda de comunicación de los suscriptores de la red celular, los proveedores implementan técnicas de acceso múltiple para optimizar el espectro radioeléctrico que pueden explotar. Existen diferentes técnicas de acceso múltiple con diferentes procedimientos para lograr la repartición del espectro entre los suscriptores, sin embargo en este trabajo sólo se tratarán las técnicas de acceso múltiple por división de frecuencia y por división de tiempo.

1.9.3. Técnica de acceso múltiple por división de tiempo y de frecuencia

La técnica de acceso múltiple por división de frecuencia (FDMA, *Frequency Division Multiple Access*) fue la primera técnica usada para transmitir señales de radio, de televisión y en la primera generación de las comunicaciones celulares. En esta técnica la banda de frecuencias disponible es dividida en subgrupos de bandas de frecuencias, suficientemente apartados unos de otros para evitar interferencia, llamados canales. Un ejemplo donde se ocupa la técnica FDMA es la transmisión de señales de televisión, en dicha transmisión sólo existe una banda de frecuencias dividida en varias frecuencias que conocemos como canales televisivos. Así mismo, en este tipo de transmisión todos los aparatos receptores pueden obtener la información de un canal simultáneamente, sin interferir con otros equipos ni afectar la información del canal. Otro ejemplo de aplicación de la técnica FDMA es la comunicación celular, en la cual, a diferencia de la transmisión de señales de televisión, se ocupan dos bandas de frecuencias una llamada banda de enlace de subida (de la MS a la red) y la otra de enlace de bajada (de la red a la MS). Durante la transmisión de información un canal de cada banda se asigna a un usuario, estos canales no pueden ser ocupados por otro usuario, y en caso de que dos usuarios usaran el mismo canal se generaría interferencia, es decir que es lo contrario a lo que sucede en el caso de las señales televisivas. En la figura 1.22 se puede ver como en la técnica FDMA una banda de frecuencias se divide en subgrupos de bandas de frecuencia, cada subgrupo tiene su propia frecuencia central. Es importante mencionar que la técnica FDMA es la técnica de acceso múltiple más sencilla de implementar en los sistemas de comunicación, su gran desventaja consiste en que el número de usuarios al que puede servir al mismo tiempo es reducido, no obstante este modelo sigue vigente en las comunicaciones inalámbricas sirviendo como complemento a otras técnicas.

Por otro lado, la técnica de acceso múltiple por división de tiempo (TDMA, *Time Division Multiple Access*) tiene como principio de operación la compartición de un mismo canal de comunicación, durante la transmisión de información, entre varios usuarios. La técnica TDMA tiene sus orígenes en las comunicaciones telegráficas y fue usada en la telefonía celular por primera vez en la primera generación, como complemento de la técnica FDMA, para incrementar el número de usuarios que se podían conectar a la red. Durante la segunda generación de telefonía celular la técnica TDMA se adaptó a los sistemas digitales.

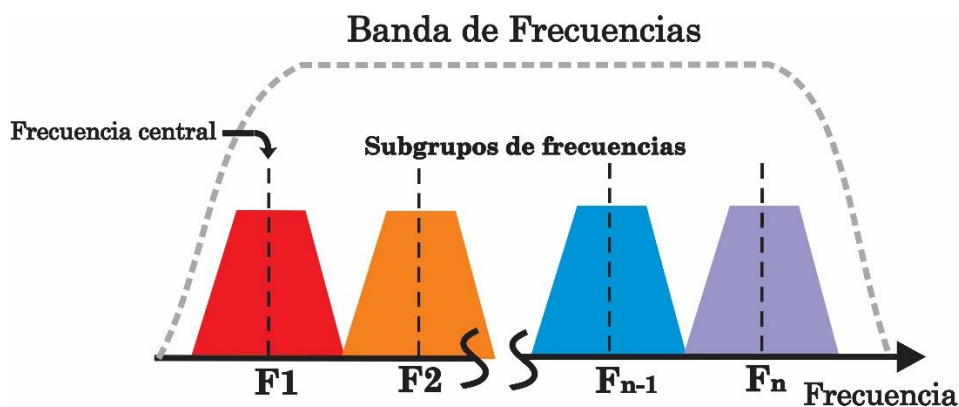


Figura 1.22. División de espectro en FDMA

En el caso de la red la red celular GSM la técnica TDMA se usa en conjunto con la técnica FDMA, es decir que los canales definidos por la técnica FDMA son optimizados por la técnica TDMA, permitiendo que más de un usuario transmita información en un mismo canal. Para lograr la compartición del canal se asigna un lapso de tiempo (bloque de tiempo o *slot*) a cada uno de los usuarios durante el cual pueden transmitir información; una vez asignado un bloque de tiempo a un usuario se mantiene dicho bloque durante toda la transmisión; el tiempo asignado para cada bloque de tiempo es de unos cuantos milisegundos (.577 [ms] en sistema GSM). Los bloques de tiempo tienen un orden definido para transmitir su información, el conjunto de bloques de tiempo ordenados forman una cadena llamada *frame*. Los *frames* tienen un tamaño definido de bloques de tiempo, ocho en el sistema GSM (lo que equivale a transmitir durante 4.615[ms]), el cual es igual al número de usuarios que pueden transmitir en el canal a la vez. Una vez que se han transmitido todos los bloques de tiempo de un *frame* la secuencia de transmisión se renueva en el mismo orden. La información de un usuario se transmite a lo largo de varios *frames*. Cada canal definido por la técnica FDMA aloja a un solo *frame*. Los *frames* de los canales de transmisión se agrupan en conjuntos de 26 a 51 *multiframes*, éstos últimos a su vez en conjuntos de 26 a 51 *multiframes* llamados *superframes* y éstos a su vez en grupos de 2048 *superframes* llamado *hyperframes*, toda esta organización permite establecer un esquema general de transmisión y la sincronización de la red celular. Lo anteriormente descrito se ilustra en la figura 1.23.

Por otro lado, en la técnica TDMA, aplicada al sistema GSM, se hace una distinción entre canales físicos y canales lógicos. Se considera que un canal es físico si cuenta con una frecuencia portadora, es decir los canales definidos por la técnica FDMA, y también a cada uno de los bloques de tiempo de un *frame*. Un canal lógico se define por la información que se transmite a través de un canal físico, las principales categorías de los canales lógicos son: canales de tráfico (usados para transmitir información) y canales de señalización (para manejar el funcionamiento de la red). Cabe mencionar que en un solo *frame* existen los canales de tráfico y de señalización, lo cual reduce el número de canales que los usuarios pueden ocupar para intercambiar información, sin embargo este problema se resuelve aumentando la infraestructura de la red celular.

La estructura FDMA/TDMA que utiliza el sistema GSM se aplica en cada una de las celdas de la red, lo cual significa que por cada celda, en teoría, se podrán usar todos los canales definidos por la técnica FDMA y por cada uno de estos canales los definidos por la técnica TDMA. Una de las prácticas más comunes para aumentar la capacidad de usuarios que simultáneamente intercambian información, en especial en zonas con alta demanda de servicios, es dividir la cobertura de una celda entre celdas más pequeñas.

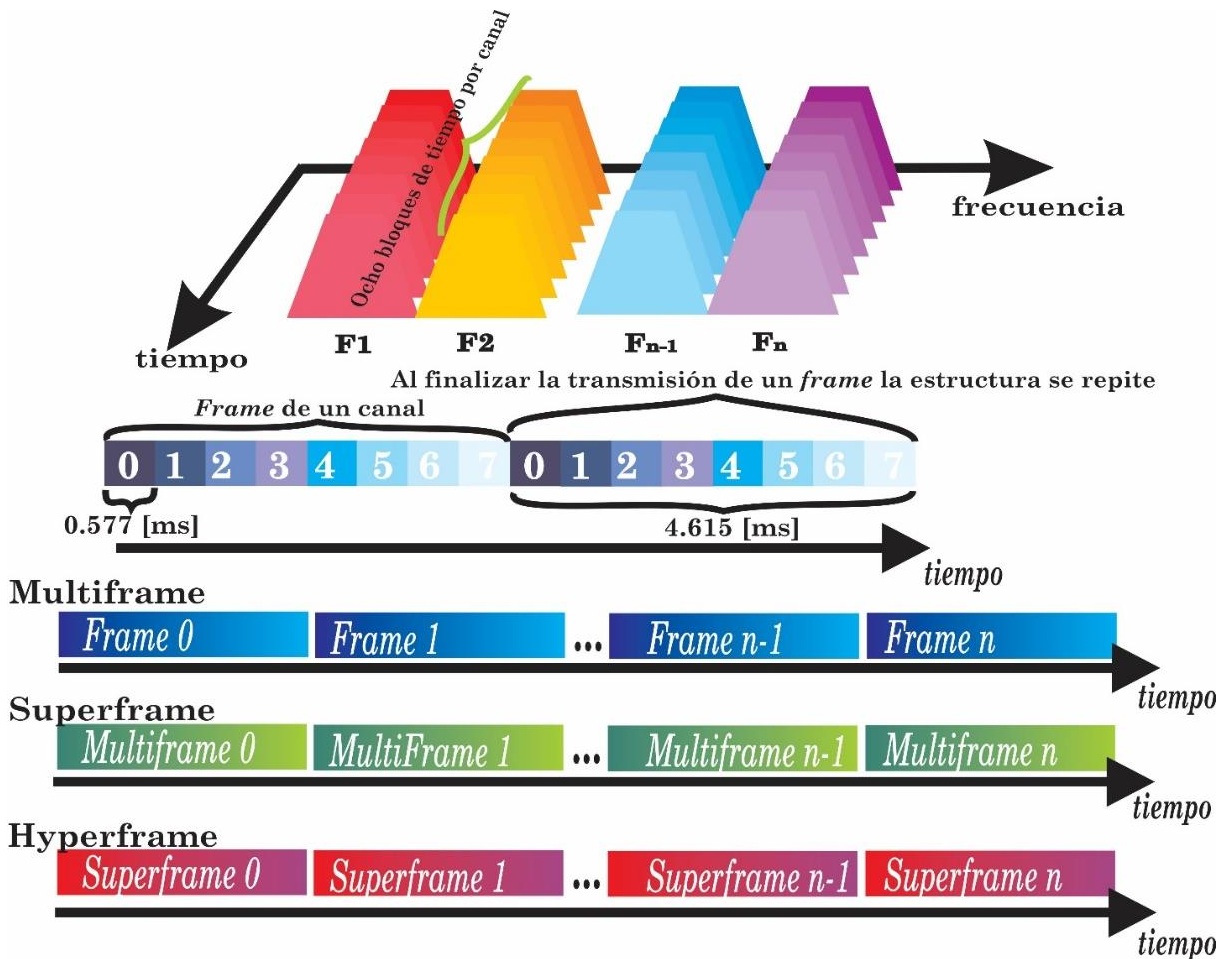


Figura 1.23. Estructura y organización de la técnica TDMA

1.9.4. Servicio General de Paquetes Vía Radio

Inicialmente el sistema GSM fue desarrollado para la transmisión de voz y para realizar dicha transmisión se requiere establecer un circuito entre el usuario que transmite y el que recibe. Aunque el sistema GSM implementa la técnica TDMA para aumentar el número de usuarios, los enlaces de transmisión de voz entre los usuarios se establecen mediante circuitos físicos; durante la transmisión de voz el operador de la red cobra el tiempo que el circuito, entre los usuarios, está activo. Por otro lado, la creciente demanda de servicios de Internet, hicieron que los operadores

de las redes celulares fijaran su atención en la transmisión de datos. Los primeros intentos para transmitir datos a través de la red celular se realizaron usando los mismos principios para la transmisión de voz, pero las velocidades de transmisión resultaron ser muy bajas para una navegación satisfactoria en Internet (14.4 [kbps] como máximo); el cobro por tiempo de conexión hacía muy costoso el servicio y los servicios de itinerancia no trabajaban muy bien con la transmisión de datos. Con el fin de sobrellevar estas dificultades y ofrecer servicios de comunicación de datos se desarrolló el Servicio General de Paquetes Vía Radio (GPRS, *General Packet Radio Service*). El servicio GPRS se desarrolló como una extensión del sistema GSM que permitiría habilitar el acceso a Internet, proveer velocidades aceptables de navegación en Internet, hacer posible que el usuario pudiese recibir llamadas y correos electrónicos en cualquier momento, así como disminuir los costos del servicio de transferencia de datos. El servicio GPRS comenzó a utilizarse comercialmente a inicios del año 2000 y representó el primer gran paso en la distribución de servicios de datos a través de las redes celulares. La implementación del servicio GPRS da inicio a la generación 2.5 de telefonía celular y es en una pieza clave para el desarrollo de las generaciones posteriores. Al igual que otros servicios de comunicación inalámbrica se desarrollaron diferentes versiones de la red GPRS, en el presente trabajo sólo se hablará del desarrollo implementado en la red GSM.

La red GSM y el servicio GPRS coexisten en la misma infraestructura, en las mismas frecuencias y en los mismos canales, pero trabajan de manera distinta. En el caso de la red GSM, como ya se ha mencionado, independientemente de si se transfiere voz o cualquier otro tipo de información se asignan recursos, bloques de tiempo, que no pueden ser usados por otros usuarios. En cambio, en el servicio GPRS se utiliza una técnica de transferencia llamada conmutación de paquetes, la cual permite asignar recursos a los usuarios mientras estén transfiriendo datos y una vez finalizada la transferencia los recursos quedan disponibles. Otras diferencias entre el sistema GSM y el GPRS son que es posible asignar varios bloques de tiempo de un solo *frame* a un usuario durante la transmisión, la transferencia de información no siempre se realiza en tiempo real, GPRS soporta los protocolo de Internet, el cobro se realiza por la cantidad de información intercambiada y los usuarios pueden estar conectados a la red GPRS indefinidamente. Cabe mencionar que la conmutación de paquetes consiste en dividir la información que se enviará en paquetes de datos con un tamaño definido a los cuales se les agrega un encabezado que indica el lugar de partida y el de llegada, esta técnica de transferencia de datos es usada en Internet. La información del encabezado es importante porque no existe un canal definido para la transferencia de información y sólo cuando la información llega o se envía se asignan los canales.

Aunque el servicio GPRS sólo es una extensión del sistema GSM, requiere de equipos, interfaces de conexión y protocolos específicos con los cuales trabajar, algunos de estos se enlistan a continuación:

- **Nodo de soporte de la compuerta GPRS (GGSN, *Gateway GPRS Support Node*):** Es el equipo encargado de direccionar los paquetes de información a la localidad de la MS, para hacerlo tiene que comunicarse con la HLR de la red y obtener la información de la MS.
- **Nodo de soporte de servicio GPRS (SGSN, *Service GPRS Support Node*):** Se encarga de realizar un seguimiento a una MS conectada vía GPRS y controlar el acceso a la

transferencia de archivos. Este nodo junto con el GGSN forman la NSS de la red GPRS. El SGSN se conecta con las BSS de la red celular.

- **Unidad de control de paquetes (PCU, *Packet Control Unit*)** Es un equipo dentro de las BTS o de los BSC que permite distinguir si una MS está transfiriendo voz o está transfiriendo datos, en el primer caso se usará la red GSM y en el segundo la red GPRS. Además se encarga de la segmentación de la información y de ensamblar los paquetes de información, control y organización de las transmisiones y de los canales.

En general, la red GPRS tiene un comportamiento similar al de la red usada para la transmisión de voz, es decir que una MS se comunica con su BTS más cercana, esta última se comunica con su BSC y aquí una PCU desvía la información al SGSN, en el SGSN se registra y verifica la localización de la MS y el GGSN permite el intercambio de información con Internet. Para conocer más sobre el servicio GPRS en conjunto con el sistema GSM se recomienda leer las normas EN (*European Standard*) 102 111-1, 301-113, 301-347, 301-344 y 301-349.

1.9.5. Descripción de la estación móvil a usar

El sistema *Cycle-DAQ* en desarrollo permitirá el monitoreo de los vehículos bajo prueba en un puesto central, para ello se enviarán, a un servidor, algunas muestras de los datos adquiridos por el equipo a través de la red GSM-GPRS.

La conexión del sistema *Cycle-DAQ* con la red GSM-GPRS se realizará usando una tarjeta de desarrollo que cuenta con un módulo transceptor celular de la compañía Quectel®, el cual soporta las bandas de frecuencia de 850, 900, 1800 y 1900 [MHz]; si el transceptor trabaja en las bandas 850 o 900 [MHz] puede llegar a transmitir con una potencia máxima de 2 [W], mientras que si opera en las bandas de 1800 o 1900 [MHz] la potencia máxima de transmisión es de 1 [W]. En la normativa GSM se establece que la potencia máxima de transmisión en cada banda de frecuencias en las que opere un transceptor debe estar en el rango de una de cinco clases (1, 2, 3, 4 o 5). Por ejemplo, el transceptor que se usará está clasificado como clase 4 para la banda de frecuencias de 850 y 900 [MHz], y 1 para las bandas de frecuencias de 1800 y 1900 [MHz]. Por otro lado, el transceptor que se usará puede trabajar con los modos de codificación para transmisión de voz aceptados para GSM *full rate*, *half rate*, *enhanced full rate* y *adaptive multirate*, estos modos de codificación están relacionados con la velocidad a la que pueden transmitir voz; el transceptor puede enlazarse a las redes GSM y GPRS al mismo tiempo, sin embargo únicamente puede transmitir en una de las dos redes a la vez, esta característica hace que sea clasificado como equipo clase B (aparte de la clase B existe la clase A y la clase C, en la clase A se clasifican los equipos que se enlazan y transmiten simultáneamente en las redes GSM y GPRS, mientras que en la clase C se considera a los equipos que sólo se pueden enlazar y transmitir en la red GSM o GPRS a la vez). El módulo transceptor soporta los cuatro esquemas de codificación (1, 2, 3 y 4) de GPRS, los cuales están relacionados con la velocidad de transmisión y la intensidad de la señal que existe entre una MS y su BTS, entre más grande sea el número se indica una transmisión de datos con mayor velocidad pero que requiere de una mejor calidad de señal. El transceptor opera con una tensión nominal de 4 [V] y puede llegar a consumir hasta 1.6 [A] durante la transmisión; tiene una salida y una entrada para audio; soporta el servicio

SMS; trabaja con tarjetas SIM que funcionen con tensiones de 3 y 1.8 [V]; requiere de una antena externa y se comunica con otros sistemas mediante comunicación serie asíncrona. En la figura 1.24 se muestra el diagrama de bloques simplificado del módulo GSM-GPRS.

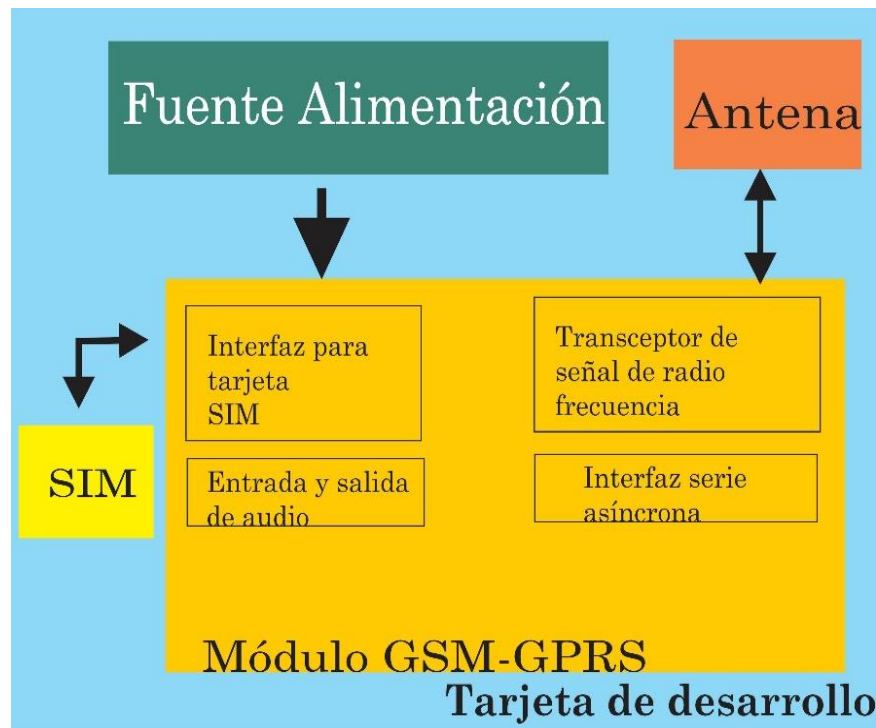


Figura 1.24. Tarjeta de desarrollo con módulo GSM-GPRS.

Como se puede ver en la figura 1.24, la tarjeta de desarrollo tiene una sección dedicada a una fuente de alimentación, dicha fuente de alimentación consiste en un convertidor de potencia DC/DC, el cual permite entregar los 4 [V] que requiere el módulo GSM-GPRS, así como la corriente de 1.6 [A], sin afectar el funcionamiento del entorno en el que se conecte la tarjeta de desarrollo. Así mismo, la tarjeta de desarrollo cuenta con una ranura para insertar tarjetas SIM (de cualquier compañía) y con una antena de montaje superficial con características especiales para comunicaciones celulares. En el presente trabajo no se utilizarán las entradas y salidas de audio pues sólo se harán transmisiones de datos usando la red GPRS.

1.10. Analizador de gases

Un analizador de gases es un instrumento que permite conocer la concentración de uno o varios gases en una muestra de interés. El análisis de la muestra se puede llevar a través de diferentes procedimientos, sin embargo en el presente trabajo se abordarán los principios de los sensores electroquímicos y los sensores infrarrojos no dispersivos.

1.10.1. Sensores electroquímicos de gases

Los sensores electroquímicos de gases permiten conocer la concentración de un gas en una muestra determinada con base en la medición del cambio de una propiedad eléctrica de una celda electroquímica (par de electrodos entre los cuales existe un material electrolítico sólido o líquido) en presencia de un gas determinado. Los sensores electroquímicos son los sensores más sencillos utilizados para medir la concentración de gas debido a que su respuesta es proporcional a la concentración gaseosa. No obstante, éstos se deterioran con el uso y el tiempo, lo cual hace que se requiera de una calibración constante o en su defecto remplazo del material electrolítico.

Debido a que los diferentes materiales electrolíticos presentan reacciones químicas distintas ante la presencia de un gas, se utilizan varias técnicas para conocer la concentración de un gas en una muestra, cuatro de las técnicas más usadas son:

- **Análisis por conductancia:** En esta técnica se mide el cambio de la conductancia del material electrolítico cuando entra en contacto con una muestra de gas, esto se debe a que el gas produce una mayor cantidad de iones en el material electrolítico, lo cual propicia que baje la conductancia. Las principales ventajas de esta técnica son una alta sensibilidad y una respuesta rápida. Sus desventajas son que la respuesta puede producirse por varios gases (en algunas mezclas de gases no se podría conocer cual gas está produciendo la reacción), depende de la temperatura y de la cantidad de gas en contacto con el material electrolítico.
- **Análisis coulombimétrico amperométrico:** En este tipo de análisis se mide la corriente consumida o producida por la reacción química entre el electrolito y la muestra. Cuando el gas entra en contacto con la celda electroquímica, la cual está conectada a una fuente de alimentación de corriente continua, la concentración de iones cambia y por ende se produce un cambio en la corriente que fluye por la celda. Este tipo de análisis permite conocer concentraciones muy pequeñas y al igual que los métodos por conducción la respuesta del sensor se puede deber a varios gases.
- **Análisis mediante transductor electroquímico:** En este tipo de análisis se mide la corriente en un electrodo producida por la oxidación de un gas. Se debe notar que a diferencia del análisis coulombimétricos la corriente no se debe a los iones por la interacción del electrolito y el gas, sino por la oxidación del gas en uno de los electrodos, los cuales también están conectados a una fuente de corriente directa. Los sensores usados en este análisis presentan una respuesta rápida, son construidos para reaccionar sólo ante ciertos gases y requieren ser calibrados constantemente.
- **Análisis mediante electrodos selectivos de iones:** Este tipo de análisis se realiza para conocer la concentración de iones de interés en una solución electrolítica. Debido a que una mezcla de gases puede producir diversas reacciones químicas en una celda electroquímica, es factible que se formen iones de diversos compuestos químicos. Para medir la concentración iónica, por ende conocer la concentración de un gas, se usan electrodos que poseen una membrana capaz de dejar pasar sólo un tipo de iones

1.10.2. Sensores ópticos no dispersivos

Los sensores ópticos no dispersivos son dispositivos que permiten medir la concentración de un gas, la cual está directamente relacionada con la disminución de la intensidad de un haz luminoso, debido al fenómeno de absorción.

El fenómeno de absorción se puede entender considerando una onda electromagnética oscilando en un medio, la cual incide sobre una partícula que también está vibrando. Si la frecuencia de la onda coincide con la frecuencia a la que vibra la partícula, la energía de la onda se absorberá por completo y ésta se convertirá en energía cinética que se disipará en forma de calor. Si consideramos que la onda a la que se hace referencia formaba parte de un haz luminoso, el haz al final de su recorrido tendrá una intensidad menor debido a la pérdida de una sus componentes.

En general los sensores ópticos no dispersivos están constituidos por una fuente luminosa incoherente (espectro de emisión es amplio), una cámara donde se hace pasar el gas y un sensor antecedido por un filtro óptico selectivo. En la figura 1.25 se puede ver la estructura de un sensor no dispersivo.

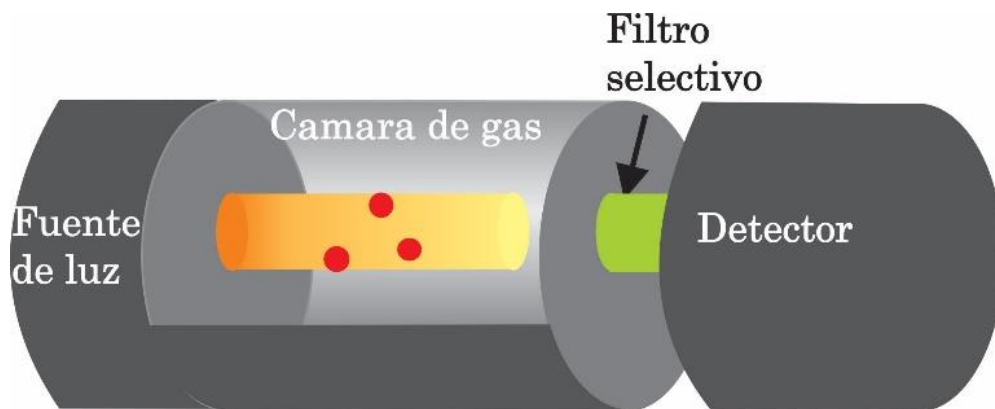


Figura 1.25. Estructura de un sensor no dispersivo.

Cabe mencionar que existen dos tipos principales de sensores ópticos no dispersivos, los de un solo haz, como el mostrado en la figura 1.25, y los de doble haz, para efectos de este trabajo sólo se describirá el sensor de un solo haz. Como se puede ver en la figura 1.25, un haz será emitido por la fuente de luz, el haz cruzará por la cámara de gas donde parte de la energía del haz será absorbida por las partículas de la muestra. Dado que el haz de luz es incoherente, el filtro selectivo se encargará de limitar el ancho de banda del haz que pasará a través de él y el cual irá al detector. Considerando que el filtro selectivo sólo deja pasar las ondas electromagnéticas cercanas a la frecuencia de absorción de un gas en específico el detector permitirá asociar la intensidad de luz, con la concentración de un gas y una magnitud eléctrica. Por otro lado, si a una estructura como la mostrada en la figura 1.25 se le agregan más filtros selectivos y más detectores es factible analizar la concentración de varios gases a la vez. Así mismo si la radiación de la fuente de luz es modulada se obtienen lecturas más exactas.

1.10.3. Descripción del analizador de gases a utilizar

El sistema en desarrollo, como ya se ha mencionado, contará con una interfaz RS-232, la cual permitirá que el sistema en desarrollo se comuniquen con un analizador de gases. El analizador de gases medirá las concentraciones de algunos de los gases emitidos por el motor de un vehículo bajo prueba, dichos gases son consecuencia del proceso de combustión que se lleva a cabo en el motor.

Cabe mencionar que idealmente los gases emitidos en un proceso de combustión son dióxido de carbono (CO_2) y vapor de agua (H_2O), no obstante el proceso de combustión nunca es ideal por lo que se producen otros gases como nitrógeno (N_2), óxidos de nitrógeno (NO_x), combustible no quemado (HC), monóxido de carbono (CO) y oxígeno (O_2). Dependiendo de las condiciones bajo las cuales trabaje el motor del vehículo se presentan dos tipos de combustión: completa, donde existe una concentración mayor de comburente que de combustible e incompleta, en la que hay más combustible que comburente. Según el tipo de combustión que se esté llevando a cabo se generan cantidades distintas de los gases antes mencionados. Las concentraciones de estos gases junto con otros parámetros permiten conocer el estado del motor.

El analizador de gases que se usará puede medir las concentraciones de HC, CO, CO_2 , NO_x y O_2 en el escape del vehículo bajo prueba. Los gases del escape del motor llegan al analizador por medio de una sonda colocada en el escape del vehículo, el otro extremo de la sonda está conectado a una bomba que se encarga de aspirar los gases y hacerlos llegar a los sensores. Es de relevancia mencionar que los gases antes de ser analizados son filtrados para eliminar el vapor de agua y partículas mayores a $0.3 \text{ } [\mu\text{m}]$. La muestra filtrada es analizada mediante un sensor óptico no dispersivo con el que se miden las concentraciones de HC, CO y CO_2 y mediante métodos electroquímicos se puede conocer la concentración de NO_x y O_2 . Cabe mencionar que cada sensor a su vez cuenta con un sensor de temperatura. La señal de salida de los sensores es convertida a una señal digital que es leída por el procesador del analizador de gases, dicho procesador se encarga de ajustar las lecturas conforme a valores de calibración (el manual de usuario indica el procedimiento de calibración), desechar las lecturas inválidas y adecuar las mediciones considerando la temperatura del gas. Una vez que se han finalizado las mediciones, los gases de prueba son liberados al medio ambiente.

Todas las mediciones realizadas se pueden transferir a otro dispositivo mediante comunicación serie asíncrona. Usando la comunicación serie asíncrona también se controla el funcionamiento del equipo, es decir que se pueden iniciar procesos de calibración, el encendido de la bomba, reinicio del sistema, entre otros. La comunicación serie asíncrona permite intercambiar secuencias de bytes; dependiendo de la acción que se esté realizando la estructura de la secuencia cambia. En el presente trabajo no se explicarán las secuencias de bytes para conocer más sobre éstos se recomienda leer el trabajo de tesis de Ramírez-Chavarría, R.G. (2013). *Adquisición y Registro de Parámetros Vehiculares para el Desarrollo de Ciclos de Manejo en el Valle de México* (Tesis de Licenciatura), Universidad Nacional Autónoma de México, Distrito Federal, México.

1.11. Microcontroladores

Los microcontroladores son sistemas computacionales constituidos en un único circuito integrado. Un microcontrolador está conformado principalmente por una unidad de procesamiento central (CPU, *Central Processing Unit*), memorias para datos y programas, puertos de entrada y salida, módulos para comunicación serie asíncrona o síncrona, convertidores analógico-digitales, entre otros. A continuación se presentará someramente algunas de las partes que integran a un microcontrolador:

- **Unidad Central de Procesamiento (CPU, *Central Processing Unit*):** La CPU es un conjunto de circuitos electrónicos con un grupo de operaciones lógicas y aritméticas predeterminadas, las cuales se utilizan para llevar a cabo una secuencia de instrucciones definidas por un programa. La CPU es la parte central de un microcontrolador debido a que es el responsable de administrar el funcionamiento y procesar la información de un microcontrolador. Los circuitos que integran una CPU típico son: la unidad aritmética lógica, encargada de realizar las operaciones asociadas a una instrucción; registros (o celdas de memoria son circuitos que pueden almacenar el estado de un byte) para las operaciones básicas de la CPU los más comunes son el contador de programa, que se encarga de almacenar la ubicación de la instrucción siguiente a ejecutar, el registro de estado que contiene la información sobre el resultado de las instrucciones ejecutadas más recientemente y el registro de pila, que guarda la dirección de la instrucción que invoca a una subrutina; registros de propósito general para guardar información de diversa índole y una unidad de decodificación y control para administrar el funcionamiento de la CPU y manejar interrupciones. El funcionamiento de una CPU al ejecutar las instrucciones de un programa consiste en tomar la dirección del registro del contador de programa, aumentar el valor en el registro del contador de programa, decodificar y almacenar la instrucción en un registro, iniciar la operación de la unidad aritmética-lógica, almacenar en los registros correspondientes el resultado de la unidad aritmética-lógica, repetir el proceso secuencialmente hasta la última instrucción del programa, tras lo cual reiniciará la secuencia. Cabe mencionar que toda la información que procesa la CPU es binaria.
- **Memoria de programa:** Es una memoria no volátil, la cual puede ser programable o programable y borrable. Esta memoria recibe su nombre porque en ella se guarda el conjunto de instrucciones (programa) que ejecuta la CPU.
- **Memoria de datos:** Es una memoria de acceso aleatorio en la que se guardan algunas variables relacionadas con la ejecución de instrucciones.
- **Puertos de entrada y salida:** Son las terminales que permiten la interacción del microcontrolador con otros elementos externos.
- **Buses¹³ de direcciones, datos y control:** Estos se utilizan para conectar las partes que forman al microcontrolador. El *bus* de dirección se usa para transferir la dirección de un

¹³ Un *bus* es un conjunto de hilos (canales físicos) que se usan para intercambiar información

registro donde se desea escribir o leer, el de control se usa para especificar si la operación es de lectura o escritura y el de datos para transferir información.

- **Base de tiempo:** Es una señal de reloj que permite mantener sincronizado a todo el sistema. La base de tiempo puede ser proporcionada externa o internamente.

Las partes antes mencionadas son fundamentales en la constitución de un microcontrolador, no obstante el avance tecnológico permitió que varios subsistemas que eran considerados periféricos se pudieran incluir en el circuito integrado. Algunos de estos subsistemas son:

- **Temporizadores:** Es un subsistema que permite realizar cuentas de tiempo sin afectar la base temporal.
- **Perro guardián (*Watchdog*):** Es una aplicación de seguridad que reinicia al microcontrolador si el programa entra en una secuencia infinita.
- **Interfaces de comunicación:** Son módulos que soportan protocolos de comunicación como I²C, SPI, serie asíncrono, *bus* serie universal (USB, *Universal Serial Bus*), entre otros.
- **Convertidor analógico-digital:** Este módulo transforma variables analógicas en valores digitales.

Los diversos microcontroladores que existen en el mercado actual se diferencian entre sí por las características de las partes que los conforman, por ejemplo el número de puertos, la cantidad de bits que procesa la CPU en una operación o las interfaces de comunicación que posee. Otro aspecto importante que diferencia a los microcontroladores es la arquitectura bajo la cual fue diseñado. Existen dos arquitecturas principales las cuales son:

- **Arquitectura Harvard:** En esta arquitectura la memoria de programas y la de datos están separadas y se conectan mediante *buses* de control, dirección y datos independientes a la CPU. La ventaja de esta arquitectura radica en su eficiencia pues permite el acceso simultáneo a las dos memorias, así mismo los *buses* entre las memorias pueden ser de distinto tamaño. Sin embargo la separación de las memorias requiere de un mayor número de conexiones físicas. Otro problema con esta arquitectura es que parte de la memoria de programas se dedica a almacenar valores constantes (pues es información no volátil) por lo que se requiere de instrucciones especiales para poder obtener los valores constantes sin confundirlos con instrucciones del programa.
- **Arquitectura von Neumann o Princeton:** En este tipo de arquitectura existe un único bloque de memoria en el que conviven la memoria de programas y de datos, además de compartir los mismos *buses* de datos, de control y dirección. Esta arquitectura es menos eficiente que la arquitectura Harvard pues se requiere de más tiempo para poder obtener información de las memorias de datos y programas, pero la gran ventaja de esta

arquitectura es que es más simple, pues no hay diferencia al trabajar con constantes y variables.

El diseño y construcción de un microcontrolador, además de la arquitectura, se basa en dos posibles estrategias de diseño que son:

- **Computadora con conjunto reducido de instrucciones (RISC, *Reduced Instruction Set Computer*):** Esta estrategia y filosofía de diseño fue desarrollada en la década de los 70's, con la finalidad de reducir el tiempo y la complejidad de las instrucciones que se llevaban a cabo en las computadoras de la época. Para lograr esto se desarrollaron sistemas cuyas instrucciones eran menos complejas y requerían menos tiempo de ejecución, con la desventaja de que los programas se volvieron más largos. La reducción en la complejidad de las instrucciones permitió aumentar el número de registros disponibles y disminuir la cantidad de circuitos dedicados a operaciones lógicas.
- **Computadora con conjunto complejo de instrucciones (CISC, *Complex Instruction Set Computer*):** Esta filosofía y estrategia de diseño es opuesta a la filosofía RISC, ya que se busca que las instrucciones engloben diversos procesos, es decir que las instrucciones se asemejen a un lenguaje de alto nivel. Con esto se busca disminuir el tamaño de los códigos de programación, sin embargo el tiempo de procesamiento y la circuitería requerida es mayor que la de la filosofía RISC.

1.11.1. Características del microcontrolador a implementar

El microcontrolador que se usará es fabricado por la compañía Atmel; los elementos principales de éste son una CPU de 8 bits, 256 [kB] de memoria de programas, 8 [kB] de memoria de datos, 32 registros de uso general y 86 puertos de entrada y salida. El circuito integrado está construido de acuerdo a la arquitectura Harvard y a la filosofía RISC. Como características adicionales, este microcontrolador cuenta con cuatro módulos para comunicación serie asíncrona, dos módulos para comunicación serie síncrona (uno para el protocolo SPI y otro para el protocolo I²C), un convertidor analógico digital de 10 bits, 6 temporizadores, 6 módulos para generar señales moduladas por ancho de pulso y un perro guardián. Por otro lado, el microcontrolador posee diversos modos energéticos de operación que permiten que el circuito integrado pueda trabajar con 1.8 [V] a 100 [nA] hasta 5 [V] a 25 [mA]. Este circuito integrado puede trabajar con señales de reloj generadas por un cristal externo o bien con señales de reloj generadas por una base de tiempo interna. Si las señales de reloj son generadas por la base de tiempo interna, la frecuencia de la señal de reloj está limitada a un máximo de 8 [MHz] y sólo con el uso de cristales se puede trabajar con frecuencias de hasta 16[MHz].

1.11.2. Lenguajes de programación

Históricamente las instrucciones que debe ejecutar un microprocesador se comenzaron a escribir en código máquina, el cual es un conjunto de datos binarios que el microprocesador puede

interpretar directamente. Sin embargo, el desarrollo de programas en código máquina era una tarea muy compleja y que consumía mucho tiempo para realizarse, por estas razones se desarrolló el lenguaje ensamblador. El lenguaje i7 ensamblador se creó para facilitar la programación de microcontroladores usando un conjunto de palabras llamadas mnemónicos, las cuales corresponden a una instrucción en código máquina. Este lenguaje permitió que el desarrollo de programas se optimizara, mas la existencia de varios lenguajes ensamblador (prácticamente uno para cada microcontrolador) y la complejidad de programas con rutinas muy extensas llevaron a que se desarrollaran los compiladores. Un compilador es un programa que permite traducir códigos escritos en lenguajes de alto nivel (p. ej. lenguaje C) a código máquina. La implementación de compiladores facilitó la creación de programas, pues algunas rutinas que en lenguaje ensamblador resultaban ser muy extensas se reducían con la utilización de funciones o estructuras de datos escritas en lenguajes de alto nivel. No obstante que se mejoró la creación de programas, la utilización de lenguajes de alto nivel aumenta el tiempo en el que se ejecuta una acción; esto se debe principalmente porque para interpretar las instrucciones del lenguaje de alto nivel se requieren realizar varias operaciones. Aunque el código máquina, el lenguaje ensamblador y el lenguaje de alto nivel son equivalentes, cada uno presenta sus ventajas y desventajas, por lo cual la implementación de un tipo de lenguaje dependerá de las necesidades del proyecto donde se usará el microcontrolador.

1.11.3. Descripción del software de desarrollo a usar

El microcontrolador con el que se trabajará se programa con un compilador diseñado por la empresa que fabrica el circuito integrado. Dicho compilador permite la programación del mismo usando los lenguajes ensamblador y C. Uno de los problemas al usar éste, y otros compiladores ofrecidos por los fabricantes de microcontroladores, es que la programación de microcontroladores es compleja. Una solución al problema antes mencionado fue ofrecida por un grupo de desarrolladores de software libre que crearon una interfaz que hace más amigable la creación de código para algunos de los microcontroladores de la compañía Atmel. La ventaja principal de usar esta interfaz es que en el desarrollo del código, la detección y la corrección de errores es más rápida. Por ésta y otras razones se optó por usar dicha interfaz a la cual también se le nombrará como software de desarrollo.

En este capítulo se han explicado someramente los temas necesarios para comprender el desarrollo y funcionamiento del sistema *Cycle-DAQ*. En el siguiente capítulo se explicará el procedimiento que se siguió para mejorar al sistema *Cycle-DAQ*.

DESCRIPCIÓN DEL PROYECTO

En este capítulo se explicará la estructura del sistema Cycle-DAQ, así como las etapas que se llevaron a cabo para integrar el hardware y software que lo componen.

2.1. Descripción general del sistema

El equipo que se desarrolló es un sistema de adquisición de datos autónomo, el cual se encargará de registrar los siguientes parámetros vehiculares: la posición (latitud, altitud, longitud e inclinación), la velocidad, la distancia recorrida y las emisiones contaminantes; la información adquirida por el sistema se ocupará para generar ciclos de manejo. Para poder medir dichas variables, el sistema contará con un sistema GPS, el cual proporcionará la fecha, la hora, la longitud, la latitud y la altitud. Si el GPS no está recibiendo información válida o no puede enlazarse, un reloj de tiempo real proporciona la fecha y la hora. Por otro lado, un acelerómetro proporcionará la inclinación del vehículo y una unidad analizadora de gases (que se conecta al sistema mediante una interfaz RS-232) las emisiones de gases en el escape de un vehículo bajo prueba. La información que se adquiera se procesará con el microcontrolador, se guardará en una memoria extraíble, se mandará vía GSM-GPRS a un servidor y algunos datos de la información adquirida se desplegarán en la pantalla táctil. Además de mostrar la información adquirida, la pantalla táctil permitirá la interacción entre el usuario y el sistema para poder configurar a este último. En la figura 2.1 se muestra el diagrama de bloques del sistema y la forma en que interactúan las partes que forman al sistema *Cycle-DAQ* que se desarrolló.

Cabe mencionar que toda la información recabada durante las pruebas (la guardada en la memoria extraíble y la almacenada en el servidor) se ingresará a una aplicación desarrollada en LabView® que permitirá desarrollar ciclos de manejo.

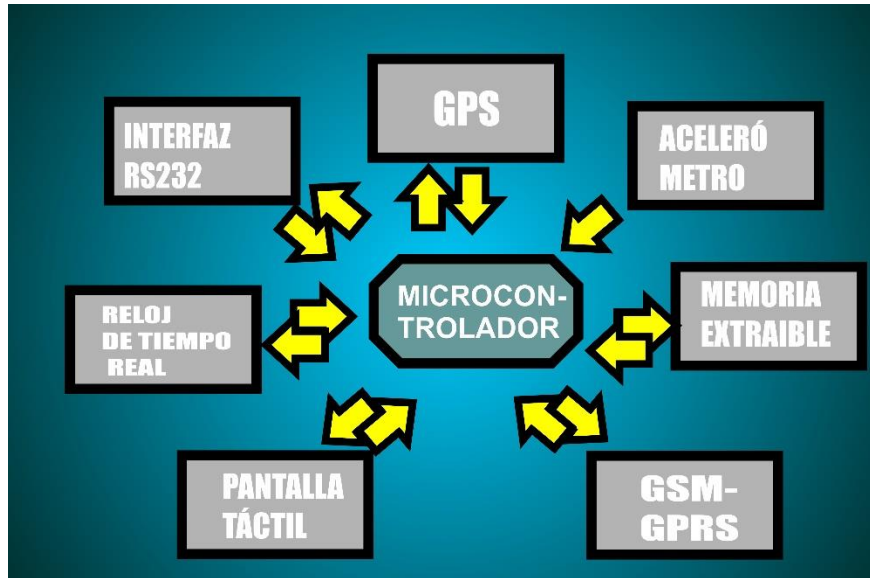


Figura 2.1. Diagrama de bloques del Cycle-DAQ.

2.2. Desarrollo del sistema

El desarrollo del *Cycle-DAQ* se llevó a cabo en varias etapas, las primeras etapas sirvieron para conocer y familiarizarse con cada una de las partes que forman al *Cycle-DAQ*. Una vez finalizadas las pruebas mencionadas se realizaron ensayos para integrar las partes y con ello construir el sistema. En los siguientes apartados se describen cada una de las etapas seguidas para desarrollar el sistema.

2.2.1. Microcontrolador

La parte del *Cycle-DAQ* con la que se comenzó a trabajar fue con la del microcontrolador. Con el fin de poder elegir el mejor microcontrolador para el sistema en desarrollo, primero se buscaron y eligieron los componentes que pudiesen cumplir con los requerimientos del proyecto. La información recabada permitió conocer el número de puertos que se requerían, el tipo de protocolos de comunicación con los que debía de contar el circuito integrado, los niveles lógicos de tensión eléctrica que debía soportar el microcontrolador, entre otros.

Conocidas las características que debía de poseer el microcontrolador, se procedió a buscar el circuito integrado más adecuado a las necesidades del proyecto. Cuando se hubo elegido el microcontrolador, se comenzó a estudiar el lenguaje de programación del microcontrolador y se desarrollaron programas sencillos para familiarizarse con el funcionamiento del circuito integrado. Cuando se consideró que se tenía la suficiente experiencia para programar rutinas complejas, se decidió empezar a comunicar el microcontrolador con las otras secciones del *Cycle-DAQ*.

2.2.2. Receptor GPS

El primer componente con el que se decidió comunicar al microcontrolador fue con el receptor GPS. El receptor GPS empleado en las primeras pruebas de comunicación fue uno que se tenía disponible, del cual se tenía una idea general de su funcionamiento y del que se sabía que se conectaba a otros componentes mediante comunicación serie asíncrona.

Antes de comenzar a programar una rutina para el receptor GPS se buscó y se revisó la hoja de datos del componente. En ésta se encontró la tensión de alimentación del receptor, la velocidad de la comunicación serie asíncrona y el tipo de información que el receptor entrega. Después de esta búsqueda se escribió la rutina mostrada en la figura 2.2.



Figura 2.2. Diagrama de flujo de la primera rutina para el receptor GPS.

En la figura 2.2 se observa que el primer paso del programa consiste en iniciar la comunicación del microcontrolador con el receptor GPS. La inicialización se realiza haciendo que el bloque de comunicación serie asíncrona del microcontrolador esté a la misma velocidad que el bloques con los que se comunica, en este caso 4800 [Bd] para el receptor GPS. Es de suma importancia que los dispositivos que se conectan mediante comunicación serie asíncrona estén a las mismas velocidades, de no ser así la información intercambiada se interpretará como basura.

Una vez establecidas las velocidades de comunicación, se reserva espacio en la memoria de datos del microcontrolador para almacenar la información entregada por el receptor GPS, es decir que se crean variables para guardar datos. Tras esto se comienza a leer la información dada

por el receptor GPS, dependiendo del tipo de información recibida se puede saber si el receptor GPS está conectado y recibiendo información de los satélites GPS. Cuando el receptor no está enlazado a la red GPS, el microcontrolador lee información basura. En caso de que el receptor esté enlazado entrega cadenas de información GGA, GSA, GSV, RMC, VTG y GLL (correspondientes al protocolo NMEA 0183). Con la información recibida se obtiene exclusivamente la cadena GGA que se guardará en una de las variables creadas. Independientemente si es información basura o la cadena GGA, se mostrará el resultado en una ventana de comandos.

La rutina de la figura 2.2 es muy sencilla pero permite conocer si el receptor GPS está trabajando correctamente; sin embargo, la cadena GGA se debe procesar para poder obtener los valores individuales de la longitud, la latitud, la altitud, la fecha y la hora; es necesario obtener por separado los primeros tres parámetros para poder calcular la distancia recorrida y la velocidad. Además de separar la cadena en sus valores individuales, la rutina 2.2 se puede mejorar si adicionalmente se discriminan las cadenas con información no válida (por ejemplo, aquellas que no contengan información sobre las coordenadas del equipo). La rutina con estas tres mejoras (separación de la cadena GGA, cálculo de distancia y velocidad y eliminación de cadenas no válidas) se muestra en la figura 2.3.

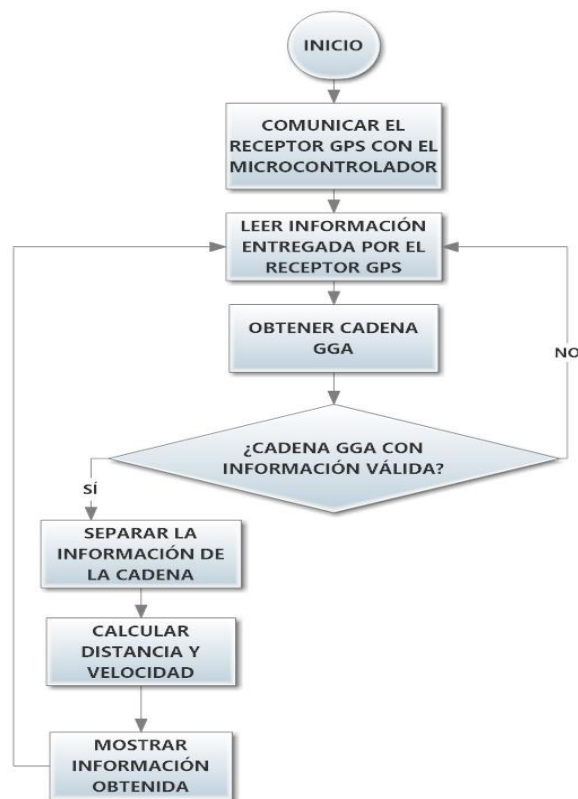


Figura 2.3. Diagrama de flujo de la rutina completa para el receptor GPS.

En la figura 2.3 se observa que una vez que se ha validado la cadena GGA, ésta se separa en las componentes de interés, se calculan otros parámetros y se despliega la información en una pantalla de comandos. La información mostrada en la pantalla de comandos son los parámetros de interés y los datos calculados exclusivamente.

La rutina de la figura 2.3 se puso a prueba durante tres días, tras los cuales se concluyó que la rutina era funcional, pero se descubrió que el receptor GPS requería de 10 a 25 minutos para poder enlazarse a la red satelital GPS, esto aunque el dispositivo se encontrase en cielo abierto. Cuando se descubrió la falta de sensibilidad se buscaron otras opciones de receptores, tras la búsqueda se encontró un receptor con características superiores al modelo utilizado y que era compatible con las conexiones del receptor GPS con el que se contaba. El nuevo receptor GPS se puso a prueba con las rutinas antes mencionadas y en ambas se obtuvo una conexión más rápida con los satélites GPS, por lo cual se optó por usar el segundo receptor.

2.2.3. Acelerómetro

En la búsqueda de información sobre cómo medir la posición angular de un cuerpo, se encontró que se pueden usar giroscopios o acelerómetros. Elegir uno u otro para medir variación angular depende principalmente del tipo de desplazamiento que se requiere medir. En nuestro caso se eligió el acelerómetro, dado que éste ofrece la información necesaria para conocer la inclinación del *Cycle-DAQ* con respecto a la superficie terrestre.

Cuando se definió que se usaría un acelerómetro se buscó información para conocer y determinar las características con las que debía de contar el acelerómetro. De acuerdo con el *Manual Normativo del Programa de Asistencia Técnica en Transporte Urbano para las Ciudades Medias Mexicanas Tomo IV* las carreteras mexicanas no sobrepasan los 20 grados de inclinación vertical, además de acuerdo a Gad-el-Hak (2006) para medir la inclinación de un vehículo se requiere de acelerómetro de tres ejes que cuente con un rango de ± 2 g, considerando estos datos se buscó el acelerómetro más adecuado.

El acelerómetro seleccionado, además de cumplir con los requisitos de medición, entrega sus medidas de manera digital mediante el protocolo de comunicación I²C. Debido a que en el protocolo I²C se requieren conocer las direcciones de los dispositivos, así como de los registros se procedió a leer la hoja de datos del acelerómetro e información en otras fuentes para comunicar al microcontrolador con el acelerómetro. Usando la información adquirida se comenzaron a programar diferentes rutinas en las cuales se probó la conexión acelerómetro-microcontrolador. La rutina final se muestra en la siguiente figura.

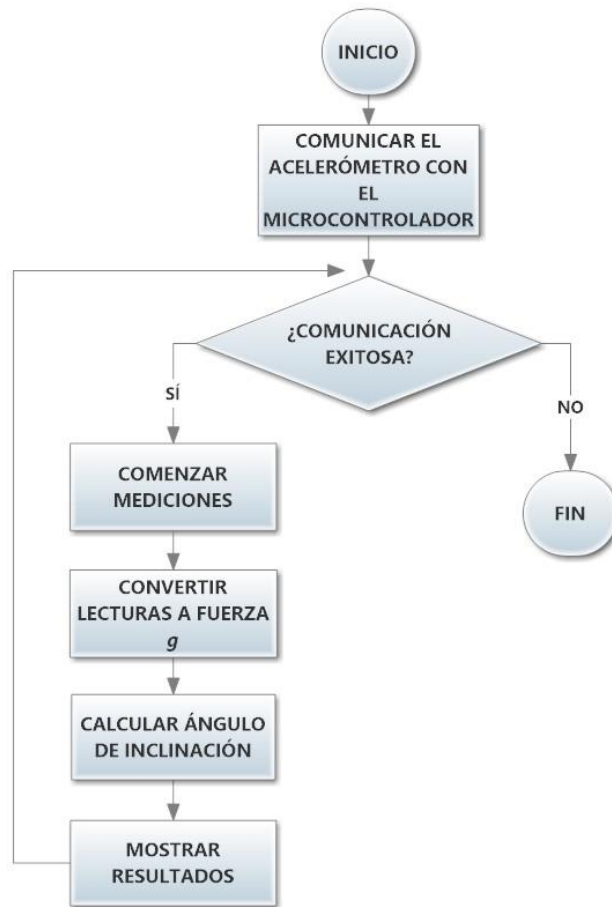


Figura 2.4. Diagrama de flujo de prueba de acelerómetro

En la figura 2.4 se puede ver que se inicializa la comunicación entre el microcontrolador y el acelerómetro, dicha comunicación se establece mediante el protocolo I²C. Mediante el protocolo I²C se mandan secuencias de bytes para preguntar si el acelerómetro está presente, si el acelerómetro no responde con el bit de reconocimiento el programa termina. Sin embargo, si el acelerómetro responde con el bit de reconocimiento se mandan nuevas cadenas para adquirir la información del acelerómetro. La información retribuida por el sensor sólo representa un valor numérico proporcional a la aceleración medida en cada uno de sus ejes, este número no representa fuerza *g*. La conversión del número obtenido del acelerómetro se debe multiplicar por un factor de escalamiento dado por el fabricante. Una vez hecha la conversión a fuerza *g* se utilizan estos valores en ecuaciones trigonométricas y se obtiene la inclinación en grados. Los valores en grados obtenidos se presentan en una ventana de comandos y la rutina vuelve a iniciar.

2.2.4. Analizador de gases

El analizador de gases que trabajará con el *Cycle-DAQ* fue seleccionado por los representantes del LCE. Los representantes del LCE determinaron que se usara un analizador de gases portátil,

con el cual ya habían realizado pruebas y cuyo funcionamiento lo conocían muy bien. Debido a que en el LCE ya se habían realizado diversas pruebas con el analizador de gases, ellos nos proporcionaron una rutina para poder comunicar al analizador con un microcontrolador similar al elegido para el presente tema de tesis.

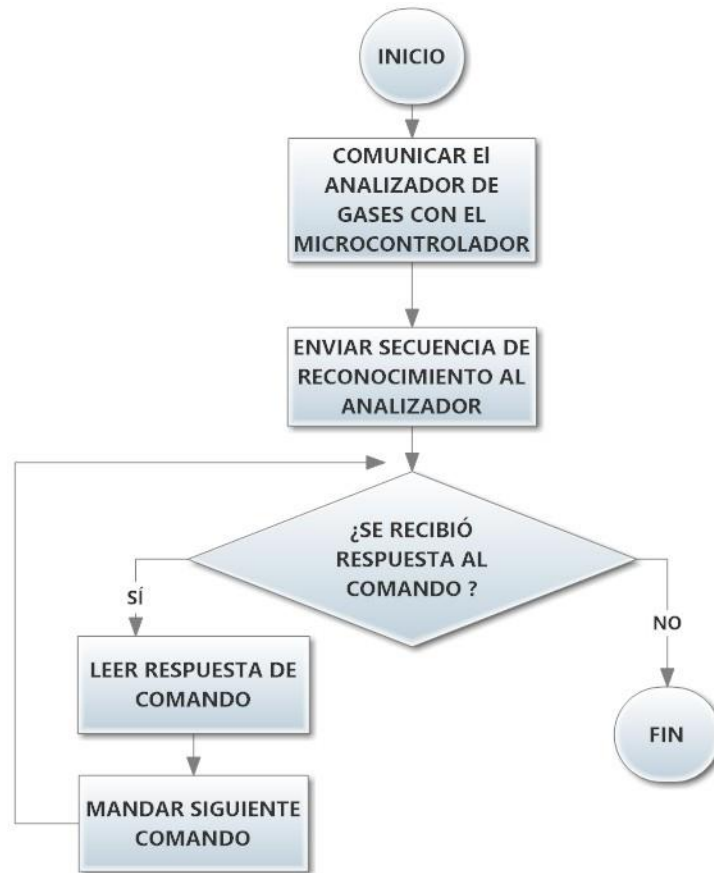


Figura 2.5. Diagrama de flujo del analizador de gases.

En la figura 2.5 se observa que el primer paso es comunicar al microcontrolador con el software de desarrollo y con el analizador de gases, en este caso ambas comunicaciones se realizarán a través de protocolos serie asíncronos. Cabe mencionar que la banca analizadora de gases trabaja con niveles lógicos del protocolo RS-232 por lo cual se requirió usar una interfaz de hardware para adecuar los niveles de tensión eléctrica a los que el microcontrolador puede soportar. Una vez que se han inicializado las comunicaciones se mandan cadenas de bytes para poder comenzar a trabajar con el analizador de gases. Además de la cadena de inicialización se pueden enviar cadenas para configurar, calibrar y obtener mediciones del analizador de gases (las cadenas de bytes están descritas en el manual de usuario del analizador de gases). Independientemente del tipo de cadena que se envíe al analizador de gases éste responde con una cadena. Si se envían cadenas de bytes y el analizador de gases no manda una respuesta significa que el analizador de gases no está funcionando correctamente, cuando esto sucede se termina la comunicación con el

analyzer de gases. Es importante mencionar que a la rutina proporcionada se le hicieron las adecuaciones necesarias para que fuese compatible con el microcontrolador del *Cycle-DAQ*.

2.2.5. Reloj de tiempo real

El *Cycle-DAQ* obtendrá información de vehículos en movimiento y es posible que las características de algunos de los lugares donde se realicen las pruebas (p. ej., zona boscosa o túneles) causen la pérdida del enlace de la señal de GPS. Previendo esto se propuso incluir un reloj de tiempo real para que proporcionara la estampa de tiempo, y así mantener el orden cronológico de la información adquirida (inclinación del vehículo y emisiones contaminantes). Lo que significa que el RTC operará siempre que el receptor GPS no esté enlazado con los satélites de posicionamiento.

Para elegir un RTC, entre la amplia gama que hay en el mercado, se compararon las características de diversos relojes. En la tabla 2.1 se muestra el cotejo de los cuatro dispositivos cuyas características mejor se adaptaban a las necesidades del proyecto, siendo uno de los cuatro el dispositivo que se eligió para ser implementado.

PARÁMETRO	ELEGIDO	OPCIÓN 1	OPCIÓN 2	OPCIÓN 3
Encapsulado	TSSOP	TSSOP	TSSOP	SO
No. De terminales	8	20	20	20
Tensión de alimentación	4.5 - 5.5V	2 - 5.5V	1.7 - 5.5V	2 - 5.5V
Tensión Entrada 1 Lógico (mín.)	2.2 - 7.3V	2 - 7.3V	3.5 - 7.3V	3.5 - 7.3V
Tensión Entrada 0 Lógico (máx.)	0.3 - 0.8V	0.3 - 0.8V	0.3 - 1.5V	0.3 - 1 V
Tensión Salida 1 Lógico (mín.)	1.6V	1.6V	2.4V	2.5V
Tensión Salida 0 Lógico (máx.)	0.4V	0.4V	0.4V	0.4V
Corriente de alimentación (máx.)	1.5mA	1.28mA	600uA	700uA
Corriente de Salida (máx.)	1.5mA	4 mA	3mA	3mA
Tensión de batería	2 - 3.5V	2.2 - 5V	1.3 - 5.5V	2 - 3.8V
Corriente alimentación en modo batería	800nA	400nA	700nA	2.5uA
SPI/I2C	I ² C	SPI	SPI	SPI
Velocidad de transmisión	100KHZ	2MHZ	4MHZ	4MHZ
Compensación año bisiesto	Sí	Sí	Sí	Sí
Precisión de Reloj	±1 min/mes	±1 min/mes	±1 min/mes	±1 min/mes

Tabla 2.1. Comparación de relojes de tiempo real.

Las primeras dos características enlistadas en la tabla 2.1 corresponden a las dimensiones físicas de los componentes (encapsulado y número de terminales). En la tabla se muestra que el dispositivo elegido tiene un encapsulado TSSOP (*Thin-Shrink Small Outline Package*,

Encapsulado de Contorno Pequeño Delgado-Reducido) de ocho terminales y con ello el dispositivo más adecuado para diseñar una tarjeta de circuitos impresos de un tamaño reducido.

Aunque el tamaño del componente fue fundamental para la elección del dispositivo a implementar, se verificó que las características eléctricas de cada elemento fuesen compatibles con las del sistema *Cycle-DAQ*. Respecto a los parámetros enlistados cabe comentar lo siguiente:

- **Tensión de alimentación:** Este parámetro corresponde a la tensión eléctrica requerida para que el circuito integrado trabaje. En la tabla se muestra que el componente elegido tiene un rango de tensión de alimentación más limitado que el de sus competidores, sin embargo la tensión de alimentación del dispositivo seleccionado se encuentra dentro de los rangos que se manejan en el sistema y en este proyecto no se ocupan otros niveles de tensión.
- **Tensión de entrada:** La tensión de entrada es el valor mínimo que se reconocerá como un 1 lógico y el valor máximo que se identificará como 0 lógico. En la tabla se exhibe que estos parámetros son muy similares para los cuatro componentes y todos ellos se encuentran dentro de los límites que se manejan en este proyecto.
- **Tensión de salida:** La tensión de salida es el parámetro que indica cual es el valor mínimo de voltaje que el circuito integrado entregará para representar un 1 lógico y el máximo para representar un 0 lógico. En ambos casos los dispositivos tienen valores similares entre sí y todos ellos son compatibles con los admitidos por el sistema.
- **Corriente de alimentación:** Este criterio corresponde a la corriente eléctrica consumida por el dispositivo durante su operación. En la tabla se expone que la corriente requerida por el dispositivo elegido es mayor que sus contrarios. No obstante ésta no es significativa comparada con la corriente que consume el equipo en general, por lo que el valor de este parámetro es adecuado.
- **Tensión y corriente eléctrica en modo batería:** Se llama modo batería a la operación de los relojes de tiempo real cuando trabajan con la energía proveniente de una batería de respaldo cuando la fuente principal ha sido desconectada. En el caso de la tensión de la batería de respaldo se muestra que el RTC elegido tiene un rango menor que el de sus competidores, sin embargo, comercialmente las baterías más adecuadas para los RTC son de 3 [V], por lo que se considera que el rango de este parámetro, para el circuito elegido, es adecuado. Por otro lado, en la tabla se evidencia que el consumo de corriente del componente elegido está dentro de los valores intermedios y se considera que el consumo es apropiado para la aplicación.

Tras corroborar las características eléctricas se revisaron los parámetros relacionados con la interacción del RTC y el *Cycle-DAQ*. Es decir que se analizaron características como el protocolo de comunicación, la compensación por año bisiesto y la precisión del reloj, las cuales están relacionadas con las consideraciones a tomar para diseñar el hardware y software correspondientes.

En el caso del componente implementado, se muestra en la tabla que se comunica con otros dispositivos mediante el protocolo I²C, el cual en contraste con el protocolo SPI (usado por los otros dispositivos) es un protocolo lento. No obstante, tiene la ventaja de requerir una cantidad menor de líneas de comunicación y aunque ante SPI sea un protocolo lento, la velocidad de comunicación sigue siendo lo suficientemente rápida para trabajar de acuerdo a las necesidades del equipo.

En cuanto a la compensación del año bisiesto y la precisión del reloj se muestra en la tabla que los valores son iguales para todos los dispositivos. Aunque estos parámetros, en los dispositivos comparados, no aportaron ninguna diferencia, era necesario tomarlos en cuenta para tomar una decisión, pues de ellos dependía la estructuración del software asociado al dispositivo.

Una vez que se analizaron todos los parámetros se tuvo que tomar una decisión eligiendo el dispositivo cuyas prestaciones fuesen más adecuadas al proyecto. Además de las razones técnicas, las cuales se describieron conforme se presentaron los parámetros comparados, se tomaron en cuenta el precio y la disponibilidad del componente en el país. En el caso de los relojes de tiempo real, el precio y la disponibilidad en el país coadyuvaron para seleccionar el circuito integrado. Cabe comentar que el ejercicio de comparación descrito se realizó similarmente para todos los dispositivos involucrados en el proyecto.

Cuando se hubo elegido el circuito integrado con el que se trabajaría se comenzaron a realizar pruebas para aprender a comunicarse con el reloj, es decir leer y escribir la fecha y hora almacenada en el dispositivo. Tras varias pruebas, y cuando se hubo consolidado el conocimiento para trabajar correctamente con el reloj de tiempo real, se diseñaron las rutinas de operación normal del RTC.

En la figura 2.6 se muestran los diagramas de flujo de tres rutinas diferentes, las cuales son las rutinas desarrolladas para el reloj de tiempo real. Como se presenta en la figura, estas rutinas no tienen una secuencia entre sí y esto se debe a que cada una se utiliza en una sección diferente de la operación del equipo.

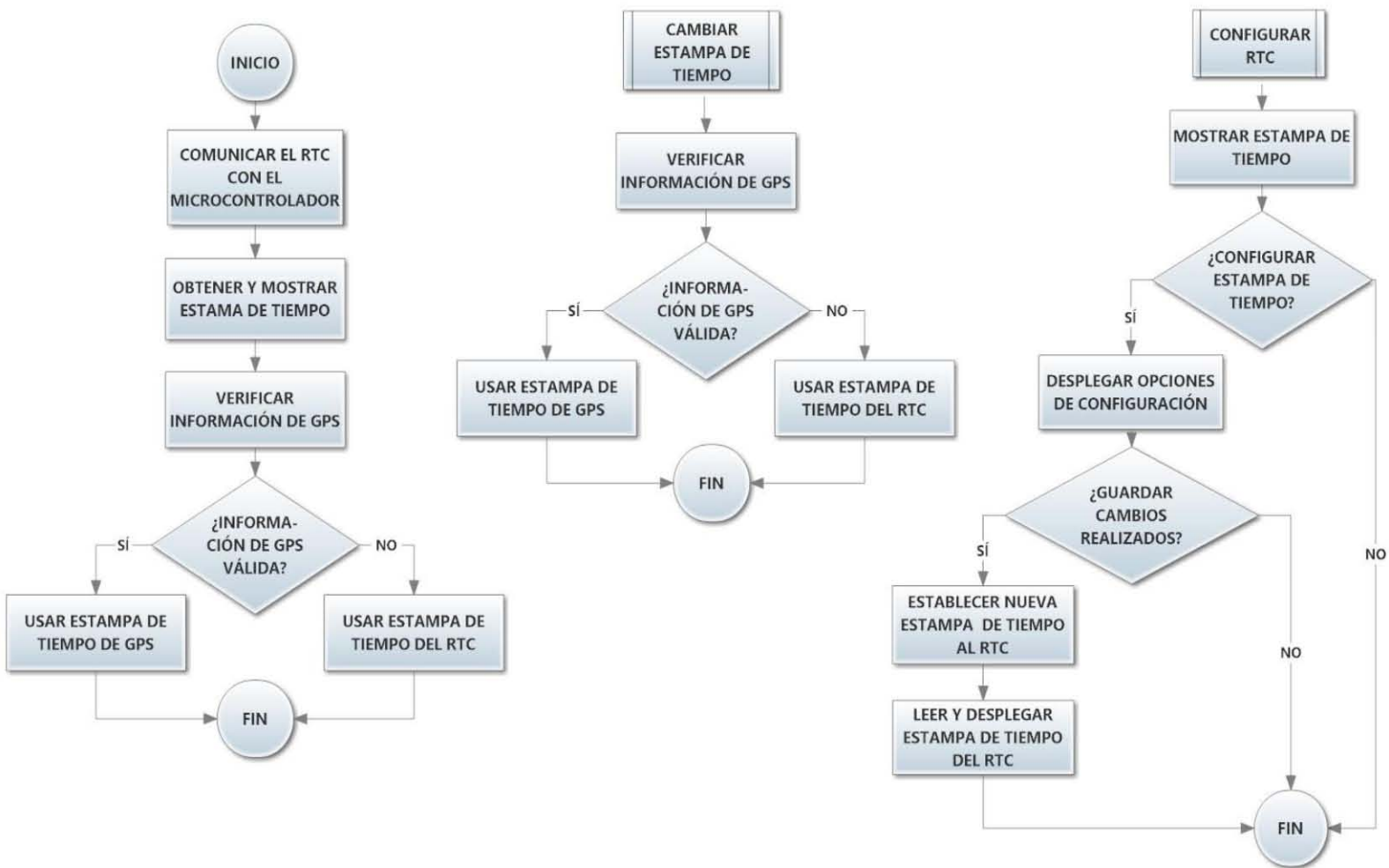


Figura. 2.6. Rutinas de reloj de tiempo real.

La primera rutina que se ejecuta es la mostrada a la izquierda de la imagen y ésta se pone en funcionamiento en la sección donde se realizan las tareas de inicialización del sistema (para mayores detalles respecto al orden de ejecución de las rutinas se recomienda consultar la sección dedicada a la pantalla táctil). Lo primero que se lleva a cabo es establecer la comunicación entre el microcontrolador y el reloj de tiempo real mediante el protocolo I²C, una vez asegurada la comunicación se recupera de los registros del RTC la fecha y la hora en los formatos dd/mm/aaaa y hh:mm:ss respectivamente. Los datos obtenidos se muestran, tal cual, en la pantalla táctil para informar al usuario los valores iniciales del reloj. Tras esto se procede a comprobar si el receptor GPS está entregando información válida o no, y dependiendo del resultado establecer cual dispositivo proporcionará la estampa de tiempo. Se hace hincapié en el hecho de que esta rutina se lleva a cabo junto con otras para preparar el equipo para obtener mediciones. Entonces, cuando el equipo comience las mediciones éstas tendrán asociadas la estampa de tiempo entregada por el dispositivo seleccionado durante la inicialización.

El diagrama de flujo del centro muestra una subrutina llamada *Cambiar estampa de tiempo*. Ésta se ejecuta antes de obtener cualquier medición y su función principal es la de determinar de cual dispositivo se obtendrá la estampa de tiempo para la medición. Esto debido a que cada los datos adquiridos deben tener asociada una estampa de tiempo, y con el fin de mantener la secuencia de tiempo, el sistema conmuta entre la estampa de tiempo del reloj de tiempo real y la del receptor GPS. La conmutación se llevaba a cabo considerando la validez de la información proporcionada por el receptor GPS. Si la información es válida, siempre se tomará la estampa de tiempo del GPS. Por el contrario, si la información no es válida siempre se usará la del RTC. Se recuerda que la información GPS no es válida cuando no tiene el formato establecido por la NMEA y esto normalmente sucede cuando el receptor no está enlazado a la red GPS.

La tercera rutina se llama *Configurar RTC* y corresponde a las acciones que se llevan a cabo para configurar la estampa de tiempo del RTC. Esta subrutina se ejecuta cuando el sistema está operando en el modo configuración y se elige configurar la estampa de tiempo del RTC. Cuando se inicia la ejecución de esta subrutina se presenta la estampa de tiempo actual con la que cuenta el reloj y se espera a que el usuario confirme que desea modificar la estampa de tiempo. Si el usuario decide no modificar la información del reloj la subrutina acaba. Mientras que si el usuario decide modificar la información se activarán las opciones para cumplir la tarea deseada. Una vez que el usuario ha terminado la configuración debe de confirmar las modificaciones, los nuevos valores se actualizarán en el reloj de tiempo real, para corroborar el éxito de la operación se mostrarán los cambios en pantalla y se finalizará la rutina.

2.2.6. Pantalla táctil

Dos características con las que debe de cumplir el sistema *Cycle-DAQ* son: mostrar información y permitir la interacción del usuario con el sistema. Inicialmente se pensó que la forma más viable para cumplir con ambas características era empleando una pantalla gráfica LCD en blanco-negro para desplegar la información y una botonera para lograr la interacción entre el usuario y el equipo. Conforme se fue trabajando con esta idea se observó que la implementación de la tecnología propuesta implicaría dificultades en la construcción de un equipo portátil, tales como: espacio insuficiente para distribuir adecuadamente los componentes, se requerían muchas adecuaciones para instalar los componentes en un gabinete y se necesitaba mucho cableado. Con el fin de dar solución a estas problemáticas se buscaron mejores opciones, siendo la más adecuada la utilización de una pantalla táctil. Cabe mencionar que la elección de este tipo de tecnología se debe a que las pantallas táctiles están conformadas por una pantalla LCD a color con una membrana táctil superpuesta a la pantalla (formando un único dispositivo), lo cual hace posible que el espacio, las adecuaciones necesarias y el cableado requerido para montar el dispositivo en un gabinete se reduzca considerablemente. Además de que con un sólo componente se cubren las dos características mencionadas al inicio del párrafo.

La pantalla táctil que se decidió utilizar es un LCD completamente a color, con una resolución de 320 por 240 píxeles y una membrana táctil de tipo resistiva. Este dispositivo está colocado en una tarjeta de circuito impreso, en la que también se encuentra una ranura para alojar una tarjeta de memoria y los puertos necesarios para comunicarse con los elementos de la tarjeta. Debido a que la tarjeta es el intermediario entre la pantalla, la membrana táctil y la ranura se tuvieron que hacer pruebas para familiarizarse con el funcionamiento individual y grupal de la tarjeta. Tras familiarizarse con el equipo se obtuvieron las siguientes conclusiones:

- **La unidad mínima de trabajo es el pixel:** Toda información que se desee mostrar en pantalla se construye modificando el color de cada uno de los píxeles en un área determinada. Cabe recordar que cada pixel está formado por tres subpíxeles de colores rojo, verde y azul y que el color de un pixel está determinado por la combinación de cada uno de éstos.
- **Cada pixel es independiente:** Esta característica implica que al modificar el color de un pixel, cualquiera que sea su posición en pantalla, no se alterará el color de ningún otro pixel.
- **La membrana táctil es independiente de la información en pantalla:** Cada vez que se toca sobre la membrana táctil se obtienen las coordenadas del lugar donde se llevó a cabo el toque, las coordenadas se basan en un sistema de referencia fijo y propio de la membrana. Es decir que no importa la orientación que tenga la información en pantalla el marco de referencia de la membrana táctil no se verá afectado.
- **La tarjeta de memoria es independiente de la pantalla táctil:** Como se mencionó, la tarjeta donde está la pantalla táctil cuenta con una ranura para alojar tarjeta de memoria microSD®. Cuando se introduce una memoria en la ranura, no importa la información

que contenga la tarjeta de memoria, ésta no afectará el funcionamiento de la pantalla táctil ni viceversa.

Estos cuatro puntos definieron las pautas para desarrollar el código asociado a la pantalla táctil y, dado que este dispositivo es la interfaz del equipo con el usuario, también definieron, en algunos aspectos, el comportamiento general del equipo. Cabe mencionar que la mayor parte del código asociado a la pantalla táctil está orientado a la interacción con el usuario, por esta razón y para tener una mejor comprensión del software se explicará el funcionamiento normal del *Cycle-DAQ*.

Cuando el sistema se enciende en pantalla se despliega una imagen de presentación, la cual permanece durante 5 segundos. Pasado ese tiempo se despliega en pantalla una segunda imagen, sobre la que se despliega el estado del equipo (que dispositivos están conectados, el brillo de la pantalla, la fecha y la hora del RTC) y una leyenda en la que se le pide al usuario que toque la pantalla si desea evitar que el equipo intente conectarse a la red celular. El equipo permanecerá aproximadamente 20 segundos esperando el toque sobre la pantalla. En caso de que el usuario toque la pantalla dentro del tiempo establecido, se desplegará la imagen correspondiente al modo de adquisición de datos (modo en el que se obtiene la información del vehículo bajo prueba y se guarda) y se mostrarán los valores de las mediciones realizadas por el equipo. Por otro lado, si no se tocó sobre la pantalla se desplegará una imagen y el estado de la conexión a la red celular (conectado o desconectado). Si se ha conectado a la red el sistema entra al modo de adquisición de datos. En este modo el equipo permanecerá indefinidamente, recopilando y guardando nueva información cada segundo. Cuando se han obtenido 60 mediciones el equipo intentará enviar la información de la medición a través de la red celular, tras lo cual se reiniciará la cuenta y se tendrán que obtener 60 nuevas mediciones para volver a enviar información por este medio. Cabe comentar que en la imagen de fondo del modo de adquisición de datos hay un ícono que dice salida. Cuando el usuario presiona sobre dicho ícono, el equipo finaliza el modo de adquisición de datos y se despliega el menú principal. En tal menú se muestran dos íconos, uno de ellos está asociado al modo de adquisición de datos y el otro a las opciones de configuración. Cuando se presiona sobre el ícono del modo de adquisición se realizan las acciones, descritas con anterioridad, mientras que si se presiona sobre el ícono de configuración se presentará en pantalla el menú de configuración. En éste hay cuatro opciones disponibles: conectar el equipo a la red celular, configurar el reloj de tiempo real, modificar el brillo de la pantalla y regresar al menú principal. Si el usuario presiona sobre la opción para conectar el equipo a la red celular pueden suceder dos cosas: 1) Sólo se despliega la leyenda *Conectado* o 2) El equipo muestra en pantalla la imagen y realiza las mismas rutinas para conectar al equipo como si el equipo se estuviera inicializando. Por otro lado, si el usuario presiona sobre la opción para configurar el reloj de tiempo real, se desplegará un submenú en el que se le dan al usuario las diferentes opciones para configurar tanto la hora como la fecha del reloj de tiempo real. Así mismo, si se toca sobre la opción de brillo, el brillo de la pantalla podrá aumentarse o disminuirse a gusto. Finalmente si el usuario presiona sobre la opción de regresar al menú principal, hará que el equipo termine con los procesos asociados al menú de configuraciones y se mostrará el menú principal.

En la figura 2.7 se muestra el diagrama de flujo correspondiente a las primeras actividades que se ejecutan con la pantalla táctil. Como se muestra en la figura, la primera acción que se lleva a cabo es establecer la comunicación entre la pantalla táctil y el microcontrolador (mediante el

protocolo SPI). Realizado lo anterior se procede a utilizar la subrutina *Desplegar Imagen* (el funcionamiento de esta subrutina se explica más adelante) para mostrar en pantalla la imagen llamada *Primera_img* (todas las imágenes que se colocan en pantalla se muestran posteriormente), la cual sirve como una presentación del sistema en la que se da información sobre las instituciones involucradas en el desarrollo del sistema, el nombre del equipo y la versión de software. Mientras esta imagen está en pantalla, se crea un archivo de texto llamado *ARCXX.txt*, del cual se comenta lo siguiente:

- El archivo se almacenará en la tarjeta de memoria y tendrá como fin guardar la información recopilada por el equipo.
- Cada vez que se prende el equipo se crea un archivo nuevo, tal archivo sólo podrá ser utilizado en la sesión en la que fue creado.
- El nombre que se asigna a los archivos está conformado por el prefijo ARC y un número entre 00 y 99 (estos números sustituyen a la doble equis). El nombre es irrepetible y se crea siguiendo una numeración ascendente.

Tras crear el archivo se vuelve a ocupar la subrutina *Desplegar Imagen* para mostrar la imagen *Segunda_Img* la cual, a diferencia de la imagen anterior, sólo servirá como un fondo de pantalla. En cuanto se termina de cargar la imagen, el microcontrolador ejecuta rutinas para detectar cuales dispositivos (receptor GPS, acelerómetro, reloj de tiempo real, etc.) están conectados a él y muestra el resultado en pantalla, indicando el nombre del dispositivo y si está conectado o no; a esta información se le conocerá como estado del equipo. En seguida, se consulta si se ha tocado la pantalla táctil y dependiendo del resultado el equipo llevará a cabo alguna de las siguientes acciones. Si no se tocó sobre la pantalla táctil, se mostrará en pantalla la imagen *Tercera_img*, la cual funge como un fondo de pantalla y sobre ella se escribirá el estado de la conexión a la red celular (conectado o desconectado) y permanecerá en esta sección hasta que el equipo quede conectado. Una vez enlazado a la red el equipo desplegará la imagen *Datos_img*. Por otro lado, si se tocó la pantalla, el equipo inmediatamente mostrará la figura *Datos_img*. Antes de continuar con la descripción del software, se debe aclarar que cuando se mencione que se consulta si la pantalla táctil fue tocada o no, significará que el microcontrolador se comunicará con un circuito integrado, asociado a la membrana táctil, para conocer la información que éste contiene y así saber si la membrana fue tocada o no, es decir que se realizan *poleos* para conocer el estado de la membrana táctil.

Siguiendo con el diagrama de la figura 2.7, se puede ver que la acción que se lleva a cabo tras cargar la imagen *Datos_img* se llama *Quitar de la pantalla la información anterior*. Esta actividad consiste en borrar de la pantalla la información de las mediciones anteriores. Sin embargo, cuando no se han obtenido mediciones, por ende no se han desplegado en pantalla, no se observará ningún cambio en pantalla. Tras eliminar la información de la pantalla, el equipo ejecuta la siguiente acción, que consiste en obtener la información más reciente, obtenida por los demás componentes del sistema. Todos los datos obtenidos se agrupan en una variable, la cual se guarda en el archivo de texto *ARCXX.txt* correspondiente. Al finalizar el almacenamiento, se muestra en pantalla los datos obtenidos, Inmediatamente se aumenta en una unidad y se revisa el

valor de una variable que está asociada al número de ciclos¹⁴ que se han llevado a cabo. Como se vé en la figura 2.7, si el valor de la variable es menor a 60 se iniciara nuevamente el ciclo y en cada nuevo ciclo se incrementará el valor del contador hasta llegar a sesenta. Cuando se alcancen los 60 ciclos se mandará a través de la red celular la última cadena de datos obtenida a un puesto central, al terminar el envío se pondrá en cero al contador y el ciclo se repetirá indefinidamente.

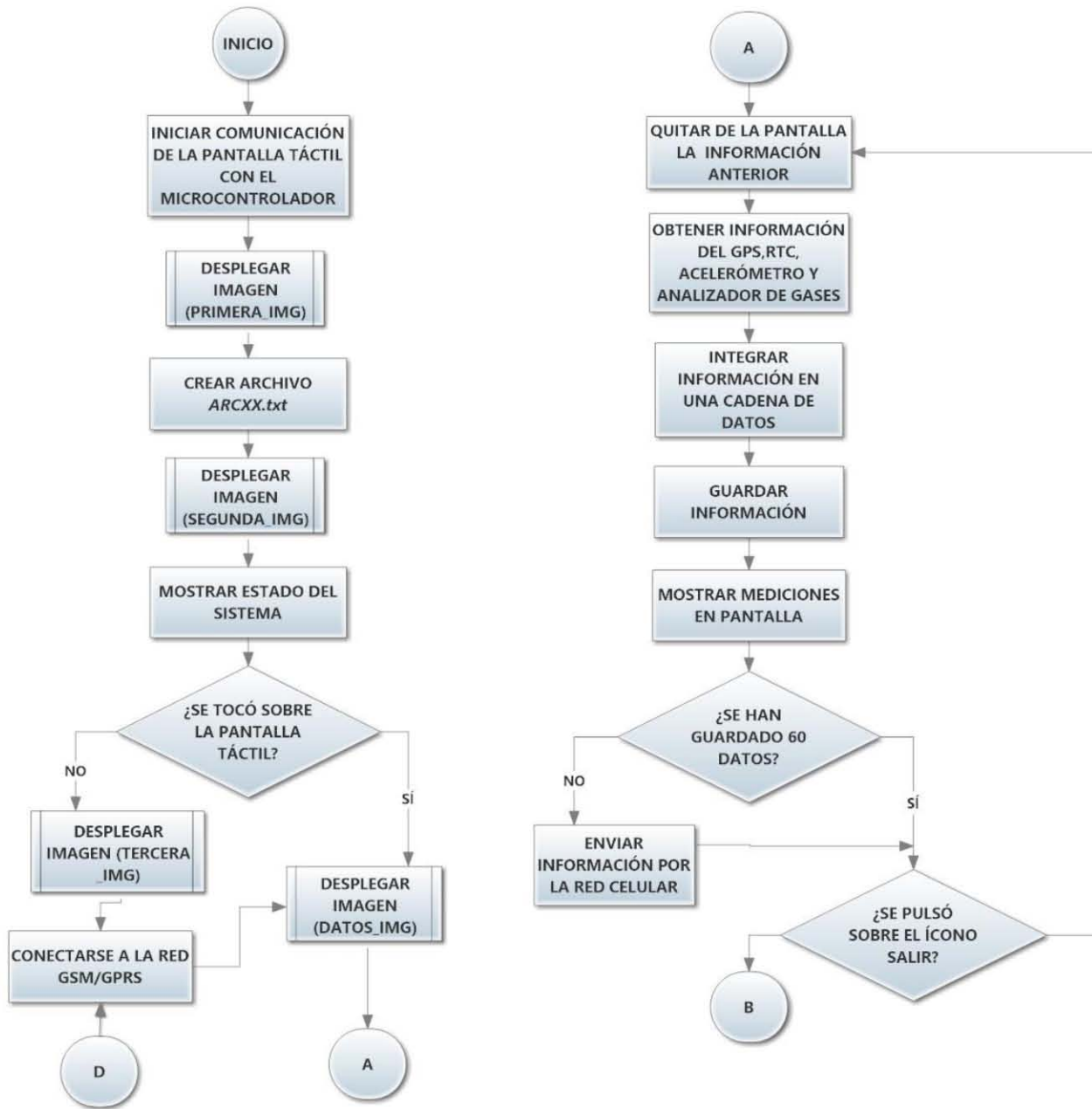


Figura 2.7. Diagrama funcionamiento de la pantalla táctil (Parte 1).

¹⁴ Un ciclo está formado por las acciones de borrar de pantalla la información, obtener nuevos datos, guardar la información y mostrar en pantalla la información adquirida.

Cabe comentar que todas las actividades descritas tras colocar en pantalla la imagen *Datos_img* constituyen lo que hemos denominado como el modo de adquisición de datos y todas las actividades fundamentales del *Cycle-DAQ* se llevan a cabo en este modo. No obstante lo anterior, y como se muestra en la figura 2.7, el microcontrolador consulta si el usuario ha presionado sobre un ícono que indica salida antes de reiniciar con el ciclo. Si se ha presionado sobre tal ícono el equipo realizará las acciones mostradas en la figura 2.7, mientras que si no se registró ningún toque se reiniciará el ciclo normalmente. Se debe aclarar que la información sobre si tocó la pantalla táctil se almacena en el circuito integrado asociado a la membrana y se renueva en cada ciclo.

Antes de continuar con la descripción del software se debe tener en cuenta que la figura 2.7 y 2.8 son complementarias y ambas forman el diagrama de funcionamiento de la pantalla táctil.

En la figura 2.7 se puede ver que cuando se detectó el toque sobre el ícono salir, lo primero que se hace es colocar en pantalla la imagen *Menu_img* y se comprueba si la pantalla ha sido tocada. En caso de que no se haya tocado la pantalla, se establece un ciclo para conocer si en algún momento se toca la pantalla. Cuando se registra un toque sobre la pantalla se llevan a cabo acciones para determinar la sección donde se realizó el toque. Dependiendo de donde se llevó a cabo el toque, se podrá regresar al menú de adquisición de datos, ingresar al menú de configuraciones o bien regresar al ciclo para esperar otro toque. Cabe comentar que mientras la imagen *Menu_img* esté desplegada en pantalla, se conocerá a esta sección cómo menú principal.

Como se puede ver en la figura 2.8, cuando se presiona sobre el área de configuraciones, el equipo despliega la imagen *Ajustes_img* y tras cargar dicha imagen se comenzará un proceso similar al descrito para la imagen *Menu_img*. Es decir que el equipo esperará un toque y una vez que se lleve a cabo un toque, se ejecutarán las acciones asociadas a la sección donde se tocó. En est caso se determinaron cuatro diferentes áreas, las cuales son:

- **Conectar a la red celular:** Al presionar sobre esta sección se inquiera si el equipo ha sido conectado a la red celular con anterioridad. Si el equipo no se ha conectado a la red celular, se regresará a la sección donde se carga la imagen *Tercera_img*, figura 2.7, y se llevaran a cabo todas las acciones subsecuentes descritas.
- **Modificar el brillo:** Al pulsar sobre esta opción el usuario podrá incrementar o reducir el brillo de la pantalla de acuerdo a sus necesidades.
- **Salir del menú de configuraciones:** Si se toca sobre este ícono, el equipo regresará al menú principal.
- **Configuración del reloj de tiempo real:** Cuando se presiona sobre esta área el equipo ingresa a un submenú en el que se podrá modificar la fecha y la hora del reloj de tiempo real.

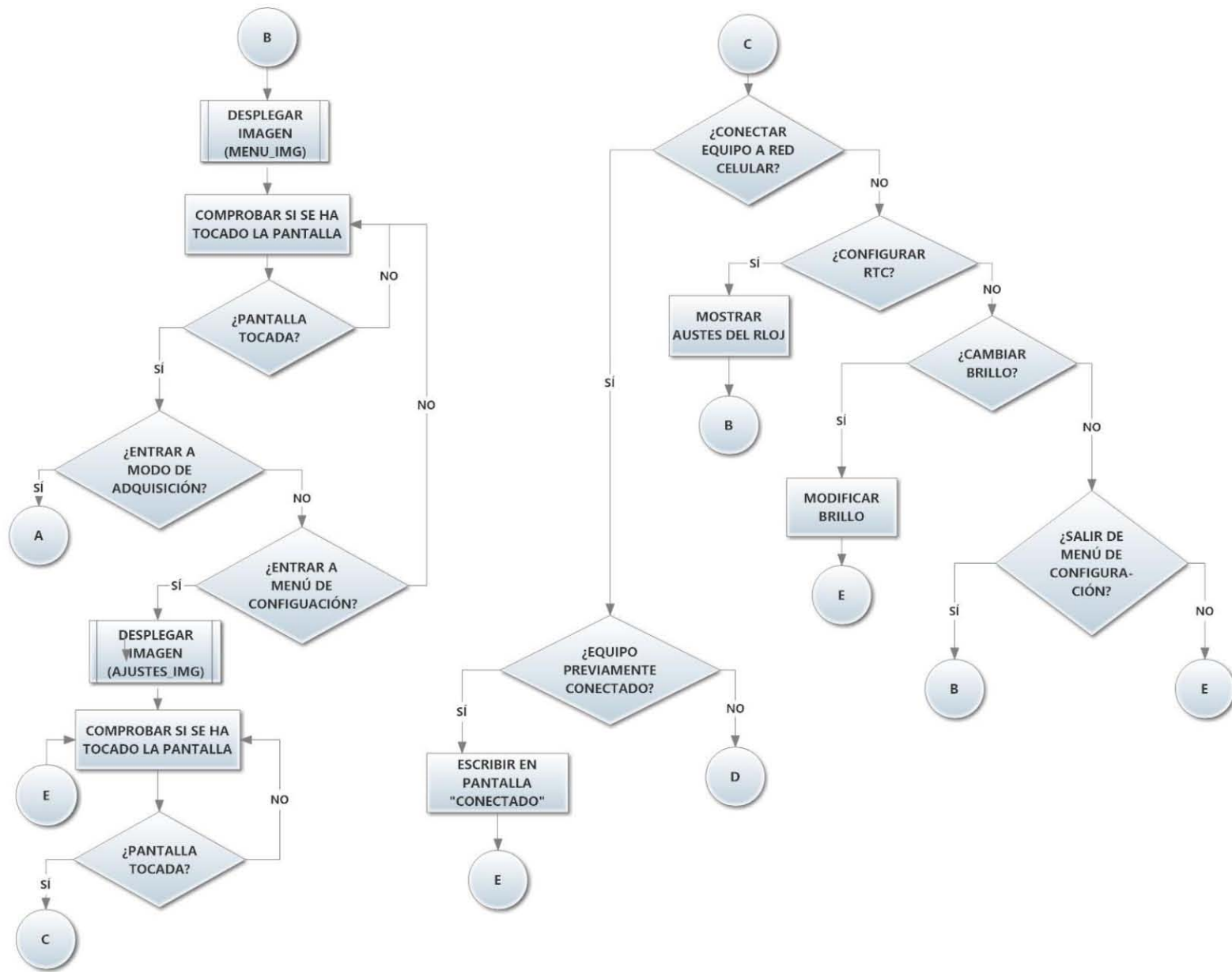


Figura 2.8. Diagrama funcionamiento de la pantalla táctil (Parte 2).

Mientras la imagen *Ajustes_img* se encuentre en pantalla, se considerará que el equipo está en el modo de configuración. Es de mencionar que todos los cambios realizados en el menú de configuración son almacenados de manera automática.

A lo largo de la descripción del software de la pantalla táctil se mencionó la utilización de una subrutina llamada *Desplegar Imagen*. En la figura 2.9 se muestra el diagrama de flujo del funcionamiento de la subrutina. Como se puede ver en el diagrama, al invocar esta subrutina se requiere indicar el nombre de la imagen que se desea desplegar, con el nombre dado se iniciará a buscar la imagen. La búsqueda se realiza dentro de una carpeta alojada en la tarjeta de memoria. Cuando la imagen es encontrada se comienza a leer la información de la imagen y conforme se va leyendo se va asociando la información a un pixel de la pantalla. Es relevante recordar que una imagen no es más que un archivo que contiene información sobre el orden y los colores de los pixeles que la forman. Por otro lado, cuando se realiza la búsqueda y la imagen no se encuentra, se procede a poner la pantalla en color negro en sustitución de la imagen. Indistintamente de si se cargó la imagen o se puso en negro la pantalla, al terminar de asignarle a los pixeles un color la subrutina termina y permite continuar con los demás métodos. En otras cuestiones, en la figura 2.10 se muestran todas las imágenes desplegadas en el sistema, cada una identificada con su nombre. Cabe comentar que en la figura 2.10 hay una imagen llamada *Reloj_img*, de la cual no se ha discutido, tal imagen se despliega cuando se ingresa al menú de configuración del reloj de tiempo real.

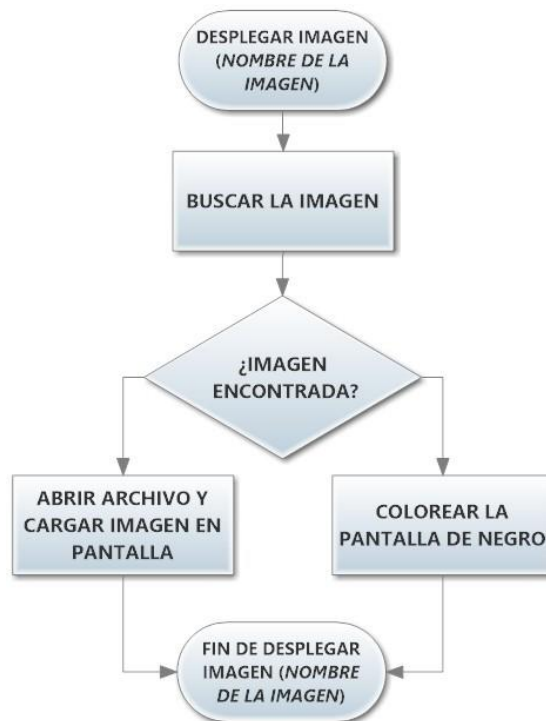


Figura 2.9. Diagrama de flujo de rutina para mostrar imagen.



Figura 2.10. Imágenes desplegadas en pantalla.

Antes de finalizar esta sección se mencionan algunos puntos de interés para complementar la descripción del software realizado:

- Debido a que la tarjeta de memoria alojará los datos recopilados del equipo, no se podrá operar el sistema si no hay una tarjeta de memoria funcional colocada en la ranura correspondiente.
- La tarjeta de memoria colocada debe de almacenar las imágenes mostradas en la figura 2.10, esto para la operación óptima del equipo.
- Se hace hincapié en el hecho de que el software de la pantalla táctil define el comportamiento general del equipo. A su vez el software del equipo está limitado por las cuatro primicias enunciadas. Ejemplo de esto último se puede corroborar con la capacidad de escribir letras sobre una imagen sin alterar el contenido de la imagen o bien la capacidad de guardar y extraer información de la tarjeta de memoria sin afectar a la pantalla ni a la tarjeta de memoria.

2.2.7. Medio de almacenamiento

Existen diferentes medios de almacenamiento disponibles en el mercado, dentro de los medios de almacenamiento que más auge han tenido en los últimos años están las memorias flash, las cuales tienen entre otras características la capacidad de almacenar una gran cantidad de información en un espacio reducido y no son tan sensibles a las vibraciones mecánicas como los medios magnéticos u ópticos.

La clasificación de memorias flash incluye varios tipos de medios, entre los que destacan (por ser los más comerciales) las tarjetas de memoria MS[®] (*Memory Stick*), CF[®] (*Compact Flash*), SD[®] (*Secure Digital*), XD[®] (*Extreme Digital*), M2[®] (*Memory Stick Duo*) y las llaves de memoria USB[®] (*Universal Serial Device*). Con el fin de elegir el medio de almacenamiento más adecuado al proyecto, las memorias flash antes mencionadas se analizaron desde varios puntos de vista, considerando para ello la capacidad de almacenamiento, la disponibilidad en el mercado, los precios y los requerimientos técnicos para poder almacenar información. Tras el análisis de la información técnica de las memorias, se determinó que las tarjetas SD eran el medio más adecuado para trabajar con el equipo. Las razones para tal elección se basaron en la tensión eléctrica de trabajo (lógica de 3.3 V), la programación a través del protocolo SPI, sus diferentes tamaños (estándar, mini y micro), la disponibilidad en el mercado y su precio accesible.

El software desarrollado para la tarjeta de memoria está enfocado a cumplir con dos funciones: 1) proporcionar información y 2) guardar información. La primera función a la que se hace referencia nos permite mostrar imágenes almacenadas, en la tarjeta de memoria, en la pantalla. Mientras que la segunda función nos permitirá almacenar los datos adquiridos, durante las pruebas, en un archivo de texto. La rutina desarrollada para proporcionar información corresponde a la subrutina llamada *Desplegar Imágenes*, la cual se detalla su funcionamiento en la sección 2.2.5. *Pantalla Táctil*. En tanto que las rutinas para guardar información se describen a continuación.

En la figura 2.11 se muestran dos subrutinas. La primera de ellas, llamada *Crear Archivo ARCXX.TXT*, tiene la finalidad de crear un archivo en el que se almacenará la información recopilada por el equipo. Esta subrutina se ejecuta durante la inicialización del sistema *Cycle-DAQ* y como se muestra en la figura 2.9 el primer paso que se lleva a cabo es establecer la comunicación entre la tarjeta de memoria y el microcontrolador. Es posible que al realizar este paso se tenga que la comunicación no ha sido exitosa, lo cual se debe principalmente a la ausencia de la tarjeta de memoria o bien algún daño en la tarjeta de memoria. Cuando alguna de las dos situaciones se presenta, el sistema entra en un ciclo para tratar de establecer una comunicación exitosa y tal ciclo se mantendrá hasta que la comunicación microcontrolador-tarjeta de memoria se realice exitosamente. De lo anterior se debe comentar que, antes de que el

equipo ingrese al ciclo referido, se escribe en pantalla una notificación que le indica al usuario que debe revisar la tarjeta de memoria. Una vez que se inserta un dispositivo de almacenamiento en condiciones operativas, se procede a buscar, en una carpeta específica, los archivos previamente almacenados en la tarjeta de memoria y con ello determinar el nombre del nuevo documento. Todos los archivos que están guardados, y los que se crean, reciben un nombre cuya estructura es *ARCXX.txt*, en donde la doble equis corresponde a un número entre 00 y 99. En la figura 2.11 se muestra que la búsqueda de los archivos comienza por el archivo *ARC00.txt* y en caso de que el archivo exista, se inicia una búsqueda cíclica hasta obtener el nombre de un archivo que no se encuentre almacenado. Una vez que se tiene el nombre se crea el documento de texto en la memoria y se termina esta subrutina. Es importante aclarar que la búsqueda se lleva a cabo aumentando, en cada ciclo, una unidad el numeral del último archivo encontrado.

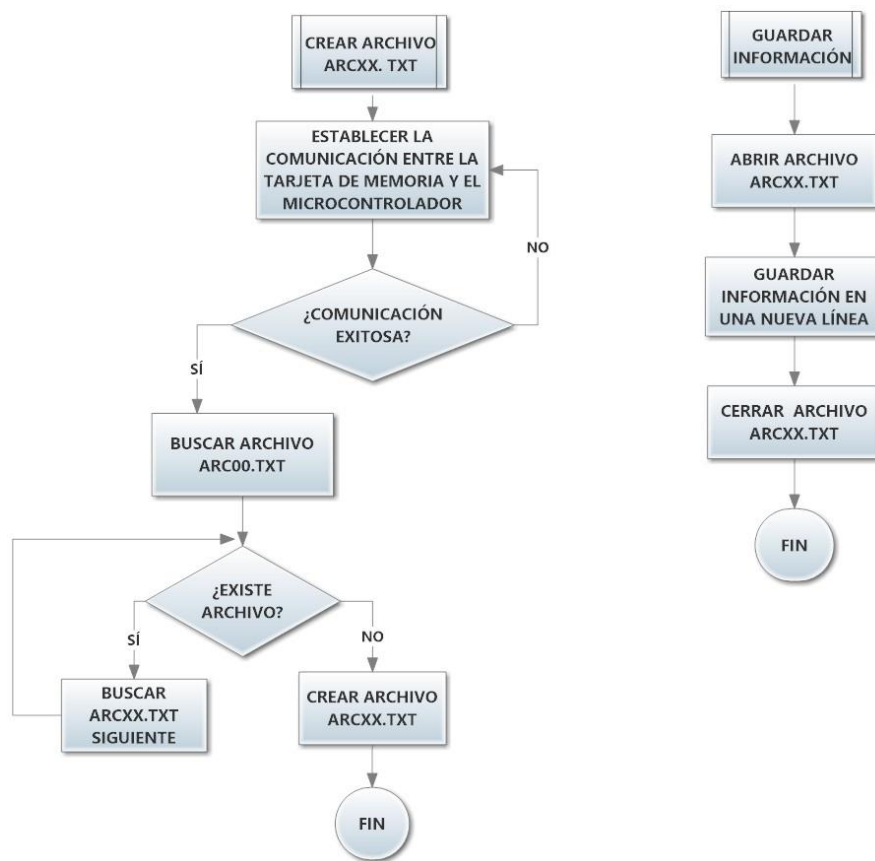


Figura 2.11. Diagramas de flujo de rutinas para almacenar información.

Se debe comentar que cada vez que se inicialice el equipo, es decir se encienda el equipo, se generará un nuevo archivo que sólo será usado en la sesión en que fue creado (hasta que el equipo se apague).

Por otro lado, si al inicializar el equipo se descubre que la carpeta, dedicada al almacenamiento de los archivos de texto, contiene todos los documentos con numerales consecutivos del 00 al 99, el equipo mostrará en pantalla una leyenda para indicarle al usuario esta situación y el equipo no realizará ninguna otra acción. Cuando esta situación se presente y para tener que utilizar una tarjeta de memoria diferente, el usuario deberá de retirar la tarjeta y en un equipo externo tendrá que retirar algunos o todos los archivos almacenados en la carpeta de archivos. Se resalta que retirar los archivos no necesariamente implica eliminar los documentos, basta con que se almacenen en otro medio o se guarde en una carpeta distinta.

La subrutina mostrada en la figura 2.11, llamada *Guardar Información*, se ejecuta exclusivamente cuando el equipo está en el modo de adquisición de datos y permite almacenar la información recopilada por el equipo en el último archivo creado por la subrutina *Crear Archivo*. Como se presenta en la figura, lo primero que realiza esta subrutina es abrir el archivo, lo que significa que se deben emplear variables y funciones especiales para dicho fin. Cuando el archivo se ha abierto, se ejecutan comandos para escribir, en una nueva línea del documento, la cadena con datos de la información recopilada por el equipo. Finalizada la escritura, se procede a cerrar el archivo y con ello termina la subrutina.

Cabe comentar que al comenzar a desarrollar el software para almacenar información se utilizaron tarjetas de memoria SD[®] modelo estándar. Sin embargo, al adquirir la pantalla táctil se cambió el modelo y se ocupó el modelo micro. Esta modificación se llevó a cabo para aprovechar que el módulo de la pantalla táctil contaba con una ranura para alojar una memoria microSD[®]. Tal modificación resultó ser bastante beneficiosa debido a que no se requirieron hacer grandes adecuaciones al software y se ahorró una cantidad considerable de espacio al momento de diseñar la tarjeta de circuitos impresos del sistema *Cycle-DAQ*.

2.2.8. Transceptor GSM/GPRS

La característica distintiva del sistema *Cycle-DAQ* es su capacidad para enviar información, de manera inalámbrica, a un puesto central mientras los datos son recopilados. Para dotar con esta característica al equipo, se estudiaron diversos aspectos de las tecnologías inalámbricas existentes en el mercado, donde se consideraron algunos puntos como los siguientes:

- **Infraestructura:** Se tuvieron que identificar los componentes que integraban la cadena de comunicación como antenas, emisores, receptores, repetidores, entre otros. Así mismo, se estudió la infraestructura que se debía instalar o bien los requisitos para tener acceso a infraestructura ya instalada.

- **Complejidad para enviar información:** Se analizaron los lenguajes de programación que empleaba cada una de las tecnologías, la dificultad para emplear dicho lenguaje y los protocolos requeridos para intercambiar información entre el emisor y el receptor.
- **Cobertura:** Con el fin de conocer la distancia máxima a la que el equipo podría enviar información al puesto central, se compararon las zonas con cobertura que ofrecía cada tecnología.
- **Costo:** Se analizaron las ventajas y desventajas económicas de implementar cada una de las tecnologías considerando la inversión necesaria para comunicar a los emisores y receptores.

Tras el análisis realizado, se determinó que la mejor tecnología de la que se podía disponer era la red celular GSM/GPRS, esto debido a que era la tecnología menos costosa de implementar, con una amplia infraestructura instalada, con una cobertura superior a las demás tecnologías inalámbricas consideradas y con un lenguaje de comunicación de mediana complejidad. Para ocupar la tecnología celular y enviar la información recolectada, se tuvo que adquirir una estación móvil (transceptor) que soportara la comunicación mediante los protocolos GSM y GPRS, y una tarjeta SIM de un proveedor que tuviera el servicio de red GPRS. Es importante mencionar que el servicio de GPRS se emplea para intercambiar información empleando protocolos de redes informáticas como TCP (*Transmission Control Protocol*, Protocolo de Control de Transmisión), IP (*Internet Protocol* Protocolo de Internet) y HTTP (*Hypertext Transfer Protocol*, Protocolo de transferencia de hipertexto). Así mismo, cabe aclarar que la red GPRS es un complemento de la red GSM y por ende no se pueden desligar.

Con la estación móvil y la tarjeta SIM (colocada dentro de la estación móvil) adquiridas se comenzaron a realizar ensayos, los cuales nos ayudaron a entender la manera más adecuada para manejar la estación móvil usando un microcontrolador. En dichas pruebas la mayor parte del tiempo se dedicó a comprender el lenguaje de comunicación de la estación móvil (comandos AT).

Los comandos AT son un conjunto de instrucciones que reconoce el transceptor, reciben este nombre debido a que la sintaxis de la mayoría de las instrucciones involucra las letras *AT* antes de la instrucción. Cada comando AT tiene una función única, específica e independiente de los demás comandos. Tras cada comando enviado la estación móvil regresará una respuesta, las cuales varían de acuerdo al comando enviado pero en general indican si el comando se ejecutó exitosamente o no. Es importante mencionar que hay varias funciones que puede realizar la estación móvil (realizar una llamada, enviar un mensaje, consultar páginas de internet) que requieren que los comandos sean ejecutados en un orden específico.

Así, tras varios ensayos se consolidó la rutina mostrada en la figura 2.12, en la que se muestra que lo primero que se realiza al ejecutar la rutina es comunicar el dispositivo celular (estación móvil) con el microcontrolador, lo cual se realiza mediante comunicación serie asíncrona. Tras esto se comprueba la comunicación y para ello se envían las letras *AT* a la estación móvil. Si la comunicación se realizó exitosamente, la estación móvil mandará como respuesta las letras *OK*, en el caso contrario se recibirán caracteres aleatorios. La respuesta será leída por el microcontrolador y tras ella decidirá si continuará la ejecución de la rutina.

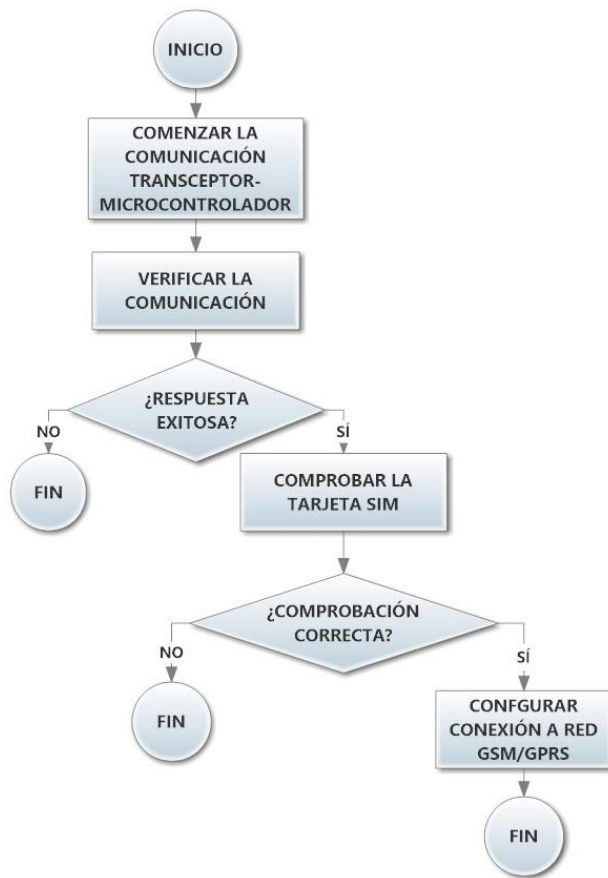


Figura 2.12. Diagramas de flujo de rutinas para almacenar información.

En la figura 2.12 se presenta que al finalizar la comprobación de la comunicación, se procede a verificar que haya una tarjeta SIM instalada. Con tal fin se envía a la estación móvil el comando *AT+CPIN?*. Este comando está diseñado para preguntar si la tarjeta SIM cuenta con un código de bloqueo PIN (*Personal Identification Number*, número personal de identificación), PUK1 (*Personal Unlock Key 1*, número personal de desbloqueo 1), PUK2 o no. No obstante, se utiliza

para saber sobre la presencia de una tarjeta SIM, puesto que si no se encontrase una tarjeta SIM la respuesta del comando corresponderá a un código de error y en el caso contrario se obtendrá el tipo de bloqueo de la tarjeta. Cabe comentar que para el presente proyecto se consideró que sólo se utilizarían tarjetas SIM sin ningún código de bloqueo.

Si tras ejecutar el comando *AT+CPIN?* se comprobó la existencia de la tarjeta SIM, se checa que dicha tarjeta esté conectada a la red celular de un proveedor, para ello se manda el comando *AT+CREG?*. La respuesta de este comando puede ser un número del uno al cinco, si el número de la respuesta es uno o cinco se considera que la tarjeta SIM está registrada en una red celular en cualquier otro caso se considera que no está registrada. En caso de que la tarjeta no se encuentre registrada la rutina termina. Se debe tener en cuenta que tanto el comando *AT+CPIN?* como el comando *AT+CREG?* conforman el bloque de comprobación de la tarjeta SIM y si alguno de los dos no da una comprobación correcta se detendrá la rutina.

El siguiente paso en la rutina mostrada en la figura 2.12, cuando todos los pasos anteriores han tenido respuestas positivas, consiste en configurar la estación móvil para que se enlace a la red GPRS. Para ello se tienen que realizar, en el orden mostrado, las siguientes acciones:

- **Iniciar la conexión al servicio GPRS:** Lo primero que se debe hacer para comenzar a usar la red GPRS, es indicarle a la estación móvil (ME) que debe prepararse para intercambiar información con la red celular usando el servicio GPRS. Con tal fin se manda el comando *AT+CGAT=1*. La respuesta de este comando indicará el éxito o el fallo de la acción.
- **Configurar el perfil de trabajo:** Cuando se han inicializado correctamente los servicios de la ME para conectarse a la red GPRS. Se debe configurar la estación móvil para que se pueda intercambiar información mediante protocolos de redes informáticas antes mencionados. Para ello se debe de elegir uno de los dos posibles perfiles configurables con los que cuenta la ME. Después se debe indicar el punto de acceso (que brindará los servicios de red informática), usuario y contraseña del punto de acceso (ésta información es proporcionada por la compañía que ofrece los servicios de la red celular y finalmente se configuran algunas opciones respecto al intercambio de información. Todo lo anterior se realiza utilizando los comandos
 - a) *AT+QIFGCNT=0* Con este comando se selecciona uno de los dos perfiles configurables con los que cuenta la estación móvil
 - b) *AT+QICSGP=1,APN, USER, PSWD* Con esta instrucción se le indica a la estación móvil que debe conectarse mediante el servicio GPRS al punto de

acceso indicado por el APN (*Access Point Name*, Nombre del punto de acceso), con el nombre de usuario y contraseñas indicadas.

- c) *AT+QIMUX=0* Se le indica a la ME que el intercambio de información se debe hacer por una sola conexión TCP/IP. Para ahondar más respecto a las ventajas o desventajas de abrir una o varias conexiones se recomienda leer el artículo publicado por la compañía *AT&T Multiple Simultaneous TCP Connections*.¹⁵
 - d) *AT+QIMODE=1* Este comando le indica a la estación móvil que cuando se inicie el intercambio de información vía GPRS, los datos que el microcontrolador le enviará a la estación móvil corresponderán únicamente a los datos y no requieren de una validación por medio de otros comando AT.
 - e) *AT+QINDI=1* Con este comando se le indica a la estación móvil que cada vez que vaya a entregar la información que recibió al microcontrolador, como resultado del intercambio de información vía GPRS, debe avisar.
- **Conectarse a la red GPRS:** Tras haber configurado a la estación móvil, se procede a iniciar la conexión GPRS. Cuando se ha iniciado la conexión GPRS, el dispositivo está listo para enviar y recibir información. Se debe recordar que cuando está activa la conexión GPRS la SE ME está reportando en la red celular como un dispositivo activo y sólo cuando hay intercambio de información se le asignan recursos de la red celular. Para comenzar la comunicación se deben ejecutar los siguientes comandos en el estricto orden en que son presentado *AT+QIREGAPP*, *AT+QIACT*, *AT+QILOCIP* y *AT+QIDNSIP=1* y lo que hacen estos comandos es indicarle a la estación móvil que se registre en la red GPRS usando el punto de acceso indicado, tras ello se confirma que la conexión inalámbrica se hará usando el servicio GPRS, después se pide que el punto de acceso asigne una dirección IP a la estación móvil y finalmente se indica que la información que se enviará tendrá como destinatarios direcciones IP.

Respecto a las acciones enlistadas se tienen que tener en cuenta que la ejecución de cada uno de los comandos debe ser en el orden mostrado ya que si no se hace así es muy probable que no se pueda enviar información vía GPRS. Por otro lado, aunque no se muestra en el diagrama mostrado en la figura 2.12, cuando se ejecutan los comandos *AT+CGAT*, *AT+QICSGP* o *AT+QIREGAPP* y la estación móvil responde con un código de error la rutina finaliza. Así mismo se debe comentar que la rutina mostrada en la figura 2.12 se ejecuta dentro de un ciclo que se finaliza cuando el sistema se ha enlazado correctamente al servicio GPRS o bien se han

¹⁵ El recurso podrá ser encontrado en la página web <http://developer.att.com/application-resource-optimizer/docs/best-practices/multiple-simultaneous-tcp-connections>

realizado más de 10 iteraciones no exitosas. Si se llegan a las 10 iteraciones y el equipo no se ha conectado se le notifica al usuario, se pide que reinicie el sistema *Cycle-DAQ* e intente cambiar de zona de conexión.

Cuando la rutina se ejecuta exitosamente en todos sus pasos, el equipo queda preparado para enviar información a través de la red GPRS. El envío de información se realiza únicamente cuando el equipo está en el modo de adquisición de datos y se lleva a cabo empleando el comando *AT+QIOPEN="TCP", DIRECCION_IP, PUERTO* . En donde *DIRECCION_IP* corresponde a la dirección del servidor donde se enviará la información y *PUERTO* corresponde al número de puerto del protocolo TCP a través del cual se intercambiara información. Cuando se ha terminado de enviar toda la información se debe de escribir el comando *AT+QICLOSE*. Para mayor información sobre los comandos AT se recomienda leer la bibliografía de este trabajo de tesis.

2.2.9. Microcontrolador

Los microcontroladores son circuitos integrados conformados por bloques funcionales de procesamiento, memoria, comunicación, con periféricos diversos, convertidores analógicos digitales y digitales analógicos, entre otros, capaces de llevar a cabo tareas previamente programadas. Las tareas que pueden llevar a cabo los microcontroladores, y la complejidad de éstas, dependen exclusivamente de la capacidad y la diversidad de los bloques funcionales que integren al microcontrolador.

En el mercado se ofrecen una variedad de microcontroladores y de ellos el mejor dispositivo es aquel que se adapte mejor a las necesidades de la aplicaciones. En nuestro caso el microcontrolador que se eligió es un microcontrolador de 8-bits, construido bajo la arquitectura Harvard, de conjunto de instrucciones reducidas, con una memoria para programas de 256 [KB], una memoria de acceso aleatorio de 8 [KB], frecuencia de reloj de hasta 16 [MHz], 86 puertos de propósito general de entrada y salida, 32 registros de propósito general, tensión eléctrica de trabajo 5 [V], cuatro bloques de comunicación serie asíncrona, un bloque para comunicación SPI, un bloque para comunicación I²C, un detector de caída en la tensión eléctrica de alimentación, y otros más. Cabe comentar que entre las razones para elegir este microcontrolador se encuentra la diversidad y cantidad de módulos de comunicación (necesarios para las diferentes conexiones), la velocidad de su reloj de trabajo, el número de puertos disponibles y su bajo costo en comparación con otros circuitos integrados.

El microcontrolador es el elemento más importante del *Cycle-DAQ*, ya que éste se encarga de coordinar el funcionamiento de los dispositivos que conforman el equipo, procesar la información que se obtiene y alojar las rutinas que se deben ejecutar. Las rutinas que se describieron en secciones anteriores, además de indicar el comportamiento de los dispositivos, muestran las acciones que realiza el microcontrolador. Debido a esto último, en esta sección no se describirá

una rutina específica para el microcontrolador, en cambio se discutirán algunas cuestiones sobre las conexiones que se implementaron para tener un dispositivo funcional.

En la figura 2.13 se muestra el diagrama de conexiones del *Cycle-DAQ*, en el centro de la figura se representa el microcontrolador y a los alrededores de él todos los componentes que se le conectaron.

Para operar los microcontroladores requieren de una señal de reloj. Entre más elevada se la frecuencia de la señal de reloj, más rápido se ejecutarán las operaciones que lleva a cabo el microcontrolador y mayor es el consumo de corriente. En el caso de nuestro microcontrolador la señal de reloj se puede obtener usando medios internos o externos. Si la señal se obtiene con medios internos, el microcontrolador usará sus recursos (temporizadores) para generar la señal. En cambio si se decide obtener la señal con medios externos, se tendrán que conectar otros componentes al microcontrolador. En la figura 2.13, en la parte central izquierda, se muestra un arreglo capacitor-cristal-capacitor, que el arreglo que requiere nuestro microcontrolador para obtener las señales de reloj por medios externos. Cabe recordar que el arreglo mostrado funciona debido a que internamente el microcontrolador tiene construido un bloque que trabaja como un oscilador tipo Pierce. Esto implica que las terminales a las que está conectado el cristal, los valores de los capacitores y la frecuencias de los cristales compatibles están determinadas por el fabricante. Se comenta que en este caso, las razones para preferir usar los medios de generación de señal de reloj externos sobre los internos fue que las frecuencias de reloj (por ende la velocidad de procesamiento) alcanzadas por los cristales eran superiores a las de los medios internos, además como el microcontrolador tiene terminales exclusivas para conectar el cristal, no impacta en las terminales disponibles para conectar otros dispositivos. Respecto a este último punto se debe comentar que, en algunos tipos de microcontroladores, las terminales para conectar un cristal pueden ser empleadas para realizar otras funciones, en estos casos se tiene que evaluar si es mejor tener mayor velocidad de procesamiento o poder utilizar dos terminales más.

En la mayoría de los microcontroladores hay una terminal llamada reinicio (*Reset*), la cual debe estar en un nivel lógico específico para permitir que el dispositivo trabaje. Si la terminal reinicio no se encuentra en el nivel lógico requerido, el microcontrolador no operará. Esta terminal se llama reinicio porque cada vez que hay una transición de un nivel lógico en el que el microcontrolador está deshabilitado a un nivel en el que sí está habilitado, las rutinas programadas en el microcontrolador se ejecutarán desde el inicio, es decir desde la primera línea programada. En la figura 2.13, también en la sección central izquierda, se puede ver que al microcontrolador se conectan la resistencia R8 y el capacitor C11, los cuales sirven para poner la terminal de reinicio a un nivel alto y con ello habilitar la operación del microcontrolador. Cabe comentar que para poner en el nivel lógico adecuado a la terminal de reinicio, basta con colocar una resistencia entre la terminal y el nivel requerido de tensión eléctrica, sin embargo se incluye el capacitor para evitar que ante caídas de tensión muy cortas el equipo se reinicie, ya que el microcontrolador cuenta con un módulo de detección de caída de tensión eléctrica.

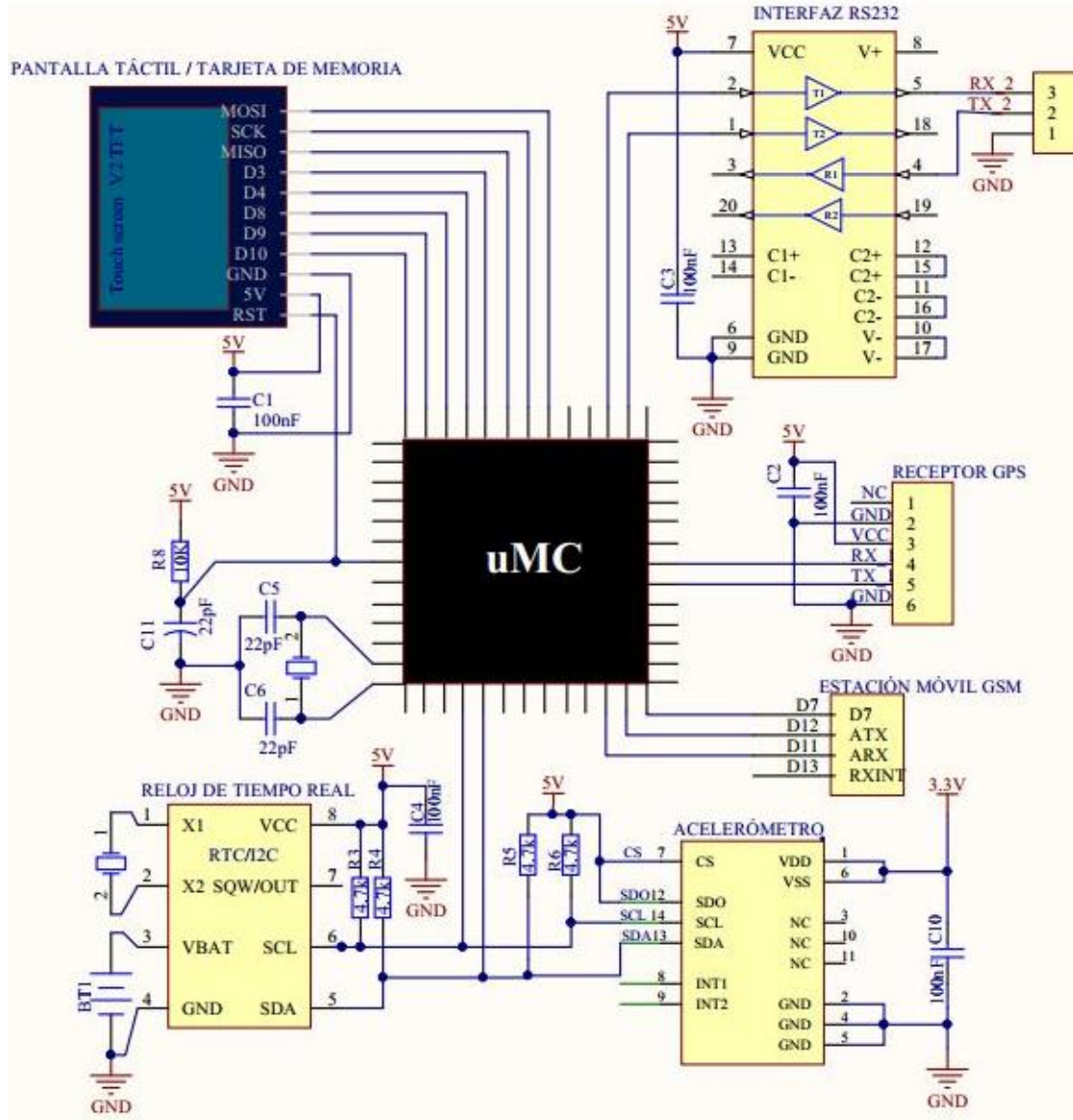


Figura 2.13. Diagramas de conexiones del sistema.

En la parte inferior de la figura 2.13 se representan las conexiones del reloj de tiempo real y del acelerómetro. Estos dos dispositivos comparten dos líneas para comunicarse con el microcontrolador, llamadas SDA y SCL, las cuales corresponden al protocolo I²C. Respecto a estos dispositivos se tiene que tener en cuenta lo siguiente:

- El cristal que está conectado al reloj de tiempo real tiene la función de proporcionar la señal para llevar la cuenta de tiempo, y forma parte de un oscilador tipo Pierce, como el que tiene el microcontrolador, con la única diferencia de que los capacitores empleados en el arreglo capacitor-cristal-capacitor están contruidos internamente.

- La batería del RTC es una fuente de alimentación de respaldo, la cual como se explicó en secciones anteriores, alimenta al circuito cuando la fuente principal falta y con ello se mantiene la cuenta de tiempo del RTC.
- Entre las líneas de SDA y SCL, tanto del reloj como del acelerómetro, y las tensiones de alimentación de los dispositivos (5 [V] para el reloj de tiempo real y 3.3 [V] para el acelerómetro) se conectan un par resistencias, las cuales se conocen como resistencias de *Pull-Up* (levantamiento). La función de estas resistencias es poner en un nivel alto las terminales SDA y SCL con la finalidad de evitar que alguna señal parásita produzca resultados no deseados.
- Al observar la conexión I²C se podría pensar que la tensión de *Pull-up* a la que se conectan las terminales del acelerómetro podrían causar problemas al dispositivo. Sin embargo, las salidas del acelerómetro tienen especificaciones que le permiten manejar las condiciones de conexión mostradas sin riesgo.
- Otro punto que se destaca es que se podría pensar en prescindir del par de resistencias de *Pull-up* del acelerómetro, debido a que la tensión de 5 [V] mantendría las líneas del protocolo del I²C en un nivel adecuado. Sin embargo, se colocaron puesto que la distancia real entre los dos circuitos es suficientemente grande como para considerar la posibilidad de la inducción de señales parásitas.

En la parte derecha de la figura 2.13 se muestran las conexiones de la estación móvil GSM/GPRS, el receptor GPS y la interfaz RS232. Estos tres dispositivos se comunican con el microcontrolador a través de comunicación serie asíncrona. Dado que el microcontrolador tiene tres módulos para comunicación serie asíncrona, cada uno de los dispositivos listados se conecta directamente a un módulo. En secciones anteriores se comentó que para implementar la comunicación serie asíncrona, entre dos sistemas, se debía de establecer la misma velocidad de transferencia de información; si tenía uno o dos bits de parada y si contaba o no con bit de paridad. En nuestro caso, los módulos de comunicación del microcontrolador se configuraron a velocidades de 115200, 4800 y 19200 baudios para conectar los elementos antes citados. En todos los casos la información sólo se envía con un bit de parada y sin bit de paridad. Se recuerda que la comunicación serie asíncrona requiere de exclusivamente dos líneas para trabajar, una de recepción (Rx) y una de transmisión (Tx). Las conexiones de estas líneas se hacen uniendo la línea de recepción de un dispositivo con la línea de transmisión del otro dispositivo y viceversa.

En el caso de la estación móvil GSM/GPRS se unen tres líneas con el microcontrolador, en este caso la línea adicional corresponde a una conexión que le permite al microcontrolador encender o apagar la estación móvil. Cabe comentar que en el diagrama de la figura 2.13 no se muestra la tensión eléctrica con la que trabaja la estación móvil celular, esto se hizo intencionalmente para evitar confusiones, ya que este equipo ocupa una tensión de alimentación de 5[V] así como la de 3.3 [V].

En la parte superior izquierda de la figura 2.13 se representan las conexiones de la pantalla táctil con el microcontrolador. La pantalla táctil se comunica a través del protocolo SPI con el

microcontrolador. El protocolo SPI requiere de cuatro líneas de comunicación para trabajar, las cuales son MISO, MOSI, SCK y CS. Como ya se explicó, MISO y MOSI son las líneas en las que viaja la información en el protocolo SPI; SCK corresponde a la señal del reloj usada en el protocolo (en este caso de 8 [MHz]) y CS sirve para habilitar el dispositivo esclavo con el que se desea comunicar. En este caso se tienen tres dispositivos esclavos con los que se comunica el microcontrolador, los cuales son: la pantalla, la membrana táctil y la memoria. Además de las líneas mencionadas, el microcontrolador ocupa dos líneas más, una para indicar si la información que se le envía a la pantalla son comandos o datos que debe desplegar y la otra línea permite controlar el brillo de la pantalla.

En todos los dispositivos mostrados, salvo la estación móvil, se muestra un capacitor conectado entre la alimentación y tierra. Dicho capacitor tiene la función de reducir el ruido producido por las señales de alta frecuencia y con ello evitar el dispositivo se prenda y se apague por una señal parásita. No se colocó un capacitor a la estación móvil porque ésta ya contaba con uno.

2.2.10. Placa de circuito impreso

En las secciones anteriores se describió el funcionamiento individual de los bloques que conforman el *Cycle-DAQ*. Aunque las rutinas asociadas a cada bloque se desarrollaron de manera individual, todas ellas se diseñaron pensando en que posteriormente se integrarían en una única rutina.

Cuando se terminaron de desarrollar todas las rutinas, se procedió a unir todas las secciones tanto en hardware como en software. La unión de los componentes se fue haciendo progresivamente, es decir primero se realizaron las conexiones pertinentes entre el microcontrolador y dos dispositivos, así mismo las rutinas asociadas de dichos dispositivos se integraron en una sola rutina y el arreglo se puso a prueba. Cuando se tuvo un comportamiento adecuado, al arreglo anterior se le agregó un nuevo dispositivo, la rutina asociada al nuevo componente se integró a la rutina anterior y nuevamente se comprobó el funcionamiento. Así se fueron integrando todos los dispositivos hasta obtener el primer prototipo del *Cycle-DAQ* y una sola rutina. Cabe destacar que para lograr que los dispositivos trabajaran correctamente en conjunto, se tuvieron que hacer cambios y adecuaciones como las mencionadas a continuación:

- **Estructuración de la rutina general:** Como se mencionó, la rutina general se fue construyendo conforme se iban uniendo más dispositivos. Estas rutinas se fueron ordenando de tal manera que su comportamiento se asemejase a lo descrito en la sección dedicada a la pantalla táctil del presente trabajo.
- **Reasignación de terminales:** Cuando se desarrolló el software para cada uno de los dispositivos, se asignaron terminales del microcontrolador a las que se debían de conectar los dispositivos. En algunos casos la asignación de terminales se tuvo que cambiar para favorecer la distribución espacial de los componentes y para evitar que más de un dispositivo se conectara a las mismas terminales del microcontrolador, salvo las conexiones de los protocolos I²C y SPI.

- **Cambio de nombre a las variables:** En algunos casos, los nombres que se les asignaron a las variables se tuvieron que cambiar, pues su nombre y su función llegaban confundirse con las variables de otras secciones.

El último dispositivo que se agregó al *Cycle-DAQ* fue el tranceptor GSM/GPRS, y al igual que se realizó con los demás dispositivos, se realizaron las conexiones necesarias entre el microcontrolador y se integró su rutina con las demás. Sin embargo, cuando se realizaron las pruebas de funcionamiento, se encontró que el prototipo presentaba dificultades para conectarse y mantener comunicación con la red celular. Aunque se intentaron varios ajustes para solventar esta deficiencia, como cambiar las terminales de conexión con el microcontrolador, cambiar su posición en el prototipo y optimizar su código asociado, ninguno dio resultados favorables. A pesar de lo anterior, se decidió comenzar con el diseño de una placa de circuitos impresos (*PCB, Printed Circuit Board*) para el *Cycle-DAQ*. Esta decisión se tomó ya que se pensó que la manera en que se construyó del prototipo estaba afectando el comportamiento de la antena de la estación móvil, y se pensó que si el sistema se instauraba en una PCB el problema quedaría resuelto, pues entre otras cosas se lograría una mejor distribución de los componentes. Así mismo el diseño de la PCB nos ayudaría a cumplir con el objetivo del proyecto de generar una placa de circuitos impresos para el sistema *Cycle-DAQ*.

El diseño de la placa de circuitos impresos se llevó a cabo en dos etapas principales. La primera etapa estuvo dedicada al aprendizaje y a la planeación, ya que se estudiaron reglas generales para el diseño de tarjetas, se aprendió a usar el entorno de desarrollo del programa donde se crearía la placa de circuitos impresos y se proyectó en maquetas la distribución de los componentes para obtener la mejor distribución. La segunda etapa se destinó a desarrollar la placa de circuitos impresos y en dicho desarrollo se reflejaron los conocimientos adquiridos en la primera fase. Los aspectos más relevantes de ambas etapas se resumen a continuación:

- Es altamente recomendable proyectar la tarjeta de circuitos impresos en maquetas. Debido a que de antemano se puede visualizar la conveniencia, o no, de colocar un dispositivo en cierta posición. Además, la buena distribución de componentes permitirá que el diseño de la tarjeta sea más sencillo y rápido. Cabe comentar que la proyección de la tarjeta no tiene que ser muy detallada, en general es suficiente con generar representaciones planas (círculos y rectángulos) de los componentes más relevantes y de los más grandes. Se aconseja que las figuras empleadas tengan dimensiones reales.
- Es importante conocer *grosso modo* la función de cada una de las terminales de los dispositivos implementados. Con ello se logra mejorar la distribución de componentes y permite conocer si hay componentes adicionales que se requieran para hacer funcionar adecuadamente al dispositivo.
- La mayoría de los programas de diseño de tarjetas de circuitos impresos tiene herramientas para la creación de símbolos para componentes y la creación de marcas de soldado (*footprints*). Conocer dichas herramientas y saberlas manejar son de mucha ayuda durante la creación de la tarjeta.

- Los programas de diseño de tarjetas normalmente tienen dos plataformas de trabajo, una dedicada a la creación de diagramas esquemáticos (diagrama con símbolos que representan a los componentes y sus conexiones) y otra para el desarrollo como tal de la placa de circuitos impresos. Para desarrollar una PCB, es recomendable diseñar primero el diagrama esquemático, ya que éste servirá como guía al momento de diseñar la PCB.
- Los programas de diseño de PCB manejan dos unidades de medidas, métricas e inglesas. Aprender a cambiar entre ambos sistemas de medidas facilita el diseño de la PCB. Así mismo la mayoría de los programas maneja cuadrículas que sirven como guías y facilitan la colocación de símbolos, es muy recomendable aprender a cambiar el tamaño de éstas.
- El primer paso en la creación de una placa de circuitos impresos debe ser la delimitación del área de la tarjeta. Posteriormente se deben distribuir los componentes, en el área de la tarjeta, de acuerdo a la distribución planeada o bien buscando las posiciones más convenientes.
- Una vez que se han posicionado los componentes se comienzan a generar las pistas (conexiones) entre los componentes. Una tarjeta de circuito impreso se considera terminada cuando todas las terminales requeridas están conectadas entre sí.
- Se debe conocer como cambiar las reglas de diseño del programa y adecuarlas a las necesidades del proyecto. Se debe tener en especial cuidado en los parámetros de separación mínima entre pistas y el mínimo grosor de pistas.
- Algunas consideraciones que se deben hacer al diseñar una tarjeta de circuitos impresos son:
 - 1) Las rutas entre los componentes deben colocarse buscando que se generen las trayectorias con menor distancia posible.
 - 2) Las trayectorias deben ser lo más rectas posibles. Sin embargo, si se tienen que hacer cambios de dirección en las trayectorias, éstos deberán de hacerse en ángulos de 45° .
 - 3) Debe evitarse generar rutas que tengan cambios de dirección en ángulos de 90° . Es posible conectar dos pistas y que éstas formen ángulos de 90° , siempre y cuando en la intersección la pista se remate con chaflanes en ángulos de 45° a ambos lados de la pista que corta.
 - 4) Las pistas deben tener trayectorias lo más cortas posibles, esto con el fin de que la resistencia, inductancia y capacitancia de la pista no sean significativas. Además se deben generar pistas con un ancho adecuado para soportar la corriente que conducirá la pista, manejar adecuadamente el tipo y velocidad de la señal que conducirá. Elegir un ancho de pista inadecuado puede producir

malfuncionamientos en el circuito, para conocer más sobre las dimensiones recomendadas para las pistas se recomienda leer la norma IPC (*Institute of Printed Circuits*. Instituto de Circuitos Impresos)-2221^a.

- 5) Las pistas con mayor ancho deben ser aquellas que distribuyen la alimentación del circuito.
- 6) Una pista debe mantener su ancho a través de toda su trayectoria.
- 7) Es deseable que la tarjeta cuente con un plano de tierra. Un plano de tierra consiste en un polígono distribuido a lo largo de la tarjeta de circuitos impresos y el cual está conecta a la referencia de tensión eléctrica. La razón de hacer esto es para hacer que se la referencia de tensión eléctrica sea robusta, es decir que no fluctúe su valor.
- 8) Las pistas que lleven señales deben de incluir puntos de prueba. Los cuales se pueden colocar en cualquier parte de la pista, sin embargo se recomienda que se coloquen cerca de las terminales de los dispositivos que generan la señal.
- 9) Cuando se diseñan PCB de dos o más capas, se recomienda que los dispositivos que disipan más calor se coloquen en la cara frontal de la PCB.
- 10) Las terminales de los cristales (para los dispositivos que así lo requieran) deben de colocarse lo más cerca posible a las terminales del circuito integrado que así lo requieran. Las pistas que se dibujen deben ser lo más directas posibles.
- 11) Para mantener claridad en el diseño se recomienda que los componentes pasivos se coloquen lo más cerca posible al circuito integrado correspondiente y de ser posible que todos los componentes pasivos asociados a un circuito integrado tengan la misma orientación.
- 12) Es deseable que la tarjeta tenga cierta estética y simetría. En la mayoría de las ocasiones la estética y la simetría están relacionadas con diseños ordenados y trabajar más adecuadamente que los dispositivos.
- 13) Los símbolos que se colocan en la PCB deben de contar con identificadores. Se aconseja que todos los identificadores tengan la misma orientación.
- 14) Para facilitar el soldado de componentes se recomienda que los *footprints* de los componentes de agujeros pasantes (*through-hole*) sean de forma ovalada y en el caso de los *footprints* de los componentes de montaje superficial, tengan dimensiones ligeramente superiores a las indicadas por el fabricante
- 15) La tarjeta de circuitos impresos debe incluir perforaciones para facilitar su montaje en gabinetes.

- 16) Se recomienda evitar colocar pistas cerca del borde de la tarjeta de circuitos impresos.
 - 17) Se recomienda que todos los circuitos integrados cuente con un capacitor de 100 nF entre las terminales de alimentación y la referencia de voltaje. Dicho capacitor se conoce como capacitor de desacoplamiento, éste cumple con las funciones de mantener el nivel de la tensión de alimentación ante cambios rápidos en la tensión y también disminuye los efectos del ruido producido por señales de alta frecuencia en la alimentación del circuito integrado.
- Si se conoce un experto en el diseño de tarjetas de circuitos impresos, es recomendable presentarle el diseño de la tarjeta y realizar las adecuaciones que el experto considere son necesarias.
 - Es muy importante saber como obtener los archivos de fabricación de la tarjeta en diferentes formatos.

Los puntos antes mencionados son resultado de la experiencia adquirida al trabajar en el proyecto, sin embargo, para ampliar sobre el diseño de tarjetas de circuitos impresos se recomienda leer la norma IPC-2221A y el libro *Fabricating Printed Circuit Boards*, ambos parte de la bibliografía de este trabajo de tesis.

Como se comentó, parte del proceso del desarrollo de la tarjeta de circuitos impresos incluyó la revisión del diseño por parte de un experto. La dinámica que se siguió con el experto fue la de mostrarle el diseño de la tarjeta de circuitos impresos, respecto a esa tarjeta el experto emitía sus recomendaciones y éstas se llevaban a cabo. El circuito impreso corregido se mostraba nuevamente al experto y se implementaban las observaciones que él realizaba. Este ciclo se repitió en cuatro ocasiones hasta que se obtuvo el circuito impreso con el mejor diseño. En la figura 2.14 se muestra una sección de la tarjeta de circuitos impresos, antes de aplicar las observaciones del experto y después de llevar a cabo las correcciones propuestas.

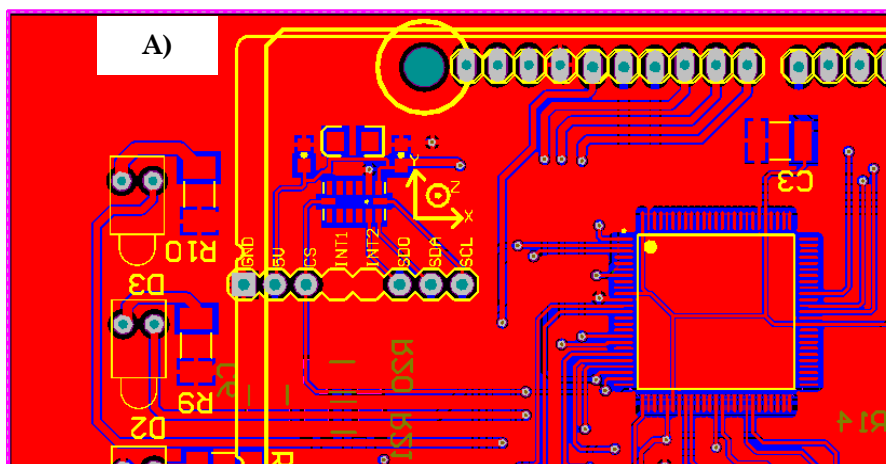
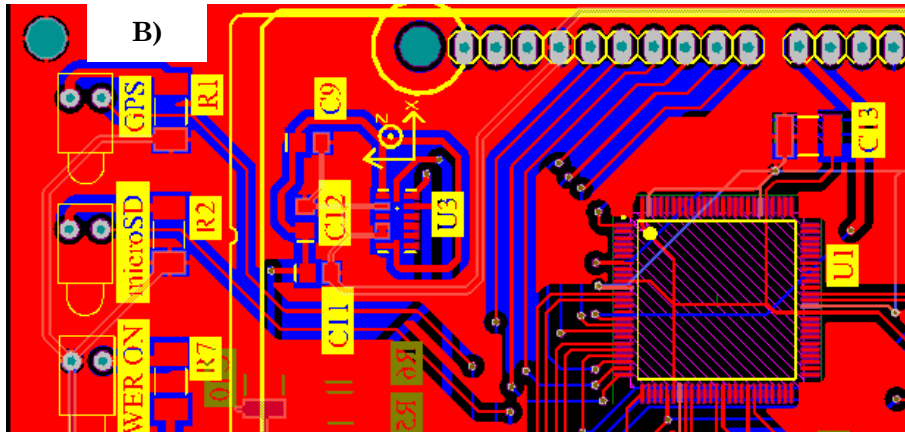


Figura 2.14. Mejoras al diseño de PCB (continúa).



La figura 2.14 está conformada por dos imágenes, que corresponden a la misma sección en la placa de circuitos impresos, siendo la imagen superior la imagen (imagen A) la correspondiente al primer diseño presentado ante el experto y la imagen inferior (imagen B) la correspondiente al diseño obtenido tras aplicar las correcciones emitidas por el experto. Algunas de las cuestiones que se corrigieron se comentan a continuación:

- **El grosor y la ruta de las pistas:** Si se compara la imagen A con la B, se puede ver que las pistas más cercanas al microcontrolador tienen un grosor menor en la imagen B, así mismo algunas de las rutas se cambiaron en la figura. Un claro ejemplo de esto se muestran en las pistas que están en la parte central superior de ambas imágenes (conectadas a las terminales de soldado de un conector de tecnología de agujeros pasantes).
- **Identificadores más claros y con una única orientación:** En la imagen A se exhibe que los identificadores de los componentes (por ejemplo D3, R9 y C3) tienen orientaciones distintas. En cambio, en la figura B, se muestra que todos los identificadores tienen una sola orientación, además se cambió la tipografía utilizada en los identificadores, lo cual se hizo con la finalidad de facilitar la identificación de los componentes.
- **Mejor agrupación de componentes:** Para obtener un mejor diseño de la tarjeta se tuvieron que relocalizar algunos componentes activos y sus componentes pasivos asociados. Un ejemplo de esto se puede ver en el circuito integrado marcado como U3 en la imagen B, al cual se le cambió su orientación respecto a la mostrada en la imagen A. También, se modificó la localización de los componentes C11, C12 Y C9.
- **Adición de perforaciones de fijación:** En la imagen B se muestra que en la esquina superior izquierda se agregó una perforación, la cual es una de cuatro perforaciones que se agregaron (una por cada esquina de la tarjeta) para poder sujetar la tarjeta a su gabinete.
- **Modificación en las propiedades del plano de tierra:** En la figura B se muestra que las propiedades del plano de tierra fueran modificadas para que existiese un espacio mayor entre las pistas y el plano de tierra. Este cambio se hace notorio en el circuito integrado U1 y en las pistas a su alrededor. Este cambio se hizo principalmente para facilitar la fabricación de la tarjeta. Cabe comentar que el achurado que se presenta en el circuito integrado U1 indica que

en esa área no se puede colocar otro componente. Este achurado no se observa en la imagen A, ya que se confunde con el plano de tierra.

Adicionalmente a lo comentado, en la figura 2.14 se puede corroborar que la estética y la simetría en el circuito impreso son indicadores de un diseño más adecuado.

Cuando se tuvo el diseño final de la tarjeta de circuitos impresos, se mandó a fabricar a una empresa especializada. Cuando se nos entregó el trabajo terminado se corroboró que la construcción de la tarjeta correspondiera a lo diseñado, para ello se utilizó un multímetro y se comprobó que hubiera continuidad entre los extremos de las pistas de acuerdo al diseño original. Cuando se constató que la fabricación había sido adecuada se procedió a soldar componentes.

En nuestro caso la primera sección que se soldó fue la sección de alimentación, después el microcontrolador, su cristal, capacitores y puerto de programación. Una vez que se tuvo el microcontrolador soldado, éste se programó con una rutina sencilla para corroborar que estuviese trabajando adecuadamente, comprobado esto se soldaron los demás dispositivos. Los otros dispositivos se fueron soldando uno a uno. Cuando se concluía el soldado de un dispositivo éste se comprobaba comunicándolo con el microcontrolador, una vez que se tenía buena comunicación con el microcontrolador se tomaba otro dispositivo y se repetía el proceso, hasta que se hubieron soldado todos los dispositivos. Cabe comentar que en este procedimiento sólo se verificaba el funcionamiento individual de las partes, el comportamiento en conjunto del sistema se dejó para etapas posteriores. Por otro lado, en el proceso de soldado se aprendieron diversas cuestiones, a continuación se enlistan los puntos más relevantes:

- Si no se tiene mucha experiencia soldando componentes se recomienda practicar en tarjetas de aparatos descompuestos. Es recomendable que se practiquen técnicas de soldado con cautín y con pistola de aire caliente.
- Para el soldado de componentes de montaje superficial (SMD, *Surface Mounting Device*) es recomendable tener un cautín con una punta lo más delgada posible y de preferencia que sea de temperatura regulable. También se recomienda contar con una pistola de aire caliente, soldadura con un diámetro menor a un milímetro, malla para desoldar, pinzas de punta delgada, alcohol isopropílico y flux.
- Al soldar cristales se recomienda que el soldado de las terminales se haga rápidamente y si se tiene acceso al ajuste de temperatura, el soldado se debe realizar con la temperatura más baja a la que se funde la soldadura. Un exceso de calor sobre las terminales del cristal puede implicar un daño irreversible. En todos los casos se recomienda leer las recomendaciones del fabricante.
- Dependiendo del tipo de encapsulado se tendrá que elegir entre usar un cautín, usar una pistola de aire caliente o ambos. La pistola de aire caliente se puede emplear para soldar la mayoría de los componentes de montaje superficial, siendo esencial para soldar componentes cuyo encapsulado sea BGA (*Ball Grid Array*, Arreglo de Cuadrícula de Esferas), LGA (*Land Grid Array*, Arreglo de Cuadrícula de Terminales Ocultas) o con

terminales en la parte inferior del componente. Por otro lado, el cautín se puede emplear para soldar la mayoría de los componentes de montaje superficial cuyo encapsulado tenga terminales en la periferia del componente.

- Independientemente de la herramienta que se decida usar para soldar se recomienda estañar las terminales de los *footprints*, antes de colocar los componentes. Hecho esto, se sugiere que se tome el componente a soldar con una pinza adecuada, una herramienta de succión o un lápiz adhesivo (sólo si la superficie del componente es mayor que el diámetro del lápiz adhesivo) y se coloque en la posición de soldado. Una vez en posición se deben de calentar las terminales del componente. En la mayoría de los casos el estañado permitirá que el componente se fije al *footprint*, en caso de que no suceda esto se debe de colocar un poco más de soldadura en el *footprint* y repetir los pasos anteriores. Una vez que el componente este fijado se debe de revisar que las terminales estén bien soldadas, en caso contrario se debe corregir el soldado, agregando más soldadura o retirándola con la malla, según sea el caso.
- Para ejemplificar la metodología del punto anterior, se describirá el proceso seguido para soldar el circuito integrado mostrado en la figura 2.15.
 - a) El primer paso consistió en estañar todas las terminales del *footprint*, para realizar esto se colocó un poco de flux sobre todas las terminales. Después se colocó la punta del cautín sobre una de las terminales, se acercó la soldadura a la punta del cautín, se fundió un poco de soldadura en dicha terminal y se esparció sobre toda la terminal. El proceso descrito se debe repetir hasta estañar todas las terminales del *footprint*. Sin embargo, para agilizar este proceso se puede seguir la siguiente técnica: Normalmente al fundir soldadura queda un excedente en la punta del cautín, este excedente se puede aprovechar para estañar las demás terminales, pasando la punta del cautín sobre las terminales no estañadas como si se estuviese coloreando. Esto se hace hasta que el exceso de soldadura en la punta del cautín se agote, cuando el excedente se termina se funde más soldadura y se repite el procedimiento.
 - b) Al terminar de estañar las terminales, se usó la malla para limpiar el exceso de soldadura que quedó. Al usar la malla se procuró que todas las terminales del *footprint* tuvieran una capa delgada de soldadura, es decir, que no hubiesen protuberancia.
 - c) Enseguida, se procedió a tomar el circuito integrado con un lápiz adhesivo, para ello el lápiz adhesivo se colocó sobre la superficie delantera del componente (donde está la matrícula) hasta que éste se adhirió al lápiz adhesivo. Hecho esto se acomodó el componente en la posición en que se debía soldar, con el cautín se calentaron un par de terminales del circuito integrado hasta que este quedó fijo.
 - d) Una vez afianzado el componente se procedió a colocar un poco más de flux para soldar las demás terminales. A continuación se describen las metodologías

seguidas que se usaron para soldar dicho componente usando diferentes herramientas:

- I. **Con un cautín de punta fina o especial y soldadura de diámetro inferior a un milímetro.** Se colocó el cautín en cada una de las terminales, de tal manera que tocara una parte de la terminal y una parte de la pista. Después se acercaba la soldadura y se dejaba fundir un poco de soldadura
- II. **Con un cautín de punta fina o especial y soldadura de un milímetro.** Es similar al proceso anterior, ya que el procedimiento hace terminal por terminal. A diferencia de la técnica anterior, la soldadura se fundió primero en la punta del cautín y posteriormente este se acercaba a la terminal.
- III. **Cautín de punta regular y cualquier tipo de soldadura.** Para usar esta técnica se colocó soldadura sobre las terminales del circuito integrado (se hizo como en caso I y II y se obtuvieron los mismos resultados). Debido a que la punta del cautín era más grande que en los casos I y II, se tuvo un exceso importante de soldadura (comparado con el obtenido en los casos anteriores), dicho exceso se esparció arrastrando el cautín por las demás terminales, de tal manera que al cautín tocara una parte de la terminal del *footprint* y del componente. Una vez que todas las terminales estuvieron soldadas, se agregó un poco más de flux y se usó la malla para quitar el exceso de soldadura y garantizar que ninguna terminal quedara en corto circuito con sus vecinas. Una vez que se pasó la malla, se volvió a pasar el cautín, sin soldadura, por todas las terminales.
- IV. **Pistola de aire caliente (frontal).** Una vez que el circuito integrado se fijó, se usaron unas pinzas para sujetar el componente. Las pinzas se colocaron de tal manera que agarraban al componente por dos esquinas diametralmente opuestas. Una vez asegurado el componente con las pinzas, se colocó la pistola de aire caliente a una distancia entre 5 y 10 centímetros sobre el componente y se aplicó aire caliente sobre el componente realizando movimientos circulares suaves, de tal manera que el aire impactara sobre todas las terminales del circuito integrado hasta lograr que la soldadura de los *footprints* llegara a su punto de fusión. Una vez que la soldadura llegó a su punto de fusión se apagó la pistola y se sujetó el componente hasta que la soldadura se solidificara nuevamente. Cuando se usa una pistola de aire caliente se debe tener especial cuidado de no mover el componente mientras se está enfriando la soldadura, ya que de lo contrario se tendrá que repetir el proceso. Por lo que se recomienda buscar la posición más cómoda para sostener el circuito integrado. Es importante mencionar que si no se usan las pinzas, el aire de la pistola puede hacer que el circuito integrado sea retirado completamente

de su posición. Así mismo, al usar la pistola de aire caliente se debe ser muy cuidadoso, ya que los dispositivos circundantes al componente a soldar podrían ser afectados por el aire caliente y desoldarse. Para evitar esto es recomendable limitar el área de soldado con papel aluminio pegándolo a la placa con cinta de enmascarar.

- V. **Pistola de aire caliente (inferior).** En este método se siguieron los primeros pasos y precauciones descritas en el punto anterior, con la diferencia de que el aire se aplicó por la parte inferior de la placa, en la zona donde estaba el componente. Al usar este método se tuvo una mayor comodidad para sujetar el componente con las pinzas. Se hace notar que para aplicar este método se tuvo que usar un soporte donde pudiese yacer la placa para poder aplicar el aire caliente por la parte inferior. También, se debe tener en cuenta que dependiendo de los materiales con los que está construida la tarjeta y la calidad de los mismos se puede dañar la placa si se excede el calor aplicado.



Figura 2.15. Circuito integrado de encapsulado cuadrado plano de 100 terminales.

- Aunque los métodos discutidos se describen para un solo componente, éstos pueden extrapolarse para soldar la mayoría de los circuitos integrados y componentes de montaje superficial. Dependiendo del componente a soldar se recomienda elegir el método más sencillo para soldar dichos componentes.
- Para circuitos integrados de encapsulado BGA, LGA o similares, se debe usar una máquina especial para soldar dichos componentes. Sin embargo, si no se cuenta con una máquina, se pueden soldar los componentes siguiendo la técnica de soldado con aire caliente aplicado por la parte inferior.
- Recuerde que para soldar componentes de agujeros pasantes es recomendable que doble la terminal, del componente a soldar, 45 grados en dirección de la pista donde se soldará. Hecho esto, recorte la terminal del componente para que sobresalga un milímetro o dos

sobre la perforación. Para soldar se debe colocar la punta del cautín de forma que toque la terminal del *footprint* y la terminal del componente a soldar. Después se debe acercar la soldadura en donde está la punta del cautín, se deja fundir un poco de soldadura, se retira la soldadura, y sin retirar el cautín se coloca la soldadura en el extremo contrario de donde está la punta del cautín y se deja fundir un poco más de soldadura. Se retira la soldadura y manteniendo el cautín en posición, se deja fundir soldadura en la punta de la terminal del componente soldado. Terminado esto se retira la soldadura y después el cautín. El procedimiento descrito debe realizarse rápidamente.

- Nunca sople o aplique aire frío para enfriar una soldadura.
- Para soldar componentes se recomienda usar una pulsera antiestática y ropa de algodón para evitar descargas sobre los componentes.
- Use extractores de humo para evitar respirar directamente los vapores derivados del proceso de soldado.
- Siempre trabaje en lugares bien iluminados y use lentes de aumento cuando suelde secciones muy pequeñas.
- Una vez terminado el soldado es recomendable retire el exceso de soldaduras con malla y limpie la placa con un poco de alcohol isopropílico.

Cuando se terminó de armar la tarjeta, ésta se colocó en un gabinete junto con una batería (como sistema de alimentación principal) y con ello se obtuvo el primer sistema *Cycle-DAQ*. En la sección siguiente se comentaran los resultados tras poner a prueba el equipo que se armó, así como las soluciones que se tuvieron que proponer para que el equipo fuera completamente funcional.

PUESTA A PUNTO DEL SISTEMA

En este capítulo se comentarán los resultados de las pruebas realizadas al sistema desarrollado, los problemas que surgieron al poner a trabajar el equipo y las soluciones que se dieron a dichos problemas.

3.1. Primeras pruebas

En el primer *Cycle-DAQ* que se armó, es decir el equipo montado en la PCB y un gabinete, se probó que todas sus partes trabajaran individualmente, pero no se había comprobado el funcionamiento del equipo haciendo que todas sus partes trabajaran al mismo tiempo. Por esta razón, el equipo fue sometido a un conjunto de pruebas cuya finalidad era darnos información sobre el desempeño del sistema, las limitaciones y las fallas del mismo. Cabe comentar que estas pruebas se llevaron a cabo en condiciones de laboratorio, lo que significa que el equipo se alimentó con una fuente de alimentación de laboratorio, el área de trabajo tenía condiciones ambientales regulares (humedad relativa, temperatura y presión similares para todos los días de prueba) y la superficie sobre la que se colocó el equipo era plana y sin vibraciones.

En el capítulo anterior se comentó que antes de contar con el equipo armado en una PCB se hubo construido un prototipo. En él se probó la rutina general, mostrada en capítulos anteriores, y se obtuvo información que permitió caracterizar el comportamiento del prototipo. Debido que la rutina que se probó en el prototipo se programó tal cual en la tarjeta de circuitos impresos, se esperaba que el equipo tuviera un comportamiento similar o mejor. Por ello, se llevaron a cabo pruebas generales para corroborar el funcionamiento del equipo. Los resultados de estas pruebas nos mostraron que el equipo era funcional, pero no había mejorado su funcionamiento respecto al prototipo, ya que el problema de conexión a la red celular se seguía presentando.

Como se comentó, la idea bajo la que se diseñó la tarjeta era tener un diseño en el que los dispositivos y conexiones interfirieran lo menos posible con la antena del dispositivo celular. Sin embargo, como los resultados de las pruebas generales nos mostraron que se tenía la misma falla, se decidió que se debían probar a fondo todas las secciones del *Cycle-DAQ*, con la finalidad de descartar, o encontrar, si el problema de conexión se estuviese generando por las otras secciones del sistema, ya fuese en cuestiones de software o hardware.

Para realizar las comprobaciones mencionadas se estableció la siguiente metodología:

- Se analizarían cada una las fallas y se buscarían las causas más probables de cada falla. De acuerdo a las causas se establecerían hipótesis que se tratarían de comprobar mediante pruebas.
- Se realizarían ciclos de pruebas, es decir que se harían varias pruebas para comprobar una sola hipótesis.
- Las pruebas tendrían una duración mayor a tres horas. Cabe comentar que dicho tiempo se estableció para simular un tiempo prolongado de funcionamiento; puesto que, los ciclos de manejo estándar no duran más de dos horas (para mayor referencia se recomienda leer la sección de este trabajo de tesis dedicada a ciclos de manejo).
- La estación móvil celular estaría deshabilitada durante todos los ciclos de prueba.

El primer ciclo de pruebas que se llevó a cabo tenía la finalidad de mostrarnos el funcionamiento general del equipo sin la estación móvil celular. Este ciclo de pruebas duró dos días y las pruebas tuvieron una duración de cinco horas cada una. Durante este ciclo de pruebas se encontró que el equipo tenía problemas para desplegar información en pantalla, pues existían caracteres que nunca se borraban, el tamaño de letra era muy pequeño para desplegar algunas variables y en algunas secciones se encimaban caracteres. Este malfuncionamiento se presentó a lo largo de cada prueba de este ciclo y para solucionar estas problemáticas se revisaron minuciosamente las conexiones del equipo y el código correspondiente al despliegue de información. Tras la revisión se realizaron los siguientes ajustes:

- **Aumentar el tamaño de letra y limitar número de caracteres desplegados.** Cuando se diseñó la rutina para desplegar información, se pensó que las mediciones deberían de mostrarse en pantalla con un alto grado de exactitud. Sin embargo, tras el primer ciclo de pruebas se comprobó que sería mejor mostrar en pantalla las mediciones con una menor exactitud y con ello ganar espacio para mostrar los datos con un tamaño de letra mayor. Respecto a esto se debe recalcar que disminuir la exactitud de las mediciones mostradas en pantalla no afecta la exactitud de las mediciones almacenadas en el archivo de texto generado por el equipo.
- **Desechar las primeras lecturas recibidas del receptor GPS.** Al analizar el comportamiento del equipo se descubrió que cuando el receptor GPS se enlaza por

primera vez a la red satelital manda información basura, ésta tiene la estructura de una cadena GPS válida, pero los datos contenidos en ella son inservibles. Cuando el equipo recibía las cadenas basura las interpretaba como válidas y desplegaba su información en pantalla. Los datos basura por lo general necesitaban un área mayor para ser presentados en pantalla, de la que se había considerado para desplegar la información válida. Por esto, una parte de dicha información llegaba a desplegarse en áreas donde la rutina de borrado de información en pantalla no estaba definida para ejecutarse. Debido a lo anterior, y a que durante la ejecución de la prueba el receptor GPS llegaba a perder conexión varias veces, se producían zonas donde los caracteres se encimaban o bien se mostraban caracteres que nunca se borraban. Tras considerar diversas soluciones al problema, se encontró que la solución más viable era la de discriminar las primeras diez cadenas de información recibidas del receptor GPS tras conectarse y con ello asegurar que la información mostrada en pantalla y guardada es información válida. Cabe comentar que ignorar estas cadenas no es significativo, ya que equivale a perder menos de 2 segundos de información.

Tras realizar los ajustes se comenzó el segundo ciclo de pruebas, el cual duró tres días, y cuyas pruebas se realizaban en lapsos de tres horas cada una. En este caso se eligió disminuir el tiempo de duración de las pruebas y aumentar el tiempo del ciclo, y con ello tener un mayor número de archivos de datos generados por el equipo. Con la información recuperada de los archivos, se deseaba conocer el desempeño de las partes del equipo, ya que se pensó en que si hubiese un mal funcionamiento con cualquier parte (salvo la pantalla táctil), éste se vería reflejado en los datos almacenados en los archivos, ya sea como datos basura o fuera del rango esperado.

A la mitad del ciclo de pruebas, se comenzó con el análisis de la información generada hasta ese momento. La información de los archivos nos mostró datos alarmantes, ya que se encontraron archivos en los que después de cierto tiempo, todas las cadenas de información almacenadas estaban truncas. Así mismo, se encontraron archivos que tenían información que indicaba que el equipo no había recolectado información durante toda su operación. Si bien, se había considerado que unas cuantas cadenas almacenadas tuviesen diferencias con respecto al grueso de la información generada en cada prueba, nunca se pensó que el equipo presentase un comportamiento como el indicado. Debido a los problemas encontrados, se decidió ocupar la segunda mitad del ciclo de pruebas para probar diferentes hipótesis y soluciones a los conflictos encontrados.

Durante las pruebas anteriores se monitoreo la información desplegada en pantalla, esto para corroborar que los ajustes realizados hubiesen sido adecuados, pero en ese periodo no se notó el comportamiento indicado en los archivos de texto. Por lo anterior, se planteó probar que los datos de la pantalla eran incorrectos y que la información almacenada describía el comportamiento del equipo. Bajo esta premisa se determinó probar dos cosas: 1) que las mediciones mostradas en la pantalla eran la repetición de una sola medición y 2) corroborar que el monitoreo de la pantalla había sido correcto. Para comprobar estas ideas se realizó un monitoreo intensivo de la información mostrada en pantalla, lo cual se realizó registrando cada dos minutos, en un documento externo, la información desplegada. Así mismo, uno de los parámetros que se podían modificar libremente, en las condiciones de laboratorio antes descritas, era la pendiente. Debido

a esto, se decidió que la pendiente se utilizaría como un parámetro adicional para verificar la diferencia entre lo almacenado y lo desplegado en pantalla. Cabe comentar que la metodología seguida y los resultados obtenidos de las mediciones de ángulos se tratarán con mayor amplitud en secciones posteriores.

Tras realizar un par de pruebas se verificó la información del archivo de texto de las pruebas y se cotejó contra lo recabado en el registro externo. Las conclusiones del cotejo fueron que la información almacenada en el archivo y la desplegada en pantalla, eran congruentes entre sí, sin embargo la información en pantalla no mostraba los problemas que se presentaban en los archivos de texto, además se probó que la información en pantalla no era la repetición de una medición. Debido a estos resultados se planteó investigar el problema en la sección de almacenamiento.

Las premisas bajo las que se indagaron las posibles fallas en el medio de almacenamiento fueron que la tarjeta de memoria podía estar dañada o bien no se había programado correctamente el código para guardar información. Para descartar la primera suposición se revisó el medio de almacenamiento haciendo uso de herramientas informáticas que nos revelaran daños en los sectores lógicos y físicos de la tarjeta de memoria. Las pruebas realizadas nos indicaron que el dispositivo de almacenamiento estaba en un estado óptimo, tras esto se procedió revisar el código asociado al almacenamiento. No obstante de que se realizó una revisión minuciosa, no se encontró ningún error sintáctico ni lógico, pero para descartar por completo esta sección de código se modificó la rutina general para que hacer que el equipo guardara y mostrara en pantalla la información de una cadena de datos predeterminedada. Con estas modificaciones se puso a trabajar el equipo y tras algunas pruebas se revisaron los resultados, los cuales nos mostraron que el dispositivo de almacenamiento y su código asociado estaban trabajando correctamente. Debido a que las hipótesis anteriores no nos condujeron a la raíz del problema, se tuvieron que plantear otras probables causas que estuviesen afectando el almacenamiento de la información. De las nuevas propuestas se eligió indagar sobre aquellas que parecían ser más probables, las cuales fueron:

- **Diseño, construcción o montaje deficiente de la PCB:** Si bien, el diseño de la PCB se rigió por normativas internacionales y fue revisado por un experto en la materia, no era improbable que el diseño fuese el causante de la falla. De la misma manera, el problema se pudo generar por la utilización de elementos de baja calidad en el proceso de fabricación o bien por haber soldado de manera inadecuada los componentes. Para saber si alguna de las primeras dos suposiciones era la causante del conflicto, se decidió comprobar la respuesta en frecuencia de las pistas entre el microcontrolador y la tarjeta de memoria. La respuesta en frecuencia de las pistas nos indicó que cerca de la frecuencia de trabajo (8 [MHz]) no había pérdidas significativas. En cuanto a la tercera suposición se verificaron nuevamente todas las terminales soldadas, y salvo unas pocas soldaduras (de las que se corrigió su soldado) no había problemas con las soldaduras. Ya que en este caso sólo se realizaron verificaciones y correcciones menores, no se llevaron a cabo pruebas para corroborar si las mejoras implementadas habían surtido efecto.

- **Diseño de software y procesamiento de información deficiente:** Debido a que el hardware había sido verificado, sin que se encontrara el origen de los problemas de almacenamiento, no quedaba más que explorar a fondo todo lo relacionado al software, en especial en las secciones involucradas en el procesamiento de información. Las acciones empleadas, para verificar el software de manera general, consistieron en revisar la rutina general para encontrar secciones y variables que no se ocuparan, buscar procesos cuya codificación pudiese optimizarse y comprobar que la lógica de trabajo programada fuese adecuada. Si bien se hicieron mejoras, ninguna de ellas fue una modificación mayor, es decir no se cambiaron cosas que modificaran la estructura del programa. Antes de probar si los cambios habían sido efectivos, se le dio seguimiento puntual a todas las secciones relacionadas con el procesamiento de información. Una de las secciones de procesamiento que más nos generó duda fue la sección en donde los datos adquiridos por los sensores se integraban en una sola cadena de información. Se consideró la opción de que la variable donde se integraba la cadena de información, no tuviese la asignación suficiente de memoria RAM. Antes de intentar algún cambio se investigó sobre cómo se podía conocer la cantidad de memoria asignada una variable, sin embargo durante la búsqueda nos encontramos con el texto *Memories of an Arduino* de Bill Earl¹⁶. En el texto, el autor explica sobre los conflictos que presentan los microcontroladores de la plataforma Arduino® cuando la memoria RAM no está trabajando adecuadamente. Aunque los microcontroladores tratados en el texto no correspondían al empleado en el presente trabajo de tesis, los problemas descritos y los posibles causantes se asemejaban a los del *Cycle-DAQ*. Debido a esto se consideraron algunas de las recomendaciones dadas en el texto y finalmente se hicieron los siguientes ajustes: se quitaron la mayor cantidad de variables globales y se convirtieron en variables locales, se quitaron la mayoría de las variables constantes, se decidió que la cadena de texto se integraría a partir de dos cadenas menores y posteriormente se integrarían en una sola variable, la cual una vez conformada inmediatamente sería guardada. Con las adecuaciones realizadas se puso el equipo a trabajar y tras dos días de pruebas se comprobó la información recolectada, en ella se encontró que los problemas descritos se habían corregido. Para corroborar la solución se probó el equipo durante un día más, al finalizar las pruebas no se presentó ningún problema.

Con las acciones anteriores, se lograron resolver todos los problemas del *Cycle-DAQ*, sin la estación móvil, y dado que ya se habían verificado las soluciones ofrecidas a cada problema se dio por finalizado el segundo ciclo de pruebas. Entonces, se determinó que se deberían empezar a realizar pruebas al equipo con la estación móvil habilitada. La metodología seguida para las pruebas de la estación móvil y los resultados obtenidos se describen en la siguiente sección.

3.2. Puesta a punto de la estación móvil celular

En la sección anterior se describió la metodología desarrollada para buscar y corregir los errores asociados a la mayoría de los dispositivos del *Cycle-DAQ*. Sin embargo, no se debe olvidar que

¹⁶ Consultado el 26 de octubre de 2015 en <https://learn.adafruit.com/memories-of-an-arduino/arduino-memories>

en parte esto se realizó para asegurar que los problemas de conexión con la red celular se debiesen exclusivamente a la estación móvil. Así, cuando se finalizó la corrección de los errores de almacenamiento, se decidió reconectar y habilitar el código asociado a la estación móvil. Tras ello, el equipo se sometió a ciclos de pruebas como los descritos previamente, con la diferencia de que si el equipo al inicializar no se conectaba a la red celular tras cinco minutos, se daría por concluida la prueba y se iniciaría una nueva.

El primer ciclo de pruebas al que se sometió el *Cycle-DAQ*, tras habilitar la estación móvil celular, duró dos semanas, y a lo largo de éste, se buscaron las condiciones espaciales, ambientales y técnicas en las que el equipo presentaba una mayor probabilidad de enlazarse a la red celular por primera vez. Con tal fin, el equipo fue colocado en diferentes posiciones y localizado en varias áreas del laboratorio, así mismo se registraron las condiciones climáticas del día de la prueba, la hora a la que se realizaba la prueba, el tiempo que tardaba en conectarse a la red, las variaciones en la corriente y voltaje durante la conexión a la red. Al finalizar el ciclo de pruebas, se hizo un análisis completo de los datos, los cuales nos permitieron llegar a las siguientes premisas:

- Las condiciones ambientales y la hora del día no afectaban a la comunicación celular.
- Mientras el equipo intenta enlazarse a la red, ya sea para conectarse por primera vez o durante la transmisión de datos, el equipo llegaba a consumir hasta un ampere en lapsos muy cortos. En cuanto a la tensión eléctrica, no se encontró variación significativa. Este último resultado se tomó con reserva ya que el equipo estuvo alimentado por una fuente de laboratorio y no por una batería, como el equipo debía ser alimentado.
- Si tras dos minutos el equipo no se enlazaba a la red celular, era muy probable que el equipo ya no se conectara.
- La posición del equipo no fue un factor que mejorara considerablemente la conexión. Sin embargo, la localización de la estación móvil resultó ser fundamental. Se obtuvieron mejores resultados cuando el equipo se encontraba en las secciones del laboratorio que tenían vista al exterior.
- Los cambios llevados a cabo en secciones anteriores mejoraron las probabilidades de conexión.

Los resultados obtenidos no nos permitieron determinar si los problemas de conexión ya se habían solucionado. Pues, aunque el equipo había tenido un mejor comportamiento que en otras ocasiones, seguía presentando un comportamiento errático al momento de conectarse con la red. En principio se pensó que quizá la ubicación del laboratorio y la estructura del mismo eran los causantes del problema. Puesto que se comparó el comportamiento para conectarse a la red GSM/GPRS del sistema en desarrollo y el de una estación móvil comercial. Los resultados nos mostraron un comportamiento similar entre ambos dispositivos, esto nos convenció de probar el equipo fuera de laboratorio.

Con la finalidad de probar que los problemas de conexión habían quedado resueltos y que el equipo era funcional, continuamos con lo siguiente:

- En todas las pruebas el equipo fue alimentado exclusivamente con baterías. Inicialmente se le dotó de una batería de 4 [Ah] a 6 [V].
- El equipo se probó estáticamente. Se llevó a dos localidades diferentes de las del laboratorio, se colocó en una superficie plana y se puso a trabajar.
- También se llevaron a cabo pruebas dinámicas al equipo, es decir que se el sistema se colocó en un vehículo y se sometió a pruebas.
- Las pruebas dinámicas se llevaron a cabo en rutas preestablecidas, para comprobar la repetitividad de resultados, y también en rutas aleatorias, para recolectar la mayor información sobre la operación posible. Cabe comentar que las verificaciones realizadas se llevaron a cabo tanto en ciudades como en carreteras.
- Si al inicializar el equipo y tras dos minutos el sistema no se enlazaba a la red celular la prueba se cancelaba y se comenzaba una nueva.
- Se usó una estación móvil comercial como sistema de referencia, con la finalidad de saber si en las zonas de prueba había problemas con conexión a la red.

El equipo se puso a prueba durante un mes, y se procuró recopilar la mayor cantidad de información posible respecto a la capacidad del mismo de conectarse a la red celular. En este periodo se observó que en las pruebas estáticas el equipo presentaba su mejor comportamiento, ya que el sistema se conectó a la red celular el noventa por ciento de las ocasiones, la información recolectada no presentaba problemas de visualización ni de almacenamiento, los datos enviados al servidor web eran correctos y el equipo funcionaba sin mayor novedad. Sin embargo, en las pruebas dinámicas el comportamiento era distinto, ya que sólo el sesenta por ciento de las ocasiones el equipo lograba enlazarse a la red celular. En estas pruebas la ubicación del equipo resultó ser crucial, ya que si el sistema intentaba conectarse a la red en sitios donde la estación móvil comercial indicaba una recepción regular o baja el equipo no se conectaba (se utilizó la estación móvil comercial para conocer la recepción de una zona. La recepción era indicada a través de barras. Cuatro barras indicaban la mejor recepción, tres a dos barras indicaban una recepción regular media y regular baja respectivamente, una barra indicaba una mala recepción y cero barras que no había recepción). Así mismo, si el equipo se había enlazado a la red y se intentaba enviar información en una zona de recepción regular baja o menor, el equipo dejaba de realizar sus funciones, puesto que el equipo mostraba únicamente la información que se había intentado enviar y no llevaba a cabo ninguna otra acción. Así mismo, al revisar la información almacenada en la tarjeta de memoria, correspondiente a las pruebas en que el equipo dejaba de trabajar, sólo se encontraban los datos recopilados hasta el momento en que el equipo intento enviar la información y en el servidor web nunca se recibieron dichos datos. Cuando el equipo dejaba de trabajar era necesario reiniciarlo y buscar una mejor zona de conexión. En algunos

casos, en especial en carreteras, en zonas con muy baja recepción o cuando la tensión la batería era cercano a los 5.5 [V], el sistema se reiniciaba.

Si bien el funcionamiento del equipo había mejorado con respecto a lo observado en el laboratorio, el comportamiento del equipo no era satisfactorio, ya que con la probabilidad de conexión obtenida en las pruebas dinámicas no podíamos asegurar que el equipo fuese funcional. Además, al comparar las probabilidades de conexión del *Cycle-DAQ* y la estación móvil comercial, se notó un comportamiento muy superior en la estación móvil comercial; ya que en las pruebas estáticas la probabilidad de conexión fue de cien por ciento y en las pruebas dinámicas del noventa y cinco por ciento. Considerando todo esto se planteó buscar que el *Cycle-DAQ* tuviera el mejor desempeño posible al enlazarse a la red celular. Para cumplir con dicho objetivo se analizaron con mayor detalle los datos recabados, buscando en ellos los causantes más factibles de las deficiencias en la conexión. Tras el análisis de los datos se establecieron tres puntos de investigación, los cuales se detallan a continuación:

- **Proveedor de servicios comunicación celular:** En las pruebas mencionadas, la estación móvil comercial y el *Cycle-DAQ* usaron tarjetas SIM de proveedores de servicios de red distintos. Al principio se pensó que esto no sería un problema, puesto que en las zonas de prueba los proveedores indicaban, en sus mapas de cobertura (los proveedores de telefonía celular cuentan con mapas de cobertura, en los que se detallan la cobertura de sus servicios ofrecidos. Los mapas se pueden consultar en las páginas web de los proveedores), que había cobertura completa para la comunicación GSM/GPRS. Sin embargo, como los resultados de conexión eran distintos entre nuestro equipo y la estación móvil comercial, se decidió utilizar el mismo proveedor para ambos equipos. Se estableció que se ocuparían tarjetas SIM del proveedor de servicios de telefonía celular más grande del país, ya que se pensó que con esto se tendría un mayor éxito al enlazarse a la red. Para comprobar nuestra suposición se adquirió una tarjeta SIM para el *Cycle-DAQ* (del mismo proveedor que el de la estación móvil comercial), se realizaron los ajustes necesarios para que el equipo trabajara con la nueva tarjeta SIM y el sistema fue sometido a pruebas estáticas y dinámicas. Los resultados de las pruebas nos indicaron que se había mejorado la probabilidad de conexión a la red (en cinco por ciento para las pruebas estáticas y en diez por ciento en las pruebas dinámicas). A pesar de las mejoras registradas, el equipo siguió presentando problemas con la conexión al inicializar si la cobertura registrada por la estación móvil comercial era regular o baja. De la misma manera, si el lugar donde se intentaba enviar no tenía buena cobertura el equipo dejaba de trabajar o se reiniciaba. Antes de sacar conclusiones respecto a lo observado y buscar la raíz del problema en otra parte, se realizó la siguiente comprobación: Debido a que se tenían tres tarjetas SIM (SIM A, SIM B y SIM C), dos de un mismo proveedor (SIM A, perteneciente a la estación móvil comercial y SIM C, correspondiente a la segunda tarjeta que se colocó en el *Cycle-DAQ*) y una tarjeta de un proveedor distinto (SIM B, correspondiente a la primera tarjeta usada en el *Cycle-DAQ*), se decidió intercambiar las tarjetas entre las estaciones móviles y conocer el comportamiento de las mismas, con las distintas tarjetas SIM. Entonces, como la estación móvil comercial sólo había trabajado con la SIM A, se le retiró ésta y se le colocaron las tarjetas SIM B y C. Mientras que al *Cycle-DAQ* se le colocó la SIM A. La información recolectada tras este experimento nos

dio información interesante, dado que por un lado nos indicó que el equipo tenía un mejor rendimiento con la SIM A, en tanto que la estación móvil comercial no presentaba ningún conflicto de conexión con al usar las SIM B y C. Cabe comentar que aunado al experimento, se contactó al centro de atención de clientes del proveedor de las tarjetas SIM A y C, para preguntar si los usuarios con mayor antigüedad alquilando sus servicios tenían preferencia sobre aquellos usuarios de menor tiempo de afiliación, para poder ocupar la tecnología GSM/GPRS. Ante la pregunta la compañía nos respondió que todos los usuarios afiliados a su red tenían la misma preferencia para transmitir-recibir información y que normalmente los problemas para enlazarse a su red provenían de deficiencias en las estaciones móviles. Así mismo, se le preguntó a la compañía, si existía alguna diferencia en la transmisión-recepción, si se ocupaban modelos distintos de tarjetas SIM (las compañías de telefonía celular cuentan con tarjetas SIM que vienen marcadas con diferentes modelos, por ejemplo v3, v4 y v5. Con la respuesta recibida y tras investigar un poco más de ello se supo que el modelo indicado en las tarjetas SIM estaba de acuerdo a cambios en la apariencia y dimensiones físicas de la misma), a lo cual nos respondieron que tampoco debería de existir diferencia. Con toda la información que obtuvimos y las respuestas de la compañía de servicios celulares, no logramos obtener conclusiones importantes respecto a este punto, por lo que se decidió parar la investigación de este punto y continuarla hasta que se revisaran las otras líneas de investigación. Es importante mencionar que para las pruebas posteriores el *Cycle-DAQ* se probó con la SIM C y la estación móvil comercial se quedó con la SIM A.

- **Alimentación del equipo:** La segunda línea de investigación que se siguió estaba relacionada con la energía suministrada por la batería. Ya que se pensó que los reinicios frecuentes del equipo se debían a la incapacidad de la batería para suministrar energía o bien algún problema con los reguladores de tensión eléctrica usados para adecuar la tensión de la batería a los niveles requeridos por los dispositivos del *Cycle-DAQ*. Con la finalidad de corregir esta deficiencia primero se consultó la hoja de datos de los reguladores para saber si eran capaces de soportar las condiciones de carga a las que estaba sometido. Al leer la hoja de datos corroboramos que las condiciones de carga a las que se estaba sometiendo el regulador eran adecuadas para los reguladores utilizados, sin embargo se notó que habíamos cometido un error al utilizar una batería de 6 [V], ya que el regulador permitía el máximo flujo de corriente si la tensión de entrada era cercana a los 10 [V]. Entonces se adquirió y se colocó una batería de 1 [Ah] a 12 [V] al *Cycle-DAQ* para probar su desempeño. Con la nueva batería el equipo mejoró su probabilidad de conexión en las pruebas dinámicas (alcanzando una probabilidad de setenta y cinco por ciento). Aunque la mejora en la probabilidad de conexión fue un gran punto, los reinicios del equipo se seguían presentando y debido a ello no podíamos aseverar que la batería tuviese la capacidad para suministrar la energía requerida por el equipo. Antes de adquirir una batería con una capacidad de corriente mayor, se consideró buscar más factores involucrados en los reinicios. El primer elemento que se consideró fue el microcontrolador, ya que era probable que durante la conexión a la red celular o durante el envío de información, el voltaje llegaba a caer lo suficiente como para producir el reinicio. Con la finalidad de corroborar nuestra hipótesis se comenzó a medir y monitorear la tensión eléctrica que presentaba la terminal de reinicio del microcontrolador. Se

encontró que efectivamente el voltaje en la terminal de reinicio se podía medir en algunas ocasiones una caída de hasta 4.5 [V] durante la comunicación con la red celular. Con tal información se buscó en la hoja de datos, si dicha tensión eléctrica era suficiente para provocar el reiniciar el equipo. En la hoja de datos del microcontrolador se encontró que dicha tensión eléctrica era suficiente para activar el detector de caída de tensión eléctrica, el cual si tenía esa lectura de tensión eléctrica por un cierto tiempo producía un reinicio en el equipo. Cabe comentar que desde el inicio del desarrollo del sistema se sabía de la existencia del detector, pero no se había leído lo suficiente respecto al mismo y desde el principio se prefirió mantener la configuración de configuraciones de fábrica del detector. Una vez que nos dimos cuenta de nuestra omisión procedimos a leer todo lo referente al detector que había en la hoja de datos. En ésta se explicaba como configurar el detector, el tiempo que debía de mantenerse una lectura baja para provocar el reinicio, cuáles eran los niveles de tensión en los que se activaba el detector y las razones para mantener activo el mismo (una de las razones para dejar activo el detector, es para prevenir que la información almacenada en el microcontrolador se vea corrompida por una deficiencia en la alimentación). Con la información recopilada se descubrió que el microcontrolador tenía preestablecida una configuración para ocupar el detector de caída de tensión, y la cual resultó ser la más adecuada para las características de nuestro sistema. Una vez que se definió la mejor configuración del detector para nuestras necesidades, se llegó a la conclusión de que el equipo se estaba reiniciando porque pasaba el tiempo suficiente, con lecturas menores a los 4.5 [V], para causar el reinicio. Como parte de la búsqueda de información se consultó en Internet, y también cómo otras personas habían lidiado con problemas similares. Se encontró que las personas que habían trabajado con el mismo microcontrolador que nosotros no presentaban el problema de reinicio, aun cuando trabajaban con cargas similares a las nuestras, usando un arreglo resistivo capacitivo como el mostrado en la figura 2.13, y comentado en otra sección. Se decidió entonces implementar el arreglo en el equipo y tras probar el desempeño del sistema con la adecuación se encontró que el problema de reinicio había quedado resuelto. Sin embargo, el equipo seguía presentando problemas para enlazarse a la red o dejaba de trabajar.

- **Estación móvil celular:** En las líneas de investigación anteriores logramos mejorar la conexión del equipo e inclusive resolver el problema de reinicio automático del equipo. No obstante, no se había logrado que la conexión con la red celular tuviera un comportamiento similar al presentado en estaciones móviles comerciales. Por ello, tras arreglar el problema de reinicio, se decidió analizar profundamente toda la información disponible de la estación móvil, procurando conocer más sobre el hardware de la estación móvil. Después de la búsqueda documental, se realizaron comprobaciones a la tarjeta, es decir que se revisó el estado de los componentes (capacitores, diodos, transistores, fusibles, osciladores, componentes asociados a la antena) que al sufrir un daño pudiesen generar un comportamiento como el observado. Mas las pruebas sólo nos confirmaron que los componentes de la estación móvil estaban en buen estado. Concluida la comprobación de componentes, se observó con mayor detalle el comportamiento de la estación móvil antes, durante y después de la conexión. Al realizar esta revisión, nos llamó la atención el comportamiento de un LED, el cual indicaba si el transceptor de la estación móvil estaba encendido o apagado (no se debe perder de vista que el transceptor

es sólo una parte de la estación móvil. En este caso, aunque la estación móvil estaba encendida, el LED nos indicaba que el transceptor no lo estaba). En teoría dicho LED, por ende el transceptor, siempre debía de estar encendido, no obstante éste se apagaba cuando el equipo no podía enlazarse a la red y éste no se volvía a prender hasta que se reiniciara el *Cycle-DAQ*. Por lo observado, se buscó información del fabricante que indicara algo relativo al comportamiento mostrado por el sistema, sin embargo no se encontró información documental que nos orientara sobre lo sucedido. Entonces, como se contaba con el diagrama esquemático de la tarjeta, se optó por medir las tensiones eléctricas de la estación móvil y compararlas con las indicadas por el fabricante y corroborar el problema nos fuese una falla eléctrica. En los puntos medidos no se encontró discrepancia respecto a lo esperado, a pesar de ello se decidió darle un seguimiento exhaustivo al sistema de encendido del equipo y comprobar que esta sección no estuviese provocando el apagado del transceptor. La información recabada en nuestro estudio nos llevó a concluir que el transceptor tenía el siguiente funcionamiento: Al encender la estación móvil, el transceptor quedaba energizado automáticamente y esperaba una señal para encender todas sus funciones. Dicha señal de encendido correspondía a un nivel lógico bajo en el colector del transistor, ya que sólo trabajaba como un interruptor digital. Entonces, cuando la señal de encendido era enviada por el microcontrolador del *Cycle-DAQ*, el transceptor la detectaba y alistaba su operación para la conexión a la red. Posteriormente, si el equipo intentaba conectarse a la red y no lograba la conexión, reiniciaba todas sus funciones, lo cual hacía que volviera a esperar la señal de encendido, sin embargo, como no recibía dicha señal, porque efectivamente no se mandaba, el equipo transceptor no hacía más. Por otro lado, el programa de conexión estaba diseñado para mantener un ciclo hasta que el sistema se conectara a la red celular, pero como el transceptor no hacía nada el equipo se quedaba en el ciclo indefinidamente. Así mismo, cuando el equipo trataba de enviar información y el transceptor no encontraba una red válida, éste último reiniciaba sus funciones y no hacía nada más. Con la finalidad de comprobar nuestras suposiciones se decidió mantener la señal de encendido durante todo el funcionamiento del equipo, es decir que se mantendría un nivel lógico bajo en todo momento en la terminal de encendido. Cabe comentar que antes de realizar esto nos aseguramos, en la información proporcionada por el fabricante, que no existiera una contraindicación de mantener la señal de apagado, dado que no se encontró nada, se procedió a conectar un puente entre el colector del transistor y tierra. Haciendo la adecuación mencionada el equipo se puso a funcionar, tras varias pruebas se encontró que el equipo había mejorado completamente su funcionamiento, ya que su probabilidad de conexión era comparable al presentado al de la estación móvil comercial, ya no presentaba los problemas de conexión y tampoco quedaba pasmado. Una vez encontrada la raíz del problema se quitó el puente y se acondicionó el software para que enviase la señal de encendido en cada iteración que el equipo no quedaba enlazado a la red celular.

Con todas las acciones que se llevaron a cabo, se pudieron corregir todos los problemas que impedían tener un equipo completamente funcional y confiable. No obstante, se llevaron a cabo pruebas adicionales para constatar que el comportamiento del equipo, tras las correcciones realizadas. En el siguiente capítulo se presentaran los resultados de las pruebas mencionadas. Así mismo se presentarán las conclusiones del proyecto.

RESULTADOS Y CONCLUSIONES

En este capítulo se muestran los resultados de las pruebas finales realizadas al sistema Cycle-DAQ. También se dan las conclusiones del proyecto.

4. Resultados

Las últimas pruebas a las que se sometió el equipo tuvieron la finalidad de permitirnos constatar que se tenía un sistema confiable, funcional y robusto, es decir que pudiera ser usado inmediatamente por los alumnos e investigadores del LCE. A continuación se detallan los resultados obtenidos.

4.1. Medición de Ángulos

Uno de los parámetros que mide el sistema *Cycle-DAQ* es la inclinación frontal del vehículo con respecto a un plano horizontal. Este parámetro, como se mencionó en el apartado *3.1 Primeras Pruebas*, fue una parte importante para resolver uno de los problemas de almacenamiento que presentó el *Cycle-DAQ*, ya que al variar la inclinación (ángulo) del equipo pudimos comprobar una hipótesis que nos llevaría a encontrar el origen de los fallos de almacenamiento de información.

Las primeras pruebas que se realizaron, respecto a la medición de ángulos, consistieron en variar aleatoriamente la inclinación del equipo. Las mediciones desplegadas en la pantalla táctil se registraban y se analizaban para corroborar que fueran consistentes con las inclinaciones realizadas, esto significa que si en una medición el equipo se había inclinado menos que en otra se verificaba que la diferencia entre ambas lecturas fuera congruente. Aunque las pruebas realizadas nos habían dado información valiosa, nos cuestionamos si las mediciones obtenidas eran válidas. Para conocer y asegurar que la información que se había obtenido era completamente confiable, se decidió que se debía medir la inclinación del sistema y las mediciones registradas debían de ser comparadas con un patrón.

Las pruebas para verificar las mediciones de ángulo reportadas por el sistema desarrollado se llevaron a cabo en el Laboratorio de Mecánica de la Facultad de Ingeniería de la UNAM, ya que en él contaban con el material y las instalaciones para poder realizar nuestras comprobaciones.

En la figura 4.1 se muestra la manera en que se dispusieron los diferentes elementos involucrados en las pruebas. Así mismo, se representa el material que se utilizó para verificar dicho funcionamiento, el cual consistió en un riel con soporte y un indicador de ángulo. En la figura 4.1 se puede ver que el *Cycle-DAQ* se colocó sobre el riel y que este último se elevaba en uno de sus extremos para obtener el ángulo de inclinación deseado.

Cabe comentar que el indicador de ángulo está formado por un transportador graduado que tiene atado a su centro el extremo de una cuerda delgada y ésta a su vez una masa en su otro extremo. La medición de los ángulos realizada de esta forma es posible debido a que sin importar la inclinación del transportador, la cuerda siempre mantendrá su posición vertical, a causa del peso de la masa. Entonces, cuando el transportador cambia su inclinación, la cuerda coincide con una de las graduaciones del transportador, y al calcular el valor absoluto de la diferencia entre el ángulo inicial y el ángulo final se obtendrá el ángulo de inclinación.

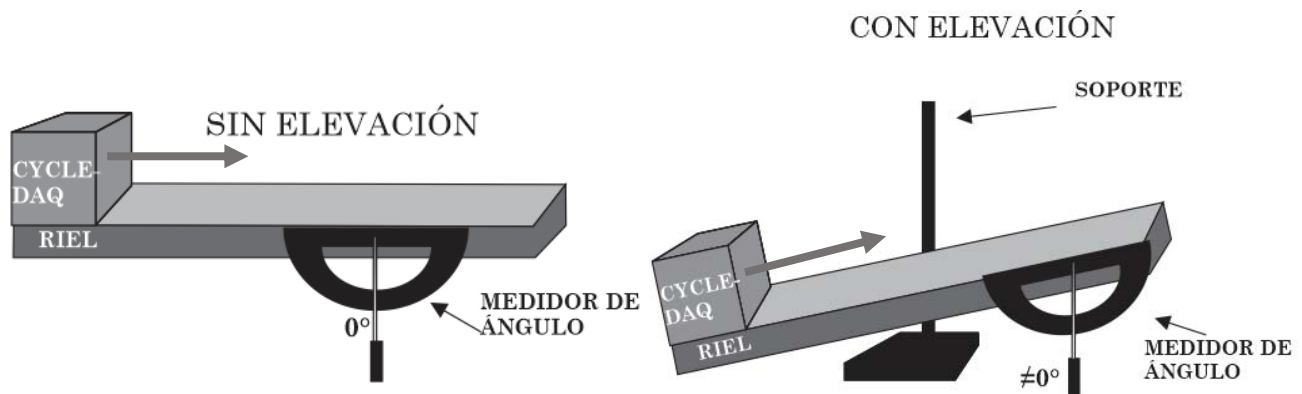


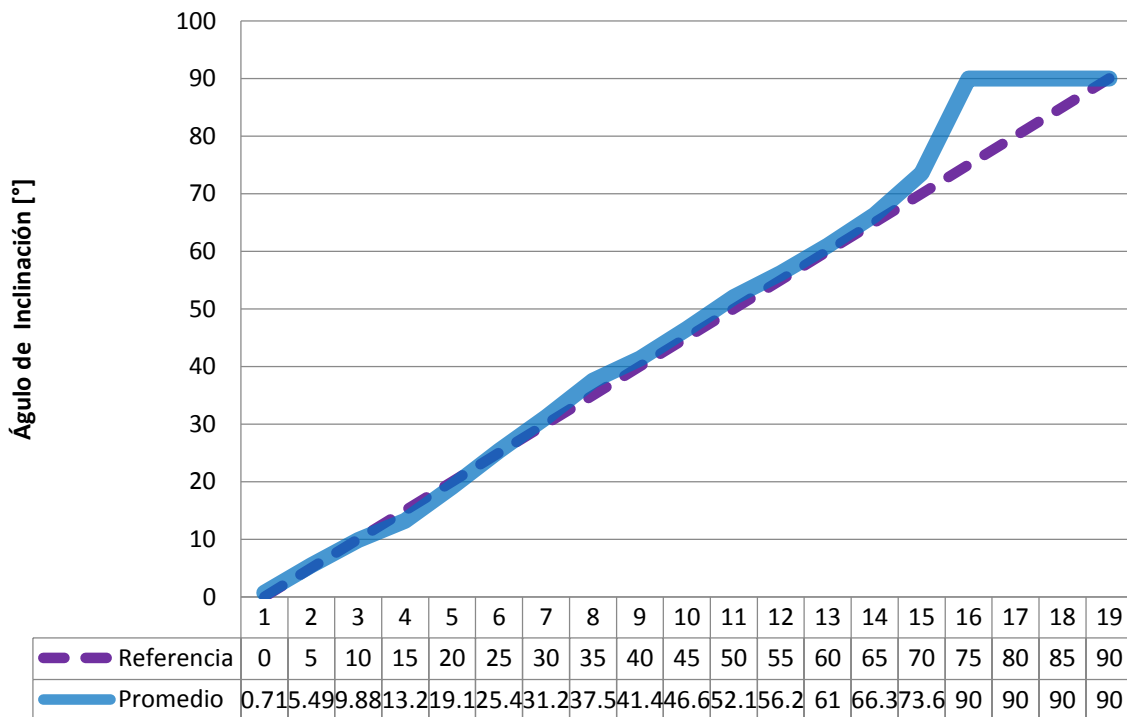
Figura 4.1. Arreglo para la medición de ángulos.

Con un arreglo como el de la figura 4.1 se realizaron mediciones de ángulos en el rango de 0 a 90 grados, en pasos de 5 grados. Así mismo, en las pruebas que se ejecutaron se realizaron comprobaciones tanto para la medición de pendientes positivas como de pendientes negativas.

Se debe tener en cuenta que la pendiente positiva hace referencia a la inclinación que tiene un vehículo cuando asciende y la pendiente negativa a la inclinación del vehículo cuando desciende.

En la figura 4.1 se indica con una flecha la inclinación que debe tener el equipo para medir una pendiente positiva, si el equipo se inclina en la otra dirección se medirá una pendiente negativa. Es importante mencionar que al revisar las mediciones de pendiente negativa se obtuvo un comportamiento similar al de la pendiente positiva, por lo que para fines prácticos se consideró el valor absoluto de las mediciones de la pendiente negativa y se analizaron todos los datos como positivos.

En la gráfica 4.1 se muestran dos líneas, la línea recta representa el comportamiento ideal del equipo y la otra corresponde al promedio de los datos obtenidos en los ensayos realizados. En la parte inferior de la gráfica, se muestra la tabla con los datos graficados. Si se observa el comportamiento de la información, ya sea en la tabla o en la gráfica se nota que el equipo tiene un comportamiento aceptable en cuanto a la medición de ángulos, ya que en la mayor parte del rango de medición, los datos obtenidos son cercanos a su comportamiento ideal. Se nota que a partir de los 75 grados de inclinación el equipo presenta serios problemas para proporcionar una medición confiable. No obstante lo anterior, ningún vehículo bajo prueba superará los 15 grados de inclinación, es decir que nuestro rango de interés es de 0 a 15 grados y, como se muestra en la gráfica, dentro de este rango el equipo ofrece información confiable para realizar ciclos de manejo.



Gráfica 4.1. Caracterización de la medición de ángulos.

4.2. Batería

Como se comentó a lo largo del texto el equipo *Cycle-DAQ* es un equipo autónomo y portátil, ya que no depende de otros sistemas para su funcionamiento y se puede transportar fácilmente. Sin embargo, como todos los equipos autónomos y portátiles, tiene un tiempo limitado de operación, el cual está relacionado con la capacidad de la fuente de alimentación de proporcionar energía al equipo.

En nuestro caso, el equipo requiere dos niveles de tensión eléctrica, uno de 3.3 [V] y el otro de 5 [V]. Con la finalidad de evitar usar dos fuentes de alimentación, se prefirió utilizar una sola y que los niveles de tensión eléctrica requeridos fueran dados por reguladores. Sin embargo, como se comentó anteriormente, por las características eléctricas de los reguladores se tenía que usar una fuente de alimentación cuyo voltaje nominal fuese superior a los 10 [V] e inferior a los 15 [V], con la finalidad de que los reguladores suministrasen la mayor cantidad de corriente posible. Por lo anterior, se decidió usar una batería de 12 [V], ya que en el mercado existe una amplia oferta de este tipo de baterías.

Las baterías cuentan con dos parámetros importantes a considerar: es la tensión nominal y la cantidad de corriente por hora nominal que puede suministrar. Dado que la tensión nominal ya había sido definida, faltaba por determinar la corriente de la batería. Este último parámetro es de suma importancia debido a que a mayor cantidad de corriente por hora el equipo adquiere un tiempo mayor de operación, pero más corriente por hora implica baterías más pesadas y voluminosas. Para seleccionar la batería más adecuada se realizó una estimación usando la siguiente fórmula:

$$I [Ah] = 0.4 * \frac{5}{6}x + 2 * \frac{1}{6}x \quad (4.1)$$

Donde I es la capacidad de corriente eléctrica por hora que debe tener una batería, para que el equipo opere x número de horas. La fórmula se obtuvo de la siguiente manera:

- Sabemos que el tiempo que tardará en descargarse una batería (x) es directamente proporcional a su capacidad de corriente en ampere-hora (I) e inversamente proporcional al consumo de corriente del equipo (i)¹⁷. Es decir que:

$$x(\text{tiempo de descarga})[h] = \frac{I (\text{Valor nominal}) [Ah]}{i (\text{Consumo de corriente}) [A]} \quad (4.2)$$

- Si de la expresión 4.2 despejamos I tendremos que:

$$I [Ah] = i [A] * x [h] \quad (4.3)$$

¹⁷ Boylestad, R. (2004). *Introducción al análisis de circuitos* (p.43). México: Pearson Educación de México, S.A. de C.V.

- Para conocer el consumo de corriente del equipo, se hicieron mediciones en el sistema funcionando y se obtuvo que si el equipo no estaba transfiriendo información a través de la red celular consumía 0.4 [A], mientras que si hacía intercambio de información con la red celular podía usar hasta 2 amperes. Cabe comentar que en las mediciones de corriente que se realizaron no obtuvimos una medición superior a los 1.5 [A], sin embargo como el fabricante indicaba que podía llegar a consumir hasta 2 [A] se prefirió usar este dato para la estimación. Además, se debe aclarar que durante el tiempo en que el equipo transmite información la corriente ocupada es variable, pero para simplificar la fórmula se consideró que consumía 2 [A] durante todo ese lapso.
- Se debe recordar que el equipo transmite información hacia un puesto central cada minuto, y al hacer mediciones del tiempo que pasaba transmitiendo información, llegamos a la conclusión de que el equipo pasaba 10 segundos enviando información. Entonces consideramos que el equipo pasaba 50 segundos sin usar la red celular y 10 segundos usando la red celular. Ahora, si sumamos la cantidad de segundos que el equipo pasa sin usar la red en una hora tendremos 3000 segundos, y a lo largo de una hora el equipo pasa 600 segundos usando la red celular. Estas cantidades representa 5/6 y 1/6 de hora respectivamente.
- Finalmente si sumamos el consumo del equipo sin usar la red celular más el consumo del equipo usando la red celular, obtendremos la ecuación 4.1, la cual si se simplifica queda como:

$$I [Ah] = .4 * \frac{5}{6}x + 2 * \frac{1}{6}x = \frac{2}{6}x + \frac{2}{6}x = \frac{4}{6}x = \frac{2}{3}x \quad (4.4)$$

Usando la fórmula 4.4 pudimos estimar que una batería de 1 [Ah] nos duraría 1.5 [h] y una de 4 [Ah] nos duraría 6 [h]¹⁸. Con esta información decidimos que para ciclos de manejo estándar se podían utilizar baterías de 1 [Ah] y para ciclos de manejo más largos sería conveniente usar baterías de 4 [Ah] o superiores. Cabe comentar que a lo largo de las pruebas de funcionamiento sólo se usó una batería de 1[Ah] a 12 [V], la cual nos dio un tiempo de autonomía ligeramente superior a la hora y media calculada. Por ello, podemos decir que la expresión 4.4 nos da una estimación correcta del funcionamiento del equipo.

4.3. Pruebas de funcionamiento

Tras haber solucionado los problemas de conexión, el sistema *Cycle-DAQ* se puso a prueba para corroborar su funcionamiento. En la figura 4.2 se muestra una sección de uno de los archivos donde se almacena la información recopilada. En la misma figura, en la parte superior izquierda, se muestra el nombre del archivo, el cual como se comentó en el apartado 2.2.5. *Pantalla Táctil*, está formado por el prefijo *ARC*, por dos números (64) que indican el número consecutivo de los archivos almacenados y con extensión *.txt* para indicar que se trata de un archivo de texto.

¹⁸ Los valores nominales de corriente, para baterías de 12 V, más comunes en el mercado son de 1 [Ah] y de 4 [Ah].

En la parte central de la figura se muestra como se almacenan las cadenas de información en el archivo. Como se muestra en la figura, cada cadena ocupa un solo renglón, cada segundo se está guardando una cadena de información distinta y cada cadena está formada por la unión de todos los datos recopilados. Como se comentó en el capítulo anterior, a la información mostrada en la pantalla se le redujo la precisión, sin embargo en el archivo de texto se conservó la precisión de los datos recopilados. Cabe comentar que al final de la cadena de datos se presentan una serie de ceros, los cuales corresponden a la información adquirida del analizador de gases, sin embargo como no se contó con dicho equipo no se pudo recopilar información de emisión de gases. Así mismo, es importante mencionar que el número 6.2 que se escribe al final de las cadenas es un identificador para saber de cual equipo proviene la información almacenada, ya que se pensó en construir más de un equipo *Cycle-DAQ*.

```

ARC64.TXT: Bloc de notas
Archivo Edición Formato Ver Ayuda
28/1/16,16:18:20,19.442368,-99.215454,2303.20,0.00,0,0,0.72,84.38,0,0,0,0,0,6.2
28/1/16,16:18:21,19.442375,-99.215431,2302.00,2.43,7,1.41,85.07,0,0,0,0,0,6.2
28/1/16,16:18:22,19.442396,-99.215416,2302.10,5.37,7,5.09,88.48,0,0,0,0,0,6.2
28/1/16,16:18:23,19.442419,-99.215378,2302.10,10.07,7,4.17,85.98,0,0,0,0,0,6.2
28/1/16,16:18:24,19.442442,-99.215347,2302.10,14.24,7,2.09,86.44,0,0,0,0,0,6.2
28/1/16,16:18:24,19.442442,-99.215347,2302.10,14.24,15,1.18,86.89,0,0,0,0,0,6.2
28/1/16,16:18:25,19.442465,-99.215302,2302.10,19.63,16,4.40,85.53,0,0,0,0,0,6.2
28/1/16,16:18:26,19.442488,-99.215256,2302.20,25.03,16,8.61,86.67,0,0,0,0,0,6.2
28/1/16,16:18:27,19.442509,-99.215210,2302.20,30.43,16,6.96,87.35,0,0,0,0,0,6.2
28/1/16,16:18:28,19.442532,-99.215149,2302.10,37.23,16,3.24,82.55,0,0,0,0,0,6.2
28/1/16,16:18:29,19.442556,-99.215103,2302.20,42.81,16,8.61,85.30,0,0,0,0,0,6.2
28/1/16,16:18:29,19.442556,-99.215103,2302.20,42.81,19,3.01,87.12,0,0,0,0,0,6.2
28/1/16,16:18:30,19.442577,-99.215057,2302.20,48.21,20,6.26,84.84,0,0,0,0,0,6.2
28/1/16,16:18:31,19.442604,-99.214996,2302.20,55.14,20,1.41,85.30,0,0,0,0,0,6.2
28/1/16,16:18:32,19.442623,-99.214943,2302.40,61.12,20,8.14,82.32,0,0,0,0,0,6.2
28/1/16,16:18:33,19.442650,-99.214897,2302.50,66.80,20,4.63,85.53,0,0,0,0,0,6.2
28/1/16,16:18:34,19.442673,-99.214844,2302.50,73.00,20,2.09,82.08,0,0,0,0,0,6.2
28/1/16,16:18:34,19.442673,-99.214844,2302.50,73.00,22,7.67,89.15,0,0,0,0,0,6.2
28/1/16,16:18:35,19.442696,-99.214783,2302.50,79.74,22,4.40,84.61,0,0,0,0,0,6.2
28/1/16,16:18:36,19.442722,-99.214729,2302.60,86.11,22,3.70,85.98,0,0,0,0,0,6.2
28/1/16,16:18:37,19.442751,-99.214668,2302.50,93.19,22,8.37,86.44,0,0,0,0,0,6.2
28/1/16,16:18:38,19.442776,-99.214615,2302.70,99.32,22,5.33,81.16,0,0,0,0,0,6.2
28/1/16,16:18:39,19.442793,-99.214554,2301.80,105.99,22,1.18,85.76,0,0,0,0,0,6.2
28/1/16,16:18:39,19.442793,-99.214554,2301.80,105.99,23,2.55,88.93,0,0,0,0,0,6.2
28/1/16,16:18:40,19.442812,-99.214493,2301.30,112.62,23,3.70,86.67,0,0,0,0,0,6.2

```

Figura 4.2. Archivo de texto generado por el equipo.

En la figura 4.3 se muestra una gráfica de velocidad contra tiempo tal y como se visualiza en la página web del servidor. La información con la que se construye dicha gráfica corresponde a la información enviada por el equipo durante las pruebas. Si se observa detenidamente los puntos graficados, éstos están separados en lapsos menores a un minuto. Cabe comentar, que en la gráfica de la figura 4.3, se presenta información recopilada a lo largo de un día, pero en distintos horarios. Es importante mencionar que la gráfica que se presentan en la página web del servidor, muestra la información de varias pruebas realizadas, como si estas hubiesen sido una sola prueba. Es importante mencionar que hasta el momento de escribir este texto no se ha

configurado la página web para que distinga las pruebas, pero si se tiene la capacidad de borrar la información de la gráfica y comenzar desde cero.

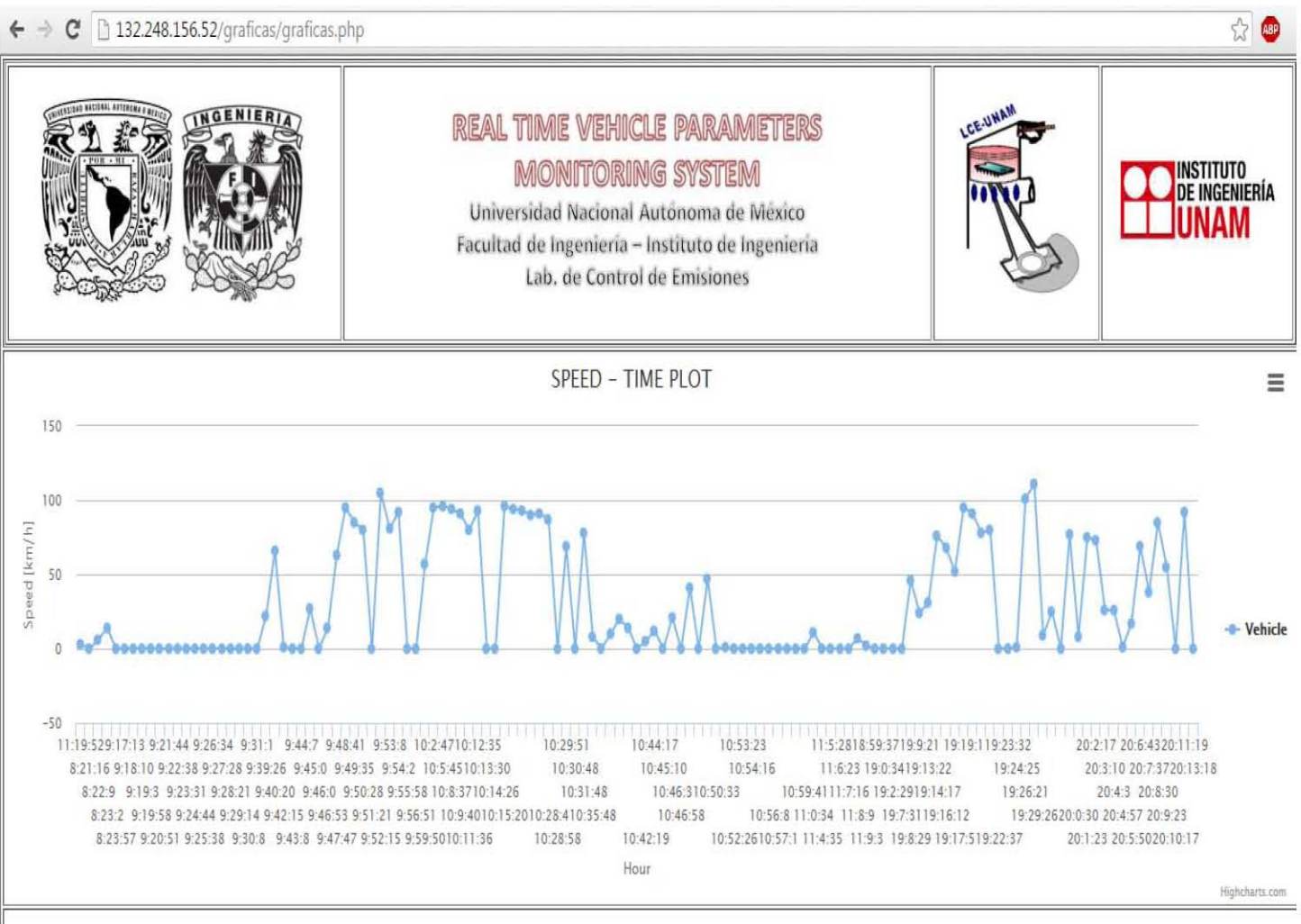


Figura 4.3. Gráfica velocidad contra tiempo mostrada en el servidor web.

En la figura 4.4 se presenta la trayectoria del vehículo cuya gráfica de velocidad contra tiempo se expone en la figura 4.3. Con esta prueba se comprobó definitivamente que el equipo era completamente funcional y robusto. Se debe hacer notar que en la página web también se muestran las trayectorias de los vehículos como se presenta en la figura 4.4. Es importante comentar que las pruebas se comenzaron en la Ciudad de México, se continuaron en la carretera México-Pachuca, posteriormente en la Ciudad de Pachuca y se terminó en la carretera Pachuca-Actopan. Como se presenta en la figura 4.4, la información enviada al servidor coincidió perfectamente con la trayectoria seguida, lo cual demuestra que el equipo tiene una muy buena precisión en la información de localización (latitud y longitud). Así mismo, se debe recalcar que pese a que las pruebas se llevaron a cabo tanto en ciudades como en carreteras federales el equipo nunca perdió su conexión con la red ni presentó problemas.

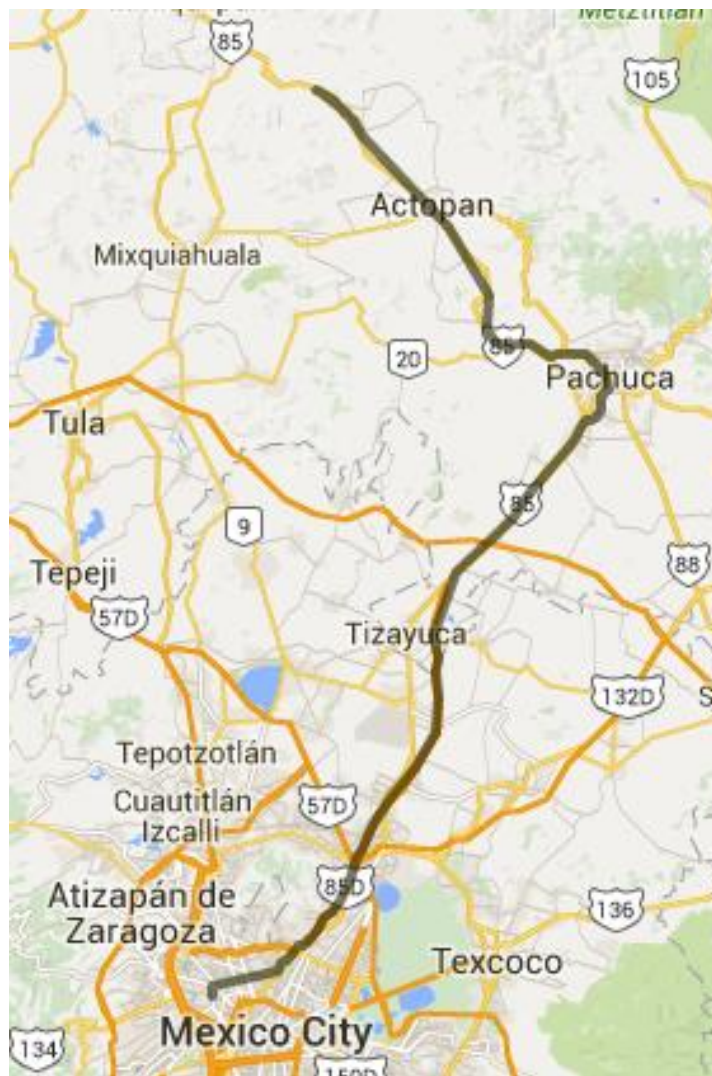


Figura 4.4. Trayectoria de vehículo bajo prueba.

Respecto a otros parámetros como la fecha, la hora, la distancia total recorrida y la velocidad, se compararon contra otros equipos. Por ejemplo, la fecha y la hora se compararon con un reloj digital, la distancia total recorrida se comparó contra la distancia calculada, por programas computacionales en la misma ruta seguida y la velocidad contra lo indicado en el tacómetro del vehículo bajo prueba. Así mismo la altitud de los sitios de prueba se comparó contra valores en tablas de altitud y el funcionamiento del reloj de tiempo real se comprobó por observación directa. Debido a que todos los parámetros se comprobaron y se observaron que eran cercanos a los valores de los patrones utilizados se determinó que ya se tenía un equipo confiable.

4.4. Conclusiones

Con los resultados obtenidos en las últimas pruebas comentadas, se dio por concluido el proyecto y respecto a lo tratado a lo largo de este trabajo de tesis se concluye lo siguiente:

- Para desarrollar un proyecto es necesario conocer y entender todos los pormenores del problema a resolver.
- Una vez que se ha comprendido el problema, se deben buscar las soluciones más adecuadas para el proyecto.
- Las soluciones más adecuadas para un proyecto serán aquellas que puedan resolver el problema del proyecto, pero que permitan respetar el tiempo de entrega y el presupuesto del mismo.
- Antes de comenzar a trabajar en cualquier proyecto se deben delimitar los alcances, objetivos y resultados esperados del proyecto.
- Para desarrollar un proyecto es importante establecer una metodología de trabajo y llevar un registro de las actividades realizadas, así como de los resultados obtenidos tras realizar aplicar cada acción. Esto último es sumamente recomendable ya que la información recopilada facilita la depuración de errores.
- Cuando en un proyecto se tengan que desarrollar sistemas que serán empleados por personas, se recomienda que todo el trabajo esté orientado para que la operación del sistema sea lo más sencillo posible.
- Si durante el desarrollo de un proyecto, se encuentra una mejor solución al problema que la planteada inicialmente, es recomendable evaluar su integración al proyecto.
- Cuando se presenten complicaciones en la aplicación de una solución y no se tiene acceso a otra opción, se recomienda analizar detenidamente cada parte de ella, medir todas las variables posibles de la solución y buscar la mayor cantidad de información posible sobre la solución empleada.
- Al desarrollar un sistema es recomendable realizar pruebas cada vez que se le haga una adecuación, esto permitirá tener certeza de que las adecuaciones están siendo efectivas.

Las conclusiones antes presentadas, engloban el conocimiento adquirido a lo largo de este trabajo y el cual fue aplicado en el desarrollo del equipo. Por lo que al finalizar el proyecto se logró obtener un equipo robusto, que cumplió con todas las características definidas al principio del proyecto, es decir que el sistema es capaz de adquirir, procesar, almacenar y enviar a un puesto central la información de la latitud, la longitud, la altitud, la velocidad, la distancia recorrida, el

ángulo de inclinación y las emisiones contaminantes de un vehículo bajo prueba. Además de que el equipo es completamente portátil, cuenta con una interfaz completamente amigable y diseñada para facilitar la operación del equipo y ofrece información confiable.

No obstante lo anterior, se reconoce que las soluciones discutidas, aunque se consideraron las mejores soluciones posibles, estuvieron limitadas por los conocimientos técnicos que se tenían, el tiempo para desarrollar el proyecto, la infraestructura con la que se contaba y el presupuesto asignado a este proyecto. Por lo anterior, a continuación se presentan algunos puntos que se sugiere sean revisados para proyectos futuros basados en este proyecto:

- **Implementar baterías de ion litio para alimentar el equipo.** Es altamente recomendable investigar el comportamiento del equipo con baterías de ion litio. Ya que a diferencia de las baterías de plomo ácido que se usaron, para las pruebas llevadas a cabo, las baterías de ion litio son ligeras y ocupan menor espacio. Entonces, si se integrasen al sistema, se podría dotar al sistema con mejores características de portabilidad.
- **Utilizar un sistema de potencia más eficiente.** Una de los puntos de mejora del proyecto, es buscar utilizar convertidores DC-DC, los cuales permitirán usar baterías con tensiones eléctricas menores y ocupar más eficientemente la energía del equipo.
- **Cambiar el reloj de tiempo real.** Al estudiar detenidamente el comportamiento del reloj de tiempo real, nos percatamos que la medición del tiempo del dispositivo, al ser alimentado por la batería de respaldo, tenía una variación de tiempo superior a la indicada por el fabricante. Por lo que se recomienda cambiar el reloj de tiempo real por uno que no requiera de un cristal externo para trabajar.
- **Modificar la tarjeta de circuitos impresos.** Se recordará que la tarjeta de circuitos impresos se diseñó inicialmente para descartar problemas de interferencia entre el GPS y la estación móvil celular. Pero como los problemas de conexión no se debían interferencias, sería conveniente cambiar el diseño a una placa de menores dimensiones.

Se considera que si se incluyesen las mejoras mencionadas, el equipo tendría prestaciones superiores.

Antes de terminar con esta sección se debe comentar que desde el punto de vista académico llevar a cabo proyectos, como el presentado, es un proceso muy formativo, ya que se aprende a llevar un proyecto desde el inicio hasta su conclusión, enfrentarse a los problemas del proyecto genera una gran experiencia para establecer metodologías de trabajo y de resolución de problemas. Así mismo, se adquieren conocimientos técnicos y habilidades que complementan la educación recibida en la carrera universitaria.

GLOSARIO DE TÉRMINOS Y ABREVIATURAS

A

AC: *Alternate Current*, Corriente Alterna

AM: Antes de Medio Día

AN: *Application Note*, Nota de Aplicación

APN: *Access Point Name*, Nombre del Punto de Acceso

AT: *Attention*, Atención

ASCII: *American Standard for Information Interchange*, Estándar Americano para el Intercambio de Información

AUC: *Authentication Center*, Centro de Autenticación

B

BCD: *Binary Coded Decimal*, Decimal Codificado en Binario

BGA: *Ball Grid Array*, Arreglo de Cuadrícula de Esferas

BSC: *Base Station Controller*, Controlador de Estación Base

BTS: *Base Transceptor Station*, Estación Transceptora Base

C

CF[®]: *Compact Flash*

CGATT: *Attach to /Detach from GPRS Service*, Vincularse al / Desvincularse del Servicio GPRS

CH: *Clock Halt*, Bit de paro

CI: Coordinación de Ingeniería

CISC: *Complex Instruction Set Computer*, Computadora con Conjunto Complejo de Instrucciones

CPHA: *Clock phase*, Fase de Reloj

CPIN: *Enter PIN*, Ingrese PIN

CPOL: *Clock polarity*, Polaridad de Reloj

CL: *Capacitive Load*, Carga Capacitiva

CPU: *Central Processing Unit*, Unidad de Procesamiento Central

CREG: *Network Registration*, Registro en Red

CS, \overline{CS} : *Chip Select*, Selección de Chip

CSD: *Circuit Switched Data*, Circuito Conmutado de Información

CTR: *Cathode Ray Tube*, Tubo de Rayos Catódicos

D

DC: *Direct Current*, Corriente Directa

DI: véase *MISO*

DOP: *Dilution of Precision*, Dilución de Precisión

DO: véase *MOSI*

E

EIA: *Electronic Industries Alliance*, Alianza de Industrias Electrónicas

EIR: *Equipment Identity Register*, Registro de Identificación de Equipo

EN: *European Standard*, Estándar Europeo

EOC: *End of Clusters*, Final de los Clusters

ETSI: *European Telecommunications Standards Institute*, Instituto Europeo de Estándares de Telecomunicaciones

F

FAT: *File Allocation System*, Sistema de Asignación de Archivos.

FDMA: *Frequency Domain Multiple Access*, Acceso Multiple por División de Frecuencia

FI: Facultad de Ingeniería

FIFO: *First Input First Output*, Primera Entrada Primer Salida

FTP: *Federal Test Procedure*, Procedimiento Federal de Pruebas

G

GGA: *Global Positioning System Fixed Data*, Información Establecida del Sistema Global de Posicionamiento

GGSN: *Gateway GPRS Support Node*, Nodo de Soporte de la Compuerta GPRS

GLL: *Geographic Position Latitude- Longitude*, Latitud y Longitud de la Posición Geográfica.

GNSS: *Global Navigation Satellite System*, Sistema Satelital de Navegación Global

GPS: *Global Positioning System*, Sistema Global de Posicionamiento

GPRS: *General Packet Radio Services*, Servicio General de Paquetes Vía Radio

GSA: *GNSS DOP and Active Satellites*, DOP y Satélites Activos del GNSS

GSM: *Global System for Mobile Communications*, Sistema Global para las Comunicaciones Móviles

GSV: *GNSS Satellites in View*, Satélites Visibles del GNSS

H

HDOP: Horizontal DOP, *véase DOP*

HLR: *Home Location Register*, Registro de Ubicación de Base

HTTP: *Hypertext Transfer Protocol*, Protocolo de Transferencia de Hipertexto

I

II: Instituto de Ingeniería

IC: *Inter-Integrated Circuits*, Circuitos Inter-Integrados

IP: *Internet Protocol*, Protocolo de Internet

IPC: *Institute of Printed Circuits*, Instituto de Circuitos Impresos

IS: *Interim Standard*, Estándar Interno

ISDN: *Integrated Services for Digital Networks*, Servicios Integrados para Redes Digitales.

L

LC: *Liquid Crystal*, Cristal Líquido

LCD: *Liquid Crystal Display*, Pantall de Cristal Líquido

LCE: Laboratorio de Control de Emisiones

LED: *Light Emitting Diode*, Diodo Emisor de Luz

LGA: *Land Grid Array*, Arreglo de Cuadrícula de Terminales Ocultas

LSB: *Less Significant Bit*, Bit Menos Significativo

M

M2[®]: *Memory Stick Duo*

MAP: *Mobile Application Part*, Parte de Aplicación Móvil

MCS: *Master Control Station*, Estación Maestra de Control

MEMS: *Microelectromechanical Systems*, Sistemas Microelectromecánicos

MI: Maestro en Ingeniería

MISO: *Master Input Master Output*, Entrada Maestra Salida Maestra

MOSI: *Master Output Master Input*, Salida Maestra Entrada Maestra

MS: *Mobile Station*, Estación Móvil

MS[®]: *Memory Stick*

MSB: *Most Significant Bit*, Bit Más Significativo

MSC: *Mobile Switching Center*, Centro Conmutador Móvil

N

NSS: *Network Subsystem*, Subsistema de Red

NMEA: *National Marine Electronics Association*, Asociación Nacional Electrónica Marítima

NOM: Norma Oficial Mexicana

P

PCB: *Printed Circuits Board*, Tarjeta de Circuitos Impresos

PCU: *Packet Control Unit*, Unidad de Control de Paquetes

PIN: *Personal Identification Number*, Número Personal de Identificación

PM: Después de Medio Día

PSTN: *Public Switched Telephone Network*, Red de Telefonía Pública Conmutada

PUK: *Personal Unlock Key*, Número Personal de Desbloqueo

Q

QIACT: *Activate GPRS/CSD Context*, Activar el contexto GPRS/CSD

QICSGP: *Select CSD or GPRS as the Bearer*, Seleccionar el CSD o el GPRS como Portador

QIDNSIP: *Connect with IP Address or Domain Name Server*, Conectarse con la Dirección IP o el Nombre del Dominio del Servidor

QIFGCNT: *Select a Context as Foreground Context*, Elegir un Contexto como Contexto de Primer Plano

QINDI: *Set the Method to Handle Received TCP IP Data*, Establecer el Método para Manejar la Información TCP IP recibida

QILOCIP: *Get Local IP Address*, Obtener una Dirección Local IP

QIMODE: *Select TCPIP Transfer Mode*, Seleccionar Modo de Transferencia TCPIP

QIMUX: *Control Whether or Not to Enable Multiple TCPIP Sessions*, Controlar si Habilitar o No Múltiples Sesiones TCP IP

QIOPEN: *Start up TCP or UDP Connection*, Iniciar una Conexión TCP o UDP.

QIREGAPP: *Start TCPIP Task and Set APN, User Name and Password*, Iniciar las Tareas TCP IP y Establecer el APN, el Nombre de Usuario y Contraseña.

R

RAM: *Random Access Memory*, Memoria de Acceso Aleatorio

RISC: *Reduced Instruction Set Computer*, Computadora de Conjunto de Instrucciones Reducidas

RS: *Recommended Standard*, Estándar Recomendado

RMC: *Recommended Minimum Specific GNSS Data*, Información del GNSS Específica Mínima Recomendada

RTC: *Real Time Clock*, Reloj de Tiempo Real

Rx: Recepción

S

SEMARNAT: Secretaría de Medio Ambiente y Recursos Naturales

SCK: *Serial Clock*, Reloj de Comunicación Serie

SCL: *véase SCK*

SD[®]: *Secure Data*

SDA: *Serial Data*, Información serie

SDSC: *SD Standard Security*, Estándar de Seguridad para SD

SDHC: *SD High Capacity*, SD de Alta Capacidad

SDXC: *SD Extended Capacity*, SD de Capacidad Extendida

SGSN: *Service GPRS Support Node*, Nodo de soporte de servicio GPRS

SIM: *Subscriber Identity Module*, Módulo de Identificación del Suscriptor.

SMD: *Surface Mounted Device*, Dispositivo de Montaje Superficial.

SPI: *Serial Peripheral Interface*, Interfaz de Comunicación Serie de Periféricos.

SPIBR: *SPI Baud Rate*, Tasa de baudios de SPI

SPICR1: *SPI Control Register 1*, Registro de Control 1 de SPI

SPICR2: *SPI Control Register 2*, Registro de Control 2 de SPI

SPIDR: *SPI Data Register*, Registro de Información de SPI

SPISR: *SPI Status Register*, Registro de Estado de SPI.

SRAM: *Static Random Access Memory*, Memoria Estática de Acceso Aleatorio

\overline{SS} : *Slave Select*, Selección de Esclavo

SS-7: *Common Channel Signaling System No. 7*, Sistema de Señalización de Canal Común No. 7.

T

TCP: *Transmission Control Protocol*, Protocolo de Control de Transmisión

TDMA: *Time Division Multiple Access*, Acceso Múltiple Por División de Tiempo

TIA: *Telecommunications Industries Association*, Asociación de Industrias de Telecomunicaciones.

TS: *Technical Specification*, Especificación Técnica

TSSOP: *Thin-Shrink Small Outline Package*, Encapsulado de Contorno Pequeño Delgado-Reducido

TWI: *Two Wire Interface*, Interfaz de Dos Hilos

Tx: Transmisión

U

UART: *Universal Asynchronous Receiver/Transmitter*, Receptor-Transmisor Universal Asíncrono.

UDDS: *Urban Dynamometer Driving Schedule*, Programa de Conducción de Dinamómetro Urbano

UDP: *User Datagram Protocol*, Protocolo de Datagrama de Usuario

UHS: *Ultra High Speed*, Ultra Alta Velocidad

UNAM: Universidad Nacional Autónoma de México

USB: *Universal Serial Bus*, Bus Universal de Comunicación Serie

V

V_{bat}: Tensión Eléctrica de Batería.

V_{cc}: *Voltage Colector-Colector*, Tensión Eléctrica Colector a Colector

V_{DD}: *Voltage Drain-Drain*, Tensión Eléctrica Drenador a Drenador

VDOP: Vertical DOP

VLR: *Visitor Location Register*, Registro de Ubicación de Visitante

BIBLIOGRAFÍA

BIBLIOGRAFÍA

LIBROS

Boylestad, R. (2004). *Introducción al análisis de circuitos*. (Décima Edición, p.43). México: Pearson Educación de México, S.A. de C.V.

Davies, J. (2008). *MSP430 Microcontroller Basics*. Estado Unidos: Newnes

Di Jasio, L. (2007). *Programming 16-bit PIC Microcontrollers in C: learning to Fly the PIC 24*. Estados Unidos: Elsevier.

El-Rabani, A. (2002). *Introduction to GPS: The Global Positioning System*. Massachusetts, Estados Unidos: Artech Hous, Inc.

Gad-el-Hak, M.(2006). *MEMS Applications*. Florida, Estados Unidos: CRC Press.z

Halonen, T., Romero, J. & Melero, J. (2003). *GSM, GPRS and EDGE Performance*.(2da. Edición). West Sussex, Inglaterra: John Wiley & Sons Ltd.

Hecht, E. (2002). *Optics*. (Cuarta Edición). Estados Unidos: Pearson Education, Inc.

Heine, G. (1999). *GSM Networks: Protocols, Terminology and Implementation*. Massachussetss, Estados Unidos: Artech House, Inc.

Ibrahim, D. (2010). *SD Card Projects Using the PIC Microcontroller*. Estados Unidos: Newnes.

Jamsa, K. (2013). *Rescued by A+ Certification Exam Preparation*.(pp. 394-395). Canadá: Thomson

Phillips, H. (2012). *New Perspectives on Microsoft Windowa 7 for Power Users, Advanced*. Boston Estados Unidos: Cenage Learning.

Purdum, J. (2012). *Beginning C for Arduino*. Nueva York, Estados Unidos: Apress.

Sánchez-Gómez, P. (2005). *Análisis del desempeño de las redes Celulares GSM-GPRS*. (Tesis de licenciatura). Universidad Nacional Autónoma de México. Ciudad de México, México.

Sanders, G., Thorens L., Reisky, M., Rulik, O. & Deylitz S. (2004). *GPRS Networks*. West Sussex, Inglaterra : John Wiley & Sons Ltd.

Sauter, M. (2011). *From GSM to LTE: an Introduction to Mobile Networks and Mobile Broadband*. West Sussex, Inglaterra: John Wiley & Sons Ltd.

Secretaría de Comunicaciones y Transportes. Manual de Diseño Geométrico de Vialidades. En el *Programa de Asistencia Técnica en Transporte Urbano para las Ciudades Medias Mexicanas* (Tomo 4, pp. 17-18). México: Autor.

Ramírez-Chavarría, G. (2013). *Adquisición y Registro de Parámetros Vehiculares para el Desarrollo de Ciclos de Manejo en el Valle de México*. (Tesis de licenciatura). Universidad Nacional Autónoma de México. Ciudad de México, México.

GUÍAS, MANUALES Y NOTAS DE APLICACIÓN

Brady, M. J. & O'Mahony, M. (2013). *The Development of a Driving Cycle for the Greater Dublin Area Using a Large Database of Driving Data with Stochastic and Statistical Methodology*. Proceedings of the ITRN2013. Dublín, Irlanda.

Irazabal, J-M. & Blozis, S. (2003). *AN10216-01 I²C Manual*. California, Estados Unidos: Philips Semiconductors.

Keithly (2001). *Data Acquisition and Control Handbook: A Guide to Hardware and Software for Computer-Based Measurement and Control*. Ohio, Estados Unidos: Autor.

Motorola, Inc. (2003). *SPI Block Guide V03.06*. Autor

Panasonic, SanDisk & Toshiba. (2013). *SD Specifications Part1 Physical Layer Simplified Specification Version 4.10*. Autor.

Quectel Wireless Solutions Co. Ltd. (2010). *M10 Quectel Cellular Engine*. Autor.

MESOGRAFÍA

LIBROS

Andrejašic, M. (2008). *MEMS Accelerometers*. Universidad de Ljubljana. Ljubljana, Eslovenia. Recuperado el 20 de mayo de 2015 de http://mafija.fmf.uni-lj.si/seminar/files/2007_2008/MEMS_accelerometers-koncna.pdf

Bangia, R. (2007). *Dictionary of Information Technology*. (p.498). Nueva Delhi, India: Firewall Media. Recuperado el 15 de marzo de 2016 de la base de datos de Google Libros

Barret, S. F. & Pack, D.J. (2006). *Microcontrollers Fundamentals for Engineers and Scientists*. DOI: 10.2200/S00025ED1V01Y200605DCS001.

Chollet, F. & Haobing L. (2015). *A (not so) short introduction to MEMS*. (Versión 5.2). Recuperado de: <http://memscopylodedia.org/introMEMS.html>

Cross, M. (2008). *Scene of the Cybercrime*. (Segunda Edición, pp.175-178). Massachusetts, Estados Unidos: Elsevier. Recuperado de la base de datos de Google Libros.

Faiz, A., Weaver, C. S. & Walsh M.P. (1996). *Air Pollution from Motor Vehicles: Standards and Technologies for Controlling Emissions*. Estados Unidos: The International Bank. Recuperado de la base de datos de Google Libros.

Richardson, J.H. & Peterson, R.V. (1974). *Systematic Materials Analysis Volume III*. (pp. 139-141). Nueva York, Estados Unidos: Academic Press, Inc. Recuperado de la base de datos de Google Libros.

Sinclair, I. (2001). *Sensors and Transducers*. (Tercera Edición). Oxford, Gran Bretaña: Newnes.
Sencar, H. T. & Memon N. (2013). *Digital Image Forensics: There is More to a Picture than Meets the Eye*. (pp. 124-127). Nueva York, Estados Unidos: Springer. Recuperado de la base de datos de Google.

GUÍAS, MANUALES, NOTAS DE APLICACIÓN, NOTAS DE CONFERENCIAS Y OTROS TEXTOS

Ababeil, C. (2013). *Lecture 12: SPI and SD cards*. Recuperado el 23 de junio de 2015 de http://www.dejazzer.com/ee379/lecture_notes/lec12_sd_card.pdf

ActWireless. *How Wireless Works*. Recuperado el 26 de julio de 2015 de <http://www.actwireless.org/coverage-and-speed/how-wireless-works/>

Burgess, P. (2014). *Adafruit GFX Graphics Library*. Recuperado el 10 de marzo de 2015 de <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-gfx-graphics-library.pdf>

Chapter 5: GSM Architecture and Interfaces. Recuperado el 28 de julio de 2015 de <https://www.pearsonhighered.com/samplechapter/0139491244.pdf>

Chapter 8: Multiple Acces Techniques. Recuperado el 27 de julio de 2015 de http://www.iitg.ernet.in/scifac/qip/public_html/cd_cell/chapters/a_mitra_mobile_communication/chapter8.pdf

Downs, R. (2005). *Using resistive touch screens for human/machine interface*. Recuperado el 12 de mayo de 2015 de <http://www.ti.com/lit/an/slyt209a/slyt209a.pdf>

Earl, B. (2013). *Arduino Memories*. Recuperado el 26 de octubre de 2015 de <https://learn.adafruit.com/memories-of-an-arduino/arduino-memories>

Fisher, C.J., Tsuzuki, T. & Lee J., *Utilization of the First In, First Out (FIFO) Buffer in Analog Devices, Inc. Digital Accelerometers*. Recuperado el 20 de marzo de 2015 de <http://www.analog.com/media/cn/technical-documentation/application-notes/AN-1025.pdf>

Henry-Labordère, A. & Jonack V. (2004). *SMS and MMS Interworking in Mobile Networks*. (pp 1- 12). Massachusetts, Estados Unidos: Artech House. Recuperado de la base de datos de Google Libros

Illuminet, *Signaling System 7 (SS7)*. Recuperado el 28 de julio de 2015 de <http://www.cs.rutgers.edu/~rmartin/teaching/fall04/cs552/readings/ss7.pdf>

Fisher, C.J., Tsuzuki, T. & Lee J., *Utilization of the First In, First Out (FIFO) Buffer in Analog Devices, Inc. Digital Accelerometers*. Recuperado el 20 de marzo de 2015 de <http://www.analog.com/media/cn/technical-documentation/application-notes/AN-1025.pdf>

Henry-Labordère, A. & Jonack V. (2004). *SMS and MMS Interworking in Mobile Networks*. (pp 1- 12). Massachusetts, Estados Unidos: Artech House. Recuperado de la base de datos de Google Libros

Illuminet, *Signaling System 7 (SS7)*. Recuperado el 28 de julio de 2015 de <http://www.cs.rutgers.edu/~rmartin/teaching/fall04/cs552/readings/ss7.pdf>

K2B2 Enviromental Equipment Services Co. *Infrared Gas Analyzers and Gas Filter Correlation*. Recuperado el 1 de junio de 2015 de http://www.k2bw.com/5_c_18.htm

Kolokowsky, S. (2007). *Touchscreens 101: Understanding Touchscreen Technology and Design*. Recuperado el 20 de agosto de 2015 de <http://www.cypress.com/file/95156/download>

Maxim Integrated. (2002). *Design Consideration for Maxim Real-Time Clocks*. Recuperado el 28 de abril de 2016 de <http://pdfserv.maximintegrated.com/en/an/AN504.pdf>

National Coordination Office for Space-Based Positioning, Navigation and Timing.(2015). *The Global Positioning System*. Recuperado el 23 de mayo de 2015 de <http://www.gps.gov/systems/gps/>

Nova Analytical Systems. *Portable Engine Exhaust Analyzers*. Recuperado el 26 de mayo de 2015 de <http://www.nova-gas.com/analyzers/engine-exhaust>

Omega. *Data Acquisition Systems*. Recuperado el 22 de enero de 2015 de <http://www.omega.com/techref/pdf/dasintro.pdf>

Poole, I. *GSM Frame Structure*. Recuperado el 29 de julio de 2015 de http://www.radio-electronics.com/info/cellulartelecomms/gsm_technical/frames-structure-super-hyper.php

Propiedades de las Mezclas. Recuperado el 13 de junio de 2015 de http://www.iessierradeguara.com/documentos/departamentos/automocion/circuitos_auxiliares/Mezclas%20y%20gases%20de%20escape/combustion_mezcla.pdf

Raivis. (2014). *MEMS (Part 1) – Guide to using accelerometer ADXL345*. Recuperado el 29 de abril de 2015 de <http://morf.lv/modules.php?name=tutorials&lasit=31>

RP Photonics Encyclopedia. *Birefringence*. Recuperado el 15 de junio de 2015 de <https://www.rp-photonics.com/birefringence.html?s=ak>

Royal Academy of Engineering. *The Mathematics of Engineering Surveying (3)*. Recuperado de: <http://www.raeng.org.uk/publications/other/10-resection>

Scattering. Recuperado el 3 de junio de 2015 de http://www.geo.mtu.edu/~scarn/teaching/GE4250/scattering_lecture_slides.pdf

Schwartz, M. (2004). *Mobile Wireless Communications*. Nueva York, Estados Unidos: Cambridge University Press.

SD Association. *About SD Card Choices*. Recuperado el 30 de julio de 2015 de <https://www.sdcard.org/consumers/choices/index.html>

Secretaría de Gobernación. (1993). *Norma Oficial Mexicana NOM-CCAT-010-ECOL/1993 que establece las características del procedimiento de medición para la verificación de los niveles de emisión de contaminantes, provenientes de los vehículos automotores en circulación que usan gasolina, gas licuado de petróleo, gas natural y combustibles alternos*. Recuperado el 13 de febrero de 2015 de http://dof.gob.mx/nota_detalle.php?codigo=4795482&fecha=22/10/1993

Silabs. (2013). AN0030- Application Note . En *Fat on SD Card*. Recuperado el 16 de abril de 2015 de <https://www.silabs.com/Support%20Documents/TechnicalDocs/AN0030.pdf>

Wireless Multiple Access. Recuperado el 28 de julio de 2015 de http://user.it.uu.se/~carle/datakomm/Notes/Networking/23_WirelessAccess.html

Wong, J.Y. (2013). *Non-Dispersive Infrared (NDIR) Gas Measurement: Past, Present & Future*. Recuperado el 1 de junio de 2015 de http://www.iaria.org/conferences2013/filesSENSORDEVICES13/Jacob_SensorDevices2013_Keypnote.pdf

WTEC Hyper-Librarian. (2000). *Multiple-Access Techniques*. Recuperado el 27 de julio de 2015 de http://www.wtec.org/loyola/wireless/02_04.htm