



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

TESIS

Diseño y desarrollo de un sistema de adquisición y envío de datos del pulso radial y movilidad

QUE PARA OBTENER EL TÍTULO DE INGENIERO ELÉCTRICO ELECTRÓNICO

PRESENTA:

Cabañas Castillo Fidel Eduardo

DIRECTOR DE TESIS:

Ing. Juan Eduardo Murrieta León

CIUDAD UNIVERSITARIA

Agosto 2014



Dedicatorias

A mis padres:

Fidel Lidio y María Teresa

Por darme siempre su amor y comprensión, por apoyarme en las decisiones que he tomado. Gracias por enseñarme que tengo que luchar por mis sueños.

A mis Hermanos:

Daniel, Jorge y Areli

Por brindarme tantas alegrías. Les agradezco su cariño y apoyo en los momentos más difíciles. Nos esperan muchas alegrías, luchen por sus sueños. Ustedes junto con mis padres son mi más grande inspiración para seguir adelante.

Agradecimientos:

A Dios por estar siempre conmigo y ayudarme en los momentos más difíciles de la vida.

A mis padres y hermanos, por ser uno de los motores más importantes de mi vida.

A la Universidad Nacional Autónoma de México, por permitirme realizar una carrera y por todos los beneficios que recibí.

A la Facultad de Ingeniería y a todos mis profesores, por todos los conocimientos que me han transferido.

Al Instituto de Ciencias Nucleares (ICN), por brindarme la oportunidad de realizar mi servicio social y este trabajo de tesis.

A mi director de tesis Ing. Juan Eduardo Murrieta León, por todo el apoyo y dedicación brindados en la elaboración de este trabajo de tesis.

Al M. F Benjamín Alejandro Morales Ruíz, responsable del área de vinculación del Instituto de Ciencias Nucleares, por darme la oportunidad de trabajar en el proyecto VACS y por el apoyo recibido.

Al M.I Enrique Patiño, por su valiosa asesoría durante mi estancia en el ICN, pero sobre todo por su amistad.

Al laboratorio de electrónica del Centro de Ciencias de la Atmosfera, por brindarme la oportunidad de dar mis primeros pasos en la electrónica.

A mis compañeros de la facultad por todas las experiencias vividas en especial: César, Valera, Héctor, Andrés, Tony, Marco, Diego, Edher, Richi, Franco, Nayeli, Daniel, Fabio, Metzli, Yesenia, Ivan, Claudia, Adrián Jiménez, Patricia, Erick y Miguel.

A mis amigos del C.C.H que me siguen honrando con su amistad: Arturo, José María, Axosco, Urrutia, Jesús, Cecilia, Yovaní, Isaac, Eduardo, Josué, Erick, Noe, Rubén.

A Javier, Omar, Néstor, Sócrates, Adrián de la Cruz, por su valiosa amistad y consejos.

A todas las personas que conocí durante mi estancia en la UNAM: Nelly, Juan Carlos, Greta, Héctor, Isaí, César Tavera, Dioscelina, Linda, Omar, Melba, Norma y Susana.

Y por último, a todos los que alguna vez caminaron conmigo que aunque hoy nuestros caminos y metas se encuentren muy separados, me alegra saber que muchos de ustedes siguen sus sueños e ideales.

Contenido

Índice de figuras	IV
Índice de tablas	VIII
1. Introducción	1
1.1. Proyecto VACS	1
1.2. Movilidad	2
1.2.1. Caídas	2
1.3. Sistema cardiovascular	3
1.3.1. Ciclo Cardíaco	5
1.3.2. Pulso arterial	6
1.3.3. Arritmias	8
1.4. Fotopletismografía	8
2. Antecedentes del proyecto	11
2.1. Sensores	11
2.1.1. Sensores fotoeléctricos	11
2.1.2. Sensores de temperatura	12
2.1.3. Acelerómetro	14
2.2. Acondicionamiento de señales	16
2.2.1. Amplificación	16
2.2.2. Filtrado	21
2.3. Conversión análoga digital	29
2.3.1. Convertidor de analógico digital de aproximaciones sucesivas	31
2.3.2. Teorema de muestreo de Nyquist	32
2.4. Comunicación serie	33
2.4.1. Estándar SPI	34
2.4.2. UART	36
2.5. GPS	38
2.5.1. Estándar NMEA	40
2.6. GPRS	43
2.6.1. Comandos AT	45
2.7. Tarjeta de memoria microSD	46

2.7.1.	Comandos CMD y respuesta	46
2.7.2.	Respuestas en modo SPI	48
2.8.	Sistema de archivos FAT	50
2.8.1.	Región de arranque	51
2.8.2.	Directorio FAT	52
2.8.3.	Directorio de raíz	53
2.8.4.	Región de datos	54
2.9.	Microcontrolador	55
2.9.1.	Arquitectura de los microcontroladores	55
2.9.2.	Familias de microcontroladores	57
2.10.	Lenguajes de programación	58
3.	Diseño y desarrollo	63
3.1.	Requerimientos y especificaciones de diseño	63
3.2.	Propuesta de solución	64
3.2.1.	Selección de dispositivos	64
3.2.2.	Sensores	65
3.2.3.	Almacenamiento de datos	67
3.2.4.	Ubicación geográfica	68
3.2.5.	Envío de datos	69
3.2.6.	Microcontrolador	69
3.2.7.	Potencia	71
3.3.	Desarrollo de hardware	72
3.3.1.	Acondicionamiento de señales	73
3.3.2.	Almacenamiento de datos	79
3.3.3.	GPS	79
3.3.4.	GPRS	80
3.3.5.	Potencia	80
3.3.6.	Integración de hardware	82
3.4.	Desarrollo de Software	84
3.4.1.	Adquisición de datos	85
3.4.2.	GPS	87
3.4.3.	Almacenamiento de datos	89

3.4.4.	Envío de datos _____	99
3.5.	Procesamiento de datos _____	104
3.5.1.	Entorno de desarrollo netbeans IDE _____	104
3.5.2.	Programa para visualización de datos _____	105
4.	Integración y pruebas _____	108
4.1.	Integración del sistema en PCB _____	108
4.1.1.	Pruebas al módulo de potencia del sistema de adquisición _____	110
4.2.	Pruebas de funcionamiento preliminares _____	110
4.2.1.	Pruebas realizadas al módulo acondicionador de señal _____	111
4.2.2.	Pruebas realizadas al módulo GPS _____	113
4.2.3.	Pruebas realizadas al módulo de almacenamiento de datos _____	114
4.2.4.	Pruebas realizadas al módulo de envío de datos _____	116
4.3.	Prueba final _____	118
5.	Resultados y conclusiones _____	123
5.1.	Resultados _____	123
5.2.	Conclusiones _____	124
5.3.	Trabajo a futuro _____	125
5.4.	Recomendaciones _____	125
A.	Anexo I _____	128
	Diseño de filtros _____	128
	Filtro paso bajas de Bessel _____	128
	Filtro paso bajas Butterworth _____	131
	Conclusiones _____	133
	Bibliografía y referencias _____	135

Índice de figuras

Capítulo 1

Figura 1.1. Sistema cardiovascular.....	4
Figura 1.2 Diagrama de Wigger.....	6
Figura 1.3. Trayectoria de luz en la fotopleletismografía.....	9
Figura 1.4. Cantidad de luz que se absorbe por las distintas partes del cuerpo.....	9
Figura 1.5. Áreas para detectar la señal de pulso, clasificadas por métodos de detección.....	10
Figura 1.6. Métodos de detección del fotopleletismógrafo.....	10

Capítulo 2

Figura 2.1. Curvas caracteristicas del fotodiodo.....	12
Figura 2.2. Esquema de un fototransistor NPN.....	12
Figura 2.3. Capacitor diferencial.....	14
Figura 2.4. Símbolos del TBJ.....	17
Figura 2.5. Amplificador emisor común.....	17
Figura 2.6. Amplificador colector común.....	18
Figura 2.7. Amplificador base común.....	18
Figura 2.8. Representación de un amplificador operacional.....	19
Figura 2.9. Circuito amplificador inversor.....	20
Figura 2.10. Amplificador no inversor.....	20
Figura 2.11. Configuración seguidor de voltaje.....	21
Figura 2.12. Tipos de filtros de acuerdo a su respuesta en frecuencia.....	22
Figura 2.13. Respuesta del filtro Butterworth.....	23
Figura 2.14. Respuestas del filtro Chebyshev.....	24
Figura 2.15. Filtro de realimentación múltiple.....	25
Figura 2.16. Diagrama General del filtro Sallen Key.....	26
Figura 2.17. Filtro paso banda de realimentación múltiple.....	27
Figura 2.18. Filtro Sallen Key paso bajas.....	28
Figura 2.19. Proceso de conversión analógico digital.....	29
Figura 2.20. Diagrama de bloques de un convertidor de aproximaciones sucesivas.....	31
Figura 2.21. Espectro de una señal continua limitada en banda.....	32
Figura 2.22. Señal muestreada siguiendo el teorema de Nyquist.....	32
Figura 2.23. Espectro de una señal muestreada limitada en banda.....	32
Figura 2.24. Señal muestreada con superposición espectral.....	33
Figura 2.25. Señal submuestreada.....	33
Figura 2.26. Conexión SPI con un solo esclavo.....	34
Figura 2.27. Conexión SPI multi esclavos.....	35
Figura 2.28. Modos de operación del estándar SPI.....	36
Figura 2.29. Conexión UART.....	36

Figura 2.30. Protocolo de comunicación asíncrona serie.	37
Figura 2.31. Principio de funcionamiento del GPS.	39
Figura 2.32. Pseudorange de tiempo GPS.	40
Figura 2.33. Formato de comandos NMEA.	41
Figura 2.34. Comando PMTK para cambiar la velocidad de transmisión.	41
Figura 2.35. Comando PMTK para habilitar mensajes.	41
Figura 2.36. Mensaje NMEA.	42
Figura 2.37. Desarrollo de generaciones móviles a través de los años.	43
Figura 2.38. Diagrama general de la arquitectura del GSM.	44
Figura 2.39. Memoria microSD.	46
Figura 2.40. Diagrama de terminales de la memoria microSD.	46
Figura 2.41. Respuesta R1.	48
Figura 2.42. Respuesta R2.	48
Figura 2.43. Respuesta R3.	49
Figura 2.44. Respuesta R7.	50
Figura 2.45. Estructura del sistema de archivos FAT.	51
Figura 2.46. Estructura del sector cero de FAT 16.	52
Figura 2.47. Directorio FAT 16.	53
Figura 2.48. Arquitectura Von Neumann.	56
figura 2.49. Arquitectura Harvard.	57
Figura 2.50. Estructura secuencial.	59
Figura 2.51. Estructura selectiva.	59
Figura 2.52. Estructura repetitiva.	60

Capítulo 3

Figura 3.1. Esquema general del proyecto VACS.	63
Figura 3.2. Diagrama conceptual del sistema de adquisición.	64
Figura 3.3. Fototransistor QRE1113.	65
Figura 3.4. Gráficas características del fototransistor QR1113.	65
Figura 3.5. Tarjeta microSD.	68
Figura 3.6. GPS Fastrax UP501.	68
Figura 3.7. Modem ENFORA GSM0308.	69
Figura 3.8. Microcontrolador Stellaris LM3S8962.	70
Figura 3.9. Batería de LIPO.	71
Figura 3.10. MCP73831.	72
Figura 3.11. Diagrama de bloques de la interacción del microcontrolador con los otros dispositivos.	73
Figura 3.12. Diagrama usado para adquirir el pulso radial.	74
Figura 3.13. Circuito de amplificación para el fotodiodo.	74
Figura 3.14. Respuesta en frecuencia del filtro paso banda.	76
Figura 3.15. Acelerómetro LIS3LV02DL.	77
Figura 3.16. Diagrama de conexión básica (rango 2°C a 150°C).	78
Figura 3.17. Circuito divisor de voltaje para la batería.	78
Figura 3.18. Diagrama de conexión memoria microSD.	79
Figura 3.19. Conexión del microcontrolador con el GPS.	80
Figura 3.20. Diagramas de conexión modem GPRS.	80

Figura 3.21. Diagrama general del subsistema de potencia	81
Figura 3.22. Diagrama de conexiones del MCP73831	81
Figura 3.23. Regulador a 1.8 [V].....	82
Figura 3.24. Regulador a 3.3 [V].....	82
Figura 3.25. Prueba de integración del sistema	83
Figura 3.26. Capa superior del circuito impreso	84
Figura 3.27. Cara inferior del circuito impreso	84
Figura 3.28. Adquisición de datos.....	87
Figura 3.29. Paquete de datos antes de obtener la ubicación	88
Figura 3.30. Paquete de datos después de establecer la ubicación	88
Figura 3.31. Diagrama de flujo para configurar el GPS.....	88
Figura 3.32. Máquina de estados para obtener la posición del GPS	89
Figura 3.33. Diagrama de flujo para inicializar la tarjeta microSD.	91
Figura 3.34. Diagrama de flujo para escribir en un sector de la memoria microSD.	93
Figura 3.35. Diagrama de flujo para leer un sector de la memoria microSD.	94
Figura 3.36. Sector cero de la microSD.....	95
Figura 3.37. Diagrama de flujo para crear un nuevo archivo.....	96
Figura 3.38. Diagrama de flujo para buscar una entrada libre	97
Figura 3.39. Directorio de raíz	98
Figura 3.40. Diagrama de flujo para escribir datos en un archivo	98
Figura 3.41. Diagrama de flujo para actualizar datos en un archivo	99
Figura 3.42. Inicializar modem	100
Figura 3.43. Función comando	101
Figura 3.44. Función busca	102
Figura 3.45. Entorno de desarrollo netbeans IDE.....	105
Figura 3.46. Plantilla del programa para vizualizar los datos	105
Figura 3.47. Graficas de las señales senoidales.....	107
Figura 3.48. Graficas de las señales senoidales con la barra desplazada	107

Capítulo 4

Figura 4.1. Capa superior del PCB del sistema	108
Figura 4.2. Capa inferior del PCB del sistema	108
Figura 4.3. Capa superior del sistema de adquisición de pulso radial	109
Figura 4.4 Capa inferior del sistema de adquisición de pulso radial	109
Figura 4.5. Diagrama de conexiones entre la sistema de adquisición (H1) y el programador XDS100V3.....	111
Figura 4.6. Conexión del programador con el sistema de adquisición de datos.....	111
Figura 4.7. Respuesta del circuito acondicionador de señal con una señal de entrada senoidal de 50[mv] y una frecuencia de 0.1[Hz]	112
Figura 4.8. Señal obtenida a la salida del filtro	112
Figura 4.9. Presión arterial promedio	113
Figura 4.10. Datos recibidos del GPS.....	113
Figura 4.11. Lugar al que correponden las coordenadas obtenidas el GPS.....	114
Figura 4.12. Directorio raíz del sistema de archivos FAT.....	115
Figura 4.13 Directorio raíz del sistema de archivos FAT en hexadxecimal	115
Figura 4.14. Información guardada del archivo TEXTO.TXT	116
Figura 4.15. Información guardada del archivo TEXTO1.TXT.....	116

Figura 4.16. Circuito recomendado para la activación de los terminales “Activacion” y “Reinicio”	117
Figura 4.17. Comandos enviados por el microncontrolador al modem para realizar una conexión al servidor de Enfora.....	117
Figura 4.18. Captura de pantalla del servidor de Enfora.....	118
Figura 4.19. Mensaje enviado por el sistema de adquisición recibido por un teléfono celular	118
Figura 4.20. Pulsera de velcro con los sensores de pulso y temperatura	119
Figura 4.21. Forma en la que se realizó la prueba.....	119
Figura 4.22. Datos obtenidos.....	120
Figura 4.23. Imagen de la señal de pulso generada con 200 muestras.....	120
Figura 4.24. Imagen de la señal de pulso generada con 1,000 muestras.....	121
Figura 4.25. Gráfica de la señal del acelerómetro.....	121
Figura 4.26. Mensaje enviado por el sistema de datos	122
Figura 4.27. Ubicación del sistema de datos	122

Anexo

Figura A.1. Señal obtenida con el filtro del sistema.....	128
Figura A.2. Simulación Filtro paso bajas Bessel a 30[Hz].	129
Figura A.3. Implementación filtro paso bajas Bessel a 30[Hz], la señal superior representa la señal de entrada al filtro mientras que la señal inferior la señal de salida.	130
Figura A.4. Simulación Filtro paso bajas Bessel a 250[Hz].	130
Figura A.5. Implementación Filtro paso bajas Bessel a 250[Hz] , la señal superior representa la señal de entrada al filtro mientras que la señal inferior la señal de salida.....	131
Figura A.6. Simulación filtro paso bajas Butterworth a 20[Hz].	132
Figura A.7. Implementación filtro paso bajas Butterworth a 20[Hz] , la señal superior representa la señal de entrada al filtro mientras que la señal inferior la señal de salida.....	132
Figura A.8. Simulación filtro paso bajas Butterworth a 250[Hz].	133
Figura A.9. Implementación filtro paso bajas Butterworth a 250[Hz] , la señal superior representa la señal de entrada al filtro mientras que la señal inferior la señal de salida.....	133

Índice de tablas

Capítulo 1

Tabla 1.1. Frecuencia cardíaca en reposo[6].	8
--	---

Capítulo 2

Tabla 2.1. Factores de amortiguamiento y razones $f_{(3\text{ dB})}/f_c$ De filtros de segundo orden	28
Tabla 2.2. Mensajes más usados por el protocolo NMEA.	42
Tabla 2.3. Estructura de los comandos CMD en modo SPI.	47
Tabla 2.4. Comandos CMD más usados en el modo SPI.	47
Tabla 2.5. Registro OCR.	49
Tabla 2.6. Voltajes aceptados.	50
Tabla 2.7. Estructura del byte de hora de creación del archivo.	54
Tabla 2.8. Estructura del byte de la fecha de creación del archivo.	54

Capítulo 3

Tabla 3.1. Comparativa de algunos acelerómetros disponibles en el mercado.	66
Tabla 3.2. Dispositivos para monitorear la temperatura.	66
Tabla 3.3. Conexiones entre los canales del ADC y las señales.	86

Capítulo 4

Tabla 4.1. Condiciones eléctricas de operación	110
Tabla 4.2. Registro de los voltajes en las baterías antes y después de una hora de ser conectarlos al circuito cargador de baterías	110

Anexo

Tabla A.1. Cálculos para un filtro Bessel paso bajas.	129
Tabla A.2. Componentes ajustados a valores comerciales.	129
Tabla A.3. Cálculos para un filtro Bessel paso bajas.	131
Tabla A.4. Componentes ajustados a valores comerciales.	131

1. Introducción

En este capítulo se mencionarán los antecedentes del proyecto, así también los conceptos teóricos del pulso radial y su relación con el ciclo cardiaco, la movilidad y las caídas: debido a que no es un trabajo dedicado a la medicina no se profundizará en los conceptos. Por último se presenta un breve resumen de fotopleletismografía.

1.1. Proyecto VACS

El proyecto Vigilancia y Análisis Continuo de Signos vitales (VACS), consiste en el desarrollo de un dispositivo de adquisición y envío de datos ligado a una plataforma de servicios de teleasistencia. Mediante la adquisición en tiempo real del pulso radial, movilidad, localización y temperatura, el sistema pretende monitorear el ritmo cardiaco y detectar caídas del usuario mediante la generación de una serie de servicios que ofrezcan seguridad y tranquilidad, tanto a los usuarios como a sus familiares. Los objetivos científicos y tecnológicos de este proyecto están encaminados a establecer un paradigma en el diseño de programas de atención preventiva para el cuidado de los adultos mayores.

De acuerdo con la organización mundial de la salud las enfermedades cardiovasculares (ECV) son la principal causa de muerte en el mundo, a tal grado que cada año mueren más personas por este tipo de enfermedades que por cualquier otra causa. Se calcula que en el 2008 murieron por esta causa 17.3 millones de personas, lo cual representa un 30% de las muertes registradas en el mundo. Las muertes por ECV afectan a ambos sexos por igual, y más del 80% se producen en países de ingresos bajos y medios. Para el 2030 se calcula que morirán cerca de 23.3 millones de personas por enfermedades de este tipo y se prevé que siga siendo la principal causa de muerte [1].

La Organización Mundial de la salud (OMS) indica que las caídas son la segunda causa de muerte por lesiones accidentales o no intencionales. Los mayores de 65 años son quienes sufren más caídas mortales, cada año se producen 37.3 millones de caídas cuya gravedad requiere atención médica y se calcula que anualmente mueren en todo el mundo unas 424,000 personas debido a caídas, lo que convierte a las caídas en la segunda causa mundial de muerte por lesiones no intencionales, por detrás de los traumatismos causados por el tránsito [2].

El proyecto impulsado por la empresa Matersys Group involucra a instituciones de educación superior como lo son: la Universidad Estatal del Valle de Ecatepec (UNEVE) responsable del desarrollo de las pruebas e investigación médica, y la Universidad Nacional Autónoma de México (UNAM), a través del Instituto de Ciencias Nucleares, este último es responsable de la instrumentación, análisis de variabilidad y del desarrollo e implementación de los prototipos.

El proyecto está pensado como un proyecto de desarrollo tecnológico que incluye los siguientes componentes:

- Brazaletes. Consiste en un dispositivo que incluye transductores de temperatura, pulso radial, patrones de movilidad, sistema de almacenamiento de datos, sistemas de comunicación inalámbrica (GPRS) y un sistema de localización (GPS). Con el fin de obtener información útil para el diagnóstico de algunas enfermedades cardiovasculares.
- Plataforma VACS. Está compuesta por varios sistemas, encargados de: la recepción de información registrada por el brazalete VACS, la validación y categorización de eventos, el manejo de las interfaces de usuario y la comunicación de alertas.

El brazalete proporcionará información oportuna del ritmo cardiaco del paciente así como la morfología del pulso que será de gran utilidad para la detección de anomalías cardiacas. De igual forma se pretende supervisar la movilidad del paciente con el fin de detectar caídas que lo puedan inmovilizar.

Los datos obtenidos proporcionarán al médico especialista información en tiempo real de la situación en la que se encuentra el paciente, con la cual contará con herramientas para un diagnóstico y/o un tratamiento más adecuado al estilo de vida del paciente. A su vez dará tranquilidad a su familia ya que sabrán en todo momento la ubicación y condiciones en que se encuentra el paciente.

1.2. Movilidad

Si bien la capacidad de moverse (movilidad) no es considerada un signo vital, también, proporciona información del estilo de vida del paciente: al contar con esta información se podrían establecer tratamientos más adecuados, no sólo para el control del sobre peso, también, para pacientes con sistemas de locomoción y/o con problemas cardiacos; en los que la detección de caídas y su atención oportuna son invaluable. De acuerdo con la Organización Panamericana de la Salud (PAHO), México ocupa el primer lugar en obesidad tanto en niños como en adultos, esto es un gran problema debido a que se estima que cada año mueren alrededor de 2.6 millones de personas a causa de la obesidad y el sobrepeso[3].

1.2.1. Caídas

La Organización mundial de la salud (OMS) define a las caídas como: *“acontecimientos involuntarios que hacen perder el equilibrio y dar con el cuerpo a tierra u otra superficie firme que lo detenga”*[2]. También puede definirse: *“como la consecuencia de cualquier acontecimiento que precipita al individuo, generalmente al piso, contra su voluntad, ésta suele ser repentina, involuntaria e insospechada y puede ser confirmada o no por el paciente o un testigo”* [4].

Las complicaciones médicas son por el propio traumatismo de la caída (fracturas, contusiones, heridas, traumatismos craneoencefálicos, etc.) O por la estancia prolongada en el suelo, al no poder levantarse el individuo por sus propios medios (destrucción muscular, úlceras por presión, deshidratación, infecciones respiratorias por bronco aspiración, hipotermia, etc.), y que se relacionan con una peor evolución y una mayor mortalidad. Aproximadamente el 30% de

los individuos mayores de 65 años se cae cada año y de éstos el 5% presenta lesiones mayores, como fracturas y otras consecuencias; el 2% precisará hospitalización y de éstos sólo la mitad sobrevivirá al año de la caída. Aproximadamente el 10% de las visitas de los ancianos a urgencias son debidas a caídas[4].

1.3. Sistema cardiovascular

El sistema cardiovascular o aparato circulatorio sanguíneo está compuesto por el corazón y los vasos sanguíneos (ver figura 1.1). El corazón es el elemento central del sistema cardiovascular ya que bombea la sangre que recorrerá todo el cuerpo. Los vasos sanguíneos están constituidos por arterias, venas y capilares; las arterias son los vasos que conducen la sangre desde el corazón hasta los distintos órganos, las venas se encargan de conducir la sangre de vuelta al corazón desde los distintos órganos, por último, los capilares son pequeños vasos a través de los cuales se filtran el plasma sanguíneo y los nutrientes hacia las células, y los productos de desecho de los tejidos hacia la sangre.

El sistema cardiovascular tiene las funciones siguientes:

- Distribuir por todo el organismo los alimentos y hormonas, así como recoger los productos de desecho del metabolismo celular y llevarlos hasta los órganos que intervienen en su eliminación.
- Distribuir el oxígeno por todo el organismo, llevando la sangre rica en oxígeno desde los pulmones hacia los espacios intercelulares y llevando el dióxido de carbono resultante del metabolismo celular a los pulmones.

Tomando como referencia el corazón, la circulación sanguínea se puede dividir en dos circuitos:

- Circulación menor o pulmonar. Se lleva a cabo en el circuito que se establece entre el corazón y los pulmones. Este circuito está constituido por el ventrículo derecho, las arterias pulmonares, los capilares pulmonares, las cuatro venas pulmonares y la aurícula izquierda. Tiene como misión captar el oxígeno en los alvéolos pulmonares y llevar el dióxido de carbono a los pulmones, para su expulsión al exterior.
- Circulación mayor o Sistémica. Es el circuito que se establece entre el corazón y los distintos órganos del cuerpo. Este circuito está formado por el ventrículo izquierdo, la arteria aorta y todas las arterias que derivan de ella, los capilares de los tejidos, el sistema venoso de retorno que confluye en las venas cavas y la aurícula derecha. Tiene como misión transportar el oxígeno y los nutrientes a las células, y recoger los productos de desecho del metabolismo para conducirlos a los órganos encargados de su eliminación.

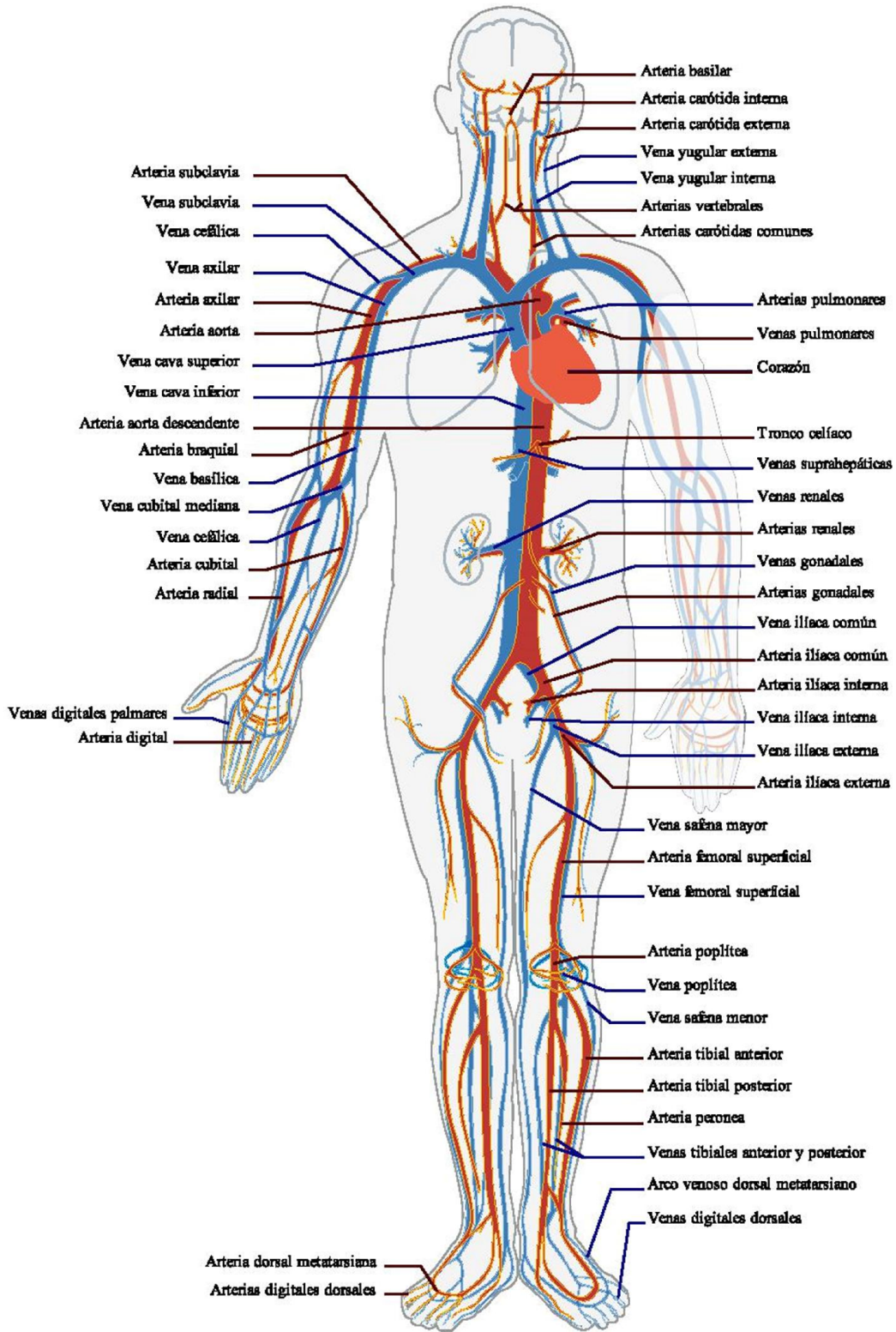


Figura 1.1. Sistema cardiovascular.

Así, por la parte derecha del corazón circula la sangre desoxigenada, también llamada sangre venosa, y por la izquierda, lo hace la sangre oxigenada, que recibe el nombre de arterial. En estado de reposo, la sangre realiza la totalidad del recorrido en un minuto y, en situación de extraordinaria actividad, puede hacerlo hasta seis veces por minuto.

Es conveniente destacar que las arterias parten siempre de los ventrículos, mientras que las venas siempre entran en las aurículas, independientemente de que lleven sangre arterial o venosa. Así, la arteria pulmonar, que sale del ventrículo derecho, contiene sangre venosa, mientras que las venas pulmonares, que llevan la sangre de los pulmones a la aurícula izquierda, contienen sangre arterial.

1.3.1. Ciclo Cardíaco

Con este término se designa al latido cardíaco completo que consiste en contracción (sístole) y relajación (diástole) de ambas aurículas, y de ambos ventrículos.

Cada contracción rítmica del corazón es consecuencia de la relación entre su actividad eléctrica y su actividad mecánica. Normalmente toda actividad eléctrica es seguida por una función mecánica (contracción o relajación) que da lugar a una onda de presión. La onda P, del complejo QRS (representación gráfica de la despolarización ventricular) y la onda T del electrocardiograma (ECG¹) son las responsables de: la contracción auricular, la activación ventricular, y la relajación ventricular respectivamente.

La duración de los impulsos eléctricos y el tiempo de transmisión a las diferentes áreas del corazón, permiten que las funciones mecánicas estén sincronizadas y generen el gasto cardíaco y las presiones necesarias para que los tejidos corporales se irriguen adecuadamente con sangre oxigenada. Una idea de esta relación entre actividades eléctricas y mecánicas del corazón se puede obtener del llamado diagrama de Wigger que se muestra en la figura 1.2, en donde se relacionan las ondas de presión y las señales eléctricas durante un ciclo cardíaco.

El ciclo empieza con la onda P, que es la señal eléctrica para la contracción auricular. La onda A de presión auricular sigue a la onda P después de 30[ms] – 50[ms]. Después del pico de la onda A, la aurícula se relaja y la presión disminuye, generando el descendente x (punto b). El siguiente evento es la despolarización ventricular durante el QRS. La presión en el ventrículo izquierdo (VI) justo después del pico de la onda A, es llamada presión final diastólica. Entre 15[ms] – 30[ms] después del QRS, los ventrículos se contraen y las presiones ventriculares izquierda y derecha aumentan rápidamente.

¹ ECG (Electrocardiograma). Representación gráfica de la actividad eléctrica del corazón.

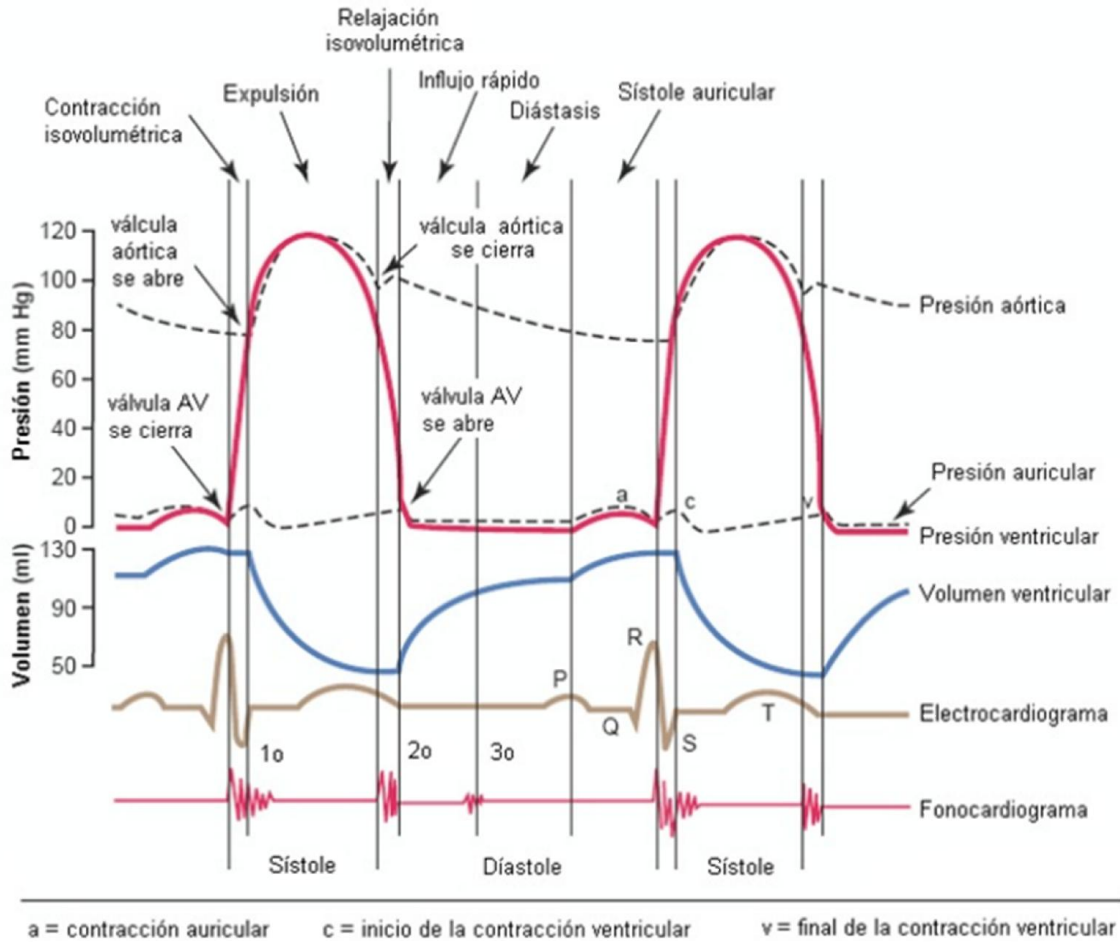


Figura 1.2 Diagrama de Wigger.

1.3.2. Pulso arterial

El propósito de las arterias es entregar sangre oxigenada a los tejidos que, a su vez, es parte fundamental de la homeostasis² necesaria para sostener el funcionamiento adecuado de todo el organismo. Dentro de las tareas que el sistema arterial desarrolla para lograr sus objetivos está el convertir el gasto cardíaco, que es fundamentalmente intermitente, en un flujo capilar continuo.

Las arterias se asemejan a tubos elásticos, y cada contracción cardíaca impulsa 30[g] – 60[g] de sangre hacia los vasos sanguíneos ya llenos. La distensión consecutiva se transmite a lo largo del árbol arterial con una rapidez aproximada de $7 \left[\frac{m}{s} \right]$ Hasta que alcanza los capilares. Aquí el pulso se pierde debido a la resistencia periférica al flujo sanguíneo y a la ausencia de elasticidad de las paredes vasculares.

²Homeostasis: Capacidad del organismo para presentar un estado físico-químico característico estable dentro de ciertos límites.

El pulso es una expansión rítmica de las arterias consecuencia del paso sucesivo de oleadas de sangre producidas por las contracciones rítmicas del corazón; los latidos cardíacos producen la circulación de la sangre, que a su vez genera una onda que viaja a través del árbol arterial. Las arterias responden al paso del flujo distendiéndose, y luego de que pasa cada perturbación recobran su forma normal por contracción del músculo liso de la pared vascular.

La tensión (T) de las paredes es gobernada por la ecuación de Young-Laplace (ecuación 1.1):

$$T \propto \frac{P \times r}{2h} = \frac{T_L}{2h} \quad \text{Ecuación 1.1}$$

Donde P es la presión sanguínea, r el radio interior y h el grosor de la pared del vaso. $T_L = P \times r$, es la extensión estimada con la ley de Laplace para un vaso de grosor despreciable. Dado que la expulsión de la mayor parte del volumen sistólico se lleva a cabo a principios y mediados de la sístole, los grandes cambios en la tensión de las paredes del vaso durante este periodo, son resultado de los cambios de volumen y la presión. Sin embargo, durante la parte final de la sístole y durante la diástole el efecto predominante se debe principalmente a cambios de la presión, aunque el volumen podría jugar algún papel menos.

La contracción del ventrículo izquierdo imparte su energía contráctil a la masa de sangre que contiene, generando y aumentando la presión para superar la presión diastólica en la aorta con el fin de abrir la válvula aórtica y expulsar la sangre hacia la aorta. A medida que el ventrículo expulsa la masa de la sangre hacia la aorta con cada sístole, se crea una presión del pulso así como un flujo pulsátil.

Lo que se siente cuando una arteria se palpa no es sólo la fuerza ejercida por la amplitud de la onda de presión, sino también el cambio en el diámetro; dado que la presión y el radio son dos factores importantes que afectan a la tensión de la pared, como se muestra por la ecuación de Laplace (ecuación 1.1). Así, lo que en realidad se percibe cuando el pulso arterial se palpa es el efecto causado por un cambio en la tensión de la pared de la arteria. Por lo tanto, la amplitud del pulso no solo dependerá de la amplitud de la onda de presión sino también del cambio en las dimensiones arteriales entre diástole y sístole, o en otras palabras, de la cantidad de cambio en la tensión sobre las paredes del vaso.

El pulso arterial tiene diferentes nombres dependiendo del lugar en el que se mide, por ejemplo, el pulso que se mide en la arteria radial llevará el nombre de pulso radial, los pulsos que se buscan usualmente son:

- Pulso carotideo
- Pulso axilar
- Pulso braquial
- Pulso radial
- Pulso femoral

- Pulso poplíteo
- Pulso pedio
- Pulso tibial posterior

Es sabido que los registros obtenidos con un transductor de pulso colocado externamente sobre una arteria, la carótida por ejemplo, tiene un contorno y forma muy similar a la curva de presión obtenida por medio de una catéter colocado dentro de la arteria y registrados con un sistema manométrico indicador de tensión[5].

1.3.3. Arritmias

Cuando el ritmo de los latidos cardiacos cambia, se produce una arritmia³; estos trastornos se deben generalmente a algún daño del músculo cardiaco o del módulo sinusal, que es el marcapasos natural del corazón. El daño puede ser causado por un ataque al corazón, defectos congénitos, válvulas cardiacas deterioradas o por alguna infección.

En la tabla 1.1 se muestran los valores normales de los latidos por minuto del corazón acorde a la edad. Las arritmias se dividen en los tres grupos siguientes:

- Bradicardia, frecuencia cardiaca menor a 60 latidos por minuto.
- Taquicardia, frecuencia cardiaca mayor a 100 latidos por minuto.
- Fibrilaciones, frecuencia cardiaca irregular.

Edad	Latidos por minuto
0 a 1 mes	70 a 90
1 a 11 meses	80 a 160
1 a 2 años	80 a 130
3 a 4 años	80 a 120
5 a 6 años	75 a 115
7 a 9 años	70 a 110
10 o más	60 a 100
Atletas bien entrenados	40 a 60

Tabla 1.1. Frecuencia cardíaca en reposo[6].

1.4. Fotopleletismografía

La fotopleletismografía es un método no invasivo que mide cambios de en el flujo sanguíneo, en diferentes partes del cuerpo, con esta técnica podemos supervisar el ritmo cardiaco; esta técnica consiste en iluminar el área con una fuente de luz, y con un sensor óptico detectar el cambio en el volumen de la arteria por medio de ondas de pulso.

³ Arritmias: son problemas de la frecuencia cardiaca o del ritmo de los latidos del corazón. Durante una arritmia el corazón puede latir demasiado rápido, demasiado despacio o de manera irregular

Los cambios en la frecuencia cardiaca y la presión sanguínea afectan las tasas de reflexión de la luz incidente. La longitud de onda de la luz emisora debe ser de $660[nm]$ ó de $940[nm]$, esta luz es emitida sobre la piel, la luz será absorbida, reflejada o dispersada en mayor o menor cantidad, dependiendo de la cantidad de sangre que esté circulando, como se muestra en la figura 1.3. Estas longitudes de onda son elegidas debido a que la desoxihemoglobina tiene una mayor absorción en torno a $660[nm]$ y en la hemoglobina oxigenada se tiene la mayor absorción entre $910[nm] - 940[nm]$. La hemoglobina oxigenada permite a la luz roja pasar y absorbe más luz infrarroja mientras que la hemoglobina desoxigenada permite pasar a la luz infrarroja y absorbe más luz roja.

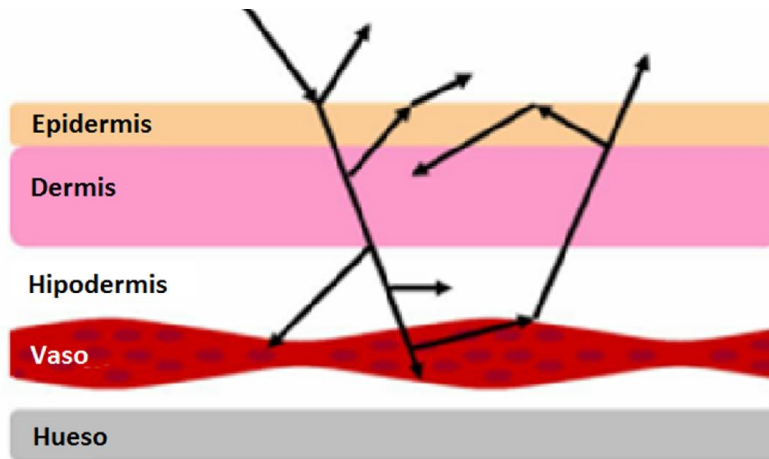


Figura 1.3. Trayectoria de luz en la fotopleletismografía.

En la figura 1.4 se muestra la cantidad de luz que será absorbida por las distintas partes del cuerpo, en la cual hay una parte de sangre pulsátil que es la señal de interés para obtener el ritmo cardiaco.

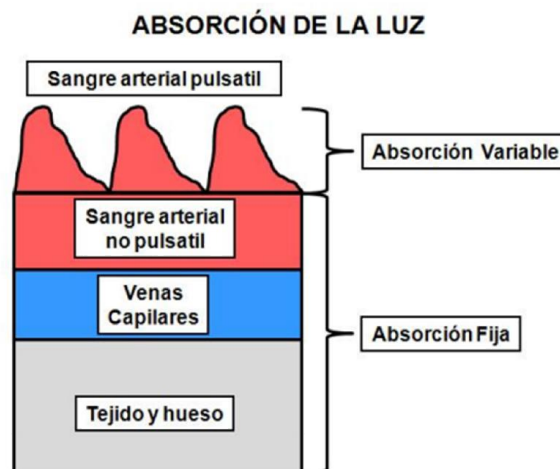


Figura 1.4. Cantidad de luz que se absorbe por las distintas partes del cuerpo.

En la figura 1.5 están representadas las posiciones en las cuales se puede obtener una señal de fotopleletismografía, los métodos de adquisición son: transmisión y reflexión. Para el método de

transmisión los lugares más comunes son: la punta del dedo, la zona más delgada es el lóbulo de la oreja y para el uso en bebés se recomienda el empeine de los pies, debido al pequeño tamaño del dedo y la oreja. Con el método de reflexión se puede aplicar en el dedo en forma de anillo, en la frente, en la muñeca, entre otros.

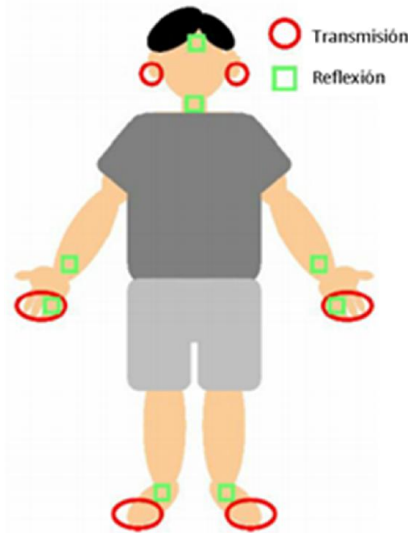


Figura 1.5. Áreas para detectar la señal de pulso, clasificadas por métodos de detección.

Como se puede observar en la figura 1.6 existen dos métodos para realizar la fotopleletismografía en la imagen (a) la luz es transmitida por lo cual el detector está del lado opuesto al emisor; normalmente este método se usa en partes delgadas del cuerpo donde la luz transmitida es más intensa que la luz reflejada obteniendo por tanto una señal más clara. En la imagen (b) la luz es reflejada, por lo tanto la luz recibida por el detector es muy poca resultando una señal débil.

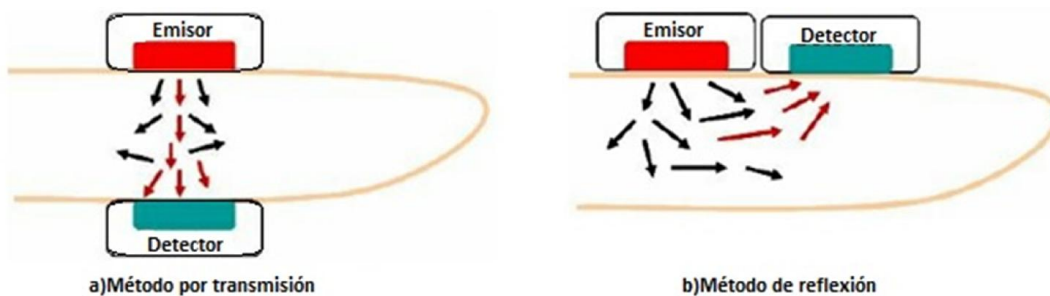


Figura 1.6. Métodos de detección del fotopleletismógrafo.

En el presente trabajo se utilizará la fotopleletismografía por reflexión para obtener la señal del pulso radial; con el fin de no limitar las actividades del usuario, ya que, la adquisición de datos se realizará en el brazo cerca de la muñeca. Con ello se pretende que el paciente lleve una vida normal tan sólo con la preocupación de recargar la batería.

2. Antecedentes del proyecto

En este capítulo se describen los conceptos teóricos básicos para el desarrollo de este proyecto, haciendo referencia a los componentes electrónicos que integran el sistema, así como también para los que componen la interfaz de usuario.

2.1. Sensores

Un sensor es un dispositivo que, a partir de la energía del medio donde se mide, da una señal transducible que es función de la variable[8]. Comúnmente este término es empleado para denominar a dispositivos capaces de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. Las variables de instrumentación pueden ser por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, movimiento, ph, etc.

A continuación se hace una descripción de los sensores que serán utilizados en el desarrollo de este proyecto.

2.1.1. Sensores fotoeléctricos

Las fuentes luminosas transmiten paquetes individuales llamados fotones; los cuales tienen un nivel de energía (E) directamente relacionado con la frecuencia (f) de la onda luminosa viajera, la cual será determinada por la siguiente ecuación:

$$E = hf [J] \quad \text{Ecuación 2.1}$$

Donde h es la constante de Planck que es igual $6.624 \times 10^{-34} [Js]$. El flujo luminoso se mide en lúmenes [lm] O watts las cuales están relacionadas por:

$$1 [lm] = 0.001464 \left[\frac{J}{s} \right] \quad \text{Ecuación 2.2}$$

La intensidad luminosa se suele medir en [$lm/\pi e^2$], candelas-pie [fc] O [W/m^2], dónde:

$$1 \left[\frac{lm}{\pi e^2} \right] = 1 [fc] = 1.609 \times 10^{-9} \left[\frac{W}{m^2} \right] \quad \text{Ecuación 2.3}$$

Para medir la cantidad de luz existen dispositivos electrónicos como son: fotorresistencias, fotodiodos o fototransistores los cuales utilizan el efecto foto-eléctrico para producir corrientes proporcionales a la intensidad lumínica a los cuales son expuestos.

Los fotodiodos son dispositivos de unión p-n cuya región de operación se limita a la región de polarización en inversa. La aplicación de luz a la unión hace que se transfiera energía de las ondas luminosas viajeras incidentes (en forma de fotones) a la estructura atómica, y el resultado es una cantidad incrementada de portadores minoritarios y un nivel incrementado de corriente en

inversa. Como puede observarse en la figura 2.1 la corriente en el fotodiodo se incrementara si aumenta la intensidad de luz; la corriente oscura es aquella que se dará sin iluminación aplicada.

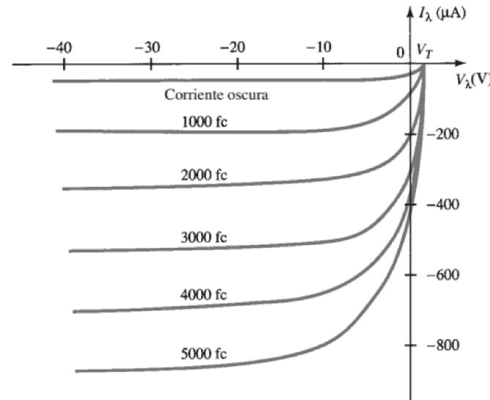


Figura 2.1. Curvas características del fotodiodo.

Un fototransistor funciona básicamente como un transistor, como se observa en la figura 2.2 los fotones que inciden en la región P del fototransistor provocaran una corriente de base, si denominamos I_B A dicha corriente la corriente en el colector (I_C) Estará dada por:

$$I_C \cong h_{fe} I_B \quad \text{Ecuación 2.4}$$

donde h_{fe} Está definido como un factor de amplificación de corriente en directa en emisor común.

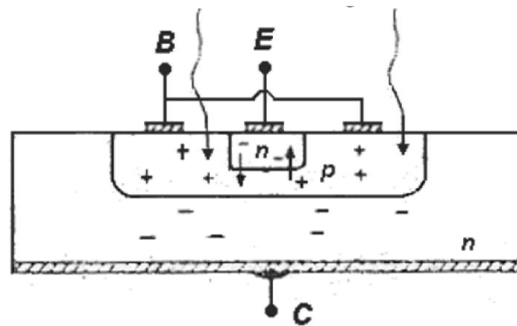


Figura 2.2. Esquema de un fototransistor NPN.

2.1.2. Sensores de temperatura

La temperatura es una variable que usualmente es medida en los sistemas ya sean mecánicos, químicos o biológicos, esto se debe a que siempre se ven afectados por la temperatura.

Se pueden encontrar distintos tipos de sensores de temperatura los más usuales son: termopares, termistores, los RTD (Resistance Temperature Detectors), monolíticos y los bimetalicos.

Termopares

Un termopar se crea cuando dos hilos metálicos diferentes los cuales se unen en un extremo. Los cambios en la temperatura de esa junta provocan un cambio en la fuerza electromotriz (FEM) entre los dos extremos. Cuando la temperatura aumenta la FEM aumenta, no necesariamente de forma lineal.

Termistores

El termistor está basado en que el comportamiento de la resistencia de los semiconductores es variable en función de la temperatura.

Existen los termistores tipo NTC y los termistores tipo PTC. En los primeros, al aumentar la temperatura, disminuye la resistencia. En los PTC, al aumentar la temperatura, aumenta la resistencia.

El principal problema de los termistores es que no son lineales según la temperatura por lo que es necesario aplicar fórmulas complejas para determinar la temperatura según la corriente que circula y son complicados de calibrar.

RTD

Un RTD es un sensor de temperatura basado en la variación de la resistencia de un conductor con la temperatura.

Los metales empleados normalmente como RTD son platino, cobre, níquel y molibdeno. De entre los anteriores, los sensores de platino son los más comunes por tener mejor linealidad, más rapidez y mayor margen de temperatura.

Sensores de silicio

Son dispositivos basados en las propiedades térmicas de las uniones semiconductoras (PN), particularmente la dependencia de la tensión base emisor (VBE) de los transistores bipolares con la temperatura cuando la corriente de colector es constante.

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

- Acondicionamiento de señales
- Correcciones de ganancia
- Linealidad
- Escalado y conversión de unidades
- Comunicación digital
- Autodiagnóstico

Los sensores digitales ofrecen una alta precisión y linealidad, pueden ser de salidas digitales o analógicas.

2.1.3. Acelerómetro

Un acelerómetro MEMS (Microelectromechanical Systems) o micro acelerómetro de silicio está basado en un capacitor diferencial, consiste en tres placas planas paralelas de poli silicio con superficie micro mecanizada y separadas por el aire. La placa superior e inferior son fijas mientras que la central se encuentra suspendida por vigas empotradas en el sustrato del sensor, la cual se comporta como una masa de prueba que es sensible a la aceleración en dirección perpendicular a las placas.

A continuación se realizara un análisis del principio de funcionamiento de un micro acelerómetro.

Capacitor diferencial

Un capacitor diferencial consiste en dos capacitores variables colocados físicamente de tal modo que experimenten los mismos cambios pero en sentidos opuestos, lo cual se observa en la figura 2.3. La capacitancia C_1 estará determinada por la ecuación 2.5 y la capacitancia C_2 por la ecuación 2.6; en dónde ϵ es la permitividad del medio, d es la distancia que existe entre las superior y la placa de en medio y que en condiciones iniciales será la misma distancia entre la placa inferior y la placa media, y x será el desplazamiento que sufrirá la placa media al ser afectada como una fuerza; el cual se tomará como positivo en el caso de C_1 cuando la placa de en medio se aleje de la placa superior y negativo en el caso contrario.

$$C_1 = \frac{\epsilon A}{d + x} \quad \text{Ecuación 2.5}$$

$$C_2 = \frac{\epsilon A}{d - x} \quad \text{Ecuación 2.6}$$

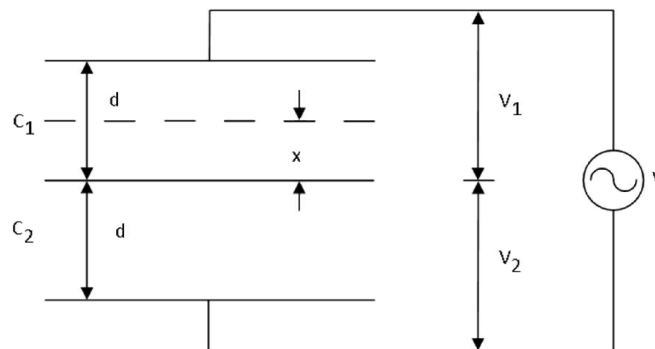


Figura 2.3. Capacitor diferencial.

También puede observarse en la figura 2.3, que la tensión que existe en las terminales de cada capacitor puede obtenerse aplicando un divisor de voltaje, quedando de la siguiente forma:

$$V_1 = \frac{V}{\frac{1}{j\omega C_1} + \frac{1}{j\omega C_2}} \frac{1}{j\omega C_1} = V \frac{C_2}{C_1 + C_2} \quad \text{Ecuación 2.7}$$

$$V_2 = \frac{V}{\frac{1}{j\omega C_1} + \frac{1}{j\omega C_2}} \frac{1}{j\omega C_2} = V \frac{C_1}{C_1 + C_2} \quad \text{Ecuación 2.8}$$

Siendo V_1 La diferencia de potencial que existe entre las terminales del C_1 Y V_2 La diferencia de potencial que existe entre las terminales del C_2 , en la ecuación 2.7 y en la ecuación 2.8 es fácil observar que la diferencia de potencial depende del valor que tome cada capacitor, por lo que para conocer la magnitud de la tensión a partir de la posición de la placa central se procede a sustituir la ecuación 2.5 y la ecuación 2.6 en la ecuación 2.7 y la ecuación 2.8 obteniendo como resultado lo siguiente:

$$V_1 = V \frac{\frac{1}{d-x}}{\frac{1}{d+x} + \frac{1}{d-x}} = V \frac{d+x}{2d} \quad \text{Ecuación 2.9}$$

$$V_2 = V \frac{\frac{1}{d+x}}{\frac{1}{d+x} + \frac{1}{d-x}} = V \frac{d-x}{2d} \quad \text{Ecuación 2.10}$$

Hemos llegado a dos expresiones en donde tenemos que el valor de la tensión está en función de x , por lo que si se restan la ecuación 2.9 y la 2.10.

$$V_1 - V_2 = \frac{V}{2d} (d+x - d+x) = \frac{V}{2d} (2x) = V \frac{x}{d} \quad \text{Ecuación 2.11}$$

Al ser d una constante se ha encontrado una relación en el cual al moverse la masa de prueba por efecto de la aceleración, un voltaje que variará de forma lineal respecto a la posición de la masa de prueba.

Sin embargo existe otra posibilidad y obtener una relación de cambio entre las tenciones lo cual quedara expresado de la siguiente forma:

$$\frac{V_2}{V_1} = \frac{C_1}{C_2} = \frac{\frac{\epsilon A}{d+x}}{\frac{\epsilon A}{d-x}} = \frac{d-x}{d+x} \quad \text{Ecuación 2.12}$$

También puede notarse en la ecuación 2.12 que esta relación no es lineal y que no ofrece ninguna ventaja sobre la ecuación 2.11 ya que para obtenerla se puede lograr de manera muy sencilla por medio de un amplificador diferencial, mientras que para obtener la ecuación 2.12 se tendrían que usar amplificadores logarítmicos y anti logarítmicos para lograr la división.

2.2. Acondicionamiento de señales

Debido a que generalmente la señal que se obtiene directamente de los sensores no es apta para ser procesada, ya que presenta problemas de ruido o a que no tiene la suficiente potencia para ser procesadas; se deben acondicionar las señales. Generalmente el proceso de acondicionamiento consta entre otras etapas:

- Amplificación
- Filtrado

2.2.1. Amplificación

Los amplificadores son circuitos que se utilizan para aumentar la potencia de la señal de entrada, con el fin de proporcionar características adecuadas para su correcto procesamiento.

En la electrónica usualmente se trabajan con amplificadores operacionales y transistores por lo que es conveniente conocer las configuraciones básicas de cada uno de ellos.

Transistor de Juntura Bipolar

El transistor bipolar de juntura (TBJ), esta construido por dos materiales tipo “N” y separados por otro tipo “P” (transistor NPN) o por dos materiales tipo “P” y separados por un tipo “N” (transistor PNP). En la figura 2.4 se muestran algunos símbolos del transistor en el cual se indican las capas que componen a cada tipo de transistor así como el sentido en que fluirá la corriente a través de el; el cual se indica con la orientación de la flecha en el emisor. Si denominamos I_B A la corriente de base, I_C A la corriente de colector e I_E A la corriente de emisor, la relación entre las corrientes transistor “NPN” estará dada por la ecuación 2.13 al igual que para el transistor “PNP”.

$$I_E = I_B + I_C \qquad \text{Ecuación 2.13}$$

El transistor puede operar en las regiones de activa, corte y saturación. En la región activa la unión base-emisor se polariza en directa, mientras que la unión colector-base en inversa, en la región de corte las uniones base emisor y colector base se encuentran en inversa, finalmente, en la región de saturación las uniones base-emisor y colector-base se polarizan en directa.

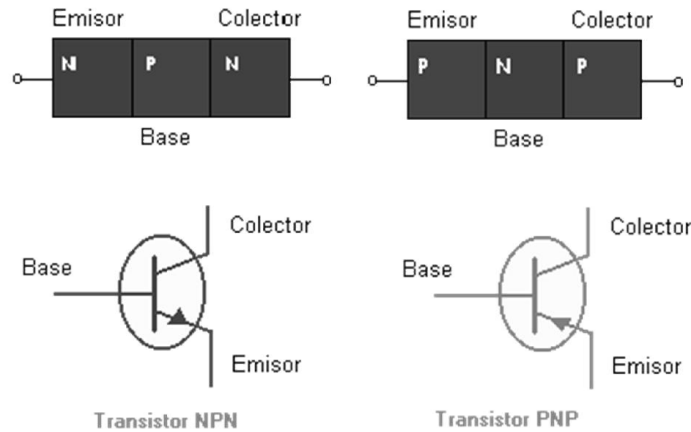


Figura 2.4. Símbolos del TBJ.

Configuraciones básicas de amplificadores TBJ de una etapa

Para que el TBJ opere como amplificador este debe operar en la región activa en la cual existe un coeficiente β que es una relación entre la corriente de base y la corriente de colector. El transistor bipolar de juntura (TBJ) tiene tres configuraciones básicas las cuales son:

- Emisor común: Recibe su nombre ya que el emisor sirve como referencia para las terminales de entrada y salida. En la figura 2.5 se muestra la configuración de un amplificador emisor común con tres resistencias.

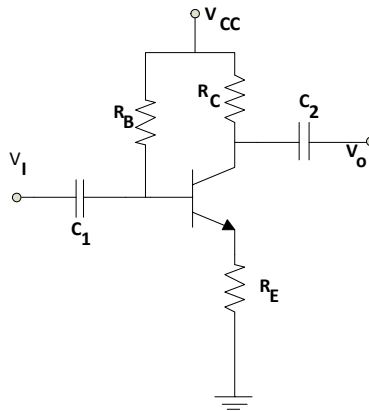


Figura 2.5. Amplificador emisor común.

- Colector común: De forma similar al anterior recibe su nombre ya que el colector sirve como referencia para las terminales de entrada y salida, como se ve en la figura 2.6. También llamado emisor seguidor debido a sus características de operación ya que su ganancia en voltaje es unitaria y no produce desfase. Otras características son: una impedancia de entrada alta y una impedancia de salida muy baja.

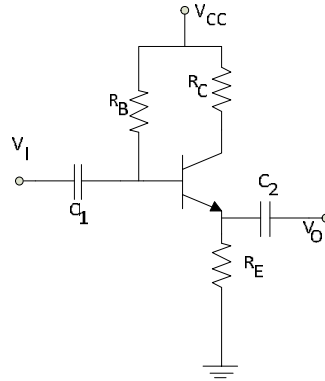


Figura 2.6. Amplificador colector común.

- Base común: La configuración en base común como se ve en la figura 2.7 se diferencia en que la señal aplicada está conectada al emisor y la señal de salida se obtiene en el colector, mientras que la base sirve como referencia a las señales de entrada y salida. Sus principales características son una impedancia de entrada baja y una impedancia de salida alta.

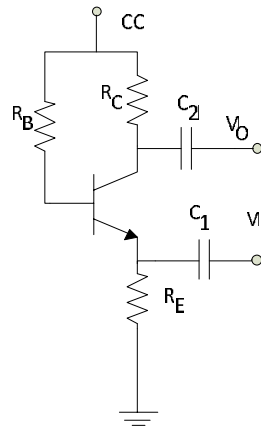


Figura 2.7. Amplificador base común.

Amplificador operacional

Los amplificadores operacionales son dispositivos comúnmente usados en el área de la electrónica, debido a que se utilizan en múltiples aplicaciones para amplificar señales muy pequeñas en procesos industriales, medición, acondicionamiento de señales, entre otras.

En la figura 2.8 se muestra la representación de un amplificador operacional, este se encuentra constituido por las señales de entrada V_n y V_p y un voltaje de salida V_o ; los cuales están de forma ideal relacionados por la ecuación siguiente:

$$V_o = a(V_p - V_n) = a\Delta V \quad \text{Ecuación 2.14}$$

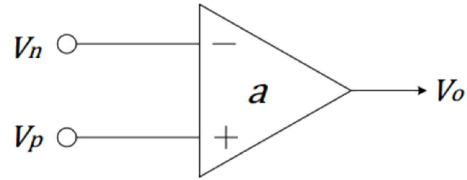


Figura 2.8. Representación de un amplificador operacional.

Entre sus principales características son:

- Impedancia de entrada elevada.
- Impedancia de salida muy baja.
- Ganancia en lazo abierto elevada, del orden de 100,000 o mayor.
- Voltajes y corrientes de polarización pequeños.
- Alto rechazo en modo común.
- *Slew rate*. Es la relación entre la variación de la tensión de salida máxima respecto de la variación del tiempo sus unidades son $[V/\mu s]$.
- Producto Ganancia – Ancho de banda (GBP). Se define como el producto de la ganancia en lazo abierto por el ancho de banda de una frecuencia específica. A ganancia unitaria, el GBP es igual al ancho de banda disponible del amplificador.

Configuraciones básicas del amplificador operacional

En el amplificador operacional se usa realimentación negativa, lo cual le da un comportamiento controlado; de esta forma se obtienen muchas ventajas, ya que la ganancia del circuito sólo depende de los componentes externos que se le conecten.

Sus tres configuraciones básicas son:

- Amplificador inversor. Este circuito se caracteriza porque la señal de salida de 180° respecto a la señal de entrada. En donde el voltaje de salida (V_o) estará dado por la ecuación 2.15, y la ganancia del circuito estará dada por el cociente de R_f Entre R_i , como se observa en la figura 2.9 la señal de entrada es conecta se introduce en R_i .

$$V_o = -\frac{R_f}{R_i} V_1$$

Ecuación 2.15

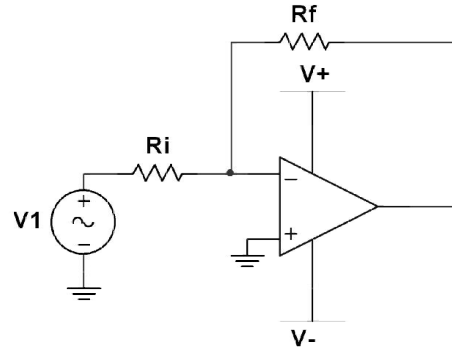


Figura 2.9. Circuito amplificador inversor.

- Amplificador no inversor. En este circuito la señal de salida se encuentra en fase respecto a la señal de entrada. Como se muestra en la figura 2.10, la señal de entrada es introducida por la terminal no inversora, la terminal inversora se conecta a tierra.

$$V_o = \left(1 + \frac{R_f}{R_i}\right) V_1$$

Ecuación 2.16

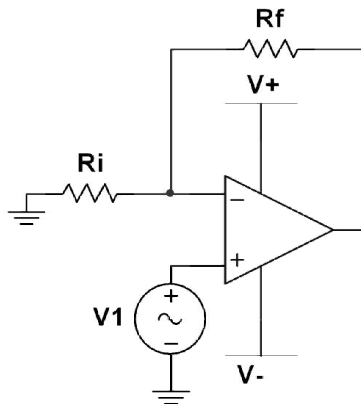


Figura 2.10. Amplificador no inversor.

- Amplificador seguidor. Este circuito se caracteriza por tener una ganancia unitaria. El voltaje en la entrada es igual al voltaje en la salida, tanto en magnitud como en signo. En la figura 2.11 se muestra el diagrama de este circuito.

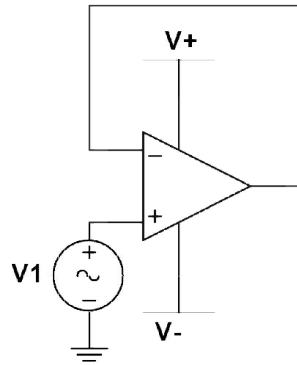


Figura 2.11. Configuración seguidor de voltaje.

2.2.2. Filtrado

Dentro de un sistema un filtro es un elemento de discriminación que se encarga de separar las señales deseadas de las no deseadas (ruido). En la electrónica un filtro es un circuito diseñado para dejar pasar una banda de frecuencias específicas.

Los filtros están conformados de al menos tres bandas:

- Banda de paso: Se define como el conjunto de frecuencias deseadas las cuales tienen la máxima ganancia permisible.
- Banda de transición: En esta banda están comprendidas las frecuencias que reciben una atenuación de 3[db] hasta una frecuencia con una atenuación de 20[db].
- Banda de rechazo: Esta banda comprende todas las frecuencias que tienen una atenuación de más de 20[db].

Los filtros pueden ser definidos por su banda de paso y de rechazo, se pueden obtener cuatro tipos de filtros los cuales son:

- Filtro paso bajas: Se caracteriza por que su banda de paso comienza en la frecuencia cero hasta una frecuencia superior (frecuencia de corte) las frecuencias mayores a esta son atenuadas (Ver figura 2.12.a).
- Filtro paso altas: Su banda de supresión comienza en las frecuencias cero hasta una frecuencia de corte, mientras que las frecuencias mayores es donde se encuentra su frecuencia de paso (Ver figura 2.12.b).
- Filtro paso banda: Como se observa en figura 2.12.c un filtro paso banda deja pasar todas las frecuencias entre una frecuencia inferior de corte f_1 Y una frecuencia de corte superior f_2 . Todas las frecuencias inferiores a f_1 Y superiores a f_2 Son atenuadas. Para este tipo de filtros así como para el filtro supresor de banda se tiene una frecuencia central f_0 la cual se considera como la media geométrica de f_1 Y f_2 la cual se calcula de la forma siguiente:

$$f_0 = \sqrt{f_1 f_2}$$

Ecuación 2.17

- Filtro supresor de banda: Contiene a dos bandas de paso una en bajas frecuencias y otra en altas frecuencias, una banda de supresión que se encuentra entre estas dos frecuencias de supresión (Ver figura 2.12.d).

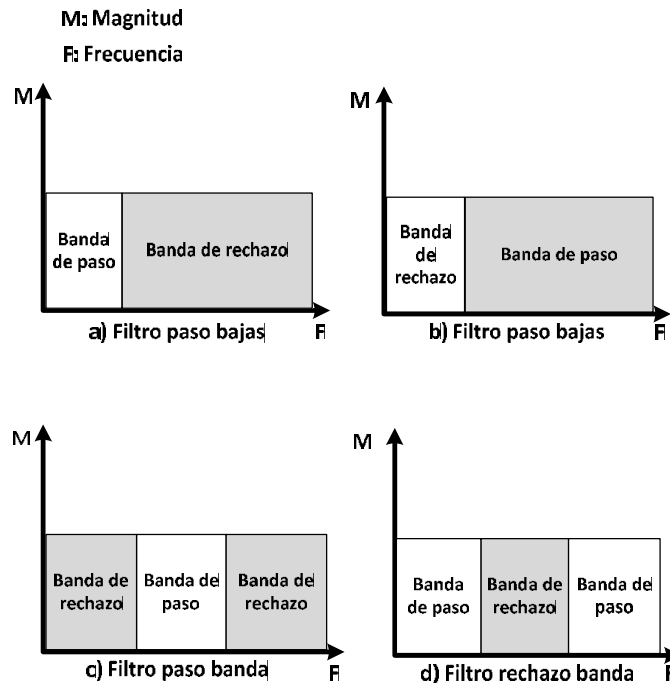


Figura 2.12. Tipos de filtros de acuerdo a su respuesta en frecuencia.

El factor de amortiguamiento α determina la forma de la región de transición y el exceso de la respuesta de banda de paso cerca de la región de transición. Por tanto, determina la forma de la respuesta del filtro y el tipo de filtro.

Para los filtros rechazo banda y el paso banda se define un parámetro llamado Factor de calidad (Q), el cual es la relación entre la frecuencia central de paso y las frecuencias de corte como se muestra en la ecuación 2.18. Para el caso de los filtros activos Q también puede obtenerse de la ecuación 2.19.

$$Q = \frac{f_0}{f_2 - f_1} = \frac{\sqrt{f_1 f_2}}{f_2 - f_1}$$

Ecuación 2.18

$$Q = \frac{1}{\alpha}$$

Ecuación 2.19

Matemáticamente los filtros se encuentran representados por su función de transferencia; las cuales están compuestas por polos y ceros. Los polos dan una atenuación de $20[db]$ y un atraso de la señal de 90° , mientras que, los zeros proporcionarían una ganancia de $20[db]$ y un adelanto de la señal de 90° .

Los filtros pueden ser activos, pasivos o digitales. Los filtros pasivos están compuestos por elementos pasivos (capacitores, resistores e inductores). Los filtros activos además de utilizar elementos pasivos utilizan amplificadores o transistores. Estos filtros se realizan a través de algoritmos numéricos.

Tipos de filtros activos

La función de transferencia es la que determina el tipo de filtro entre ellos se encuentran:

- Butterworth
- Chebyshev
- Bessel
- Cauer o Elípticos

Filtro de Butterworth

La respuesta en frecuencia de un filtro Butterworth es plana en la banda de paso. El factor de atenuación en la banda de transición es de $6[dB/octava]$ Por cada polo. Su respuesta en fase no es lineal. Se usa generalmente cuando todas las frecuencias en la banda de paso deben de tener la misma ganancia. En la figura 2.13 se muestra la respuesta en frecuencia de un filtro Butterworth de distintos órdenes los cuales están dados por n :

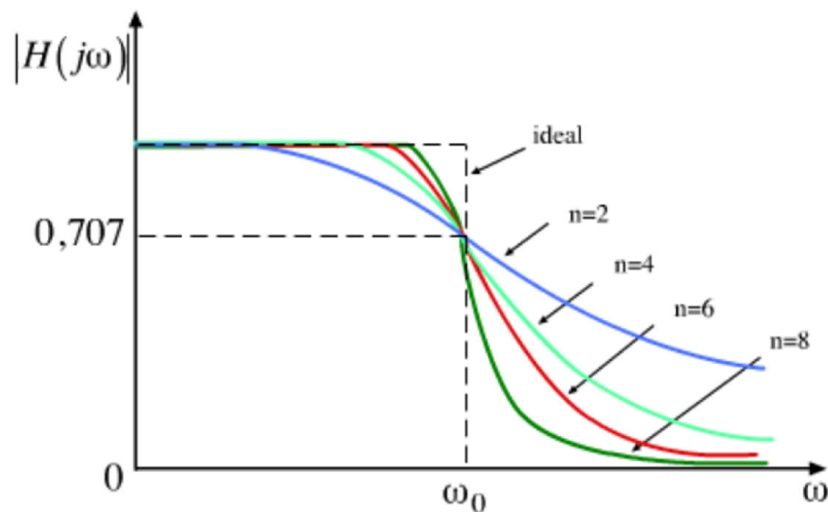


Figura 2.13. Respuesta del filtro Butterworth.

Filtro de Chebyshev

Un filtro de Chebyshev tiene ondulaciones en la banda de paso pero no en la banda de rechazo. Mientras mayor sea el orden del filtro mayor será el número de ondulaciones en la banda de paso como se observa en la figura 2.14. La amplitud de la ondulación puede establecerse en el filtro a diseñarlo generalmente se fija en $0.5[dB]$, $1[dB]$, $2[dB]$ ó $3[dB]$. Mientras más ondulación

se permita, más atenuación se obtendrá en la región de transición, esto puede observarse en la figura 2.14.

El filtro de Chebyshev tiene una variación de atenuación en la región de transición de más de $6[dB/octava]$, por cada polo.

Este filtro es útil cuando la atenuación en la región de transición debe ser muy rápida.

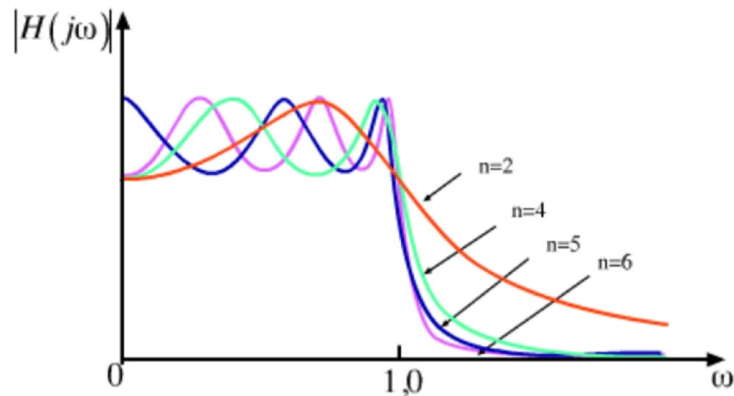


Figura 2.14. Respuestas del filtro Chebyshev.

Filtro de Bessel

Los filtros de Bessel también llamados de fase lineal o de retraso lineal en el tiempo. El retraso de fase en una señal, de la entrada a la salida, aumenta linealmente con la frecuencia. Casi no tienen exceso con una entrada de respuesta escalón. Esta característica hace que sean los mejores para filtrar ondas rectangulares sin alterar la forma de la onda.

También tienen una variación de atenuación en la región de transición de $6[dB/octava]$ por cada polo. La frecuencia de corte de un filtro Bessel se define como: “la frecuencia a la cual el retraso de fase del filtro es la mitad del retraso de fase máximo.

$$\theta(f_c) = \frac{\theta_{max}}{2} = \frac{n\pi/2}{2} [radianes] \quad \text{Ecuación 2.20}$$

Dónde:

θ : Retraso de fase

n : Orden del filtro

La frecuencia de $3[dB]$ de un filtro de Bessel no es la respuesta de corte definida.

Topologías de filtros activos

Existen distintas topologías para la implementación de filtros activos como lo son: el Sallen y Key, retroalimentación múltiple, variables de estado y los cuadráticos. Por su sencillez de

implementación se decide implementar el tipo de retroalimentación múltiple y el Sallen y Key. Por lo que se realizara una breve descripción de cada tipo de filtro así como el procedimiento que se debe de seguir para diseñar un filtro paso bajas.

Filtros de realimentación múltiple

En la figura 2.15 se ilustra las conexiones de un filtro de realimentación múltiple para un par polos complejos conjugados con zeros restringidos en el origen o en el infinito. El amplificador es usado en una configuración inversora con la entrada positiva referenciada a tierra. Cada elemento Y_i representa un simple resistor o un capacitor. La función de transferencia de este circuito es:

$$H(S) = \frac{V_o}{V_i} = \frac{-Y_1 Y_3}{Y_5(Y_1 + Y_2 + Y_3 + Y_4) + Y_3 Y_4} \quad \text{Ecuación 2.21}$$

Este filtro es normalmente usado como un filtro paso banda para factores de calidad menores a 10, ya que implementarlo es relativamente sencillo y ofrece un buen funcionamiento.

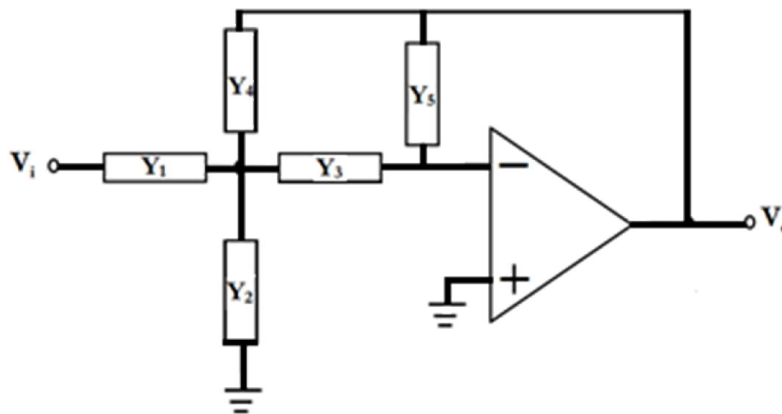


Figura 2.15. Filtro de realimnetación multiple.

Filtros Sallen Key

Este tipo de filtros también llamados VCVS por sus siglas en inglés “Voltage-Controlled Voltage Source”. La impedancia de entrada de estos filtros es de decenas de cientos de miles de mega ohms, dependiendo del tipo amplificador operacional usado, mientras que la impedancia de salida es muy pequeña, usualmente menor a $1[\Omega]$ para ganancias menores a 10^4 . Se muestra en la figura 2.16 la estructura general del filtro Sallen Key. Su ecuación característica está dada por la ecuación 2.22, en donde Y_n Corresponde a las admitancias y K es la ganancia del filtro.

⁴ Tobey, Gene E. et al., **Operational Amplifiers Design and applications**, McGRAW-HILL, United State of America, p. 295

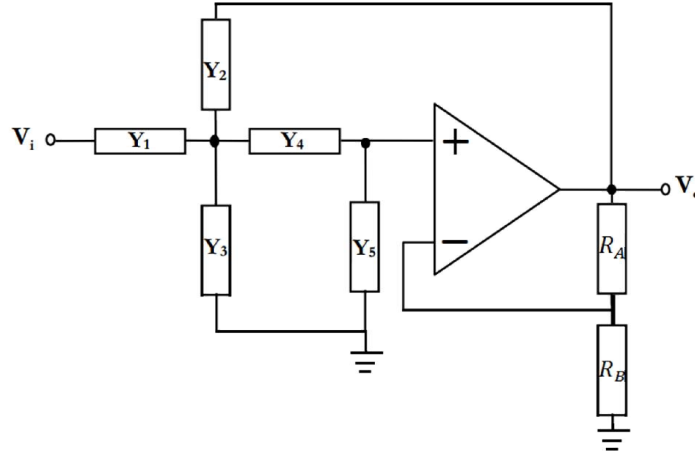


Figura 2.16. Diagrama General del filtro Sallen Key.

$$H(S) = \frac{V_o}{V_i} = \frac{KY_1Y_4}{Y_5(Y_1 + Y_2 + Y_3 + Y_4) + [Y_1 + Y_2(1 - K) + Y_3]} \quad \text{Ecuación 2.22}$$

Diseño de filtros paso bajas

Debido a que es de vital importancia conservar la forma de la onda se necesita de un filtro de fase constante en la banda de paso por lo que se utilizara un filtro de Bessel. Se tienen dos configuraciones distintas para realizar dicho filtro las dos se probaran para ver cuál de las dos ofrece mejores resultados.

Filtro de realimentación múltiple paso banda

Existen muchas configuraciones de cinco elementos que pueden ser usadas para realizar un filtro paso banda. Una de las configuraciones más prácticas se muestra en la figura 2.17, su ecuación de transferencia es:

$$H(s) = \frac{-\frac{S}{R_1C_4}}{S^2 + S \left[\left(\frac{1}{R_5} \right) \left(\frac{1}{C_3} + \frac{1}{C_4} \right) \right] + \left[\left(\frac{1}{R_5C_3C_4} \right) \left(\frac{1}{R_1} + \frac{1}{R_2} \right) \right]} \quad \text{Ecuación 2.23}$$

Mientras que la ganancia del filtro estará dada por:

$$H_0 = \frac{1}{\left(\frac{R_1}{R_5} \right) \left(1 + \frac{C_4}{C_3} \right)} \quad \text{Ecuación 2.24}$$

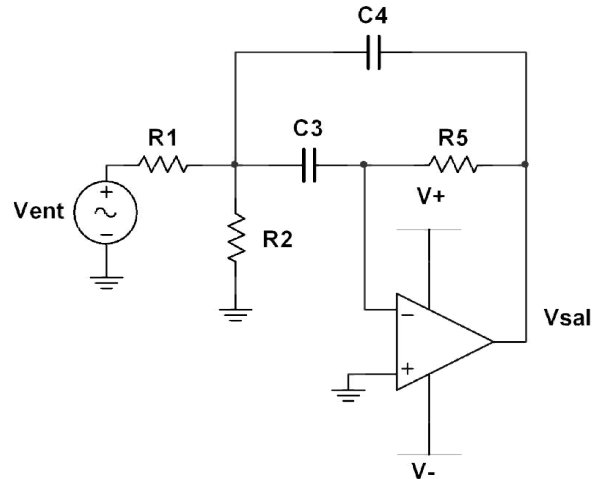


Figura 2.17. Filtro paso banda de realimentación múltiple.

Para el diseño de un filtro de este tipo se deben establecer la ganancia del filtro (H_0) el factor de calidad (Q) y la frecuencia central (f_0) y seguir los pasos siguientes⁵:

- 1) Calcular ω_0 :

$$\omega_0 = 2\pi f_0 \quad \text{Ecuación 2.25}$$

- 2) Escoger una C e igualar $C = C_3 = C_4$
- 3) Calcular R_1 , R_2 y R_5 :

$$R_1 = \frac{Q}{H_0 \omega_0 C} \quad \text{Ecuación 2.26}$$

$$R_2 = \frac{Q}{(2Q^2 - H_0) \omega_0 C} \quad \text{Ecuación 2.27}$$

$$R_5 = \frac{2Q}{\omega_0 C} \quad \text{Ecuación 2.28}$$

Filtro Sallen Key paso bajas

En la figura 2.18 se muestra el diagrama de conexión para realizar un filtro de este tipo y en la ecuación 2.29 su función de transferencia.

$$H(s) = \frac{k}{R_1 R_5 C_1 C_2} \frac{1}{s^2 + s \left(\frac{1}{R_1 C_1} + \frac{1}{R_5 C_1} + (1 - K) \frac{1}{R_5 C_2} \right) + \frac{1}{R_1 R_5 C_1 C_2}} \quad \text{Ecuación 2.29}$$

⁵ Tobey, Gene E. et al., **Operational Amplifiers Design and applications**, McGRAW-HILL, United State of America, p. 293

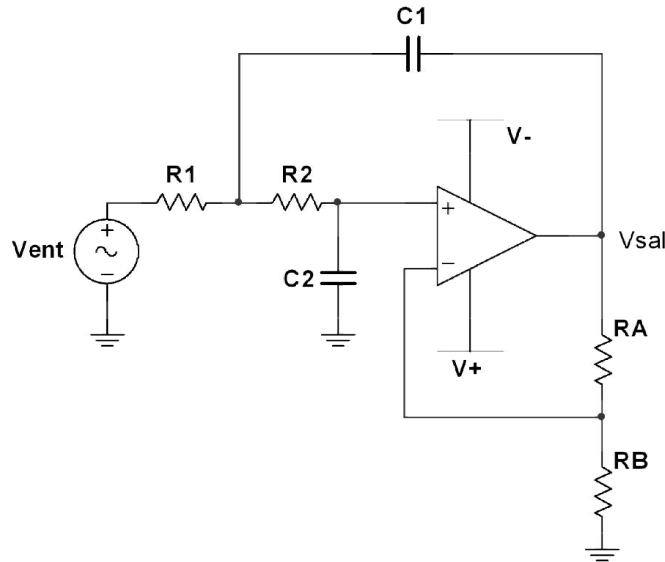


Figura 2.18. Filtro Sallen Key paso bajas.

Para diseñar este tipo de filtro se comienza seleccionando una frecuencia de corte f_c Y se debe hacer $C = C_1 = C_2$ Y $R = R_1 = R_2$ Y se deben seguir los pasos siguientes⁶:

- 1) Seleccionar el tipo de filtro y la $f_{(3\text{ dB})}$ En la tabla 2.1.

Tipo de filtro	α	$f_{(3\text{ dB})}/f_c$
Butterworth	1.414	1.000
Bessel	1.732	0.785
Chebyshev		
Ondulación de 0.5[db]	1.578	1.390
Ondulación de 1[db]	1.059	1.218
Ondulación de 2[db]	0.886	1.074
Ondulación de 3[db]	0.766	1.000
Paso bajas $f_c = f_{(3\text{ dB})}/\text{razón}$ Paso altas $f_c = f_{(3\text{ dB})}(\text{razón})$ Donde $f_{(3\text{ dB})}$: frecuencia de corte deseada de 3[db] f_c: frecuencia usada en los cálculos razón: $f_{(3\text{ dB})}/f_c$ Tomada de la tabla		

Tabla 2.1. Factores de amortiguamiento y razones $f_{(3\text{ dB})}/f_c$ De filtros de segundo orden

- 2) Seleccionar un valor de C conveniente y calcular R a partir de:

⁶ Faulkenberry. Luces M, **INTRODUCCIÓN A LOS AMPLIFICADORES OPERACIONALES CON APLICACIONES A CI LINEALES**, LIMUSA S.A de C.V., México, 1996, p. 265.

$$f_c = \frac{1}{2\pi RC} \quad \text{Ecuación 2.30}$$

- 3) Seleccionar de la tabla 2.1 el factor de amortiguación correspondiente al tipo de filtro elegido
- 4) Elegir un valor de R_A , se recomienda seleccionar el valor de R_A Y calcular el valor R_B A partir de:

$$R_B = (2-\alpha)R_A \quad \text{Ecuación 2.31}$$

- 5) Calcular la ganancia paso banda a partir de

$$A_p = \frac{R_B}{R_A} + 1 \quad \text{Ecuación 2.32}$$

2.3. Conversión análoga digital

En la actualidad el procesamiento de señales se realiza por medio de dispositivos electrónicos como los Microprocesadores, Microcontroladores, dsps, fpgas, entre otros.

Sin embargo, las variables físicas como la temperatura, presión y velocidad, son analógicas y para poder procesarlas a través con los dispositivos anteriormente mencionados necesitamos realizar una conversión del mundo analógico al mundo digital.

En la figura 2.19 se ilustra a manera de diagrama de bloques el proceso de conversión analógico digital.

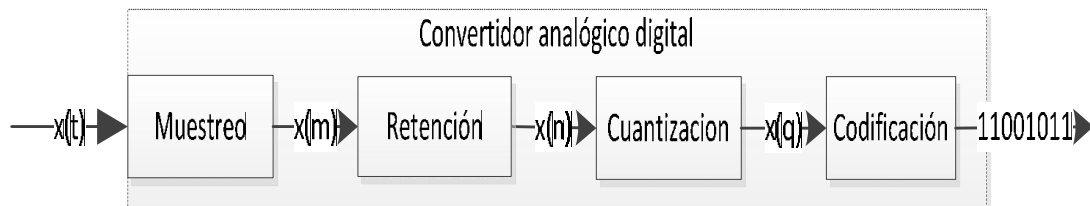


Figura 2.19. Proceso de conversión analógico digital.

El **muestreo** es el proceso que implica adquirir muestras a intervalos específicos de tiempo de una señal analógica. La velocidad con que se toman estas muestras, es decir, el número de muestras por segundo, es lo que se conoce como frecuencia de muestreo.

Retención Las muestras que se toman se deben retener cierto tiempo para que pueda ser procesada, para ello se usan circuitos de retención que mantienen el valor de voltaje por un intervalo de tiempo definido.

La **cuantificación** es la etapa en la que se convierte una sucesión de muestras de amplitud continua en una sucesión de valores discretos, preestablecidos según el código utilizado. Durante este proceso se mide el nivel de tensión de cada una de las muestras obtenidas en el proceso de muestreo y se les atribuye un valor finito (discreto) de amplitud, seleccionado por aproximación dentro de un margen de niveles previamente fijado.

Los valores preestablecidos para ajustar la cuantificación se eligen en función de la propia resolución que utilice el código empleado durante la codificación. Si el nivel obtenido no coincide exactamente con ninguno, se toma como valor el inferior más próximo. En este momento, la señal analógica (que puede tomar cualquier valor) se convierte en una señal digital, ya que los valores que están preestablecidos son finitos.

La señal que resulta de la cuantificación es sensiblemente diferente a la señal analógica que la originó, por lo que siempre va existir una cierta diferencia entre ambas, que es conocido como error de cuantificación, el cual se produce cuando el valor real de la muestra no equivale a ninguno de los escalones disponibles para su aproximación y la distancia entre el valor real y el que se toma como aproximación es muy grande. Un error de cuantificación se convierte en un ruido cuando se reproduzca la señal tras el proceso de decodificación digital.

La **codificación** consiste en la traducción de la señal ya cuantificada al sistema binario, mediante códigos preestablecidos. Finalmente la señal analógica quedará transformada en un tren de impulsos digitales.

Los criterios más importantes a tomar en cuenta al momento de elegir un convertidor analógico digital son:

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

Los convertidores ADC más comunes son:

- *ADC flash*
- ADC de aproximaciones sucesivas
- ADC comparador de tipo rampa
- ADC sigma-delta
- ADC paralelo

2.3.1. Convertidor de analógico digital de aproximaciones sucesivas

La mayoría de los microcontroladores integran un convertidor analógico digital (ADC) de aproximaciones sucesivas. En la figura 2.20 se muestra el diagrama de bloques de un ADC de aproximaciones sucesivas, el cual consta de: un convertidor digital analógico, un comparador y un registro de aproximación sucesiva (SAR). Adicionalmente se necesita: una terminal para el voltaje de entrada, así como de salidas digitales y de terminales de control⁷.

La orden de “inicio de conversión” inicia el ciclo de conversión analógico digital. El registro de aproximación sucesiva conecta la secuencia de números digitales, un número por cada bit, a las entradas del convertidor digital analógico.

El Convertidor digital analógico transforma cada número digital en una salida analógica que se compara con el voltaje de entrada al convertidor. El comparador le indica al registro de aproximaciones sucesivas cuando el voltaje de entrada es mayor o menor que la salida del convertidor digital analógico.

Las comparaciones se hacen comenzando con el bit más significativo y terminan con el bit menos significativo. La conversión inicia con el bit más significativo en estado “alto”, mientras que los demás permanecen en estado bajo: si el resultado de la comparación indica que el valor de la señal de entrada es mayor que el obtenido por el convertidor digital analógico se conserva el estado del bit de lo contrario cambiara de valor el estado, una vez determinado el estado se conserva y se proceder a realizar el mismo procedimiento con los bits restantes.

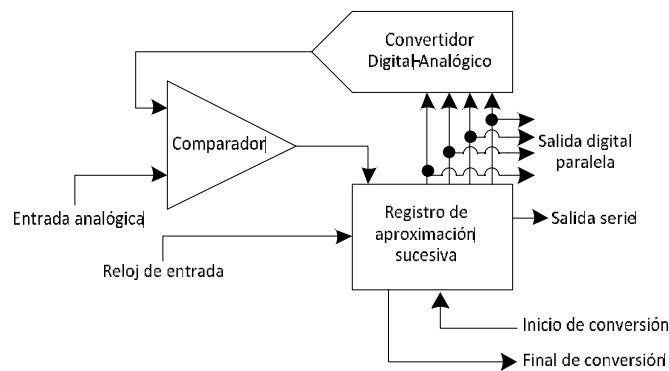


Figura 2.20. Diagrama de bloques de un convertidor de aproximaciones sucesivas.

⁷ Coughlin Robert F. *Amplificadores operacionales y circuitos integrados lineales*, Prentice Hall. México, 1999.

2.3.2. Teorema de muestreo de Nyquist

El teorema de muestreo de Nyquist establece que una señal continua en el tiempo $x(t)$ limitada en banda, puede ser reconstruida si se muestrea a una tasa tal que $f_m > 2f_{max}$, donde f_m Es la frecuencia de muestreo y f_{max} Frecuencia máxima de $x(t)$.

En la figura 2.21 se observa el espectro de una señal continua limitada en banda, donde ω representa el ancho de banda de la señal.

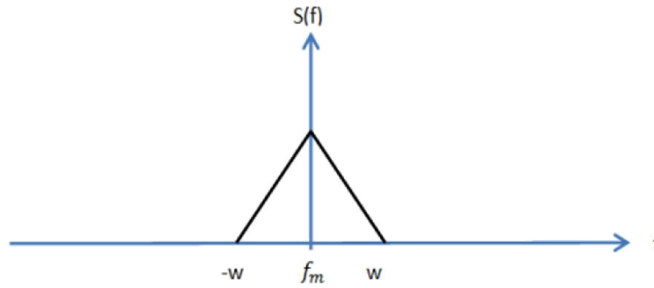


Figura 2.21. Espectro de una señal continua limitada en banda.

En la figura 2.22 se observa una señal senoidal de $100[Hz]$, la cual es muestreada a una frecuencia de $300[Hz]$, con lo cual se cumple el teorema de Nyquist.

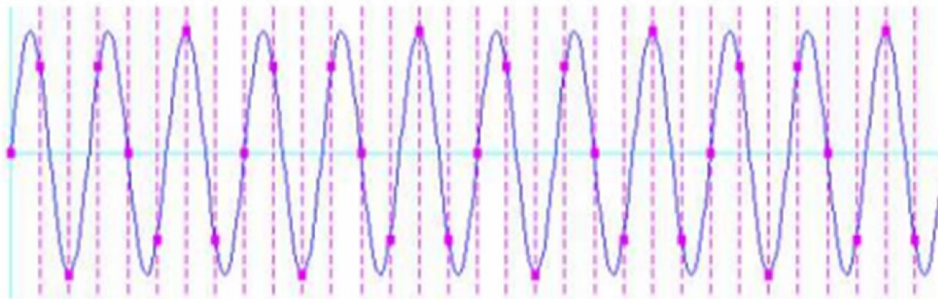


Figura 2.22. Señal muestreada siguiendo el teorema de Nyquist.

En la figura 2.23 podemos observar el espectro de la señal muestreada el cual se encuentra centrado en múltiplos de f_m . Los cuales no se traslapan ya que cumplen con el teorema de Nyquist.

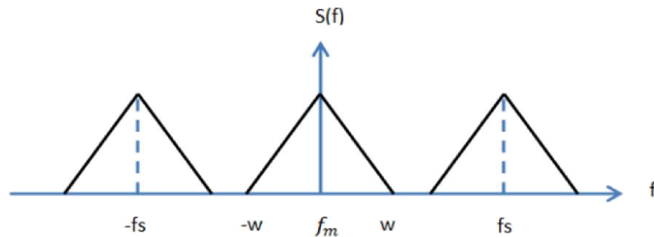


Figura 2.23. Espectro de una señal muestreada limitada en banda.

Sin embargo, cuando no es cumplido el teorema de Nyquist, es decir, cuando $x(t)$ no es limitada en banda se produce una superposición espectral lo que producirá un efecto de “aliasing”, donde los componentes en frecuencia cercanos al límite de banda se modifican como se ilustra en la figura 2.24. Para evitar el efecto de “aliasing” se debe colocar un filtro paso bajas a la señal de entrada para limitarla en banda.

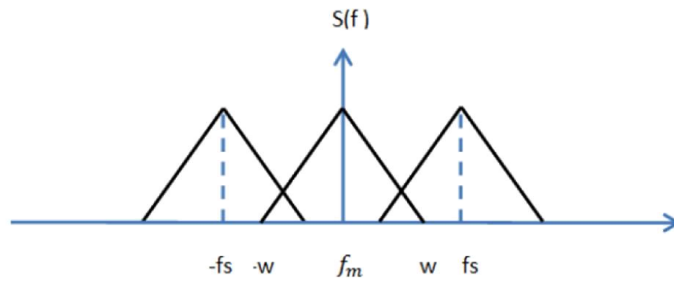


Figura 2.24. Señal muestreada con superposición espectral.

Otra forma de producir “Aliasing” es por medio de un submuestreo también producido por el solapamiento espectral, en este caso se produce una nueva señal que no tiene ninguna relación con nuestra señal muestreada, lo cual se ilustra en figura 2.25.

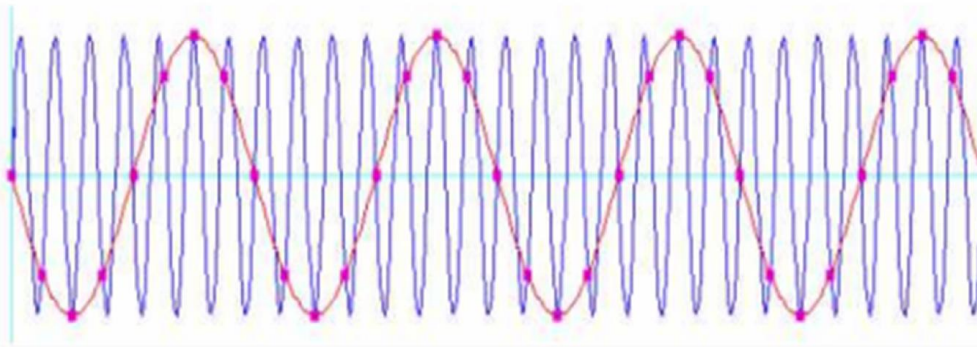


Figura 2.25. Señal submuestreada.

2.4. Comunicación serie

El intercambio de información digital entre un microcontrolador y los periféricos puede ser de forma serie o paralela. La comunicación en paralelo implica el envío simultáneo de información debido a que se utiliza una línea por bit, lo que lo hace ideal para transmitir grandes cantidades de información, sin embargo, esto implica una cantidad mayor de cables y problemas de capacitancias parásitas entre ellos, por lo que las distancias de transmisión suelen ser muy pequeñas. La comunicación serie es una forma de transmitir información con un bus mínimo de conexión, debido a que ocupan una cantidad menor de espacio y que puede transmitir información a mayores distancias de transmisión, sin embargo, la velocidad de transmisión es menor que la comunicación en paralelo.

Existen dos tipos de comunicación serie: síncrona y asíncrona. La comunicación síncrona se caracteriza por tener en alguna de sus líneas una señal de reloj que ayuda a los dispositivos al envío y recepción de datos. En la comunicación asíncrona cada dispositivo tiene su propia señal de reloj pero en la conexión física se carece de una línea que los sincronice, en cambio se establecen métodos para lograr la sincronización.

2.4.1. Estándar SPI

El Estándar SPI (del inglés *Serial Peripheral Interface*) es un estándar establecido por Motorola. La comunicación se realiza un modelo Maestro – Esclavo. El maestro selecciona al esclavo así como también es quien inicia y finaliza la comunicación. Es una comunicación del tipo serie síncrona que está constituida por un bus *full dúplex*⁸, es decir, se puede enviar y recibir información de manera simultánea lo que eleva la tasa de transferencia de datos.

Como se observa en la figura 2.26Figura 2.26 el bus de comunicación consta de 4 líneas que son:

- **Línea SCLK:** Esta línea es generada y manipulada solamente por el maestro, ésta se encarga de sincronizar la transmisión de datos, debido a que envía una señal de reloj.
- **Línea MOSI:** Se encarga de enviar los datos del maestro hacia el esclavo.
- **Línea MISO:** Se encarga de enviar los datos del esclavo hacia el maestro
- **Línea de selección de dispositivos (CS: Chip select, SS: Slave select):** Esta línea es controlada por el maestro y se encarga de seleccionar el dispositivo esclavo con el cual se desea tener comunicación cambiando el nivel lógico de “1” a “0”.

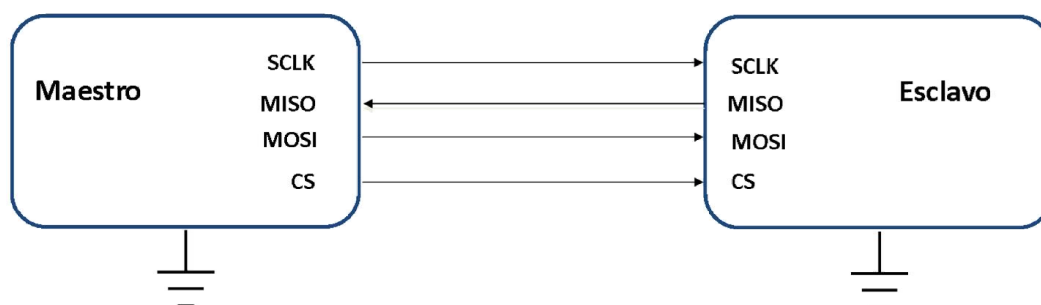


Figura 2.26. Conexión SPI con un solo esclavo.

En una red donde existen más de un dispositivo esclavo, se debe de utilizar una línea de chip select para cada uno de ellos, esto con el fin de que sólo el dispositivo que sea seleccionado tenga un intercambio de datos con el maestro, en la figura 2.27 se muestra un diagrama de conexiones de múltiples esclavos al maestro.

⁸ *Full dúplex*. En comunicaciones, término que indica la capacidad de enviar y recibir información de manera simultánea.

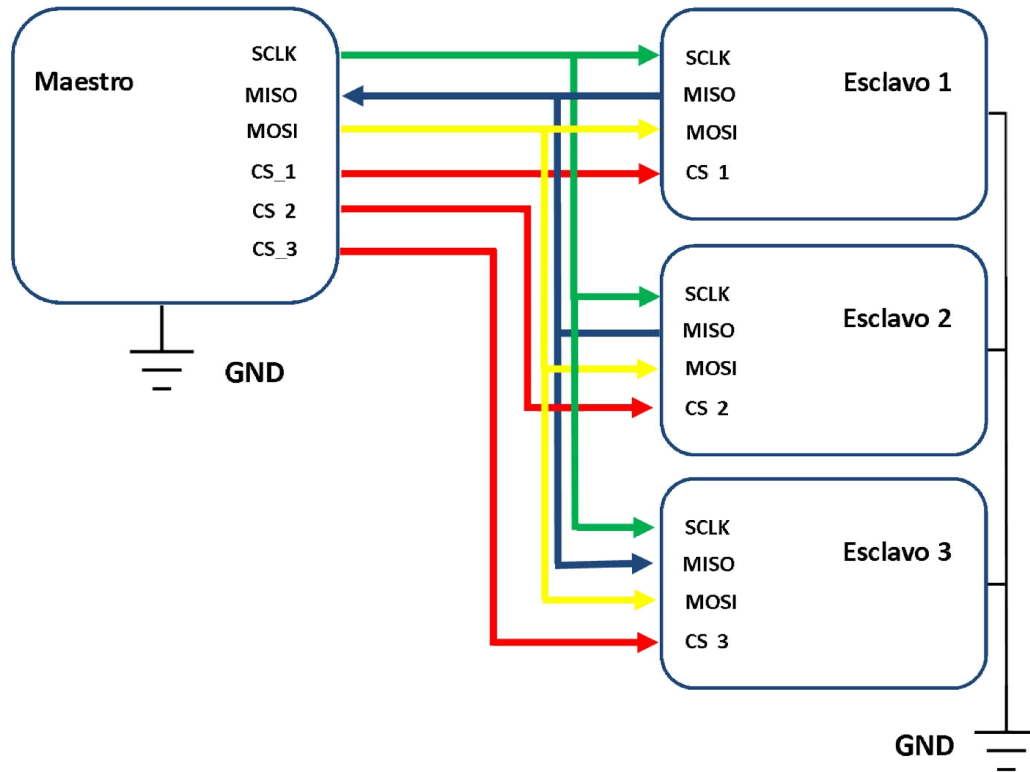


Figura 2.27. Conexión SPI multi esclavos.

Los datos son transferidos usualmente en paquetes de 8 bits, sin embargo, algunos dispositivos pueden operar con paquetes de datos de 4 bits a 16 bits.

Este protocolo cuenta con dos parámetros SPO (polaridad del reloj) y SPH (fase del reloj), con los cuales se determina con el momento en el cual se considera válido un dato de entrada o cuando se genera un dato de salida. El valor SPO determina en qué estado se encontrará la línea del reloj mientras no esté transmitiendo; si su valor es '1' SCLK estará en alto mientras no se estén transmitiendo datos, mientras que cuando este '0' se mantendrá SCLK en un estado bajo mientras. El valor de SPH determina en que flanco del reloj es adquirido el dato, cuando el bit se encuentra en bajo valor se toma en el primer flanco y cuando está en alto se toma en el segundo flanco. Con base en lo anterior se muestra en la figura 2.28 los cuatro modos de operación en los cuales se marca en que momento un dato es válido.

El algoritmo para envío y recepción de datos entre los dispositivos maestro y esclavo es el siguiente:

1. El maestro debe de habilitar al dispositivo esclavo mediante su línea de *chip select*, cambiando el estado lógico de esta; usualmente los dispositivos son seleccionados cuando esta línea cambia su estado lógico en bajo, sin embargo, existen dispositivos que se seleccionan en estado lógico alto.

2. El maestro envía las instrucciones por medio de la línea MOSI, los datos se sincronizarán por medio de la línea SCLK. Si el esclavo responde enviando datos por la línea MISO.
3. Deshabilitar el dispositivo esclavo poniendo en el estado original su línea de *chip select*.

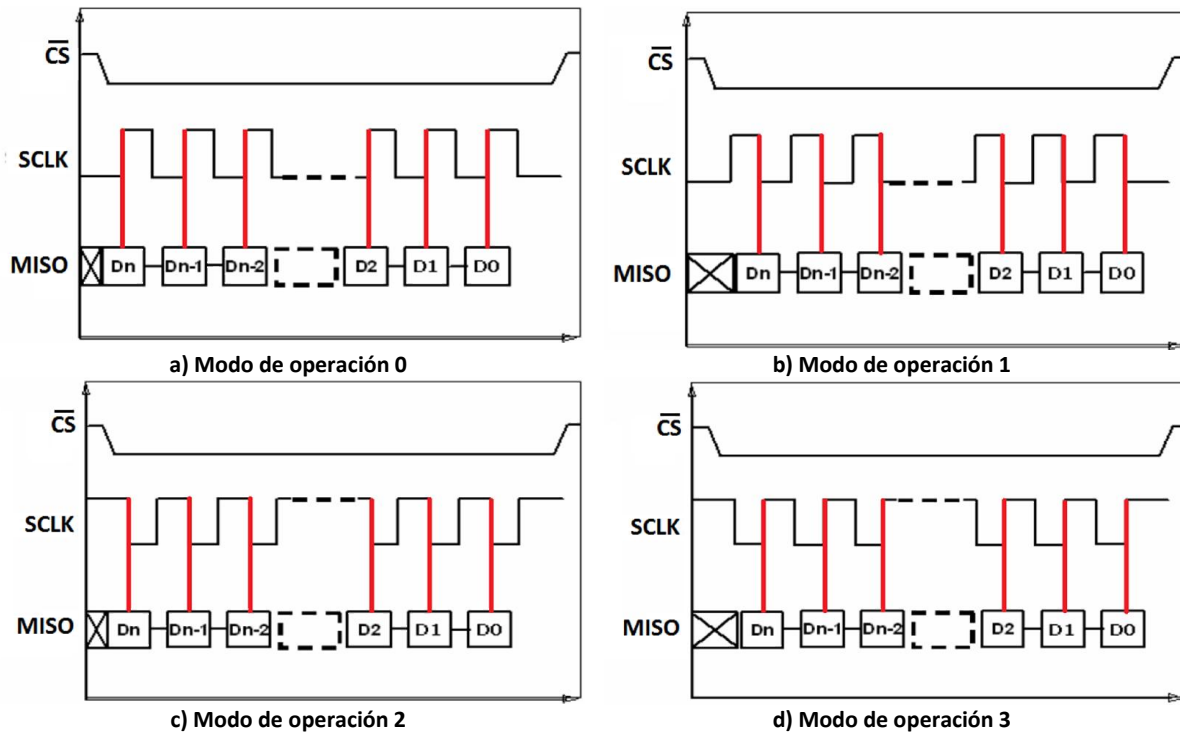


Figura 2.28. Modos de operación del estándar SPI.

2.4.2. UART

UART son las siglas de “*Universal Asynchronous Receiver Transmitter*”, es un canal de comunicación asíncrona *Full dúplex*.

Típicamente, la comunicación UART se utiliza para transmitir datos en formato ASCII. Para realizar la comunicación como se muestra en la figura 2.29 se necesitan de 3 líneas:

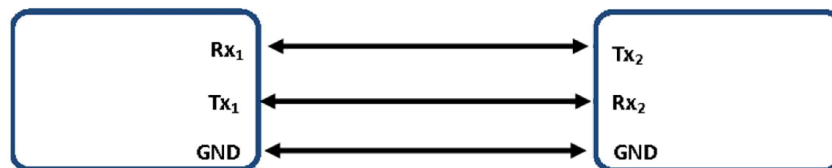


Figura 2.29. Conexión UART.

- Línea de transmisión (Tx_n)
- Línea de recepción (Rx_n)
- Tierra (GND)

Las características más importantes de la comunicación serie son:

- Velocidad de transmisión (baud rate): Indica el número de bits por segundo que se transfieren y su unidad de medición son los baudios (bauds).
- Bit de inicio: Cuando el receptor detecta el bit de inicio sabe que la transmisión ha comenzado y es a partir de entonces que debe leer la transmisión y entonces debe leer las señales de la línea a distancias concretas de tiempo, en función de la velocidad de transmisión.
- Bits de datos: Se refiere a la cantidad de bits en la transmisión y recepción de datos, la cantidad de datos no necesariamente será de 8 bits, las cantidades más comunes de bits son: 5, 7 y 8 bits. Los bits son enviados desde el menos significativo hasta el más significativo.
- Bit de Paridad: Con este bit se pueden descubrir errores en la transmisión. Se puede dar paridad par o impar. En la paridad par, por ejemplo, la palabra de datos a transmitir se completa con el bit de paridad de manera que el número de bits 1 enviados es par.
- Bit de parada: Se usan para indicar el fin de la comunicación de un solo paquete. Los valores típicos son: 1, 1.5 y 2 bits.

Los paquetes de datos se envían de la siguiente forma: de un bit de inicio, bit de datos, bit de paridad y bit de parada esto se ilustra en la figura 2.30.

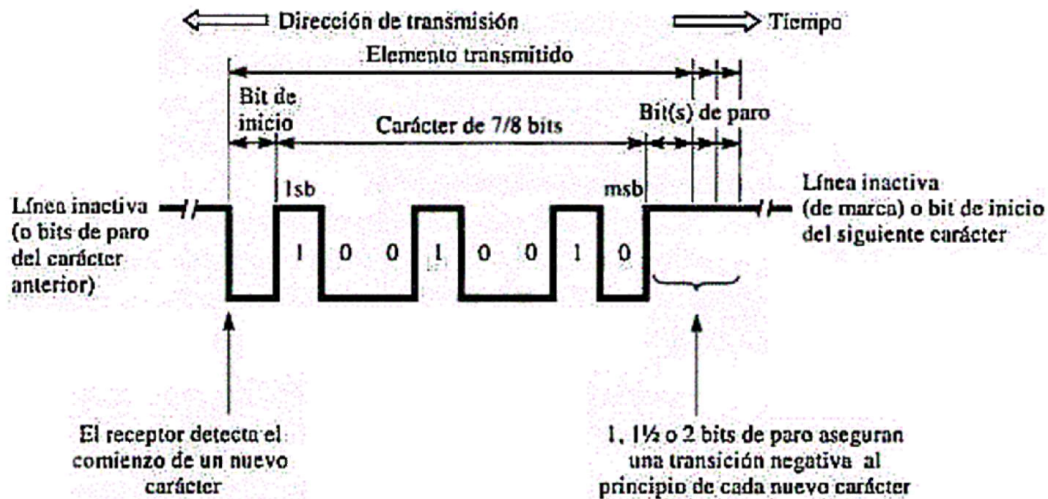


Figura 2.30. Protocolo de comunicación asíncrona serie.

Aunque existen estándares como el RS-232 o el RS-422, los dispositivos electrónicos como los microcontroladores que usan este tipo de comunicación no manejan los niveles de voltaje de estos estándares por lo que se deben adecuar las señales con el uso de *transceivers* si se requiere trabajar con los estándares. En el caso de los microcontroladores para un uno lógico el valor es GND el valor para un cero lógico suele ser el voltaje de alimentación.

2.5. GPS

GPS por sus siglas del inglés Global Positioning System. Este sistema está basado en la información obtenida de una red de 22 satélites que se encuentran en la órbita terrestre. Este sistema fue desarrollado por el Departamento de Defensa de los Estados Unidos, fue pensado inicialmente con fines militares y se denominaba "TRANSIT". Posteriormente se volvió disponible para uso civil y se conoce como "NAVASTAR".

Los satélites que forman la red orbitan a aproximadamente 20,000[km] Por encima de la superficie terrestre y están en constante movimiento, realizando dos orbitas en menos de 24 horas. Son alimentados por energía solar, se mueven con unos pequeños cohetes que los ayudan a seguir el camino correcto.

Las órbitas de los satélites GPS están inclinadas respecto al ecuador de la Tierra en alrededor de 55°. La distribución espacial de la constelación de satélites permite al usuario disponer de 5 a 8 satélites visibles en cualquier momento. El sistema está diseñado para asegurar que al menos cuatro satélites estarán visibles con una recepción configurada de la señal de 15° sobre el horizonte en un momento dado, en cualquier parte del mundo.

Cada satélite transmite una señal de radio compuesta de dos frecuencias de portadora, moduladas por dos códigos digitales y un mensaje de navegación; las dos frecuencias son generadas a 1,575.42 mhz (portador L1) y 1,227.60 mhz (portador L2). La disponibilidad de las dos frecuencias de portador permite corregir uno de los mayores errores del GPS: el retraso ionosférico. Todos los satélites transmiten las mismas frecuencias de portador L1 y L2, pero la modulación es diferente en cada satélite, para minimizar la interferencia de la señal y permitir al receptor identificar cual satélite está transmitiendo un código en particular.

La portadora L1 es modulada por dos códigos. El Código C/A o Código de Adquisición Gruesa modula a 1.023mhz (10.23/10) y el código P o Código de Precisión modula a 10.23mhz. L2 es modulada por un código solamente. El código P en L2 modula a 10.23 mhz. Cada código consiste en un flujo de datos binarios, conocidos como bits o chips. Esta modulación recibe el nombre de modulación de doble fase, ya que la frecuencia del portador varía 180° cuando los valores del código cambian de cero a uno o de uno a cero. Los códigos se conocen como códigos PRN (*Pseudo Random Number*), debido a que parecen códigos aleatorios, aunque en realidad son generados por algoritmos complejos.

El código C/A es una cadena de 1,023 dígitos binarios que se repiten cada milisegundo (1023 Mbps), mientras que el código P es una cadena muy larga de dígitos binarios que se repite cada 266 días, este código está dividido en 38 segmentos de 1 semana: 32 para cada uno de los satélites, el resto está reservado para otros usos. Los receptores GPS utilizan estos códigos para distinguir los satélites.

Principio de funcionamiento

El principio de funcionamiento del GPS es simple; se basa en la medición de la distancia desde los satélites hasta el receptor GPS en la tierra. Esta distancia hacia cada satélite puede ser determinada por el GPS. La idea básica es la siguiente: si se conoce la distancia hacia tres puntos en relación a una posición, entonces se puede determinar la posición relativa a esos tres puntos.

Como se observa en la figura 2.31, si se conoce la distancia hacia un satélite sabemos la posición del receptor debe estar en algún punto sobre la superficie de una esfera imaginaria cuyo origen es el satélite mismo. En teoría la posición del receptor se podrá determinar al intersectar tres esferas imaginarias, por lo que tendrían que ser necesarios tres satélites, sin embargo, es necesario un cuarto satélite para compensar los retrasos que pueda sufrir el receptor.

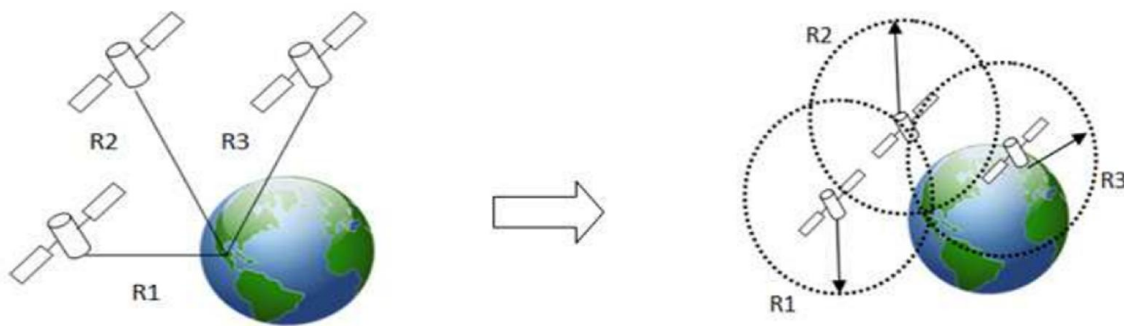


Figura 2.31. Principio de funcionamiento del GPS.

Para medir la distancia a cada satélite se utiliza ecuación 2.33 que es una de las leyes del movimiento

$$\text{Distancia} = \text{velocidad} \times \text{tiempo} \quad \text{Ecuación 2.33}$$

Donde, la velocidad es el de las señales de radio y el tiempo es el que toma a una señal de radio en viajar desde el satélite al receptor GPS. Las señales de radio viajan a la velocidad de la luz, aproximadamente a $3 \times 10^8 \left[\frac{m}{s} \right]$, el tiempo es un poco difícil de calcular, ya que se necesita conocer el momento en que la señal de radio salió del satélite y el momento en que llegó al receptor.

Suponiendo que el reloj del satélite como el del receptor están sincronizados perfectamente, cuando el código PRN (C/A o P) es transmitido desde el satélite al receptor, el receptor genera una copia exacta del mismo, después de un tiempo (equivalente al tiempo recorrido de la señal en el espacio), el código transmitido llega al receptor. El cual comparará el código transmitido con su réplica con lo que podrá obtener el tiempo de recorrido, hecho esto podrá determinar la distancia a la que se encuentra.

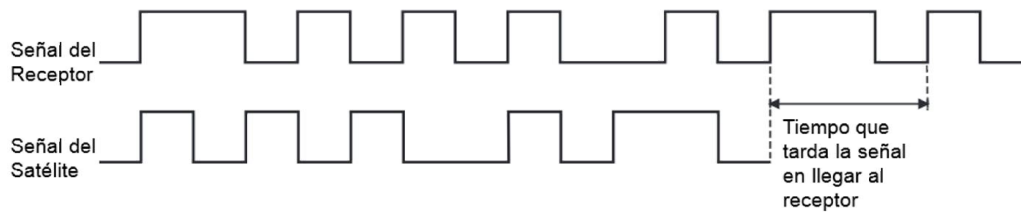


Figura 2.32. Pseudorange de tiempo GPS.

Fuentes de error

Hasta este momento, se ha asumido que la posición obtenida del GPS es muy precisa y libre de errores, pero existen diferentes fuentes de error que degradan la posición GPS desde algunos metros, hasta algunas decenas de metros. Algunas de estas fuentes de error son:

- Retrasos ionosféricos y atmosféricos
- Errores en el reloj del satélite y el receptor
- Efecto multitrayectoria
- Dilución de la precisión

2.5.1. Estándar NMEA

NMEA es la abreviatura de National Marine Electronics Association. Es una asociación fundada en el año de 1957 por un grupo de fabricantes de electrónica, con la finalidad de obtener un sistema de comunicación común entre las diferentes marcas de electrónica naval. Poco a poco se fueron sumando todos los fabricantes a este estándar, además de organizaciones oficiales y gubernamentales. NMEA fue creado para el intercambio de información digital entre productos electrónicos marinos. El primer protocolo estándar se llamó NMEA 0183, y es el que todavía se utiliza y acepta en la mayoría de equipos electrónicos destinados para la navegación como es el caso de las ecosondas, los sonares, los anemómetros, los girocompases y los receptores GPS.

El estándar NMEA consiste en una línea de texto con caracteres ASCII. La línea comienza siempre con el carácter "\$" y luego el nombre de una sentencia NMEA seguida de un número determinado de campos (los que contienen la información), los cuales, están separados por comas.

Las sentencias NMEA se dividirán en dos grupos; los primeros que van del microcontrolador hacia el GPS se denominan comandos y los segundos son los mensajes que van del GPS hacia el microcontrolador.

Comandos NMEA

Los comandos NMEA son instrucciones de configuración para el GPS con ellos se puede controlar el número de mensajes que envía el GPS hacia el dispositivo maestro, cambiar la velocidad de comunicación.

En la figura 2.33 se muestra la estructura de los comandos NMEA. El comando empieza con un preámbulo que es "\$", seguido de una etiqueta y del número de comando en formato ASCII, seguido de una coma los argumentos de la función los cuales estarán separados por comas, después sigue un * el cual determina el fin del comando, inmediatamente se encuentran dos bytes que pertenecen al checksum el cual se calcula efectuando la operación XOR con todos los caracteres que se encuentran entre el preámbulo "\$" y "*" este se envía en formato ASCII, finalmente se envía un "enter" el cual está formado por un retorno de carro (CR) y un salto de línea (LF).



Figura 2.33. Formato de comandos NMEA.

Uno de los comandos más usados es el PMTK251, el cual cambia la velocidad de transmisión, la figura 2.34 es un ejemplo para configurar el GPS a 115200[baudios], este comando puede tener como argumento las siguientes velocidades:

- 4800
- 9600
- 14400
- 19200
- 38400
- 57600
- 115200

```
$PMTK251,115200*1F
```

Figura 2.34. Comando PMTK para cambiar la velocidad de transmisión.

Otro de los comandos usados es el PMTK314 (ver figura 2.35) en el cual se pueden habilitar hasta 19 mensajes los cuales son habilitados o deshabilitados de forma ordenada; es decir, si se quiere a habilitar el mensaje 0, enseguida de la primera coma se debe colocar un número diferente de cero, este número representa el número mínimo de satélites con el que se tendrá que comunicar el GPS para obtener información.

```
$PMTK314,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0*29
```

Figura 2.35. Comando PMTK para habilitar mensajes.

Mensajes NMEA

Empiezan con las letras GP excepto las sentencias propias de los fabricantes que son adicionadas al repertorio general de la NMEA.

Estos mensajes pueden incluir información sobre la posición fecha, altitud y otras variables. De forma a los comandos, los mensajes son enviados en cadenas que comienzan con “\$”, seguido de cinco caracteres que identifican al tipo de mensaje seguido de la información separada por comas (“,”) y terminan con * el cual determina el fin del comando seguido del checksum. En la figura 2.36 se ilustra lo anterior.

\$GPRMC,220831.000,A,1919.5172,N,09908.6919,W,1.34,316.25,020913,,,A*74

Figura 2.36. Mensaje NMEA.

En la tabla 2.2 se muestran los mensajes más usados por el protocolo NMEA

Mensaje	Tipo de datos
GGA	Datos de localización.
GLL	Datos de latitud y longitud
GSA	Datos de los satélites en general
GSV	Datos de un satélite detallado
BOD	Recorrido de origen destino
RMC	Datos mínimos de posicionamiento, tiempo y velocidad
VTG	Datos del curso y velocidad con respecto al suelo
RMB	Datos mínimos de navegación

Tabla 2.2. Mensajes más usados por el protocolo NMEA.

Como se observa en la tabla 2.2 cada mensaje proporciona información diferente. Por ejemplo, la cadena GPRMC está compuesta por los parámetros siguientes:

- Hora UTC en formato de 24 horas
- Estado de datos: A (Estado correcto) o V (sin ubicación)
- Latitud
- Norte o sur (N/S)
- Longitud
- Este u oeste(E/W)
- Velocidad en nudos
- Orientación en grados
- Fecha UT
- Variación magnética
- Este u oeste(E/W)
- Cheksum

Para comprender mejor se analizara el mensaje siguiente:

\$GPRMC,220831.000,A,1919.5172,N,09908.6919,W,1.34,316.25,020913,,,A*74

- Hora UTC: **22:08:31**
- Estado de datos: A (Estado correcto)
- Latitud: **1919.5172**
- Norte (N)
- Longitud: **09908.6919**
- Oeste (W)
- Velocidad en nudos: **1.34**
- Orientación en grados: **316.25,020913**
- Fecha: **020913(2 de septiembre de 2013)**
- Cheksum: **74**

2.6. GPRS

GPRS (por sus siglas en inglés “*General Packet Radio Service*”). En los últimos años la telefonía móvil ha tenido un gran desarrollo tecnológico y un gran éxito comercial. Es común encontrar sobre todo en grandes ciudades teléfonos celulares que nos permiten: realizar llamadas, enviar mensajes de texto, recibir y enviar información de sitios web.

Desde sus inicios en el año de 1979, la telefonía móvil ha tenido etapas de desarrollo a las cuales se les conoce como generaciones. En general, el de desarrollo de los sistemas celulares, en sus diferentes generaciones, se ha dado como se indica en la figura 2.37.

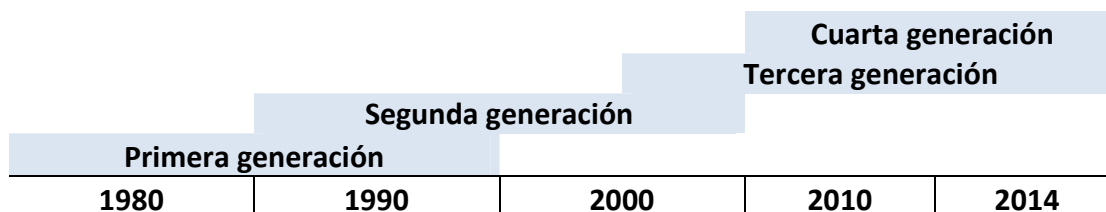


Figura 2.37. Desarrollo de generaciones móviles a través de los años.

En la Primera generación (1G) los sistemas móviles se caracterizaban por realizar la transmisión analógica de voz de baja calidad, utilizando para su funcionamiento la técnica de “Acceso Múltiple por división de frecuencia” (FDMA); que limitaba el número de usuarios y la capacidad de moverse a otra zona de cobertura (*roaming*). Se utilizaban sistemas analógicos y no existían frecuencias estándar entre los países.

Segunda generación. Ante la limitada cantidad de canales debido a la implementación de FDMA, surgen, en la segunda generación (2G) alternativas como TDMA (*Time División Multiple Access*) y FDMA (*Code División Multiple Access*).

En esta generación surge el sistema GSM (del inglés “*Global System for Mobile Communication*”), que es un sistema de comunicación basado en células de radio. Empezó a desarrollarse desde 1982, cuando se forma el “*Group Special Mobile*”, cuyo propósito era crear un

estándar europeo de telefonía móvil digital que solucionara los problemas de las redes analógicas, ya que estos no eran del todo compatibles entre sí.

GSM fue diseñado para la transmisión de voz y está basado en la conmutación de circuitos. Al realizar la transmisión mediante conmutación de circuitos, los recursos quedan ocupados durante toda la comunicación.

Arquitectura general del GSM

En la figura 2.38 se muestra de forma general las partes que componen la arquitectura de la red GSM, las cuales se dividen en:

1. Estación móvil
 - Terminal. Son los equipos como celulares o módems GSM. Esta identificado por su número IMEI (International Mobile Equipment Identity).
 - SIM (*Subscriber Identity Module*). Contiene información que posibilita la utilización de la red como:
 - Identificador de usuario
 - Claves de criptografía
 - Agenda de usuario
2. Subsistema de estación base
 - Transmisor y receptor de la estación base. Es la entidad responsable del establecimiento de las comunicaciones con las estaciones móviles que se encuentran dentro de su área de influencia.
 - Controlador de la estación base. Controla los recursos radio de las estaciones de transmisión y recepción conectadas.
3. Subsistema de red
 - Centro de registro del equipo.
 - Centro de autenticación.
 - Centro de conmutación del servicio móvil.

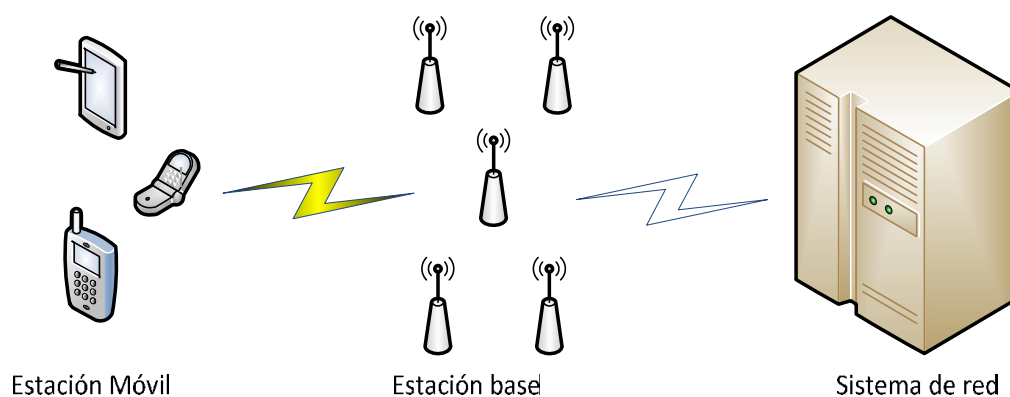


Figura 2.38. Diagrama general de la arquitectura del GSM.

Existe una etapa intermedia de 2G y 3G a esta etapa se le ha denominado como 2.5G. En esta etapa nace el GPRS el cual trabaja por medio de conmutación de paquetes de datos, lo que hace posible la conmutación de canales de comunicación y comienzan a ofrecerse servicios de navegación por internet (sólo texto sin formato).

Una conexión GPRS está establecida por la referencia a su nombre del punto de acceso (APN). Con GPRS se pueden utilizar servicios como Wireless Application Protocol (WAP), servicio de mensajes cortos (SMS), servicio de mensajería multimedia (MMS), internet para los servicios de comunicación, como el correo electrónico y la World wide web (WWW). Para fijar una conexión de GPRS para un módem inalámbrico, un usuario debe especificar un APN, opcionalmente un nombre y contraseña de usuario, y muy raramente una dirección IP, todo proporcionado por el operador de red.

Una de las principales diferencias entre el GSM y el GPRS puede ejemplificarse con las llamadas, en el GSM se establece una conexión directa entre dos dispositivos, para ello se necesita asignar un canal de comunicación, el cual estará ocupado mientras se esté conectado por lo que el costo del servicio se realiza por el tiempo de conexión. En el caso del GPRS los canales de comunicación se comparten entre los usuarios de forma dinámica, de este modo al usuario solo se le asigna un canal cuando realmente se están transmitiendo datos y el costo del servicio se realiza por la cantidad de datos intercambiados.

Las principales características de la red GPRS son:

- Velocidad de transferencia hasta 114[Kbps]
- Tiempo de establecimiento de conexión menor a un segundo
- Pago por cantidad de información intercambiada
- Conexión móvil a internet

2.6.1. Comandos AT

Los comandos AT son instrucciones codificadas que conforman el lenguaje de comunicación entre un usuario y un terminal módem; son de carácter genérico en su mayoría, ya que un mismo comando funciona en modelos de distintas marcas, haciendo que un programa basado en comandos AT sea robusto y compatible con la mayor parte de los dispositivos disponibles en el mercado. La gran parte de los módems disponibles reconocen los comandos AT más utilizados. La tecnología GSM ha adaptado el uso de estos comandos.

Los comandos AT, poseen en su mayoría un prefijo dado por 'AT'. Cada acción que se desee viene precedida por este prefijo. Algunos comandos, llevan al final un signo de interrogación (?), esto quiere decir que se está pidiendo información. Mientras que un signo igual (=) indica que se está configurando un parámetro, donde luego del signo de igual se ingresa el valor o valores de los parámetros separados por coma que se desean ajustar. La expresión igual-interrogación (=?), se usa para obtener todo el rango de valores posibles que se pueden configurar.

2.7. Tarjeta de memoria microSD

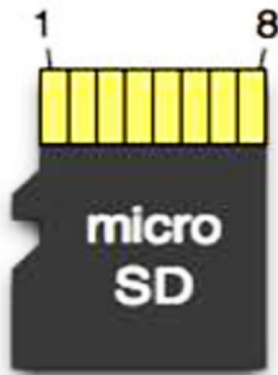
La tarjeta de memoria microSD es una tarjeta de memoria flash desarrollado por sandisk y adoptada por *SD Association* bajo el nombre de “microSD”. En la figura 2.39 se muestra la imagen de una memoria microSD, la cual por sus dimensiones que son de 15[mm] x 11[mm] x 1[mm] suelen utilizarse en dispositivos en donde factores como el tamaño es crítico, como en los celulares, reproductores mp3, consolas de videojuegos, etc.



Figura 2.39. Memoria microSD.

Las memorias microSD tienen capacidades que van desde los 16 [MB] a los 2 [GB], sin embargo, existen la memorias “SDHC” que tienen una mayor capacidad de almacenamiento que van de los 4[GB] a los 64[GB] estas tienen las mismas dimensiones que las memorias microSD y su inicialización y funcionamiento tiene muy pocas variaciones.

Soporta los buses de comunicación SD Bus y SPI. El SD Bus es el que mayor velocidad de transferencia de datos tiene llegando a $25 \left[\frac{MB}{s} \right]$ Debido a que utiliza las cuatro líneas de datos (ver figura 2.40), mientras que con el bus SPI puede alcanzar la señal de reloj un máximo de 25[Mhz].



Pin	SD Bus	SPI
1	DAT2	X
2	DAT3	CS
3	CMD	MOSI
4	DD	VDD
5	CLK	SLK
6	VSS	VSS
7	DAT0	MISO
8	DAT1	X

Figura 2.40. Diagrama de terminales de la memoria microSD.

2.7.1. Comandos CMD y respuesta

En la tabla 2.3 se muestra la estructura de los comandos CMD en modo SPI, los comandos tienen una dimensión de 6 bytes los cuales están estructurados de la siguiente forma:

	Byte 1		Byte 2-5	Byte 6		
Numero de bit	7	6	5-0	31-0	7-1	0
Valor de bit	0	1	Comando	Argumento del comando	CRC	1

Tabla 2.3. Estructura de los comandos CMD en modo SPI.

- Byte 1: En el byte 1 el bit 7 debe de tener un estado lógico 0 mientras que el bit 6 debe tener un estado lógico en 1, el resto de los bits son para indicar el número de comando. Así si se quisiera mandar el CMD0 el valor en hexadecimal quedaría dado por una operación lógica “or” entre el comando y el formato de byte 1 resultando: $0x00 \mid 0x40 = 0x40$.
- Byte 2-5: En estos bytes esta contenido el argumento del comando, como lo puede ser la dirección del sector a leer o escribir en la memoria.
- Byte 6: En este último byte el bit 7-1 deben de tener un CRC mientras que el bit 0 deberá de ser un ‘1’ lógico. Sin embargo una vez hecha la inicialización en modo SPI este byte puede contener cualquier valor.

En la tabla 2.4 se muestran los comandos CMD más usados en el modo SPI. El CMD55 se refiere al envío de comandos ACMD, este tipo de comandos son exclusivos para trabajar con memorias microSD uno de los comandos más usados es el ACMD41 el cual no necesita ningún parámetro y su propósito es iniciar el proceso de inicialización de la tarjeta.

Comando	Argumentos	Respuesta	Descripción
CMD0	No	R1	Resetea la tarjeta
CMD8	*3	R7	Configura el voltaje de operación (Solo para la Versión 2)
CMD13	No	R2	Consulta el estado de la tarjeta
CMD17	[31..0] Dirección de datos.	R1	Lee un bloque del tamaño indicado por el comando 16.
CMD24	[31..0] Dirección de datos	R1	Escribe un bloque del tamaño indicado por el comando 16.
CMD55	No	R1	Informa que el siguiente comando es un comando ACMD
CMD58	No	R3	Leer OCR
*3: Reservado(0) [31:12], voltaje de operación(1) [11:8], patrón(0xaa) [7:0]			
OCR: Registro de condiciones de operación			

Tabla 2.4. Comandos CMD más usados en el modo SPI.

2.7.2. Respuestas en modo SPI

Las respuesta a los comandos CMD se envían del bit más significativo al bit menos significativos, las respuestas más comunes son: R1, r1b, R3 y R7.

La respuesta depende del comando enviado la mayoría de los comando CMD tienen como respuesta un formato de respuesta R1, la estructura de este comando se muestra en la figura 2.41; como se observa su tamaño es de un byte en donde el bit más significativo tiene que ser cero mientras que los demás indican un tipo de error a excepción del bit 0.

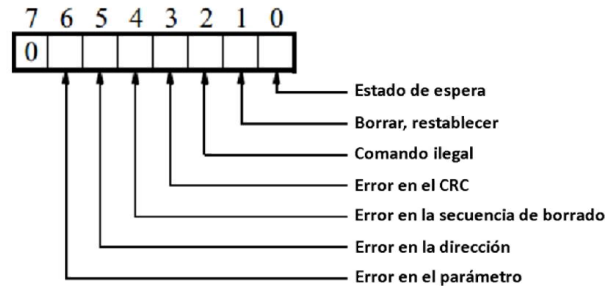


Figura 2.41. Respuesta R1.

El tamaño de una respuesta R1b también es de un byte y su función es indicar el estado de la tarjeta, es decir, si la tarjeta se encuentra ocupada o si ya está lista para recibir algún otro comando; un valor igual a cero indica que la tarjeta está ocupada y cualquier valor diferente de cero indicara que la memoria se encuentra libre para recibir comandos.

La respuesta R2 es la respuesta al comando CMD13; consta de dos bytes el primer byte es idéntico a la respuesta R1 en el segundo byte indica el estatus en el que se encuentra la memoria, en la figura 2.42 se muestra la estructura de este comando.

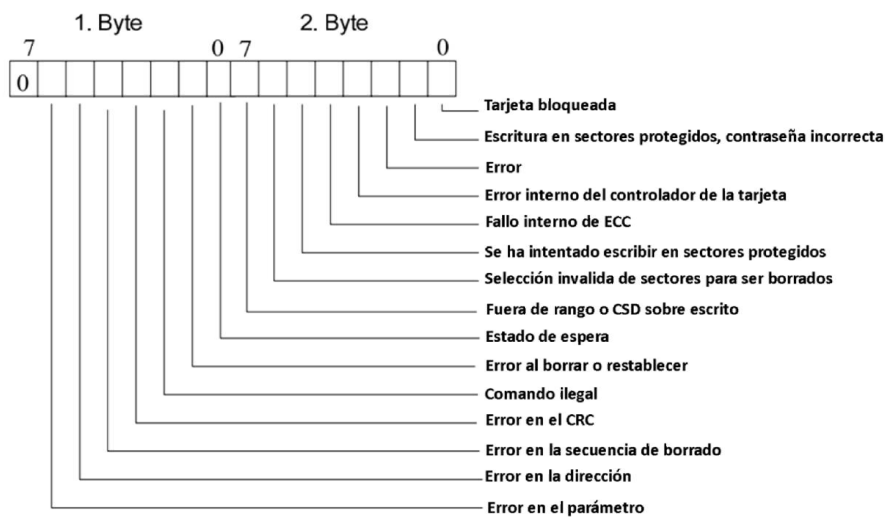


Figura 2.42. Respuesta R2.

El formato de respuesta R3 es la respuesta al comando CMD58; tiene una longitud de 40 bits (5 bytes), los bits más significativos corresponden a la respuesta R1, el resto corresponden al OCR, como se muestra en la figura 2.43.

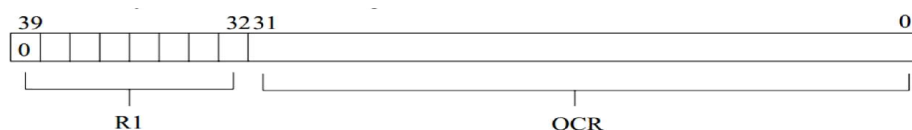


Figura 2.43. Respuesta R3.

El registro OCR indica las condiciones de operación de la microSD, los cuales se muestran en la tabla 2.5.

Posición de bits en el registro OCR	Umbral de voltaje [V]
0 - 3	Reservado
4	Reservado
5	Reservado
6	Reservado para bajos voltajes
7	Reservado
8	Reservado
9	Reservado
10	Reservado
11	Reservado
12	Reservado
13	Reservado
14	Reservado
15	2.7 - 2.8
16	2.8 - 2.9
17	2.9 - 3.0
18	3.0 - 3.1
19	3.1 - 3.2
20	3.2 - 3.3
21	3.3 - 3.4
22	3.4 - 3.5
23	3.5 - 3.6
24 - 29	Reservado
30	Este bit solo es válido cuando se establece al inicializar la tarjeta
31	Este bit se pone en bajo cuando no se ha terminado la secuencia de inicio

Tabla 2.5. Registro OCR.

El formato de respuesta R7 es enviado como respuesta del comando CMD8. La longitud de la respuesta es de 5 bytes. Su estructura muestra en la figura 2.44.

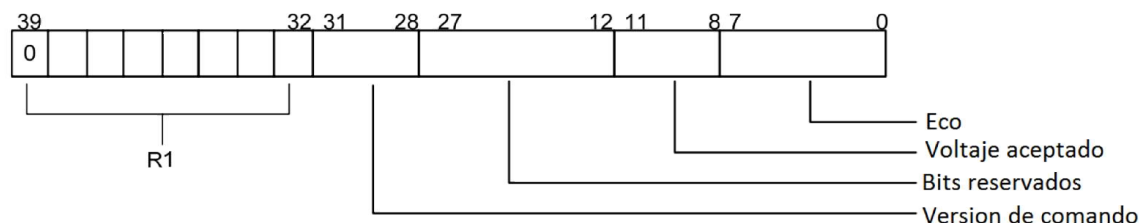


Figura 2.44. Respuesta R7.

En la tabla 2.6 se muestran los valores que puede tomar el campo de voltaje aceptado.

Voltaje aceptado (binario)	Definición de valores
0000	No definido
0001	2.7[V] - 3.6 [V]
0010	Reservado para rangos de bajo voltaje
0100	Reservado
1000	Reservado
Otros	No definido

Tabla 2.6. Voltajes aceptados.

2.8. Sistema de archivos FAT

El fin de un sistema de archivos es guardar grandes cantidades de información de forma ordenada, de forma que sea relativamente fácil el acceso a esta. Para ello un sistema de archivos debe ser capaz de: manipular, organizar y de proveer una estructura a los datos almacenados, para poder acceder a ellos desde otros equipos.

El sistema de archivos FAT (*File Allocation Table*), fue desarrollado por IBM y tiene origen entre los finales de los 70's y principios de los 80's. Fue el sistema de archivos soportado por Microsoft para MS-DOS. Fue desarrollado originalmente para discos flexibles con capacidad menor a 500[kB], sin embargo, al ser un sistema de archivos simple y de fácil implementación se siguió desarrollando para soportar una mayor cantidad de volumen.

Entre sus principales características se encuentra la utilización del formato "*Little endian*" en el directorio de archivos, es decir, el byte menos significativo se encuentra en la primera localidad de memoria, mientras que el byte más significativo se encuentra en la última localidad de esta. La unidad mínima de almacenamiento en la memoria está dada por un sector; el cual es un pequeño

paquete conformado por 512 *bytes*. Al conjunto de sectores contiguos se le denomina *cluster* y este es la unidad mínima de almacenamiento en el sistema de archivos, el número de sectores que puede tener un *cluster* siempre será una potencia de 2 (8, 12, 16, etc.) Y estará determinado por el tipo de FAT.

Actualmente existen tres sistemas de archivos FAT: FAT12, FAT16 y FAT32; siendo este último el más utilizado. La diferencia básica y razón de sus nombres, radica en el número de bits utilizados para representar a los *clusters* dentro de la FAT. Mientras que en FAT12 se tendrán 12 bits para representar a un *cluster*, en FAT16 y FAT32 se tendrán 16 y 32 bits respectivamente. Esto a su vez determinará la cantidad de información que podrá manejar el sistema, ya que indicara la cantidad de *clusters* máximo que este manejara, por lo que para FAT16 el número de bytes por *cluster* es de $2^{16} = 65536$.

El sistema de archivos FAT, como se observa en la figura 2.45 está dividido en cuatro regiones:

- Región 0. Región de inicio
- Región 1. Región FAT o directorio FAT
- Región 2. Directorio de Raíz
- Región 3. Datos

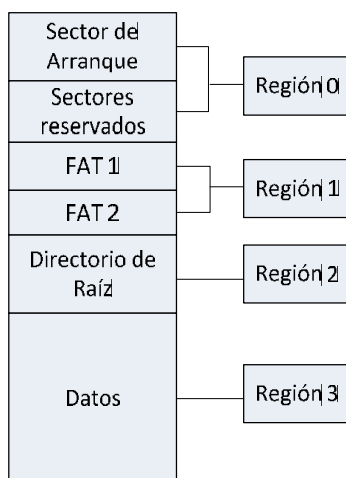


Figura 2.45. Estructura del sistema de archivos FAT.

2.8.1. Región de arranque

En la región de arranque comienza en el sector 0 se encuentra el sector de arranque (“*Boot sector*”) del sistema de archivos FAT en el cual quedan especificados todos los campos de la FAT, es decir indica el lugar en el que se encuentran la directorio FAT, el directorio de raíz, los datos; así como parámetros como: número de fats, cantidad de *bytes* por sector, número de sectores por *cluster*, la cantidad de sectores totales en el disco. En la figura 2.46, se muestran en forma ordenada los parámetros que componen el sector de arranque así como su longitud.

Sector de arranque		
512 Byte	3 Byte	Código máquina de salto (jmp short 0x3e)
	8 Byte	Cadena que identifica el fabricante del disco
	2 Byte	Bytes por sector
	1 Byte	Sectores por cluster
	2 Byte	Sectores reservados
	1 Byte	Número de fats
	2 Byte	Entradas máximas en directorio raíz
	2 Byte	Sectores totales
	1 Byte	Tipo de medio
	2 Byte	Sectores por fat
	2 Byte	Sectores por pista
	2 Byte	Número de caras
	4 Byte	Sectores ocultos
	4 Byte	Longitud total de sectores
	1 Byte	Número de unidad
	1 Byte	Banderas
	1 Byte	Firma (029h)
	4 Byte	Número de serie
	12 Byte	Etiqueta de volumen
	8 Byte	Identificador de formato
446 Byte	Código máquina del cargador de arranque	
2 Byte	Firma de unidad arrancable	

Figura 2.46. Estructura del sector cero de FAT 16.

2.8.2. Directorio FAT

El directorio FAT se puede ver como una especie de directorio de *clusters*, en este se habilitan los *clusters* que son usados en los archivos, los cuales comienzan desde el *cluster 2* hasta el 65536, esto se debe a que el número de bits con el que se representan cada cluster es de 16 bits.

En la figura 2.47 se muestra una imagen directorio FAT; donde los primeros cuatro bytes (0xf8, 0xff, 0xff y 0xff) corresponden a la etiqueta de inicio del directorio FAT, mientras los demás bytes corresponden al número de *cluster* utilizado por los archivos presentes en la memoria.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	F8	FF	FF	FF	03	00	04	00	FF	FF	FF	FF	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
90	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Figura 2.47. Directorio FAT 16.

Los sectores son ocupados de forma ordenada ascendente, si el archivo contiene menos de 65536 bits o menos de 128 sectores que es el tamaño de *cluster*, se coloca en el *cluster* correspondiente un *0xFF*, mientras que si ocupa más de un cluster; se coloca en ese lugar el número del ultimo cluster en ser ocupado, seguido de un *0xFF*, esto indica que el cluster que se encuentra a la izquierda es ultimo ocupado una vez hecho esto, el único que puede cambiar de valor es aquel que contiene el valor *0xFF*. Esto se ejemplifica en la figura 2.47, se pueden observar que se tienen almacenados dos archivos el primero empieza en el *cluster* 2 y termina en el *cluster* 4, mientras que el segundo solo ocupa el 5 *cluster*.

2.8.3. Directorio de raíz

El directorio básicamente son un conjunto de sectores que almacenan de forma ordenada las entradas de los archivos que se almacenan en la memoria. Las entradas están compuestas por 32 bytes estos contienen a los campos siguientes:

- Nombre: Longitud 8 bytes. En este campo se almacena el nombre del archivo. Al ser este el campo inicial de la entrada se cuentan con identificadores que nos permiten saber si la entrada se encuentra desocupada o no estos son: *0xe5*, *0x00*. Este campo debe ser llenado con valores mayores a *0x14* del código ASCII, la única excepción es el *0x05* que es un carácter valido en *kanji*, si el nombre no tiene una longitud de 8 bytes se deben de rellenar con espacios, también los siguientes caracteres están prohibidos: “, ¿, ?, \, >, <, |.
- Extensión: Longitud 3 bytes. Las extensiones más comunes son: *txt*, *csv*, *doc*, etc.
- Atributos: Longitud 1 byte. Atributos específicos del archivo. Los dos bits mas significativos deben de ser “0”. Los atributos que pueden tener los archivos son los siguientes:
 - *0x00*: Sin atributos
 - *0x01*: Sólo lectura
 - *0x02*: Archivo oculto
 - *0x04*: Archivo perteneciente al sistema operativo
 - *0x08*: Indica un directorio opcional al directorio de raíz
 - *0x10*: Subdirectorio (carpeta)

- 0x20: Copias de seguridad
- 0x80: Reservado
- Directorio reservado: Longitud 1 byte. Reservado para Windows NT.
- Decimas de segundos: Longitud 1 byte. Decimas de segundos de la hora en que se creó el archivo, tiene un offset de 13 y los valores que pueden tomar van de 0 -199
- Hora de creación: Longitud 2 bytes. Como se muestra en la tabla 2.7 los bits 15 al 11 contienen la hora en la que se creó en un formato de 24 horas, los bits de 10 al 5 contienen los minutos y estos van de 0 a 59 y los bits 4 a 0 contienen los segundos.

Bits	Descripción
15 - 11	Horas: 0-24
10 - 5	Minutos:0-59
4 - 0	Segundos

Tabla 2.7. Estructura del *byte* de hora de creación del archivo.

- Fecha de creación: Longitud 2 bytes. Como se muestra en la tabla2.8 los bits 15 al 9 contienen el año en que se creó el archivo comenzando desde 1980, los bits 8 al 5 contienen el mes de creación del archivo y los bits restantes contienen el día en que fue creado el archivo.

Bits	Descripción
15 - 9	Año
8 - 5	Mes: 1 -12
4 - 0	Día: 0 - 31

Tabla 2.8. Estructura del *byte* de la fecha de creación del archivo.

- Fecha del último acceso. Longitud 2 bytes. En el caso de una escritura conserva su valor.
- *Cluster* de inicio parte alta. Longitud 2 bytes. Contiene la parte alta de la palabra de 32 bits de la dirección de inicio, para FAT 16 estos dos bits son siempre 0.
- Hora de la última escritura. Longitud 2 bytes. La creación del archivo se considera una escritura.
- Fecha de la última escritura. Longitud 2 bytes. La creación del archivo se considera una escritura.
- Clúster de inicio parte baja. Longitud 2 bytes. Contienen la parte baja de la palabra de 32 bits de la dirección de inicio.
- Tamaño del archivo. Longitud 4 bytes. Tamaño de archivo en bits.

2.8.4. Región de datos

En esta región se encuentran los datos de los que están compuestos los archivos y carpetas del usuario, se encuentra inmediatamente después del directorio de raíz, que generalmente, corresponde con el cluster 2.

2.9. Microcontrolador

Un microcontrolador es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto por tres principales unidades funcionales: unidad central de procesamiento (CPU), memoria para almacenar el programa o datos y periféricos de entrada/salida.

Los microcontroladores se pueden clasificar en base con el número de bits que procesan. Se pueden encontrar desde 8 bits (siendo estos los más populares), 16 bits y 32 bits. El elegir el adecuado dependerá de la complejidad del proyecto.

Es un circuito de alta escala de integración en el cual puede incluir alguno de los siguientes dispositivos:

- Convertidor analógico digital
- Convertidor digital analógico
- Memoria RAM
- Comparadores analógicos
- Puertos de comunicación
 - UART
 - SPI
 - EUSART
 - I²C
 - CAN
- Perro guardián (*Watch dog*)
- Temporizadores
- Sensores
- Protectores de voltaje

Se pueden encontrar microcontroladores en aplicaciones como:

Automovilismo

- Sistemas de navegación
- Telemetría
- Automatización de procesos
- Telecomunicaciones

2.9.1. Arquitectura de los microcontroladores

Existen dos tipos de arquitecturas (modelos de *hardware*) diferentes para los microcontroladores, las cuales son: Von Neumann y Harvard. Las cuales se diferencian por su conexión del CPU con los bancos de memoria del microcontrolador.

Arquitectura Von Neumann

Algunos sistemas con microprocesadores basan su estructura interna según la arquitectura Von Neumann; en la cual la unidad central de proceso (CPU) está conectada a una memoria principal única, donde se guardan las instrucciones del programa y los datos. A éstos últimos se accede a través de un sistema de buses único (control, direcciones y datos). En la figura 2.48 se ilustra dicha arquitectura.

En un sistema con arquitectura Von Neumann el tamaño de la unidad de datos o instrucciones está fijado por el ancho del bus que comunica la memoria con la CPU. Así un microprocesador de 8 bits con un bus de 8 bits, tendrá que manejar datos e instrucciones de una o más unidades de 8 bits (bytes) de longitud. Si tiene que acceder a una instrucción o dato de más de un byte de longitud, tendrá que realizar más de un acceso a la memoria.

El tener un único bus hace que el microprocesador sea más lento en su respuesta, ya que no puede buscar en memoria una nueva instrucción mientras no finalicen las transferencias de datos de la instrucción anterior.

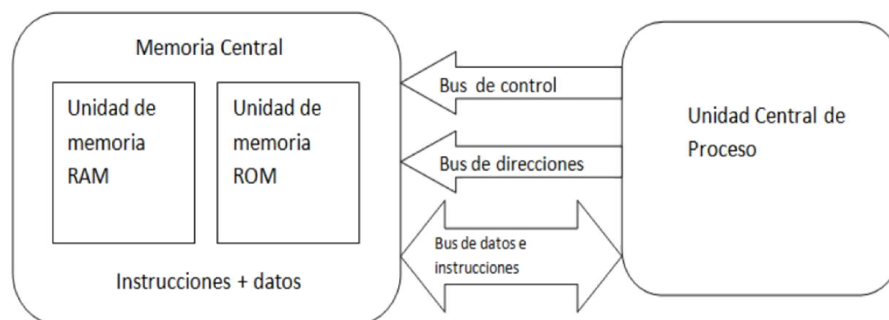


Figura 2.48. Arquitectura Von Neumann.

Arquitectura Harvard

Se caracteriza por tener la unidad central de proceso (CPU) conectada a dos memorias (una con las instrucciones y otra de datos) por medio de dos buses diferentes, como se observa en la figura 2.49.

Una de las memorias contiene solamente las instrucciones del programa (Memoria de Programa), mientras que, la otra sólo almacenara datos (Memoria de Datos). Ambos buses son totalmente independientes lo que permite que la CPU pueda acceder de forma independiente y simultánea a la memoria de datos y a la de instrucciones.

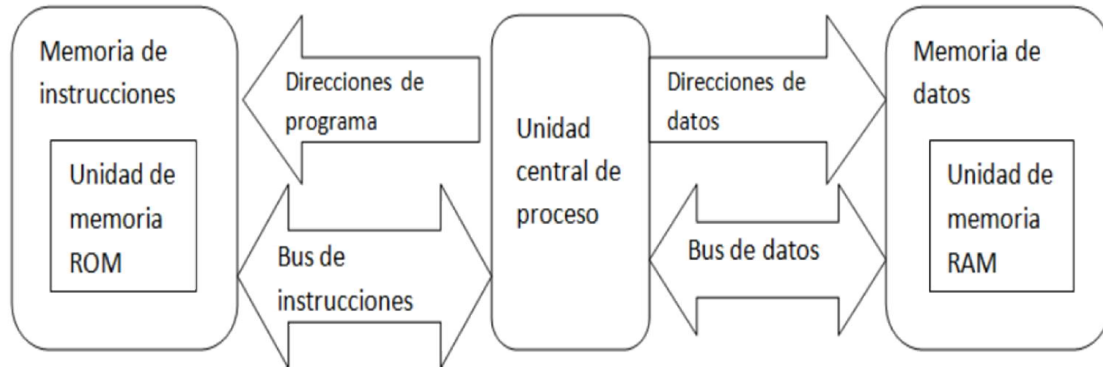


figura 2.49. Arquitectura Harvard.

Como los buses son independientes estos pueden tener distintos contenidos en la misma dirección y también distinta longitud. También la longitud de los datos y las instrucciones puede ser distinta, lo que optimiza el uso de la memoria en general.

2.9.2. Familias de microcontroladores

Actualmente existen gran variedad de microcontroladores en el mercado, siendo este resultado de los distintos productos que ofrecen las empresas como:

- **Texas instruments.** Cuenta con microcontroladores de 16 y 32 bits con familias como:
 - **MSP430.** Ideal para aplicaciones de bajo consumo de energía sin procesamiento de datos. Con velocidades de operación de hasta 25[Mhz].
 - **C2000.** Ideal para realizar procesamiento digital de señales. Con frecuencias de operación desde los 3000[Mhz] A 400[Mhz]
 - **Stellaris.** Con velocidades de operación de hasta 50[Mhz]. Son orientados a desarrollar sistemas embebidos robustos.
 - **TIVA.** Como una evolución de Stellaris es orientado al desarrollo de sistemas de monitoreo y de interfaces de usuario. Tiene una frecuencia de operación de hasta 120[Mhz].
 - **Hércules.** Esta familia está orientada al desarrollo de aplicaciones en tiempo real. Tiene una frecuencia de operación 220[Mhz].
- **Atmel.** Ofrece familias tales como:
 - AVR .Es una familia con arquitecturas de 8 y de 32 bits. Esta familia se ha popularizado debido a la introducción de la plataforma *arduino*.
 - ARM. Usa microprocesadores ARM de 32 bits, con una frecuencia máxima de operación de 120[Mhz].
- **Freescall.** Cuenta con familias ARM de 8, 16 y 32 bits.
 - Las familias de 8 y 16 bits cuenta con dispositivos como: S08, HC08, HC11, S12, HC12 y HC16. Las velocidades de operación suelen llegar a los 50[Mhz] Y con encapsulados de hasta 160 terminales.

- La familia de 32 bits consta de coldfire y Kinetis, con frecuencias de operación de hasta 50[Mhz] Y con encapsulados de hasta 144 terminales.
- **Microchip.** Cuenta con dispositivos de 8, 16 y 32 bits en familias como:
 - PIC10
 - PIC12
 - PIC16
 - PIC32
 - Dspic33
 - PIC32

2.10. Lenguajes de programación

Se puede definir como el conjunto de: palabras, símbolos y reglas sintácticas; mediante los cuales puede indicarse a un sistema de cómputo los pasos lógicos a seguir para resolver un problema, para ello, deben seguirse una serie de pasos que permiten avanzar por etapas bien definidas hacia la solución.

Etapas de programación

- Definición del problema. Es importante conocer exactamente cuál es el resultado que se desea obtener al final del proceso.
- Análisis de datos. Definir de forma precisa las especificaciones de entrada y salida, éstas deben ser descritas a detalle.
- Diseño de la solución. Esto se refiere a la obtención del algoritmo que resuelva el problema. En caso de obtenerse varios algoritmos, seleccionar uno de ellos utilizando criterios ya conocidos.
- Codificación. Crear un programa comprensible para la máquina. Para ello se debe utilizar algún lenguaje de programación.
- Pruebas. Una vez que se ha obtenido el programa ejecutable, este debe ser sometido a pruebas con el fin de determinar si resuelve o no el problema planteado.
- Depuración. Consiste en localizar los errores y corregirlos en caso de que éstos existan. Si no existen errores esta etapa puede entenderse como una etapa de refinamiento, en la que se ajustan detalles para optimizar el desempeño del programa.
- Documentación. Debido a que la utilización no podrá ser supervisada en todas las ocasiones por el programador debe crearse un manual o guía de operación que indique los pasos a seguir para utilizar el programa.
- Mantenimiento. Se refiere a las actualizaciones que deban aplicarse al programa cuando las circunstancias así lo requieran. Cualquier actualización o cambio en el programa deberá reflejarse en la documentación.

La programación pueden clasificarse en:

- Programación estructurada
- Programación orientada a objetos

Programación estructurada

La programación estructurada se basa un teorema fundamental, el cual afirma que cualquier programa, no importa el tipo de trabajo que ejecute, puede ser elaborado utilizando únicamente las tres estructuras básicas las cuales son:

- Estructura secuencial. Indica que las instrucciones de un programa se ejecutan una después de la otra, en el mismo orden en el cual aparecen en el programa. Gráficamente se representa como una caja (instrucción) después de otra como en la figura 2.50.

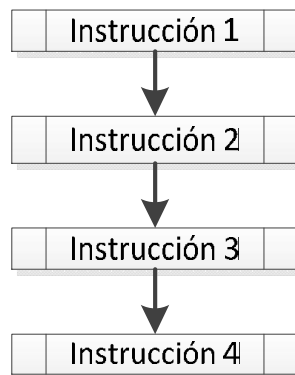


Figura 2.50. Estructura secuencial.

- Estructura selectiva. Plantea la selección entre dos alternativas con base en el resultado de la evaluación de una condición. Su representación gráfica se muestran en la figura 2.51.

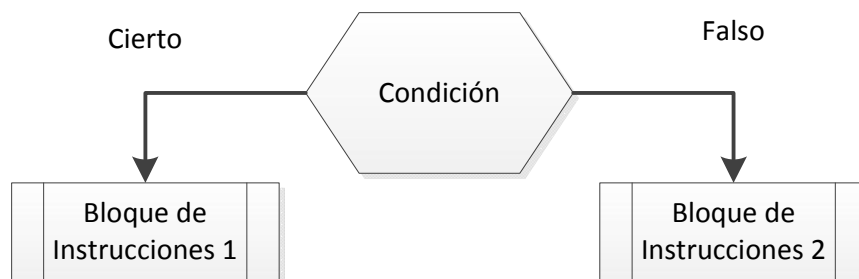


Figura 2.51. Estructura selectiva.

- Estructura repetitiva. Corresponde a la ejecución repetida de una instrucción mientras que se cumple una determinada condición. Esta estructura se muestra en la figura 2.52.

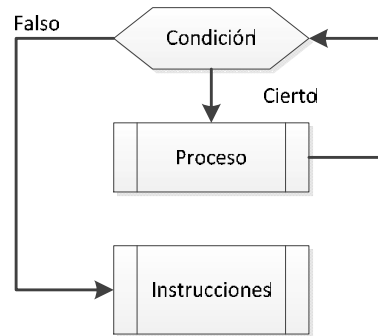


Figura 2.52. Estructura repetitiva.

La programación estructurada es la que predomina en la programación de microcontroladores, debido a que tiene las siguientes ventajas:

- Los programas son más fáciles de entender, un programa estructurado puede ser en secuencia.
- Se logra una reducción del esfuerzo en las pruebas, el seguimiento de las fallas o depuración se facilita debido a que la lógica es más clara, de tal forma que, los errores se pueden detectar y corregir más fácilmente.
- Se crean programas más sencillos y más rápidos.

Existe una gran cantidad de lenguajes de programación estructurada siendo el lenguaje C uno de los más representativos.

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

C proporciona las construcciones fundamentales que facilitan la programación en este lenguaje, entre ellas se encuentran: *if*, *else*, *switch*, *for*, *while* y *do-while*.

Por su relativa sencillez es uno de los lenguajes de programación más usados, extendiendo sus aplicaciones a la programación de microcontroladores.

Programación orientada a objetos

La programación orientada a objetos (POO) es una metodología para el diseño de software que define los programas en términos de “objetos” que interactúan entre sí.

Los conceptos fundamentales en la programación orientada a objetos son:

- Objeto. Es una instancia de una clase, que consiste en datos y todos aquellos procedimientos que pueden manipular aquellos datos.
- Clase: Es una estructura de datos que integra información y operaciones para esa información; una clase define las propiedades de un objeto pero no ocupa ningún espacio

en memoria, una clase es similar a un tipo de dato y contiene toda la información necesaria para crear nuevos objetos.

- Mensajes. Una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó.
- Atributos. Son los datos almacenados por un objeto y que lo definen de acuerdo a su clase
- Métodos. Algoritmo asociado a un objeto, cuya ejecución se desencadena tras la recepción de un "mensaje". Un método puede producir un cambio en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.

Las principales características de la programación orientada a objetos son:

- Abstracción. En el sentido más general, una abstracción es una representación concisa de una idea o de un objeto complicado. En un sentido más específico, la abstracción localiza y oculta los detalles de un modelo o diseño para generar y manipular objetos.
- Encapsulación. Es una técnica que permite localizar y ocultar los detalles de un objeto. La encapsulación previene que un objeto sea manipulado por operaciones distintas de las definidas. La encapsulación es como una caja negra que esconde los datos y solamente permite acceder a ellos de forma controlada.
- Modularidad. Denominaremos como modularidad a la subdivisión de una aplicación en partes más pequeñas (módulos), cada una de estas debe ser lo más independientes posibles de las demás. Estos pueden compilarse por separado pero deben tener conexiones con los demás módulos.
- Polimorfismo. Esta característica facilita la implementación de varias formas de un mismo método, con lo cual se puede acceder a varios métodos distintos, que tienen el mismo nombre. Existen dos tipos de polimorfismo
 - Reemplazo. Dos o más clases diferentes con el mismo nombre del método, pero haciéndolo de forma diferente.
 - Sobrecarga. Mismo nombre del método ocupado varias veces, ejecutándolo de diferente forma y diferenciándose solamente por el argumento o parámetro.
- Herencia. Es el medio para compartir en forma automática los métodos y datos entre las clases y subclases de los objetos.

Existen una gran variedad de lenguajes de programación orientado a objetos entre los que destacan C++ y Java.

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por "Sun Microsystems" que fue adquirida en 2009 por "Oracle Corporation".

Con Java, se pueden realizar una gran cantidad de aplicaciones, como son los applets, que son aplicaciones especiales, que se ejecutan dentro de un navegador al ser cargada una página HTML en un servidor WEB.

Otra de las utilidades de la programación en Java es el desarrollo de aplicaciones, que pueden ejecutarse de forma independiente de la plataforma, es decir, con java podemos crear cualquier aplicación sin preocuparnos por el sistema operativo en el cual ha sido creado, ya que este podrá ejecutarse en cualquier computadora que tenga java instalado.

3. Diseño y desarrollo

En este capítulo se abordará el diseño del brazalete, para ello se considerarán requerimientos bien definidos como lo son la adquisición, acondicionamiento, almacenamiento y envío de señales.

Este proyecto se orienta a la creación una plataforma de servicios orientados a la telemetría para pacientes de la tercera edad, esta plataforma se puede descomponer en dos grandes grupos:

- Sistema de adquisición: Estará compuesto por un dispositivo capaz de adquirir datos del pulso radial y de movilidad, procesarlos y enviarlos hacia un servidor cuando estos sean requeridos y generarlos. Dicho dispositivo tendrá que ser ligero y pequeño ya que se requiere que se localice en el brazo del paciente.
- Plataforma de servicios: Estará compuesta por varios sistemas que se encargarán de recibir la información generada por el sistema de adquisición, así como de validación y categorización de esta.

Lo anterior se muestra de forma gráfica en la figura 3.1. Sin embargo el presente trabajo se enfocará sólo al diseño del sistema de adquisición de datos del brazalete.

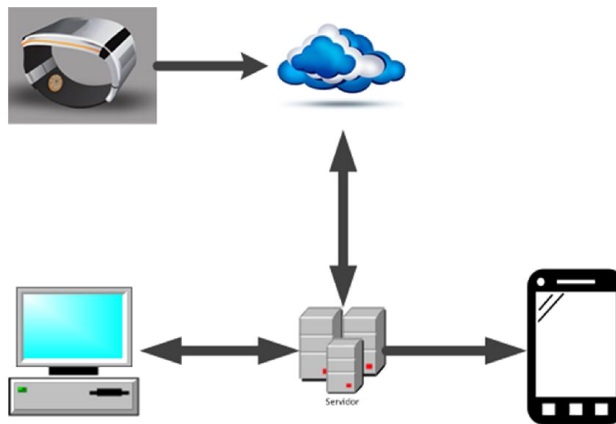


Figura 3.1. Esquema general del proyecto VACS.

3.1. Requerimientos y especificaciones de diseño

EL propósito principal del diseño y desarrollo del sistema de adquisición y envío de datos es crear un prototipo que tendrá que cumplir con las tareas siguientes: muestrear la señales de pulso radial, temperatura y aceleración; almacenar la información obtenida de las señales en una memoria; obtener la localización del sistema y enviar mensajes hacia un servidor. Ya que el sistema se planea sea portátil este debe de contar un subsistema de abastecimiento de energía.

Adicionalmente se creará una interfaz para la visualización de los datos almacenados en la memoria.

3.2. Propuesta de solución

Con base en lo anterior en la figura 3.2 se muestra de forma conceptual las tareas que deberá cumplir el sistema de adquisición. Las cuales se desglosan en los cuatro importantes bloques siguientes:

- Acondicionamiento y adquisición de señales
- Almacenamiento de datos
- Envío de datos
- Potencia

Las señales que se adquirirán son las siguientes:

- Pulso radial
- Movilidad
- Temperatura
- Ubicación geográfica
- Voltaje de la batería

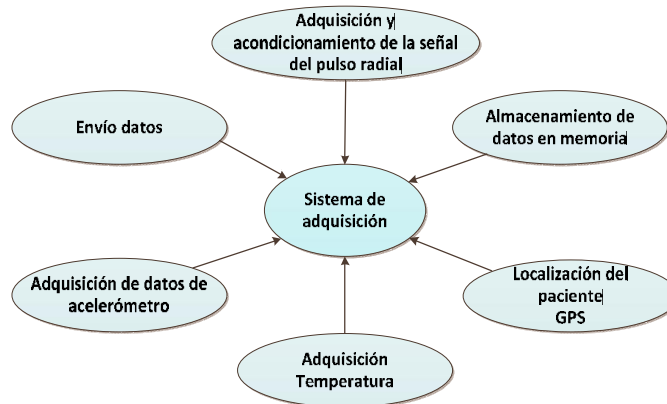


Figura 3.2. Diagrama conceptual del sistema de adquisición.

3.2.1. Selección de dispositivos

En base al diagrama conceptual figura 3.2 se procede a reconocer los dispositivos que conformarán el sistema.

Estos dispositivos deberán de ser pequeños y consumir la menor cantidad de energía posible. Para el caso de los sensores deberán de proveer una señal confiable o tener una buena resolución.

3.2.2. Sensores

Para la adquisición del pulso radial se propone implementar fototransistores, en donde los factores determinantes para la elección del sensor fue su tamaño y complejidad de instrumentación.

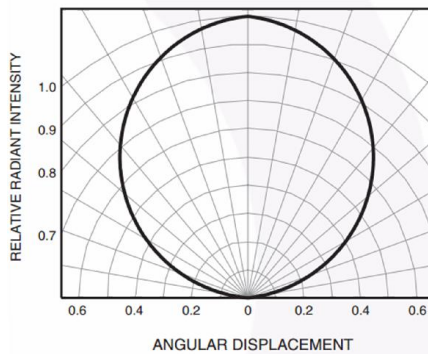
Fototransistor

Para obtener la señal de fotopleletismografía se utiliza el sensor QRE1113 (ver figura 3.3) el cual integra un fotodiodo y un fototransistor sus dimensiones son: 4.8 [mm] x 3.6[mm] x 1.70[mm].

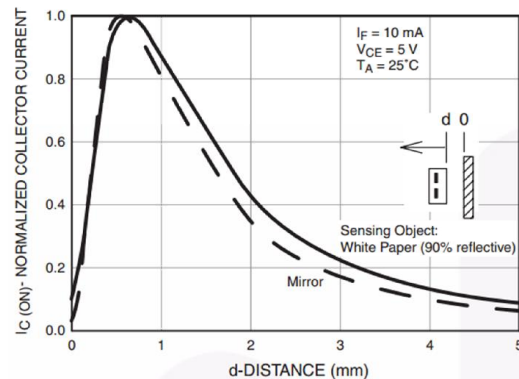


Figura 3.3. Fototransistor QRE1113.

El led emite una longitud de onda de 940[nm], sus voltajes de operación desde los 1.2[V] a 1.6[V]. Su Patrón de radiación se muestra en la figura 3.4.a. Mientras que para el fototransistor como se muestra en la figura 3.4.b se alcanza una mayor corriente en el colector alejando el objeto reflejante a una distancia de 1[mm], lo que se traduce en que el objeto y el sensor no tienen que estar en contacto directo.



a) Patrón de radiación



b) Corriente normalizada en el colector VS distancia entre el QRE1113 y el objeto reflejado

Figura 3.4. Graficas características del fototransistor QR1113.

Movilidad

Existen en el mercado varias alternativas para medir el desplazamiento de una persona, una de ellas es el podómetro. Este dispositivo funciona contando los pasos, sin embargo, el podómetro debe de ser calibrado con la longitud de los pasos del usuario lo que produce errores en la determinación de la distancia recorrida cuando la persona se mueve de manera diferente a su caminata normal.

En el mercado existe una gran cantidad de micro acelerómetros, en la tabla 3.1 se muestran algunos acelerómetros comerciales que se tomaron en cuenta.

Dispositivo	Rango máximo [G]	Tipo de comunicación	Velocidad máxima del reloj SPI	Interrupciones	Resolución [bits]	Voltaje de alimentación [V]	Consumo de corriente [μ a]	Precio [USD]
LIS3LV02DL	± 6	SPI/I ² C	8[MHz]	SI	12 - 16	2.4 - 3.6	650	12.24
ADXL345	± 16	SPI/I ² C	5[MHz]	SI	10 - 13	2.0 - 3.6	140	8.87
MMA7361L	± 6	Analógica	NO	NO	NO	2.2 - 3.6	400	NO
MMA7455L	± 8	SPI/I ² C	8[MHz]	SI	8 - 10	2.4 - 3.6	400	4.05

Tabla 3.1. Comparativa de algunos acelerómetros disponibles en el mercado.

Se decide usar el acelerómetro LIS3LV02DL debido a que de los cuatros sensores mostrados es el que mayor resolución tiene, además de que al contar con una salida de datos digital en estándar SPI y tener un velocidad de transferencia de datos alta, nos permitirá obtener una gran cantidad de datos en caso de ser necesarios. Así también cuenta con interrupciones las cuales pueden ser configuradas.

Temperatura

El registro de temperatura no es un elemento primordial del desarrollo, debido a que el propósito final del proyecto es la integración de todo el sistema en un pequeño brazalete en forma de reloj por lo que este se encontraría en alguno de los brazos; al no ser esta una de las zonas apropiadas para el registro de la temperatura (axila, boca, etc.), su información carece de utilidad médica.

Sin embargo, se puede realizar la medición de la temperatura superficial, ya que proporciona información sobre el confort del paciente.

Ante la gran gama que existe en el mercado de sensores se seleccionan algunos que se muestran tabla 3.2.

Dispositivo	Tipo de comunicación	Velocidad máxima del reloj SPI	Resolución [bits]	Voltaje de alimentación [V]	Consumo de corriente [μ a]	Precio [USD]
ADT7410	I ² C	400[khz]	16	2.7 - 3.6	5	4.47
DS1621	I ² C	4[khz]	9	2.7 - 5.5	1000	5.49
MCP9800	I ² C	400[khz]	12	2.7 - 5.5	200	3.57
LM35	Analógica	No	No	4 - 30	60	1.90

Tabla 3.2. Dispositivos para monitorear la temperatura.

Se decidió utilizar el LM35 debido a su fácil implementación y bajo consumo de corriente entre sus principales características están:

- Rango de operación de $-55[^\circ C]$ a $150[^\circ C]$
- Factor de escala lineal $10[mV/(^\circ C)]$
- Calibración en grados centígrados
- Incertidumbre de $0.5[^\circ C]$
- Impedancia de salida baja ($0.1[\Omega]$) para una corriente de salida de $1[mA]$

3.2.3. Almacenamiento de datos

El dispositivo de almacenamiento de archivos es un sistema de gran importancia debido a que este resguardara todos los datos que se generen en el sistema.

Ya que se pretende que los muestreos de las variables sean de cuando menos un día, se generara una gran cantidad de información que deberá ser almacenada. Para tener una idea clara de la cantidad de información que se espera almacenar se realiza un estimado de la memoria requerida de la siguiente forma:

- Se establecen condiciones de muestreo de las señales
 - I. Velocidad de muestreo: $200[mps^9]$
 - II. Señales muestreadas. Son 5 señales que serán muestreadas y almacenadas en memoria estas son: señal de pulso, aceleración en el eje x, aceleración en el eje y, aceleración en el eje z y temperatura.
 - III. Número de bits por muestra: 16 bits.
- Duración del muestreo. La duración del muestreo debe ser de al menos un día que son aproximadamente $86400[s]$

La cantidad de información que será almacenada se calcula utilizando la ecuación 3.1, para ello se multiplican el número de muestras por segundo, la cantidad de señales, número de bits por muestra, la duración del muestreo teniendo como resultado: $1,382,400,000[bits]$ que son aproximadamente $164[Mbytes]$ si se guardaran en forma binaria, si se requiere mostrar en un formato como el ASCII esta cantidad crece de una forma significativa.

$$Cantidad\ de\ informacion = (200)(5)(16)(86400) = 1,382,400,000[bits] \quad \text{Ecuación 3.1}$$

Otra característica que se requiere es que el dispositivo tiene que ser pequeño, de poco peso y bajo consumo de energía

Con base en lo anterior el dispositivo de almacenamiento que mejor cubre estas necesidades son las memorias microSD HC como la que se muestra en la figura 3.5.

⁹ mps: muestras por segundo.



Figura 3.5. Tarjeta microSD.

Entre sus principales características se encuentran las siguientes:

- Su voltaje de operación suele ser de 3.3[V]
- Comunicación SPI y SD BUS
- Capacidad de almacenamiento de 4[GB] a los 64[GB]
- Tasa de transferencia 25[MB/S] en modo SD BUS y aproximadamente 200[kb/S] en modo SPI

3.2.4. Ubicación geográfica

Para la ubicación geográfica se utilizará un GPS. El dispositivo a utilizar es el UP501 (ver figura 3.6) el cual tiene las siguientes características:

- Voltaje de alimentación de 3.0 [V] a 4.2 [V]
- Potencia de consumo de 75[mw]
- Antena interna
- Tamaño de 22[mm]x22[mm]x8[mm]
- Protocolo NMEA
- Precio 49.95[USD]

Ya que se debe de tener la localización del dispositivo sin importar el lugar en donde se encuentre, se utiliza un módulo GPS. Sin embargo, este será el patrón de referencia para la distancia recorrida. En un futuro se espera el GPS sólo se activara en situaciones de alarma.



Figura 3.6. GPS Fastrax UP501.

3.2.5. Envío de datos

El envío de datos se realizará a través de un modem GPRS GSM0308 este se muestra en la figura 3.7. El modem GPRS además de proporcionarnos la posibilidad de obtener datos en tiempo real, nos proporcionará una mayor independencia ya que se podrán enviar datos sin necesidad de usar algún tipo de cables o adaptador, ya que estos pueden enviarse directamente a un servidor para su almacenamiento y/o procesamiento.



Figura 3.7. Modem ENFORA GSM0308.

El modem GSM0308 tiene las características siguientes:

- Precio 54.7 [USD]
- Dimensiones 27[mm]x28[mm]x25[mm]
- Potencia de operación (3.3[V] a 4.5[V])
- Protocolos Host: Comandos AT, UDP API, CMUX, PPP
- Protocolos Internos: PPP, UDP API, TCP API, UDP y TCP
- Interfaz física: UART, USB
- Impedancia de la antena 50[Ω]
- Consumo de corriente:
 - GSM 850/900: 250[ma], corriente pico 1.6[A]
 - GSM 1800/1900: 250[ma], corriente pico 1.6[A]

Una de las razones por la que se escoge este dispositivo es porque cuenta con una API que puede manejar protocolos UDP y TCP lo que facilitará enviar datos hacia al servidor.

3.2.6. Microcontrolador

Después de revisar las especificaciones de los distintos dispositivos seleccionados que conformaran el brazaletes, se debe escoger un dispositivo de control adecuado. Para ello se listan las características deseables que debe tener este microcontrolador

- Dos canales de conversión analógico digital
- Un módulo SPI

- 2 módulos UART

Adicionalmente el microcontrolador debe contar con terminales de propósito general, interrupciones y temporizadores, para facilitar el control de procesos y dispositivos anteriormente mencionados.

Con base en lo anterior se decide utilizar el microcontrolador LM3S8962 (ver figura3.8), que además de cumplir con las características anteriormente se cuenta con experiencia previa en su manejo. Sus principales características son:

- Arquitectura de 32 bits
- Frecuencia máxima de operación de 50[MHZ]
- Memoria Ram de 64[Kbytes]
- Un ADC de 10 bits
- 4 canales de ADC
- Módulos UART 2
- Un módulo SPI
- 5 temporizadores

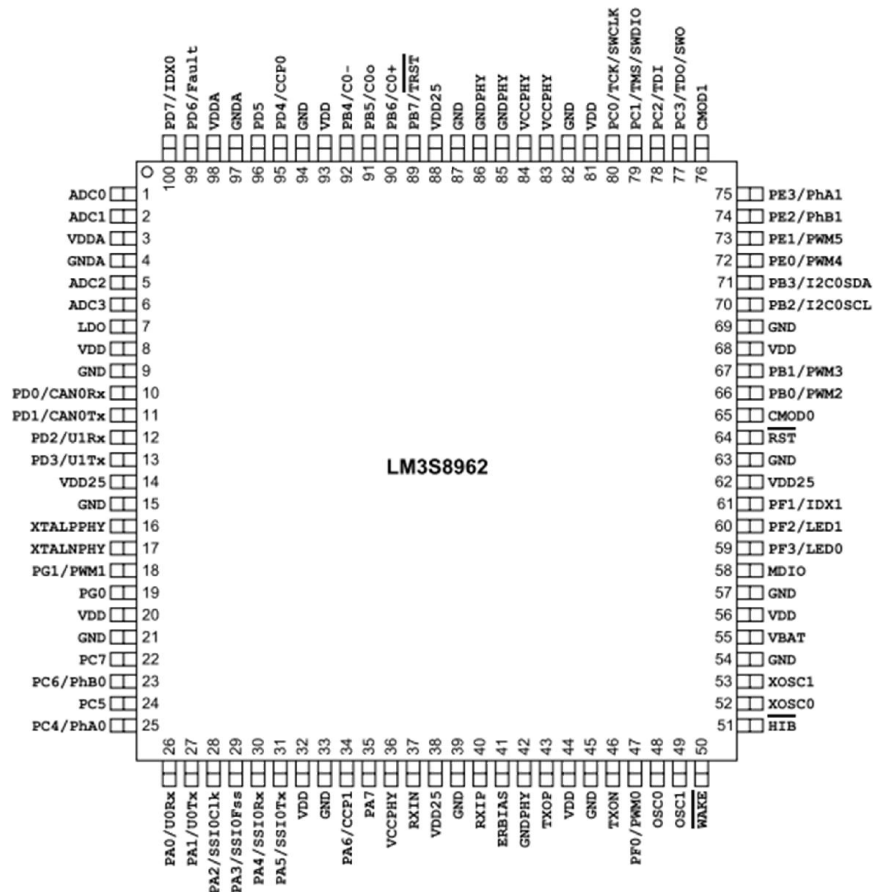


Figura 3.8. Microcontrolador Stellaris LM3S8962.

Para realizar el desarrollo de software se utilizará el entorno de desarrollo que *Texas Instruments* ofrece el cual es el *Code Composer Studio 5.5.0* así como el lenguaje de programación C.

3.2.7. Potencia

El subsistema de potencia se encargará de abastecer de energía al sistema, estará conformado por los tres bloques siguientes:

- 1) Suministro de energía. Corresponde al dispositivo que se encargara de suministrar la energía requerida para que el sistema funcione.
- 2) Reguladores. Ya que algunos dispositivos necesitan un voltaje fijo de $3.3[V]$ para operar de forma correcta.
- 3) Cargador. Se necesita tener un sistema que se encargue de recargar al dispositivo que suministre la energía.

Suministro de energía

El suministro de energía estará a cargo de una batería de polímero de litio como la que se muestra en la figura 3.9. Esta tiene las siguientes características:

- Voltaje nominal de $3.7[V]$
- Voltaje máximo $4.2[V]$
- Capacidad nominal $1000[mAh]$
- Dimensiones $6[mm] \times 34[mm] \times 50[mm]$
- Máxima corriente de descarga $1[A]$
- Impedancia de salida $50[m\Omega]$



Figura 3.9. Batería de LIPO.

Regulador

Existen en el mercado una gran variedad de reguladores de voltaje, sin embargo se ha escogido el TPS7A7300 entre sus principales características se encuentran las siguientes:

- Rango de voltaje de entrada: $1.5[V] - 6.5[V]$
- Voltaje de salida configurable en un rango de $0.9[V] - 5[V]$
- Voltaje de dropout: $240[mV]$ con una corriente de salida de $3[A]$
- Empaquetado QFN de 20 terminales, con dimensiones de $5[mm] \times 5[mm]$

Cargador

Para realizar el cargador de la batería se ha seleccionado MCP73831 el cual se muestra en la figura 3.10, por tener un tamaño reducido y por ser de fácil implementación, entre sus principales características se encuentran las siguientes:

- Voltajes de salida regulados a: $4.2[V]$, $4.35[V]$, $4.40[V]$ y $4.50[V]$
- Empaquetado SOT-23, con dimensiones de $3.1[mm] \times 3.2[mm]$
- Corriente de carga programable de: $15[mA]$ a $500[mA]$

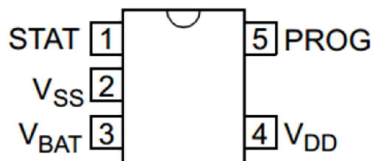


Figura 3.10. MCP73831.

3.3. Desarrollo de hardware

Una vez que se han definido los principales dispositivos con los que se realizará el prototipo. Se crea el diagrama de bloques de la figura 3.11 para establecer la forma en que estos interactuarán entre si y de esta forma empezar a seccionar las tareas que deberá realizar el microcontrolador, las cuales son nombradas como:

- **Adquisición de datos.** Consiste en la toma de muestras de los sensores de temperatura y pulso por medio del ADC, así también, la recolección de datos por medio del acelerómetro por medio del bus SPI.
- **Almacenamiento.** Guardar los datos obtenidos en la memoria microSD dentro de un sistema de archivos FAT16.
- **Localización.** Esta tarea consiste en obtener la ubicación geográfica del dispositivo por medio del GPS.
- **Comunicación.** Enviar mensajes a través del modem GSM hacia un servidor WEB.
- **Potencia.** Se encargara de abastecer la energía requerida por cada uno de los elementos del sistema.

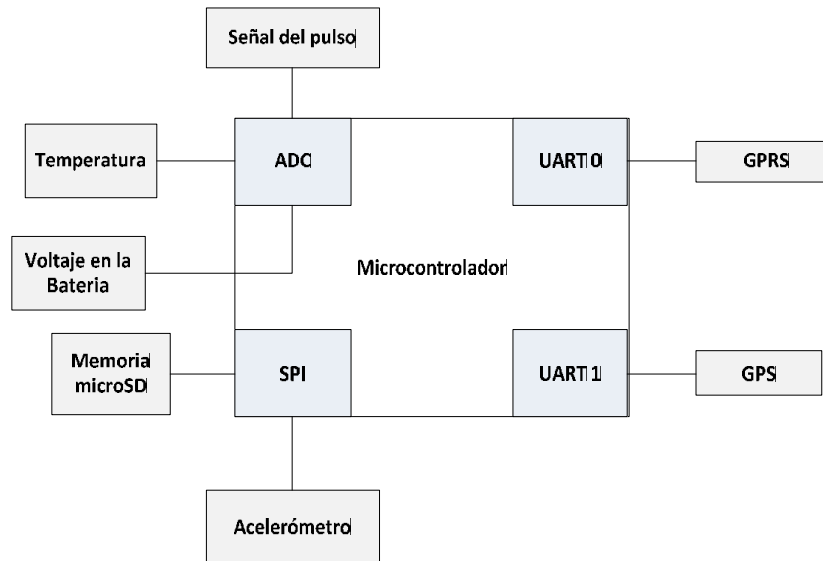


Figura 3.11. Diagrama de bloques de la interacción del microcontrolador con los otros dispositivos.

3.3.1. Acondicionamiento de señales

Como fue señalado en el apartado 2.2, las señales que se obtienen a través de los sensores generalmente no es apta para ser procesada directamente, por lo que esta tiene que ser acondicionada ya que a menudo no tiene la magnitud apropiada para ser procesada por los dispositivos electrónicos y a su vez contienen ruido. Para solucionar estos problemas la señal debe ser amplificada y filtrada.

En esta sección se explicará el funcionamiento del circuito acondicionador de señal para los sensores activos y pasivos.

Fototransistor

En la 3.12 se muestra la configuración usada en el QR1113. Se energiza el circuito con 3.3[V], mientras que R_D Limitará la corriente y el voltaje en el led , este tiene un valor de $330[\Omega]$. El transistor se encuentra en una configuración de emisor común y esta polarizado por medio de R_C ; por lo que el valor de la señal de salida será directamente proporcional al valor de R_C , para este caso $R_D = 1.5[k\Omega]$ y se ha medido una corriente $I_{CQ} = 5[mA]$. Con los datos anteriores se calcula la ganancia del circuito:

$$A_V = -g_m R_C = -(40 * 5[mA])(1.5[k\Omega]) = -300 \quad \text{Ecuación 3.2}$$

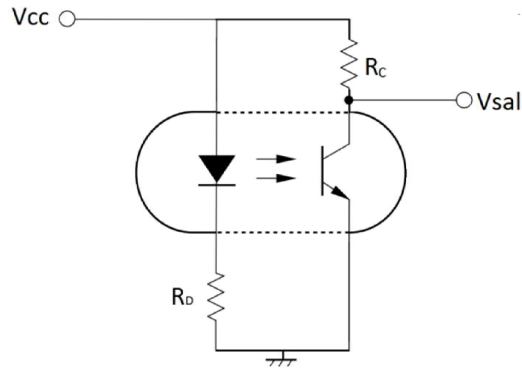


Figura 3.12. Diagrama usado para adquirir el pulso radial.

Para acondicionar la señal obtenida del fototransistor se analizaron distintas formas de realizar la instrumentación, sin embargo, se ha optado por utilizar el circuito de la figura 3.13, ya que este ya ha sido probado y se han obtenido con el buenos resultados. Este se trata de un filtro activo paso banda diferencial.

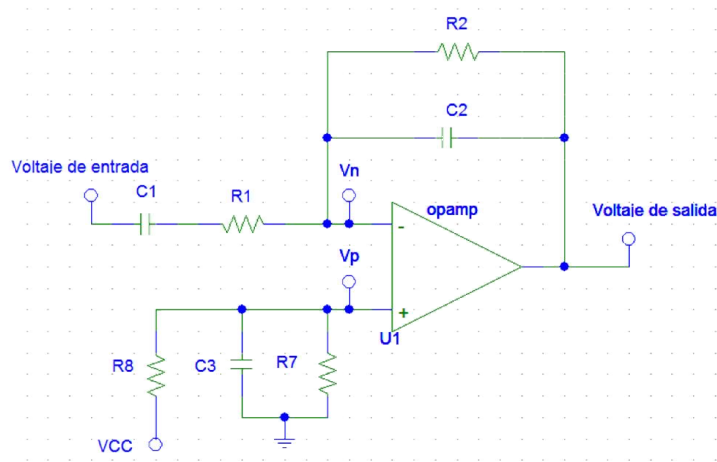


Figura 3.13. Circuito de amplificación para el fotodiodo.

Se realiza el análisis del circuito para ello se hace la consideración de que $V_n = V_p$, quedando este de la siguiente forma:

$$V_p = \frac{V_{CC}R_7}{S C_3 R_7 R_8 + R_7 + R_8} \quad \text{Ecuación 3.3}$$

Haciendo $S = j\omega$

$$V_p = \frac{V_{CC}R_7}{j\omega C_3 R_7 R_8 + R_7 + R_8} \quad \text{Ecuación 3.4}$$

Dado que V_{CC} es un voltaje de corriente directa continua se obtiene que:

$$V_p = \frac{V_{CC}R_7}{j(0)C_3R_7R_8 + R_7 + R_8} = V_p = \frac{V_{CC}R_7}{R_7 + R_8} \quad \text{Ecuación 3.5}$$

Siguiendo con el análisis:

$$Z_1 = \frac{1}{sC_1} + R_1 \quad \text{Ecuación 3.6}$$

$$Z_2 = \frac{\frac{1}{sC_2}R_2}{\frac{1}{sC_2} + R_2} \quad \text{Ecuación 3.7}$$

$$\text{Voltaje de salida} = V_p - A_1(\text{Voltaje de entrada} - V_p) \quad \text{Ecuación 3.8}$$

$$A_1 = \frac{Z_2}{Z_1} \quad \text{Ecuación 3.9}$$

$$A_1 = \frac{\frac{\frac{1}{sC_2}R_2}{\frac{1}{sC_2} + R_2}}{\frac{1}{sC_1} + R_1} \quad \text{Ecuación 3.10}$$

$$A_1 = \left(\frac{R_2}{R_1}\right) \frac{s \frac{1}{R_2 C_2}}{s^2 + s \left(\frac{1}{C_1 R_1} + \frac{1}{C_2 R_2}\right) + \frac{1}{C_1 R_1 C_2 R_2}} \quad \text{Ecuación 3.11}$$

$$V_{\text{salida}}(s) = V_p - \left[\left(\frac{R_2}{R_1}\right) \frac{s \frac{1}{R_2 C_2}}{s^2 + s \left(\frac{1}{C_1 R_1} + \frac{1}{C_2 R_2}\right) + \frac{1}{C_1 R_1 C_2 R_2}} \right] [V_{\text{Entrada}}(s) - V_p] \quad \text{Ecuación 3.12}$$

Los valores utilizados en el circuito son: $C_1 = 10[\mu F] \pm 5\%$, $C_2 = 10[nF] \pm 5\%$, $R_1 = 10[k\Omega] \pm 2\%$ y $R_2 = 10[M\Omega] \pm 2\%$ por lo que la función de transferencia queda determinada de la siguiente forma

$$H(s) = -1000 \frac{10s}{s^2 + 20s + 100} \quad \text{Ecuación 3.13}$$

Por medio de la ecuación 3.6 se obtienen las características del circuito el cual es un filtro activo paso banda el cual tiene ancho de banda y frecuencia central:

$$f_0 = \frac{\sqrt{\frac{1}{C_1 R_1 C_2 R_2}}}{2\pi} = \frac{\sqrt{100}}{2\pi} = \frac{10}{2\pi} = 1.59[\text{Hz}] \quad \text{Ecuación 3.14}$$

$$Bw = \frac{20}{10} = 2[\text{Hz}] \quad \text{Ecuación 3.15}$$

Se corroboran estos datos realizando una simulación en matlab esta se muestra en la figura 3.14. Figura 3.14

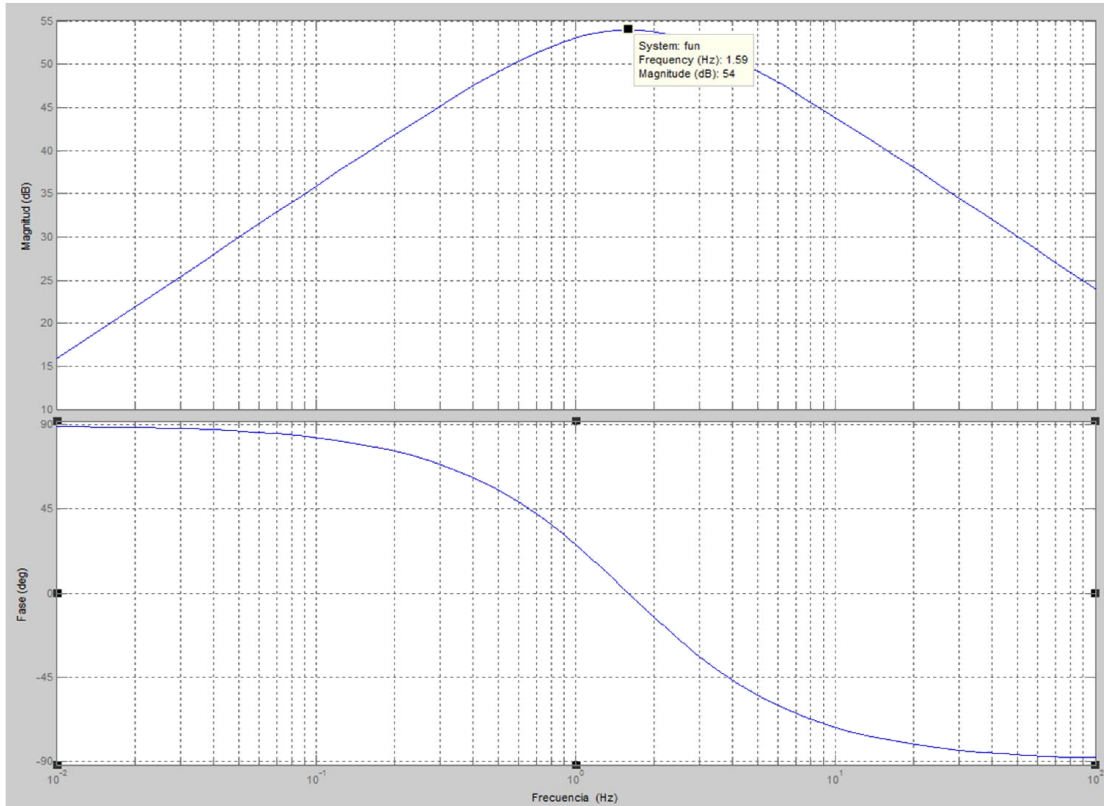


Figura 3.14. Respuesta en frecuencia del filtro paso banda.

Para calcular incertidumbre se sigue el procedimiento continuación se calcula la incertidumbre del circuito para f_0 :

$$\Delta f_0 = \frac{\partial f_0}{\partial C_1} \Delta C_1 + \frac{\partial f_0}{\partial C_2} \Delta C_2 + \frac{\partial f_0}{\partial R_1} \Delta R_1 + \frac{\partial f_0}{\partial R_2} \Delta R_2 \quad \text{Ecuación 3.16}$$

Se calculan las derivadas parciales:

$$\frac{\partial f_0}{\partial C_1} = \frac{-1}{4\pi C_1 \sqrt{C_1 C_2 R_1 R_2}} \quad \text{Ecuación 3.17}$$

$$\frac{\partial f_0}{\partial C_2} = \frac{-1}{4\pi C_2 \sqrt{C_1 C_2 R_1 R_2}} \quad \text{Ecuación 3.18}$$

$$\frac{\partial f_0}{\partial R_1} = \frac{-1}{4\pi R_1 \sqrt{C_1 C_2 R_1 R_2}} \quad \text{Ecuación 3.19}$$

$$\frac{\partial f_0}{\partial R_2} = \frac{-1}{4\pi R_2 \sqrt{C_1 C_2 R_1 R_2}} \quad \text{Ecuación 3.20}$$

Se calculan las incertidumbres

$$\begin{aligned} \Delta C_1 &= 10[\mu F] * 0.05 = 0.5[\mu F] && \text{Ecuación 3.21} \\ \Delta C_2 &= 10[nF] * 0.05 = 0.05[nF] && \text{Ecuación 3.22} \\ \Delta R_1 &= 10[k\Omega] * 0.02 = 200[\Omega] && \text{Ecuación 3.23} \\ \Delta R_2 &= 10[M\Omega] * 0.02 = 200[k\Omega] && \text{Ecuación 3.24} \end{aligned}$$

Sustituyendo:

$$\frac{\partial f_0}{\partial C_1} \Delta C_1 = \frac{-0.5[\mu F]}{4\pi(10[\mu F])\sqrt{(10[\mu F])(10[nF])(10[k\Omega])(10[M\Omega])}} = -0.0397[Hz] \quad \text{Ecuación 3.25}$$

$$\frac{\partial f_0}{\partial C_2} \Delta C_2 = \frac{-0.5[nF]}{4\pi 10[nF]\sqrt{(10[\mu F])(10[nF])(10[k\Omega])(10[M\Omega])}} = -0.0397[Hz] \quad \text{Ecuación 3.26}$$

$$\frac{\partial f_0}{\partial R_1} \Delta R_1 = \frac{-200[\Omega]}{4\pi 10[k\Omega]\sqrt{(10[\mu F])(10[nF])(10[k\Omega])(10[M\Omega])}} = -0.0159[Hz] \quad \text{Ecuación 3.27}$$

$$\frac{\partial f_0}{\partial R_2} \Delta R_2 = \frac{-200[k\Omega]}{4\pi 10[M\Omega]\sqrt{(10[\mu F])(10[nF])(10[k\Omega])(10[M\Omega])}} = -0.0159[Hz] \quad \text{Ecuación 3.28}$$

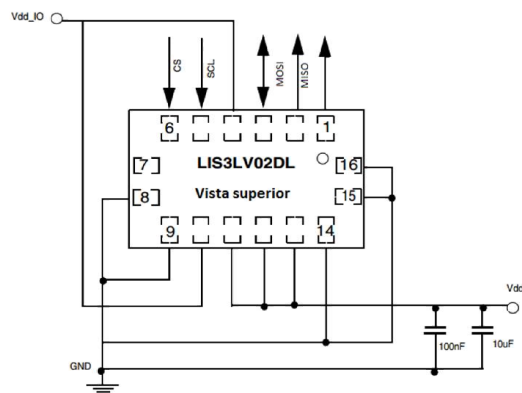
Sustituyendo en Δf_0

$$\Delta f_0 = -0.0397 - 0.0397 - 0.0159 - 0.0159 = -0.112[Hz] \quad \text{Ecuación 3.29}$$

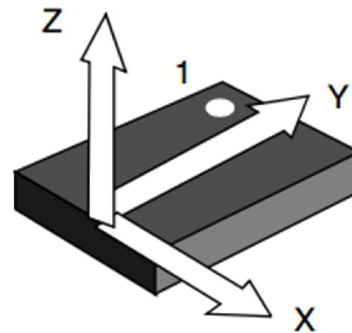
Con ello se determina que el circuito tiene una incertidumbre de $\pm 0.112[Hz]$

Acelerómetro

Para el acelerómetro se sigue el circuito sugerido en su hoja de datos el cual se muestra en la figura 3.15a. A su vez en la figura 3.15b se observan la dirección de detección del acelerómetro. Para realizar el intercambio de información se utilizará estándar SPI en su modo 3.



a) Diagrama eléctrico de conexión



b) Dirección de detección del acelerómetro

Figura 3.15 Acelerómetro LIS3LV02DL.

Temperatura

En la hoja de especificaciones del sensor de temperatura se tienen configuraciones establecidas para utilizar el sensor. La configuración que se decide usar se muestra en la figura 3.16; debido a sencillez de implementación.

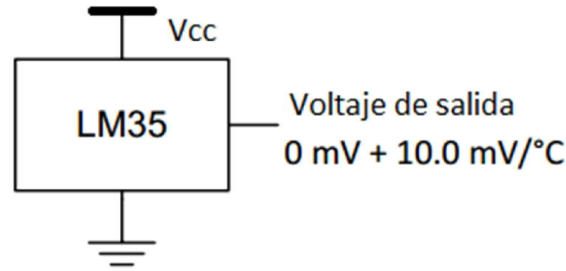


Figura 3.16 Diagrama de conexión básica (rango 2°C a 150°C).

Voltaje en la batería

Para realizar esta medición se utilizara un divisor de voltaje; debido a que casi todos los dispositivos cuentan con al menos un capacitor de acoplo para disminuir el ruido en el voltaje de alimentación.

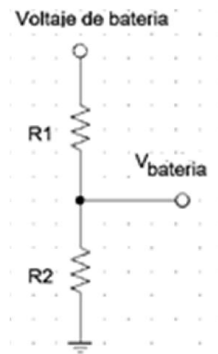


Figura 3.17. Circuito divisor de voltaje para la batería.

$$V_{bateria} = (\text{voltaje de bateria}) \left(\frac{R_2}{R_1 + R_2} \right) \quad \text{Ecuación 3.30}$$

Los valores utilizados en el circuito son: $R_1 = 100[k\Omega] \pm 2\%$ y $R_2 = 100[k\Omega] \pm 2\%$.

A continuación se realiza el análisis de incertidumbre:

$$\Delta V_{bateria} = \frac{\partial V_{bateria}}{\partial R_1} \Delta R_1 + \frac{\partial V_{bateria}}{\partial R_2} \Delta R_2 \quad \text{Ecuación 3.31}$$

Se calculan las derivadas parciales:

$$\frac{\partial V_{bateria}}{\partial R_1} = (\text{voltaje de bateria}) \frac{-R_2}{(R_1 + R_2)^2} \quad \text{Ecuación 3.32}$$

$$\frac{\partial V_{bateria}}{\partial R_2} = (\text{voltaje de bateria}) \frac{R_1}{(R_1 + R_2)^2} \quad \text{Ecuación 3.33}$$

Se calculan las incertidumbres

$$\Delta R_1 = 100[k\Omega] * 0.04 = 4[k\Omega] \quad \text{Ecuación 3.34}$$

$$\Delta R_2 = 100[k\Omega] * 0.04 = 4[k\Omega] \quad \text{Ecuación 3.35}$$

Sustituyendo:

$$\frac{\partial V_{bateria}}{\partial R_1} \Delta R_1 = \frac{-100[k\Omega](voltaje\ de\ bateria)}{(100[k\Omega] + 100[k\Omega])^2} 4[k\Omega] = \frac{-(voltaje\ de\ bateria)}{100} \quad \text{Ecuación 3.36}$$

$$\frac{\partial V_{bateria}}{\partial R_2} \Delta R_2 = \frac{100[k\Omega](voltaje\ de\ bateria)}{(100[k\Omega] + 100[k\Omega])^2} 4[k\Omega] = \frac{(voltaje\ de\ bateria)}{100} \quad \text{Ecuación 3.37}$$

Sin embargo al ser una sumatoria con signos contrarios se debe de tomar el valor de la ecuación 3.36 como positivo, para obtener el valor máximo de $\Delta V_{bateria}$ [8]:

$$\Delta V_{bateria} = \frac{(voltaje\ de\ bateria)}{100} + \frac{(voltaje\ de\ bateria)}{100} = \frac{(voltaje\ de\ bateria)}{200} \quad \text{Ecuación 3.38}$$

La incertidumbre del circuito es $\pm \frac{(voltaje\ de\ bateria)}{200}$, se expresa de esta forma porque el voltaje de la batería variara en el tiempo.

3.3.2. Almacenamiento de datos

Como se describió en el apartado 2.7 la memoria microSD tiene dos buses de comunicación se opta por trabajar con el bus SPI. En modo SPI solo soporta el modo 0 y el modo 3; debido a que se requiere optimizar el tamaño del circuito, se compartirá el bus de comunicación con el acelerómetro por lo que este trabajará en modo 3.

Su implementación es sencilla, se requiere polarizar a un voltaje de 3.3[V] y las conexiones del Bus SPI como se muestra en figura 3.18, estas deberán de tener como voltajes de operación 3.3[V] Y 0[V].

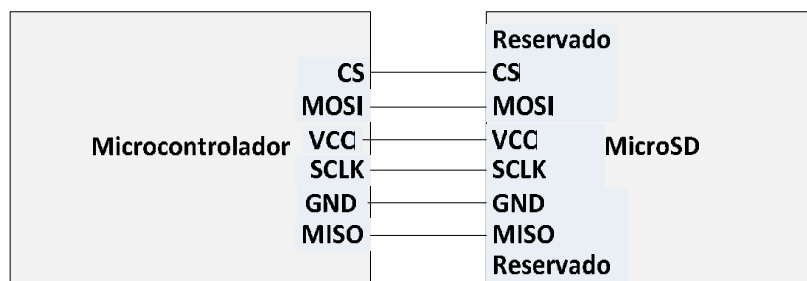


Figura 3.18. Diagrama de conexión memoria microSD.

3.3.3. GPS

El GPS UP501 requiere para operar un voltaje de alimentación que puede ir desde los 3[V] a los 4.2[V], para el intercambio de información se requiere de un módulo UART con voltajes de operación de 3.3[V] Y 0[V].

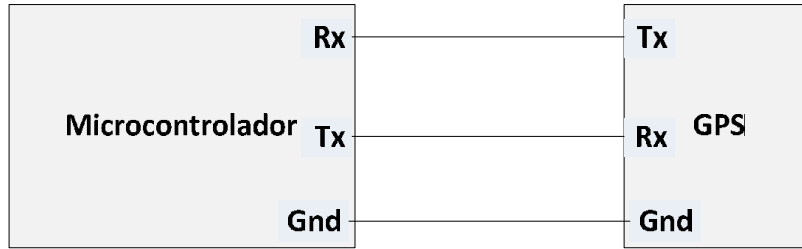


Figura 3.19. Conexión del microcontrolador con el GPS.

3.3.4. GPRS

Como se observa en la figura 3.20 el modem GPRS interactuara con el microcontrolador por medio de un módulo UART y dos señales adicionales las cuales se encargaran activar o de reiniciar el modem.

El módulo UART deberá de ser acondicionado debido a que los voltajes de lógicos de este en el GPRS son de 1.8[V] y 0[V]; mientras que para el microcontrolador son de 3.3[V] y 0[V]. Para ello se utilizara un corredor de nivel bidireccional, el cual cambiará los niveles de tensión para que sean reconocidos por ambos dispositivos de forma correcta.

A su vez las señales de “Activa” y “Reinicio” por parte del microcontrolador deberán ser terminales de propósito general que deberán configurarse en colector abierto.

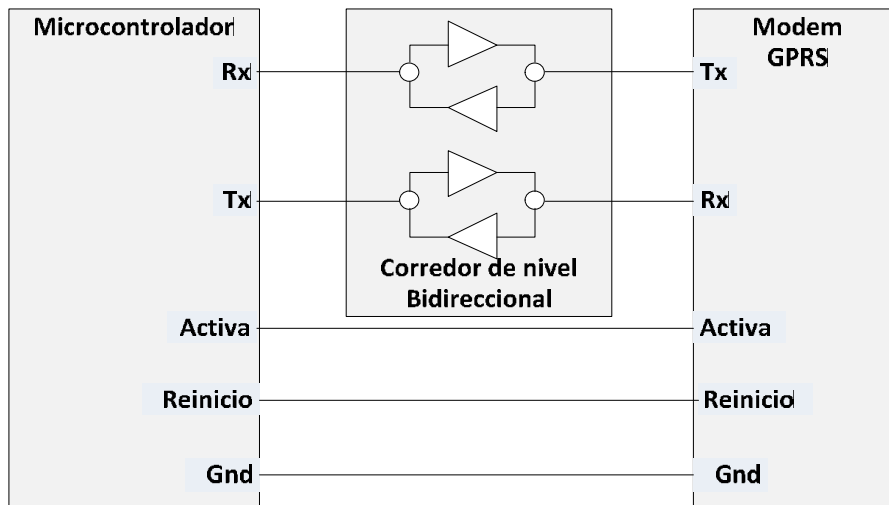


Figura 3.20. Diagramas de conexión modem GPRS.

3.3.5. Potencia

En la figura 3.21 se muestra un diagrama general del subsistema de potencia se clasifica en tres bloques los cuales son: Abastecedor de energía, reguladores de voltaje y carga. A continuación se explicarán los diagramas de conexión del cargador de baterías.

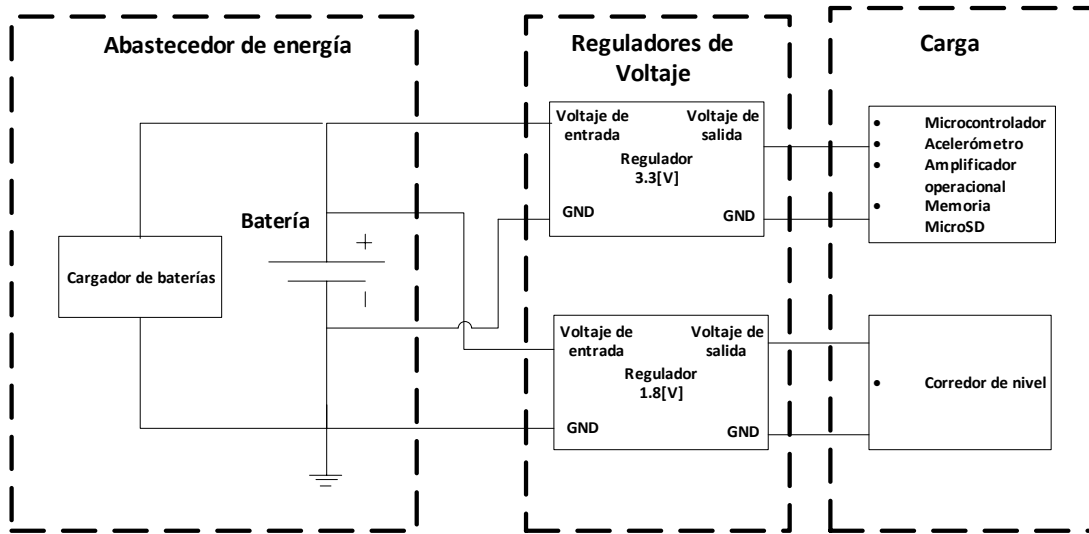


Figura 3.21. Diagrama general del subsistema de potencia.

Cargador de baterías

En la figura 3.22 se muestra la configuración usada para el cargador de baterías, como puede observarse dicha configuración posee un *led* el cual se encenderá cuando la batería este totalmente cargada. El cargador de baterías tendrá una salida de $4.2[V]$ con una corriente de salida de $500[mA]$.

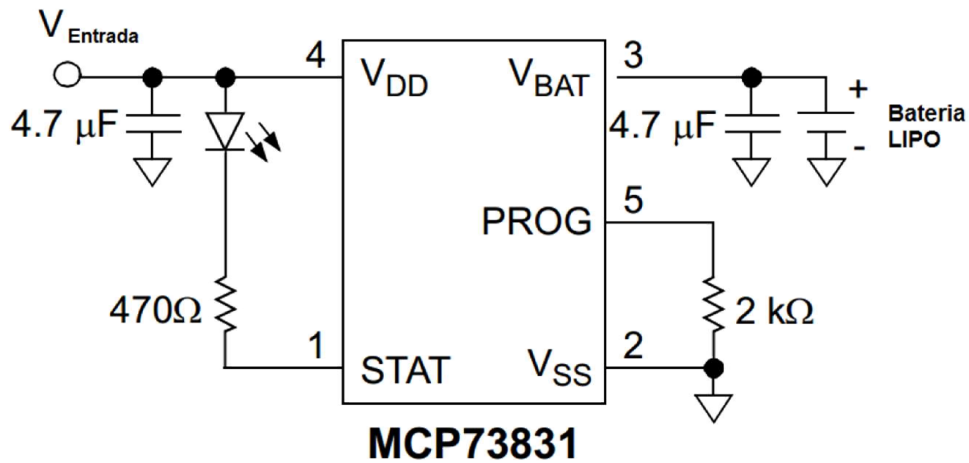


Figura 3.22. Diagrama de conexiones del MCP73831

Reguladores de voltaje

El voltaje de $1.8 [V]$ se utilizará para alimentar al corredor de nivel que permitirá el intercambio de información entre el microcontrolador y el GPRS. . En la figura 3.23 se muestra el diagrama de conexiones para que el TPS7A7300 tenga un voltaje de salida de $3.3[V]$.

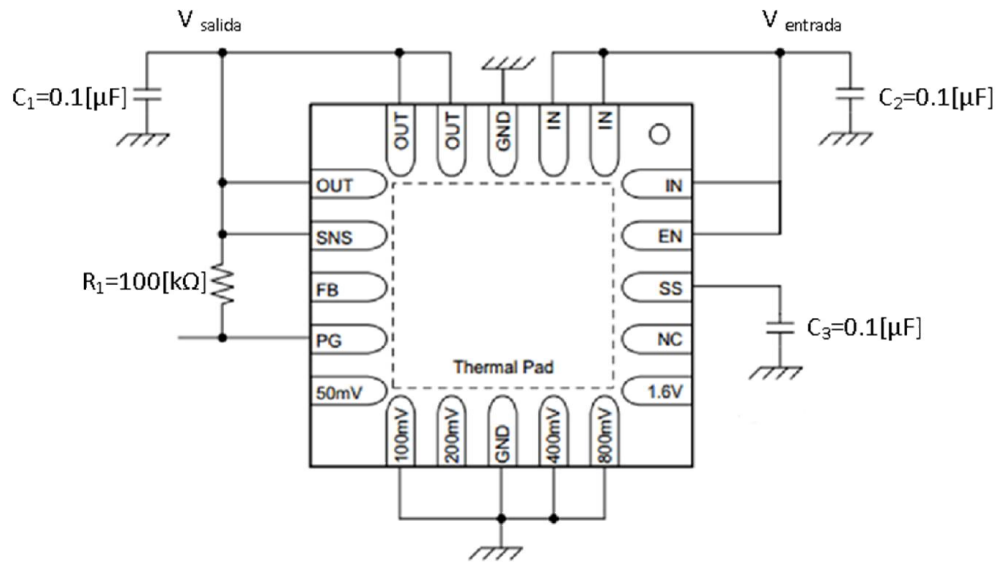


Figura 3.23. Regulador a 1.8 [V].

El regulador de voltaje de 3.3[V] se utilizara para energizar elementos del sistema como: el microcontrolador, la microSD, el amplificador operacional, acelerómetro y el fototransistor. En la figura 3.24 se muestra el diagrama de conexiones para que el TPS7A7300 tenga un voltaje de salida de 3.3[V].

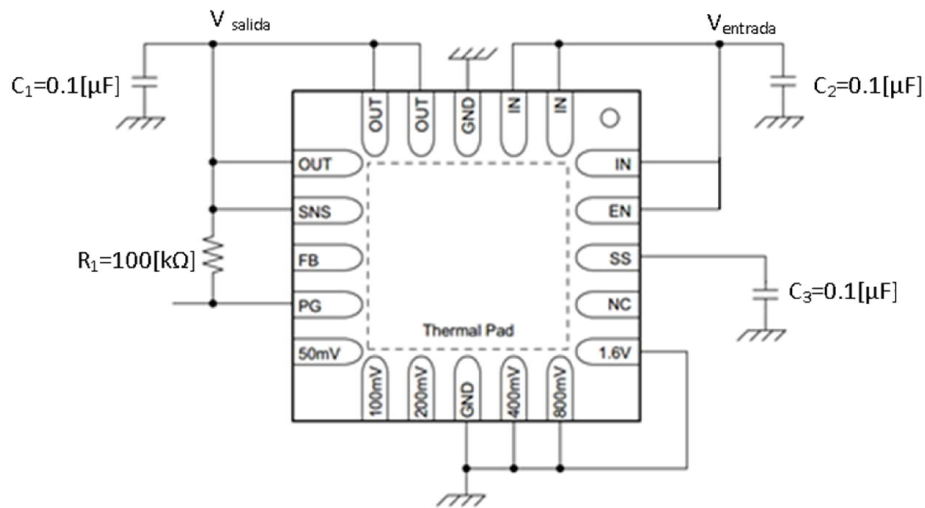


Figura 3.24. Regulador a 3.3 [V].

3.3.6. Integración de hardware

Después de haber analizado el funcionamiento de los bloques que conformarán el sistema se procedió a realizar pruebas, para ello se montaron éstos en tarjetas *protoboard* con el fin de verificar el correcto funcionamiento de todos los bloques en conjunto y, con ello localizar

problemas de integración y en el peor de los casos agregar componentes adicionales, en la figura 3.25 se muestra una de las pruebas que se realizaron.

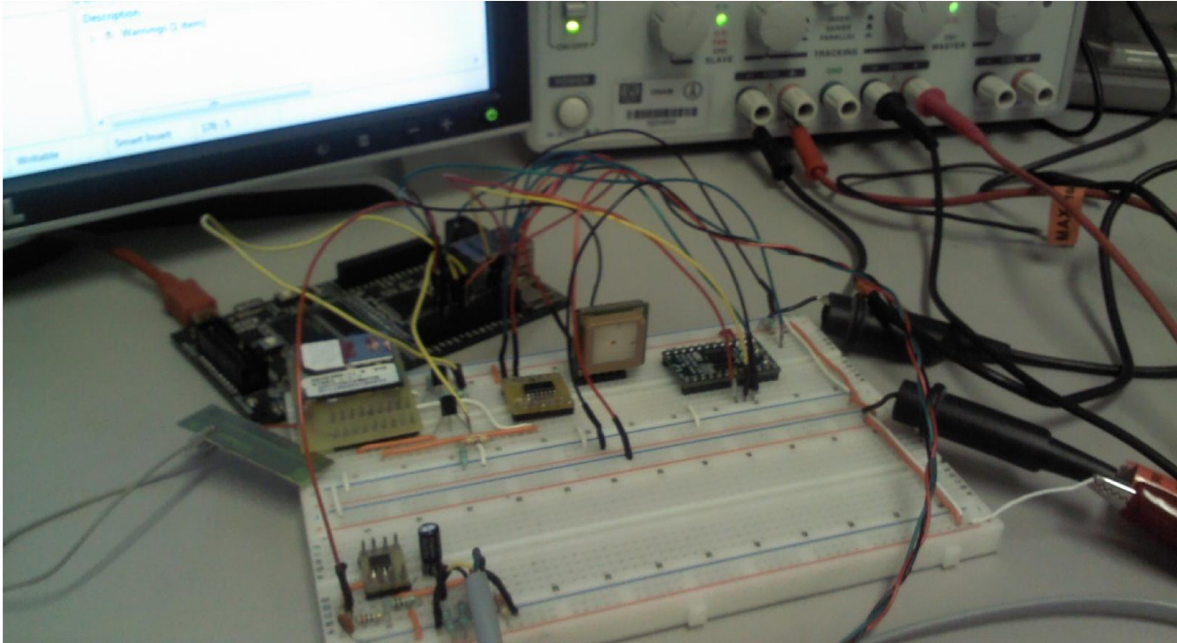


Figura 3.25. Prueba de integración del sistema

PCB del sistema

Una vez que se comprobó el correcto funcionamiento del sistema y que no se requería ningún componente adicional, se procedió a la integración del sistema en un circuito impreso (PCB: *Printed Circuit Board*), para obtener el primer prototipo final.

Algunas consideraciones que se tomaron en cuenta para la fabricación del circuito son:

- La ubicación de los componentes se realizó con la intención de tener un prototipo lo más compacto posible.
- Se optó por crear un circuito doble cara en el cual deberán de estar integrados todos los componentes a excepción de los sensores; los cuales se conectarán por medio de dos conectores Jack de 3.5[mm].
- Se colocó un interruptor el cual se encargará de encender el sistema.
- Se colocó un bloque de header para conexiones adicionales, este cuenta con canal ADC, I²C, PWM, pines de propósito general y los pines necesarios para programar el microcontrolador.

En las figura 3.26 se muestra la capa superior del circuito impreso y en la figura 3.27Figura 3.27 la capa inferior del mismo. Sus dimensiones son de 61[mm]x45[mm].

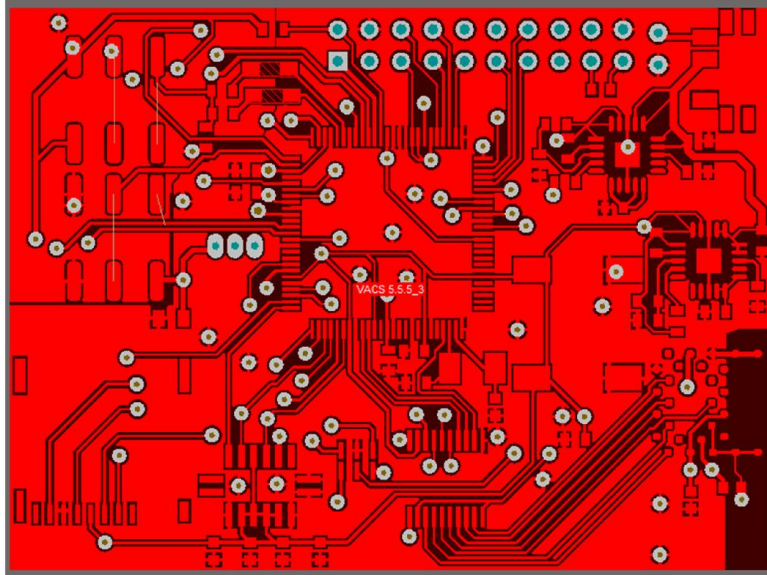


Figura 3.26. Capa superior del circuito impreso

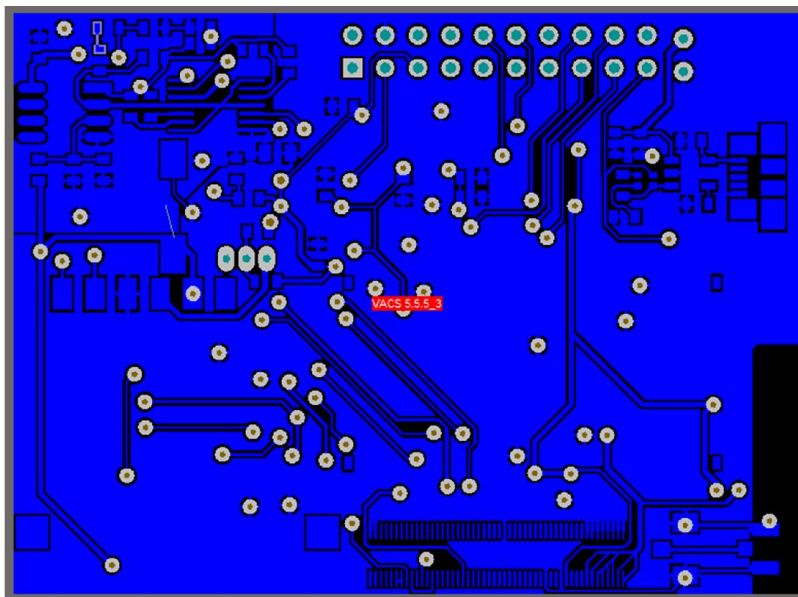


Figura 3.27. Cara inferior del circuito impreso

3.4. Desarrollo de Software

En las secciones anteriores se describieron de manera general los requerimientos físicos que tiene el hardware del sistema de adquisición de datos, para que estos puedan operar se creó un programa que se carga al microcontrolador para que este empiece a gestionar dichas tareas.

El software para el microcontrolador fue creado en el entorno de desarrollo “Code Composer Studio” en su versión 5.5.0 de Texas Instruments; este nos permite crear, transferir y depurar programas en lenguajes de programación como: ensamblador, C y C++.

Para cumplir los objetivos que se plantearon en el apartado 3.3 se secciona el software en los bloques siguientes:

- Adquisición de datos
- GPS
- Almacenamiento
- Envío de datos

En cada uno de estos bloques se encuentra una serie de rutinas las cuales nos permiten tener control sobre los dispositivos externos. Estas se describirán a continuación en las siguientes secciones.

3.4.1. Adquisición de datos

En esta sección se describe el software generado para la adquisición de los datos para su explicación este se ha dividido en los siguientes dos bloques.

Configuración

Consiste en definir las condiciones iniciales de todos los dispositivos que conforman el subsistema de sensores esto se explica a continuación.

Acelerómetro

El acelerómetro se comunicará con el microcontrolador a través del estándar de comunicación SPI. La configuración del acelerómetro se logra escribiendo en sus registros de control las características deseadas de operación, en el caso del acelerómetro LIS3102DL basta con configurar los registros de control 1 y 2; que se encuentran en la dirección de memoria 0x20 y 0x21 respectivamente.

Al enviar el dato 0x2047 se configura el registro de control 1 activando las características siguientes:

- Dispositivo activo
- Modo normal
- Decimación 512
- Activación de los tres ejes

El registro de control 2 se configura enviando el dato 0x21a0 con ello en el acelerómetro se activan las siguientes características:

- Escala $\pm 6[G]$
- Actualización continua
- Formato “*big endian*”
- Modo SPI de cuatro líneas

- Resolución 16 bits alineados a la izquierda

Pulso, temperatura y voltaje

Las señales de pulso, temperatura y voltaje serán adquiridas por el ADC del microcontrolador. El cual cuenta con un ADC que es multiplexado por los cuatro canales analógicos, cuenta con cuatro secuencias de muestreo que pueden ser programadas para obtener orden los valores de las muestras que son guardadas en una memoria FIFO (*First Input First Output*). El ADC cuenta con un voltaje interno de referencia de 3.3[V]. El muestreo puede ser activado por software o por interrupciones provocadas por los periféricos (*pwms, Timers, gpios*).

Para configurar el ADC del microcontrolador se deben activar y configurar las secuencias de muestreo; así también el evento que activará el convertidor.

Debido a que las señales están conectadas al microcontrolador de acuerdo a la tabla 3.3, se decide utilizar la secuencia de muestreo 0, la cual nos permite obtener 8 muestras en cada muestreo de estas sólo se activarán 3 muestras.

Canal del ADC	Señal conectada
Canal 0	Pulso radial
Canal 1	Temperatura
Canal 2	Voltaje
Canal 3	Sin definir

Tabla 3.3. Conexiones entre los canales del ADC y las señales

Se toman las muestras en el orden siguiente: voltaje de batería, pulso radial y temperatura. Se decide este orden ya que un valor bajo en el voltaje de batería podría ocasionar que los dispositivos no funcionen de forma adecuada.

Para obtener una imagen de buena calidad del pulso la adquisición de datos se realiza con una frecuencia de muestreo de 200[mps]0, para ello se configura el ADC del microcontrolador para que este sea controlado a través del *Timer 0* el cual se programa para que genere una interrupción de forma periódica esta activará el ADC y comenzará a tomar muestras de las secuencias de muestreo activadas.

Adquisición

Para la adquisición de datos se sigue el diagrama de flujo de la figura 3.28 la descripción de los bloques se explica a continuación:

- Limpiar la interrupción
- Obtención de datos. Esto se logra mediante la lectura de la FIFO del ADC, en el caso del acelerómetro esto se logra mediante la lectura de los registros “OUTX_L”, “OUTX_H”, “OUTY_L”, “OUTY_H”, “OUTZ_L” y “OUTZ_H”.

- Verificar voltaje. Se verifican los reguladores de voltaje ya que requieren de una tensión eléctrica mínima para mantener el voltaje a la salida, este no debe ser menor a 3.6[V].
- Almacenar datos. Si el voltaje se encuentra dentro del rango establecido se guardan los datos en la microSD de lo contrario son desechados. Estos datos serán guardados en formato ASCII.

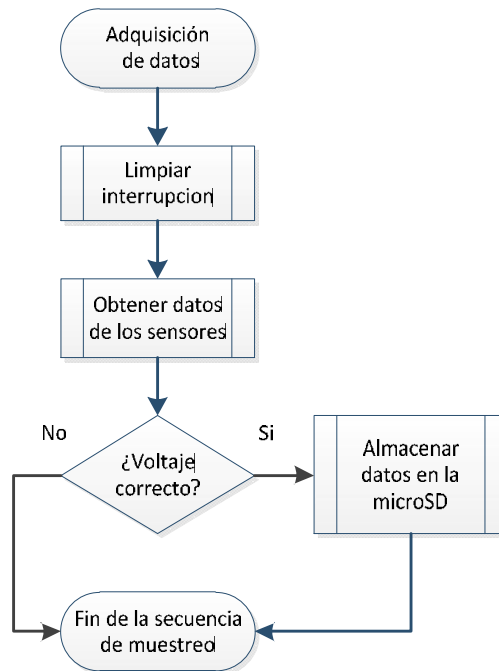


Figura 3.28. Adquisición de datos

3.4.2. GPS

Para el manejo del GPS se crearon las rutinas de inicialización y localización, estas se describen a continuación.

Inicialización

Para comenzar a trabajar con el GPS, se debe configurar el puerto UART del microcontrolador con las características de facto del GPS estas son:

- Velocidad 9600[bauds]
- 8 bits de datos
- Sin bit de paridad
- Un bit de paro

Sin embargo, como se puede apreciar en la figura 3.29, el GPS envía una gran cantidad de información lo cual no es deseable ya que el microcontrolador tiene que realizar más tareas con otros dispositivos.

```

$GPGSA,A,1,,,,,,,,,,,,,*1E
$GPGSV,1,1,01,02,,,,,19*72
$GPRMC,000002.036,V,,,,,0.00,0.00,060180,,,N*45
$GPGGA,000003.036,,,,,0,0,,,M,,M,,*4E
    
```

Figura 3.29. Paquete de datos antes de obtener la ubicación

Para aumentar la velocidad de transmisión a 115200[bauds] por lo que se envía en comando **\$PMTK251,115200*1F**. Una vez hecho lo anterior se debe cambiar la velocidad de transmisión en el puerto UART a 115200[bauds]

Una vez que se ha cambiado la velocidad y como puede apreciarse en la figura 3.30 mucha de la información que envía el GPS es redundante se decide cambiar el número de tramas de datos que envía el GPS, para ello se utiliza el comando **\$PMTK314,0,1,0,0,0,0,0,0,0,0,0,0,0,0*29**.

```

$GPGSA,A,3,02,24,10,28,12,05,17,,,,,1.30,0.96,0.87*0F
$GPGSV,2,1,08,02,61,245,42,10,39,137,21,28,35,114,15,17,33,039,25*78
$GPGSV,2,2,08,24,31,294,35,05,13,187,34,12,11,322,35,20,03,044,*75
$GPRMC,184900.000,A,1919.5306,N,09910.8431,W,0.20,261.79,180913,,,A*7C
$GPGGA,184900.000,1919.5306,N,09910.8431,W,1,7,0.96,2294.0,M,-9.0,M,,*64
    
```

Figura 3.30. Paquete de datos después de establecer la ubicación

Lo anterior se resume en el diagrama de flujo mostrado en la figura 3.31.

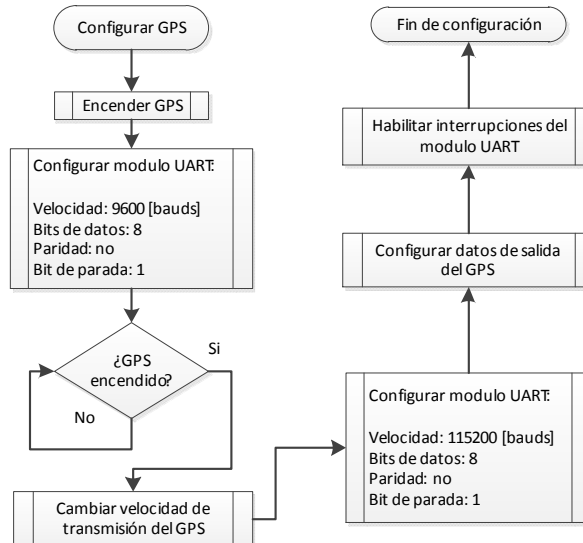


Figura 3.31. Diagrama de flujo para configurar el GPS

Localización

Para obtener la posición del sistema se hace uso de una máquina de estados la cual se muestra en la figura 3.32, con ello se espera que el microcontrolador obtenga las coordenadas de ubicación de forma óptima.

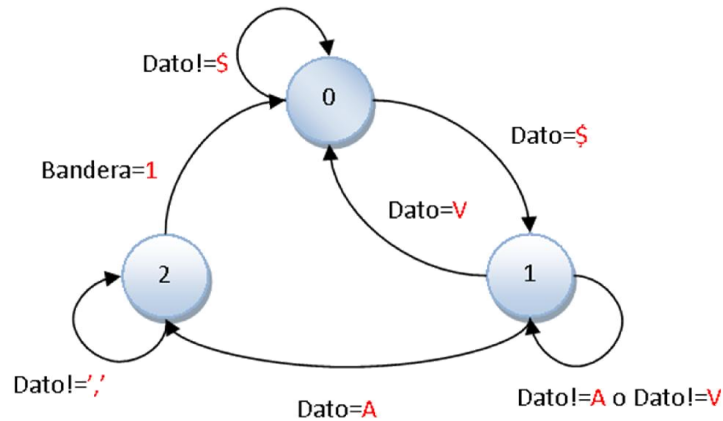


Figura 3.32. Máquina de estados para obtener la posición del GPS

Descripción de estados

- Estado 0: Es el estado inicial de la máquina de estados, el cual se mantiene hasta recibir el carácter \$ este carácter señala el inicio de la transmisión de un comando NMEA.
- Estado 1: En este estado se valida la información del GPS, es decir se verifica si el GPS obtuvo su posicionamiento.
- Estado 2: Una vez que se comprobó que el GPS ya encontró su posición se procede a tomar las coordenadas de latitud y longitud del GPS.

3.4.3. Almacenamiento de datos

Para el manejo de la memoria microSD se debe de escoger la forma en que se presentaran los datos; debido a que la información puede ser manejada por personas que no tengan conocimientos en el manejo de este tipo de memoria a bajo nivel, se propone que los datos sean guardados en forma de un archivo CSV¹⁰ o txt, lo que permitiría un fácil manejo de los datos, ya que puede abrirse desde cualquier procesador de hojas de cálculo o software especializado como Matlab.

El hecho de crear archivos implica que debemos de tener un sistema de administración de archivos, entre los cuales destacan por ser los más usuales: FAT16, FAT32 y NTFS. Se decide utilizar como sistema de archivos el FAT16 o simplemente FAT, ya que es de fácil implementación y puede

¹⁰ CSV: Valores separados por comas

soportar memorias de hasta 4[gbytes], tamaño que se considera adecuado para almacenar la cantidad de datos requerida. Este sistema de archivos puede almacenar archivos con un máximo de 2[gbytes], característica deseable para este caso. También por el hecho de ser la base de otros sistemas de archivos como FAT32 en caso de ser necesario será de gran ayuda para mudarse ha dicho sistema de archivos.

Por lo anterior se crea una librería que está conformada básicamente por seis funciones, tres que denominaremos básicas, estas son: inicialización, lectura y escritura; con estas tres funciones base se construyen las funciones: crear archivo escribir datos y actualizar archivo. Estas se describen a continuación.

Inicialización

Para empezar a trabajar con la memoria microSD, necesitamos cambiar el modo de comunicación ya que trabaja al ser polarizada; el modo de comunicación que utiliza es el de BUS SD; para cambiar el modo de comunicación de la memoria se tiene que seguir el algoritmo de la figura 3.33 el cual se explica a continuación:

- 1) Se configura periférico SPI del microcontrolador con las características siguientes:
 - a. Se configura como maestro.
 - b. Configurar la frecuencia de reloj (SCLK) a 400[khz]
 - c. Seleccionar el modo 3 de operación
 - d. Se pone el *chip select* en estado alto
- 2) Con el *chip select* en alto se envían por lo menos 74 pulsos de reloj
- 3) Activar el *chip select* (poner el *chip select* en estado bajo) y enviar el comando CMD0 el argumento deber ser 0 y su CRC 0x95 se debe enviar hasta que se reciba como respuesta 0x01.
- 4) Se envía el CMD8, el argumento debe ser 0x1aa y el CRC deber ser 0x87, esto se hace para configurar el voltaje de operación, con el fin de obtener la versión de tarjeta microSD con la que se está trabajando. Lo cual se logra de leyendo el primer byte lo cual puede indicar dos casos.
 - a. Si la respuesta del CMD8 es 0x04 indica que el comando no es válido, por lo que significa que estamos trabajando con una microSD versión 1.
 - b. Si la tarjeta arroja cualquier otro valor, la microSD es una versión 2 y tenemos que leer el resto de los bytes.
- 5) Se envía el comando CMD55 e inmediatamente el comando ACMD41 para empezar con la inicialización.
- 6) Este paso sólo debe efectuar si en el paso 4 obtenemos como resultado que la microSD es versión 3 se debe enviar el comando CMD58 con el fin de obtener las condiciones de operación.
- 7) Se finaliza con esto la inicialización de la tarjeta microSD y una vez inicializada se puede aumentar la velocidad de transmisión de datos del bus SPI.

La finalidad de encontrar la versión de la tarjeta con la cual se está trabajando es importante no sólo para lograr la correcta inicialización de la tarjeta, también, es importante debido a que cuando se escribe o lee se debe hacer un corrimiento de 9 bits en las microSD versión 1.

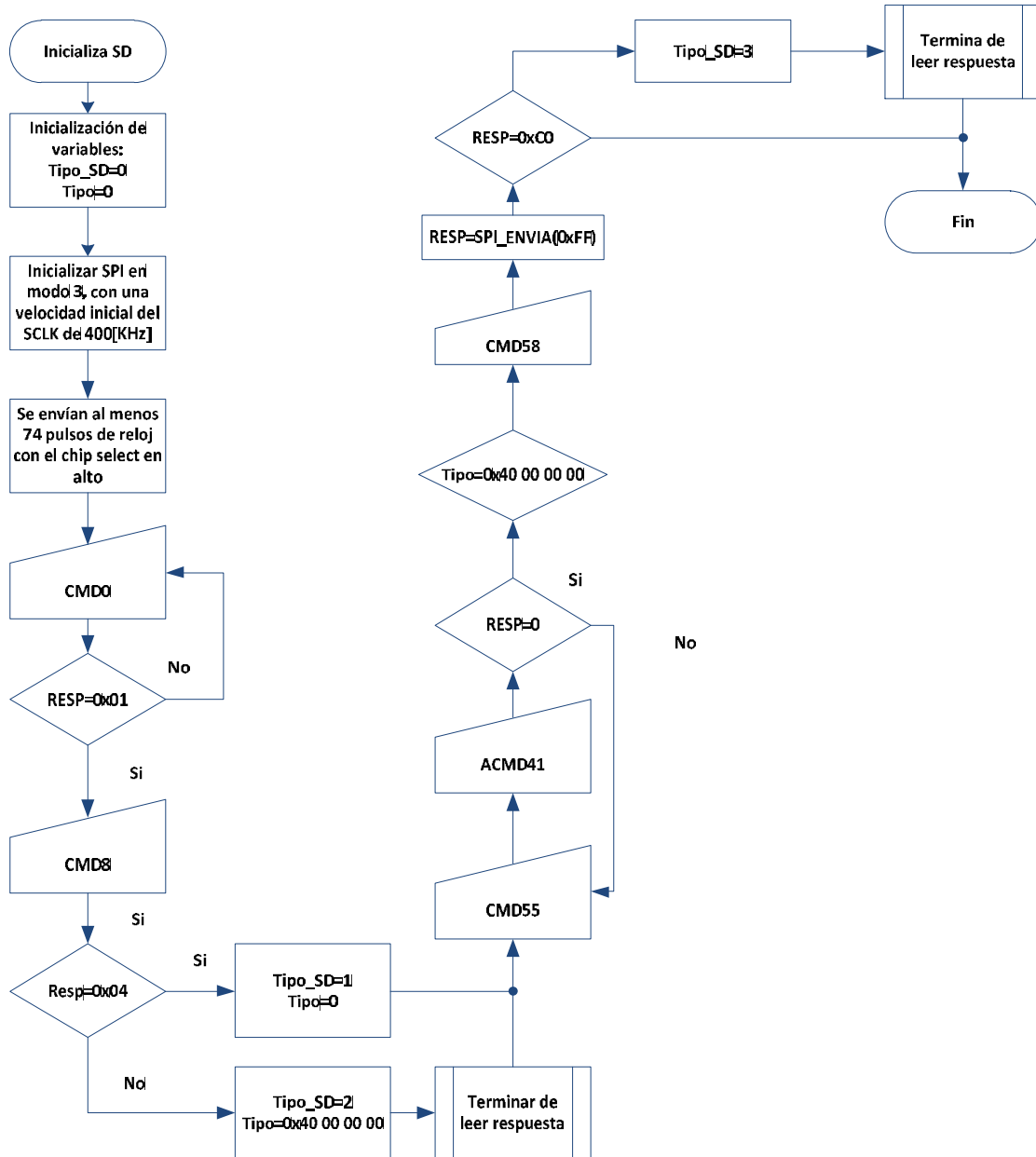


Figura 3.33. Diagrama de flujo para inicializar la tarjeta microSD.

Escritura

Como se mencionó anteriormente si se trata de la versión 1 se tiene que hacer un corrimiento de 9 bits, la dirección está dada por el número de sector en el que se desea escribir, el tamaño mínimo de bytes a escribir es 512.

Una vez aclarado lo anterior se describe el algoritmo para realizar la escritura en la microSD se muestra en la figura 3.34 y se explica a continuación:

- 1) Se coloca el *chip select* en estado bajo.
- 2) Se envía el comando CMD24 como argumento se debe enviar el número de sector en el que se desea escribir, el CRC en este comando ya no importa.
- 3) Si la respuesta es 0x01, se debe desactivar el *chip select* y se debe enviar nuevamente una secuencia de inicio. Si la respuesta es diferente de cero se debe seguir enviando el CMD24 hasta obtener como respuesta cero.
- 4) Enviar un *token* de inicio 0xfffe, este indicara a la microSD que los valores que siguen son la información que deberá almacenar.
- 5) Se envían los 512 bytes que se guardarán en la memoria.
- 6) Se deben enviar otros 512 bytes, esto se hace para darle tiempo a la memoria de almacenar los datos, pero se debe guardar el valor del tercer byte ya que este nos indicará si se han guardado los datos o no.
- 7) Se debe de enviar el CMD13 si la respuesta del último byte es diferente de cero esto indicará que la tarjeta se ha bloqueado y para desbloquear se debe de enviar nuevamente la secuencia de inicio.
- 8) Si el valor el valor del guardado en el paso 5 es diferente de 0xe5, se debe a repetir todo el algoritmo debido a que no se han guardado correctamente los valores en la memoria.
- 9) Finaliza la escritura colocando el *chip select* en alto.

Nota: Se debe esperar al menos 50 milisegundos cada vez que escribe o lee un sector de la memoria, el no esperar este tiempo, puede bloquear la memoria.

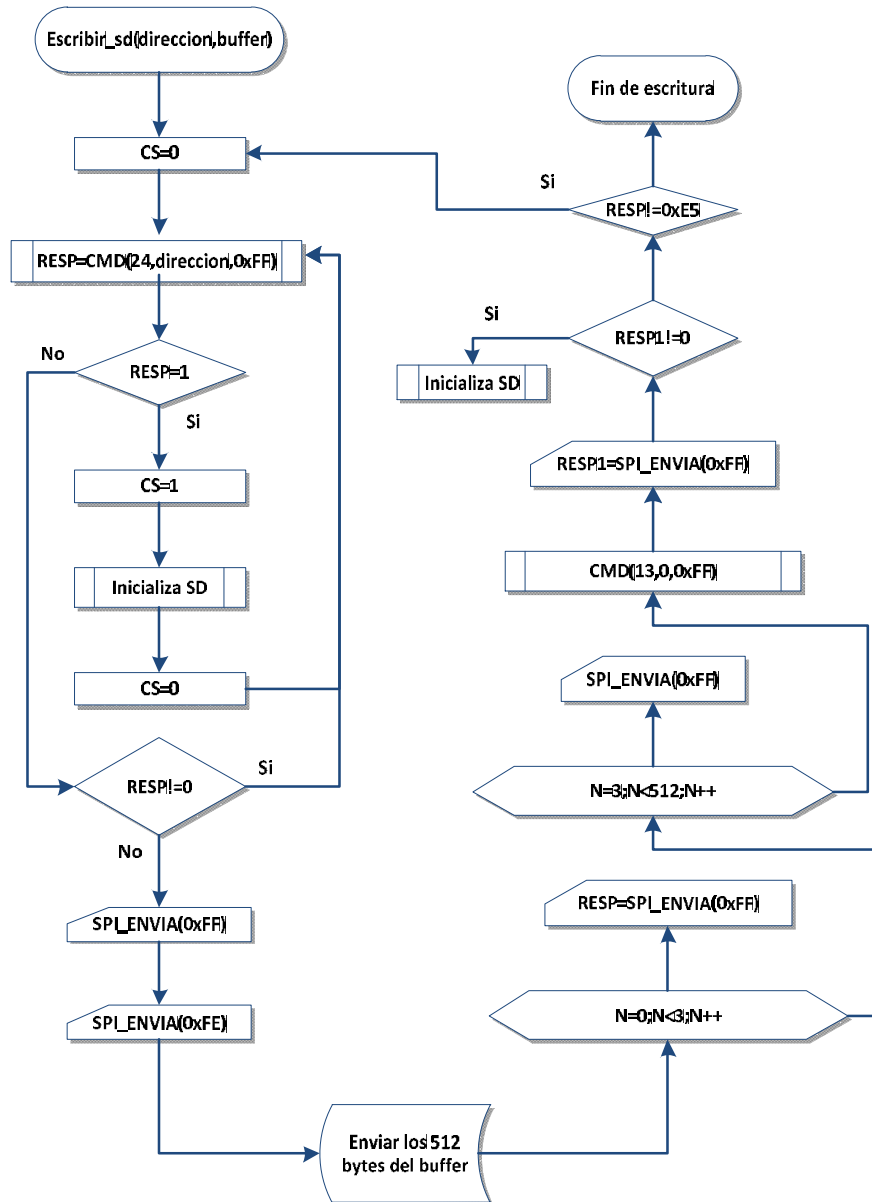


Figura 3.34. Diagrama de flujo para escribir en un sector de la memoria microSD.

Lectura

Para la lectura se siguen las mismas consideraciones que en la escritura es decir si se trata de una microSD versión 1 se debe hacer un corrimiento lógico de 9 bits hacia la izquierda.

El diagrama de flujo se muestra en la figura 3.35 los pasos a seguir son los siguientes:

- 1) Se activa el *chip select* poniéndolo en estado bajo.
- 2) Se envía el comando CMD17 el argumento es el sector de la memoria que se desea leer, hasta que se obtenga como respuesta 0x00.

- 3) Se envían el dato 0xff hasta que la tarjeta microSD responda con un 0xfe, esto indica que los siguientes 512 bytes corresponden a los datos guardados en la memoria.
- 4) Se envían 512 bytes para poder obtener los datos en la memoria.
- 5) Finalmente se envían dos bytes en 0xff.
- 6) Se coloca en el *chip select* en alto.

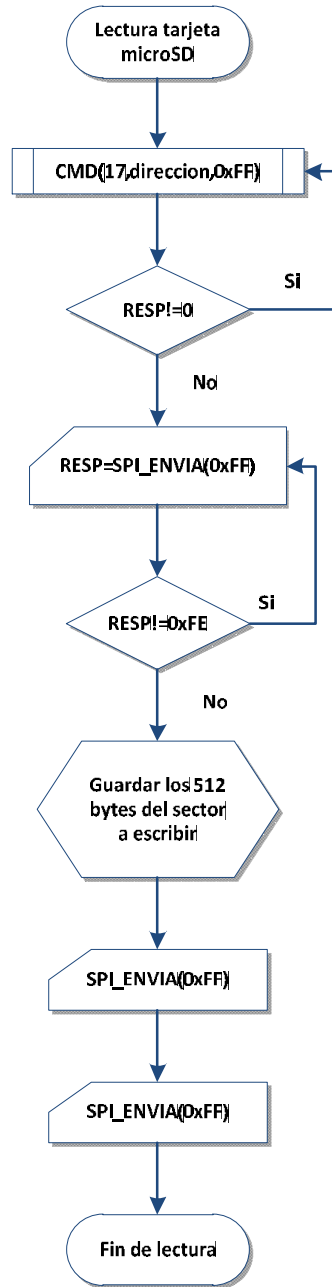


Figura 3.35. Diagrama de flujo para leer un sector de la memoria microSD.

Crear un archivo

Una vez explicadas las dos operaciones elementales que se tienen que realizar en la tarjeta microSD, se procede a explicar cómo se crea un archivo en la microSD.

Primero debemos leer el sector cero el cual se muestran en la figura 3.36, en donde se deben de leer los parámetros siguientes:

- Sectores por cluster, consta de dos bytes y se localiza en el byte 13 del sector 0.
- Sectores reservados, consta de dos bytes y se localiza en el byte 14 del sector 0.
- Número de entradas permitidas en el directorio de raíz, consta de dos bytes y se localiza en el byte 17 del sector 0.
- Sectores por FAT, consta de dos bytes y se localiza en el byte 22 del sector 0.

Offset	Title	Value
0	JMP instruction	EB 3C 90
3	OEM	MSDOS5.0
BIOS Parameter Block		
11	Bytes per sector	512
13	Sectors per cluster	128
14	Reserved sectors	6
16	Number of FATs	2
17	Root entries	512
19	Sectors (under 32 MB)	0
21	Media descriptor (hex)	F8
22	Sectors per FAT	233
24	Sectors per track	63
26	Heads	255
28	Hidden sectors	0
32	Sectors (over 32 MB)	7626752
36	BIOS drive (hex, HD=8x)	80
37	(Unused)	0
38	Ext. boot signature (29h)	29
39	Volume serial number (decimal)	2524538353
39	Volume serial number (hex)	F1 65 79 96
43	Volume label	NO NAME
54	File system	FAT16
510	Signature (55 AA)	55 AA

Figura 3.36. Sector cero de la microSD.

Una vez que se han obtenido los datos anteriores se tienen que leer el sector de la FAT y el directorio de raíz, la dirección de estos dos sectores se calculan de la forma siguiente:

- Para el sector de la FAT su dirección corresponde al número de sectores reservados, debido a que este directorio comienza después de los sectores de inicio de la FAT. Existe un segundo directorio FAT, sin embargo, este es una copia del primero, su dirección se obtiene de sumar el sector de la FAT más los sectores por FAT. Por ejemplo si tomamos los datos que se muestran en la figura 3.36, el directorio FAT1 se localizara en el sector 6 y el directorio FAT2 se encontrara en el sector 239.
- La dirección del directorio raíz se obtiene sumando la dirección de la copia de la FAT y los sectores por FAT, su longitud depende de su número de entradas; cada entrada consta de 32 bytes por lo que en cada sector se pueden almacenar 16

entradas. Si se toman nuevamente los datos de la figura 3.36 el directorio raíz se localiza en la dirección 472 y su longitud se logra dividiendo el número de entradas permitidas en el directorio de raíz entre las entradas que se pueden almacenar en un sector, así que el número de sectores que componen el directorio de raíz es 32.

- Finalmente se tienen los sectores que corresponde a la parte de datos de los archivos su dirección se obtiene de sumar la dirección del directorio de raíz más el número de sectores que componen dicho directorio. El cluster 2 es el cluster en el que inicia la parte de datos.

Una vez identificados los parámetros anteriores procedemos a seguir el algoritmo mostrado en la figura 3.37, para ejemplificar cada paso del diagrama de flujo y lograr un mejor entendimiento.

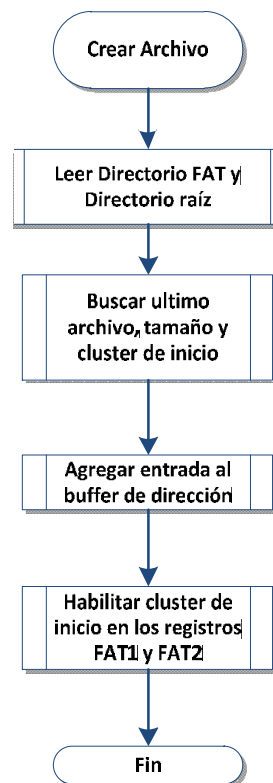


Figura 3.37. Diagrama de flujo para crear un nuevo archivo

- 1) Se leen el directorio FAT1 el cual se encuentra en el sector 6 y el directorio de raíz que se encuentra en el sector 472 y se almacenan en dos arreglos diferentes, lo anterior consumirá más memoria en el microcontrolador también reducirá el tiempo en el que interactúa.
- 2) Se busca una entrada libre en el directorio de raíz, en la figura 3.39 se muestra una imagen del directorio de raíz, se puede ver que se encuentran ya dos entradas la forma en la que se buscan las entradas es la siguiente:

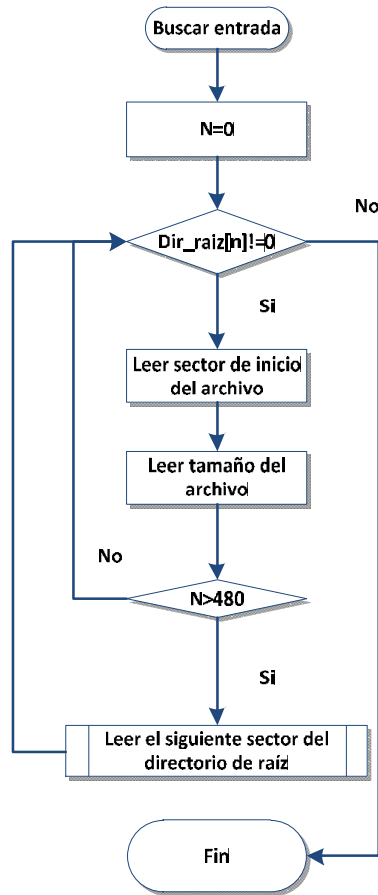


Figura 3.38. Diagrama de flujo para buscar una entrada libre

- 3) Una vez encontrado el lugar donde se pondrá la nueva entrada se tiene que calcular el *cluster* de inicio del nuevo archivo a partir del sector de inicio del archivo anterior y de su tamaño, para ello se tiene que calcular la cantidad de *clusters* ocupados con la ecuación 3.39. En el caso de que la división no sea exacta se debe tomar solo la parte entera y sumar un uno al resultado. Para calcular el *cluster* de inicio del archivo se deben sumar el *cluster* de inicio del último archivo y el número de *clusters* ocupados. El *cluster* de inicio de forma predeterminada siempre será el 2.

$$\text{Numero de clusters ocupados} = \frac{\text{Tamaño de archivo}}{65536} \quad \text{Ecuación 3.39}$$

- 4) Se agrega la entrada en el directorio de raíz y se actualizan los directores FAT1 y FAT2.
- 5) Se ingresa la entrada nueva con su respectivos parámetros, en este caso solo ingresaremos el nombre, la extensión, propiedades y *cluster* de inicio.
- 6) Se habilitan en el directorio FAT1 y FAT2 el *cluster* de inicio del archivo que se acaba de crear.

Name ▲	Ext.	Size ▲	Created	Modified	Accessed
(Root directory)		16.0 KB			
0003B000		44 41 54 4F 53 30 30 30	43 53 56 20 00 77 7D 6F		
0003B010		58 44 58 44 00 00 00 08	21 28 02 00 C0 86 01 00		
0003B020		44 41 54 4F 53 30 30 31	43 53 56 20 00 46 03 70		
0003B030		58 44 58 44 00 00 14 76	2F 44 04 00 86 E5 01 00		
0003B040		00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0003B050		00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		

Sector 472 of 7626752 Offset: 3B05F = 0 Block: 3B040 - 3B04F Size: 10

Drive F: [unregistered] 100% free
File system: FAT16
Default Edit Mode: original
State: original
Undo level: 0

Figura 3.39. Directorio de raíz

Escribir datos

Para un mejor manejo en el almacenamiento de datos se crea una función llamada *sd_print*. Esta función básicamente se encarga de ir almacenando datos que serán guardados en la parte de datos de la memoria SD cuando el buffer este lleno. Se muestra el diagrama de flujo de dicha función en la figura 3.40.

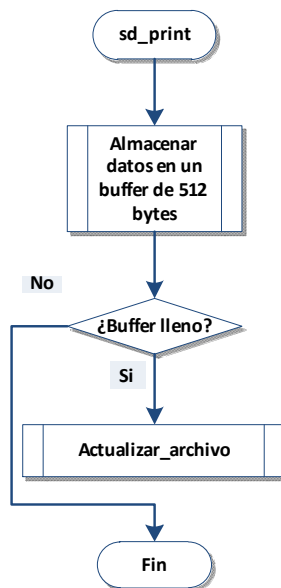


Figura 3.40. Diagrama de flujo para escribir datos en un archivo

Actualizar archivo

Esta función se encarga de actualizar el tamaño de archivo que se ha creado; lo que realiza es actualizar el tamaño del archivo en el directorio de raíz y de ser necesario actualizar el directorio FAT1 y FAT2. Se propone actualizar el tamaño de los archivos cada vez que se ocupe un *cluster* nuevo o que el archivo sea cerrado; para ello se sigue el diagrama de flujo de la figura 3.41.

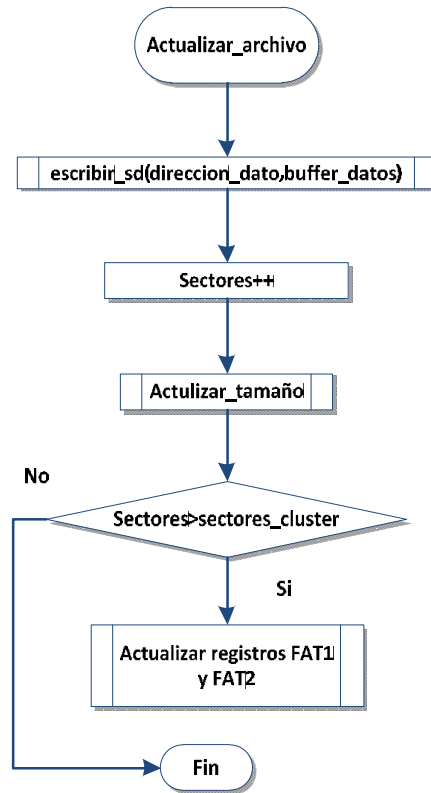


Figura 3.41. Diagrama de flujo para actualizar datos en un archivo

3.4.4. Envío de datos

Para el manejo del modem se crea una librería la cual está basada en la comunicación UART, a continuación se describen las funciones que componen.

Funciones base

De forma similar que en la librería de la memoria microSD se crean las funciones básicas que nos permiten activar y reiniciar el modem en caso de ser necesario, así también se tienen instrucciones como “comando” y “busca” que servirán como base para crear las demás rutinas de configuración, conexión a internet y envío de mensajes SMS.

Activación y reinicio

Se debe recordar que el modem además de la comunicación UART necesita de dos terminales más para su funcionamiento; éstas deben de estar en colector abierto (ésto es muy importante ya que cualquier otra configuración de los terminales impedirá que el modem pueda ser utilizado) y deben de estar en estado alto sólo debe cambiarse su estado cuando sea necesario.

La activación del modem es relativamente sencilla; como se observa en el diagrama de la figura 3.42, se compone de dos pasos estos se describen a continuación:

- **Activar comandos AT.** Para activar el modem el terminal “Activa” debe permanecer en estado bajo por al menos 50[ms], después este debe regresar a su estado alto ya que el modem GPRS verifica de forma periódica esta señal y cada vez que encuentra esta terminal en estado bajo ejecutará una rutina de inicialización.
- **Comprobar nivel de potencia de la señal.** En este paso se debe comprobar la potencia de la señal que recibe modem, este se hace mediante el envío del comando “AT+CSQ”, se debe enviar este comando hasta que la respuesta del modem sea diferente de 99; no se deben enviar comandos hacia el modem mientras este valor sea 99 ya que el modem se reiniciará una y otra vez.

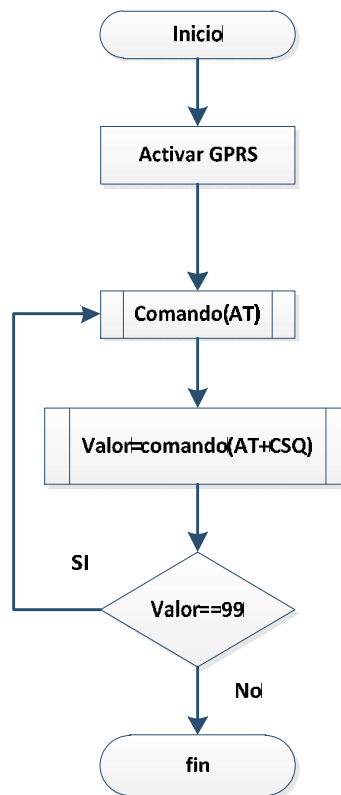


Figura 3.42. Inicializar modem

Para reiniciar el modem se debe poner en bajo la terminal reinicio del modem durante al menos 50[ms], después de eso, esta terminal debe regresar a su estado inicial ya que la secuencia de reinicio se activa por medio de un flanco de subida.

Comando

La función comando se encarga de enviar los comandos AT al modem también se asegura de que la instrucción fue recibida correctamente por él, para ello se sigue el diagrama de flujo de la figura 3.43.

Este comando se encarga de enviar una cadena de texto a través del módulo UART, si es requerido se activan instrucciones adicionales estas le agregan una cadena de texto el código

hexadecimal 0x0d0a, esperan la respuesta del modem y buscan códigos de errores en la respuesta del modem; si el comando AT fue aceptado la función termina, en caso contrario se vuelve a enviar el comando.

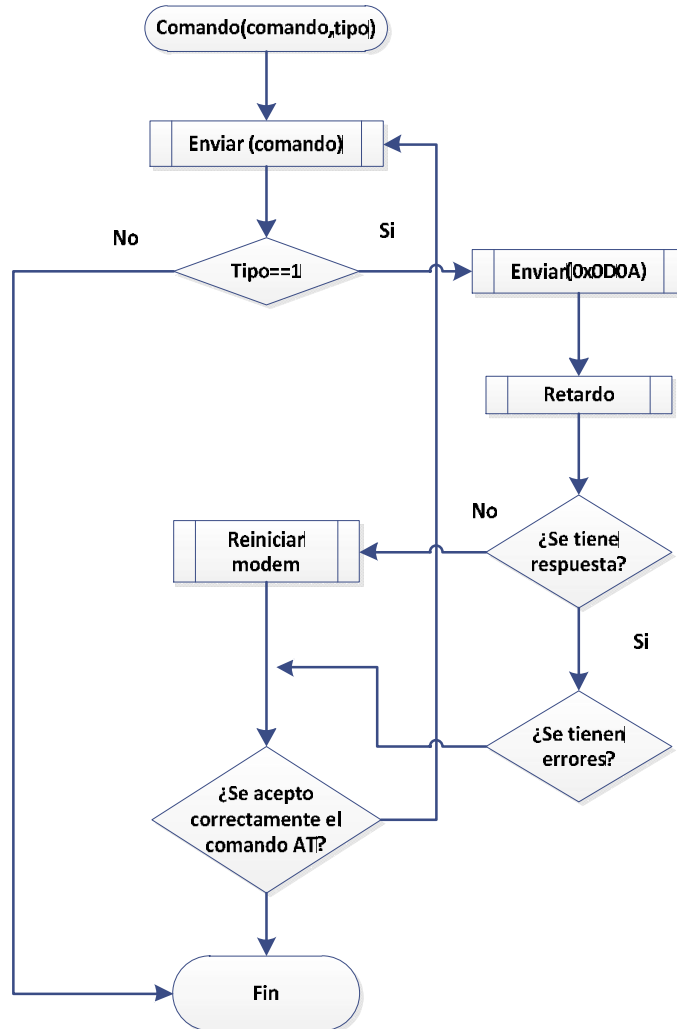


Figura 3.43. Función comando

Busca

La función busca se encarga de analizar la respuesta del modem a los comandos AT; para ello, a diferencia del GPS no se utilizan interrupciones. Esta función tiene dos argumentos “info” y “tipo”.

Tipo se encarga de activar la búsqueda: cuando tipo es igual a 0, se busca la cadena “info” en la respuesta del modem si la encuentra cambia la variable valor a 0, en caso contrario valor se hace igual a 1; cuando tipo es igual a 1 el algoritmo buscará el nombre del operador de la red y establecerá una conexión GPRS con el operador.

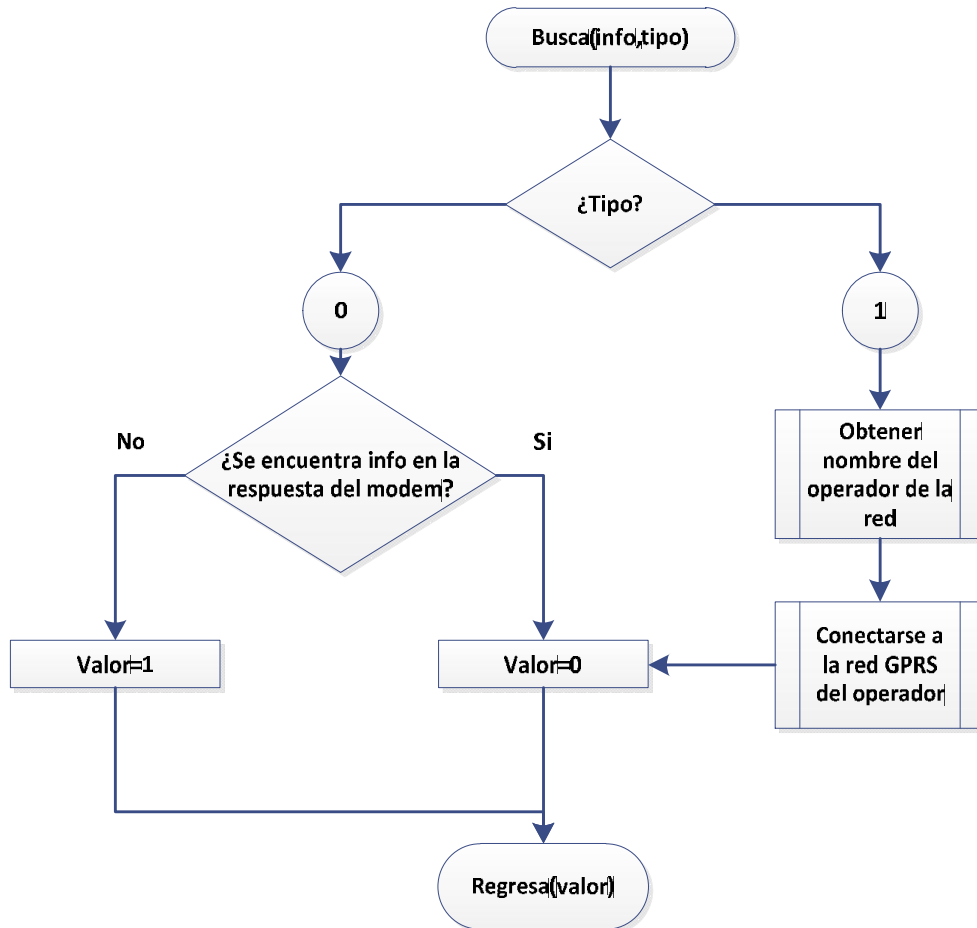


Figura 3.44. Función busca

Configuración

Para la configuración del modem se establecen dos partes importantes del código éstas son:

- **Configuración del modem.** Para la configuración del modem deben enviarse los comandos siguientes:
 - I. **AT&F.** Restablecer la configuración del modem a su configuración de fábrica.
 - II. **ATE0.** Quitar el eco de la respuesta del GPRS.
 - III. **AT\$AREG=1.** Activar el auto registro del modem.
 - IV. **AT+IFC=0,0.** Desactivar las líneas del control de la comunicación serial, el primer parámetro activa la terminal CTS, el segundo activa la terminal RTS.
 - V. **AT+IPR=115200.** Determinar la velocidad de transmisión del modem, aunque el modem tiene una velocidad preestablecida de 115200[baudios].

- VI. **AT+ICF=3.** Cambiar la configuración de la comunicación UART (número de bits, bits de paro y paridad) se escoge el tipo 3 que corresponde a 8 bits de datos, un bit de paro y sin paridad.
- VII. **AT+CMEE=0.** Desactivar expansión de los códigos de error.
- VIII. **AT&W.** Guardar la configuración del modem.
- **Registro del modem.** Para registrar el modem se envían los comandos siguientes
 - I. **AT+CREG?** Verificar que se establezca el registro del modem en la red GSM.
 - II. **AT+COPS?** Conocer el nombre de la compañía que nos provee el servicio GSM.
 - III. **AT+CGDCONT.** Establecer el nombre del contexto PDP (*Packet Data Protocol*).
 - IV. **AT%CGPCO.** Establecer contraseña para el contexto PDP
 - V. **AT+CGSN.** Obtener del número IMEI (International Mobile Equipment Identify)
 - VI. **AT\$MDMID.** Establecer el número IMEI como nombre del dispositivo
 - VII. **AT&W.** Guardar configuración.

Conexión a internet

Para realizar la conexión a internet se debe de seguir el siguiente algoritmo:

- I. **AT\$AREG=2.** Activar el auto registro de la red GPRS
- II. **AT\$FRIEND=1,1,"dirección del servidor".** Establecer conexión con el servidor con el que se desea enviar datos.
- III. **AT\$UDPAPI=,1721.** Establecer el protocolo de comunicación (UDP/TCP) y el puerto de comunicación del servidor.
- IV. **AT\$WAKEUP=1,1.** Activar envío periódico de mensajes al servidor. Esto se hace para mantener la conexión en la red.
- V. **AT&W.** Guardar configuración.

Envío de datos por GPRS

Para enviar datos a través de la red GPRS se utiliza el comando **AT\$MSGSEND=2,"Informacion"**: dentro de los paréntesis se coloca la información en formato ASCII que enviará al servidor.

Fin de conexión

Desactivar la conexión GPRS sólo se debe desactivar el auto registro, esto se logra enviando los comandos siguientes:

- I. **AT\$AREG=1**
- II. **AT&W**

Envío SMS

Para enviar mensajes de texto SMS se debe utilizar el comando siguiente:

AT+CMGS="Número"

Entre los paréntesis se debe colocar el número al cual se le enviará el mensaje de texto en formato ASCII, una vez que se ha enviado este comando se debe esperar a que el modem responda con ">"; inmediatamente se envía el texto del mensaje y se finaliza con el hexadecimal \x01a.

3.5. Procesamiento de datos

Una vez que se han obtenido los datos el siguiente paso es mostrarlos al usuario, para ello se crea un programa que permite al usuario observar los datos que han sido capturados.

El programa se desarrolla en el lenguaje orientado a objetos Java debido a que este lenguaje es multiplataforma; es decir, un programa hecho en Java puede ejecutarse en cualquier computadora que tenga instalado Java, no importando así el sistema operativo con el cual se esté trabajando.

3.5.1. Entorno de desarrollo netbeans IDE

Netbeans IDE es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está hecho para programar principalmente en Java (pero también permite programar en otros lenguajes como C, C++, etc.) Por lo que cuenta con una gran cantidad de información. Es un producto libre y gratuito sin restricciones de uso. Cuenta además de una gran cantidad de librerías que nos permiten desarrollar programas de una forma relativamente sencilla. En la figura 3.45 se muestra una imagen de netbeans.

Netbeans proporciona una gran cantidad de herramientas que nos permiten desarrollar programas desde un editor de texto o crearlos de forma gráfica; para ello genera una plantilla a la cual sólo hay que arrastrar los elementos que deseamos aparezcan en ella para después editar el código fuente.

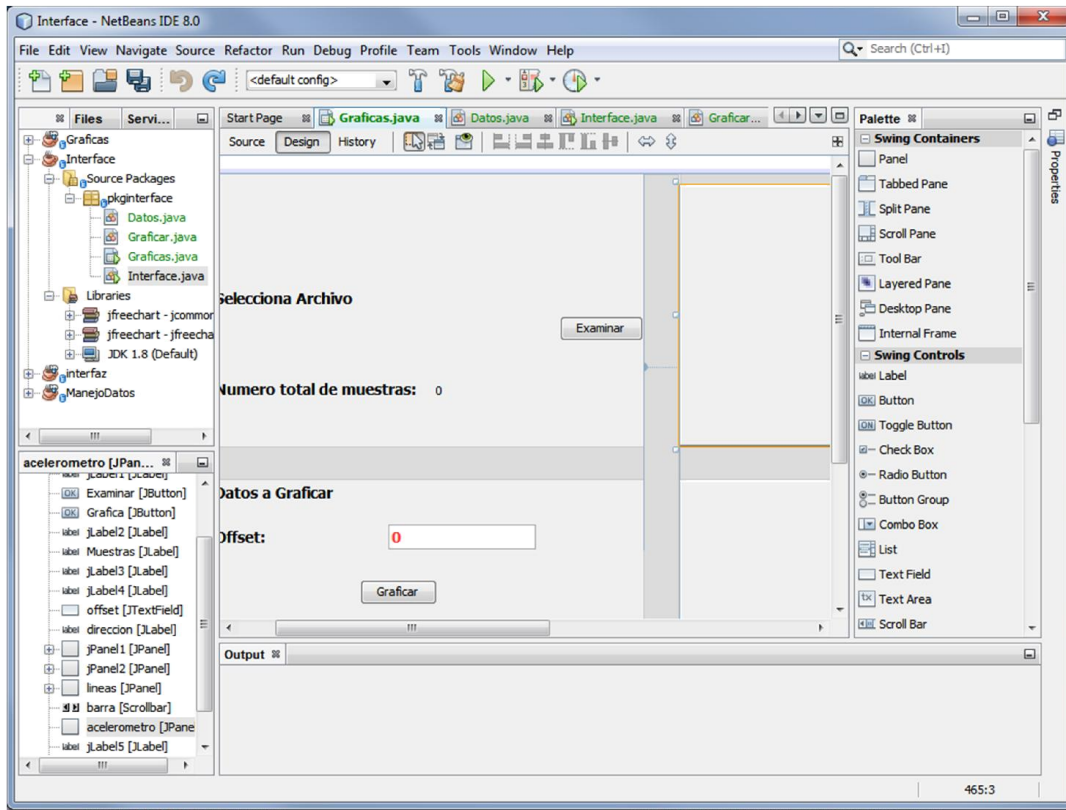


Figura 3.45. Entorno de desarrollo netbeans IDE

3.5.2. Programa para visualización de datos

El programa para visualización de datos se comienza a desarrollar de forma gráfica colocando los elementos que interactuarán con el usuario como (interfaz de usuario): cuadros de texto, barras, botones, etc. El resultado se muestra en la figura 3.46.



Figura 3.46. Plantilla del programa para vizualizar los datos

Una vez que se concluyó con el diseño de forma gráfica se escribe el código del programa, para ello se comienzan a editar las clases que fueron creadas por el programa.

El archivo que generó el sistema de adquisición de datos es un archivo con extensión CSV, el cual es un archivo que se caracteriza por separar los datos a través de comas (,).

Los atributos que contiene este programa son:

- Tamaño de archivo. Tipo entero con valor inicial de 0.
- Tipo de archivo. Tipo boolean con valor inicial "false"

Aunque prácticamente cada elemento creado es una clase diferente, sólo se tienen tres clases principales las cuales se describen a continuación.

Obtención de las propiedades del archivo

Para la obtención de las propiedades del archivo se utilizan los métodos de las clase `filereader`, `bufferedReader` y se sigue el algoritmo siguiente:

- I. Obtener la dirección del archivo.
- II. Comprobar la existencia del archivo. Si el archivo existe se sigue con el algoritmo y cambiar tipo de archivo a "true", de lo contrario se manda un mensaje de error.
- III. Obtener del tamaño del archivo.

Lectura de los datos del archivo y Generación de gráficas

Una vez que sea comprueba la existencia del archivo se deben obtener los datos que este contiene para ello se sigue el algoritmo que se describe a continuación:

- Se obtienen los datos por medio de la clase `bufferedReader`, se crean los objetos `ejex`, `eje`, `ejez` y `pulso` por medio del constructor de la clase `xyseries`. En estos se guardarán los datos que se obtengan del archivo.
- Se obtiene la información separando la información por comas y sólo se conserva el último valor de la temperatura.
- Se agregan los datos a los objetos correspondientes.
- Una vez que se han guardados los datos en la series correspondientes, se deben generara dos colecciones de datos; en una se guardan la serie de la señal pulso y en la otra se guardarán las series de las señales del acelerómetro
- Se generan las gráficas.

El programa se prueba con cuatro señales senoidales defasadas las cuales fueron generadas por medio de matlab, así también se coloca un valor arbitrario de temperatura la cual es constante. La barra permite realizar al usuario un barrido de la señal así también se permite graficar utilizando un offset. En las figura 3.47 y figura 3.48 se muestran imágenes del programa.

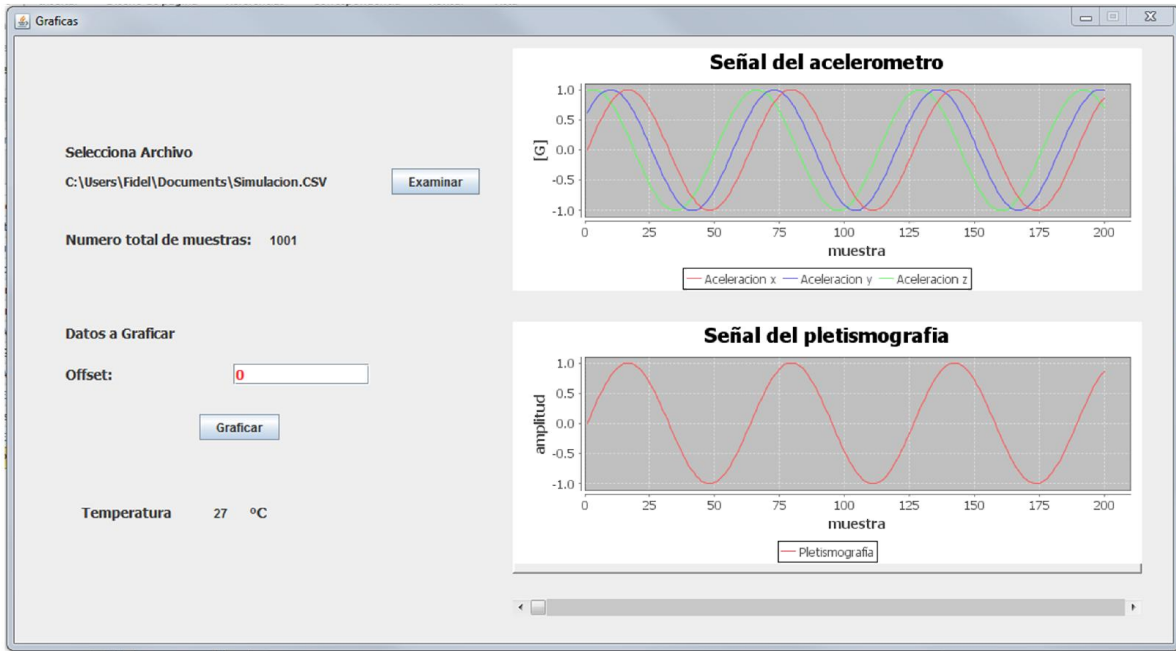


Figura 3.47. Graficas de las señales senoidales

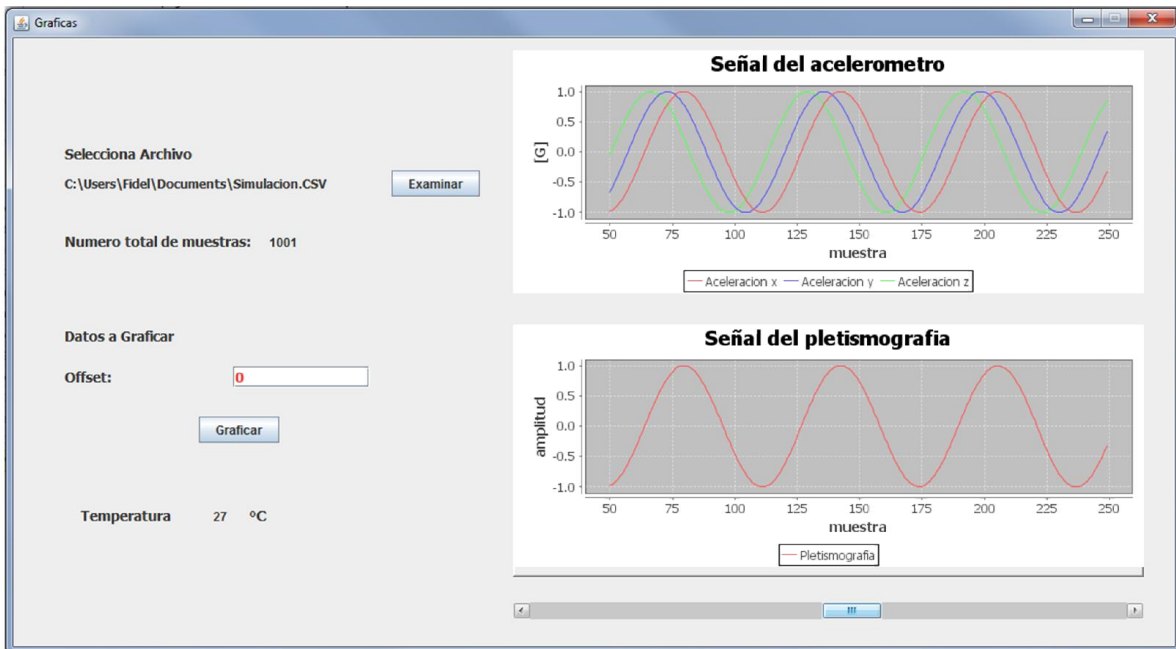


Figura 3.48. Graficas de las señales senoidales con la barra desplazada

4. Integración y pruebas

En este capítulo se abordarán las pruebas que se realizaron para la validación del prototipo. Para ello se realizaron pruebas individuales a cada subsistema y finalmente se realizó una prueba a todo el sistema.

4.1. Integración del sistema en PCB

La integración del sistema así como todas las pruebas se realizaron en el Instituto de Ciencias Nucleares de la UNAM en el laboratorio de proyectos emergentes.

El sistema de adquisición de datos fue desarrollado en una PCB de dos capas las cuales se muestran en la figura 4.1 y figura 4.2. Sus dimensiones son de $61[mm] \times 45[mm]$.

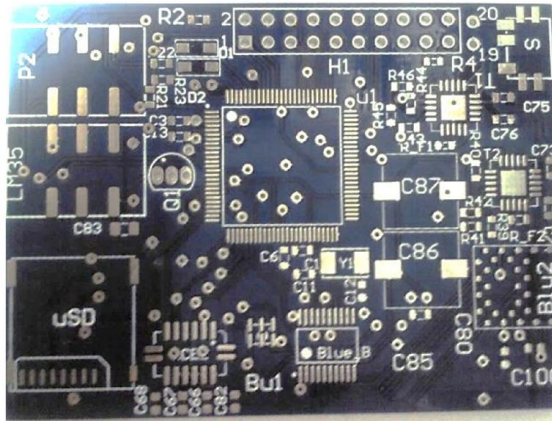


Figura 4.1. Capa superior del PCB del sistema

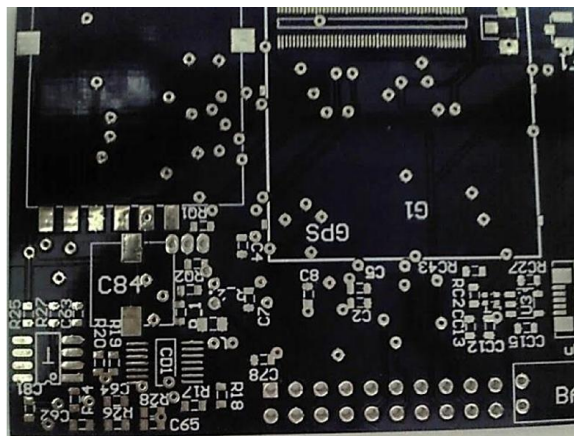


Figura 4.2. Capa inferior del PCB del sistema

El montaje de los circuitos electrónicos en la PCB se realizó por etapas para verificar el funcionamiento de los dispositivos.

En la figura 4.3 se muestra una imagen de la capa superior del PCB en el cual se pueden observar componentes como: el microcontrolador, el interruptor, los reguladores, el acelerómetro, entre otros.

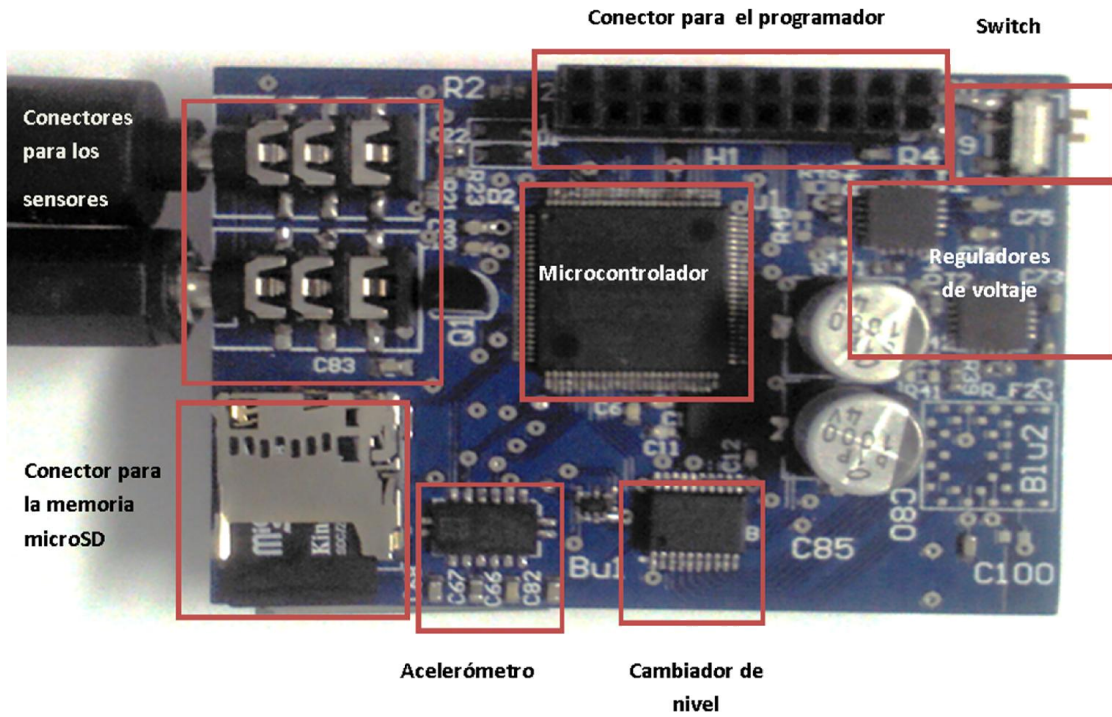


Figura 4.3. Capa superior del sistema de adquisición de pulso radial

En figura 4.4 se muestra la imagen de la capa inferior del PCB, en este se encuentran: el GPS, el modem GPRS, sistema de acondicionamiento de señales el cargador de baterías y conector de la batería.

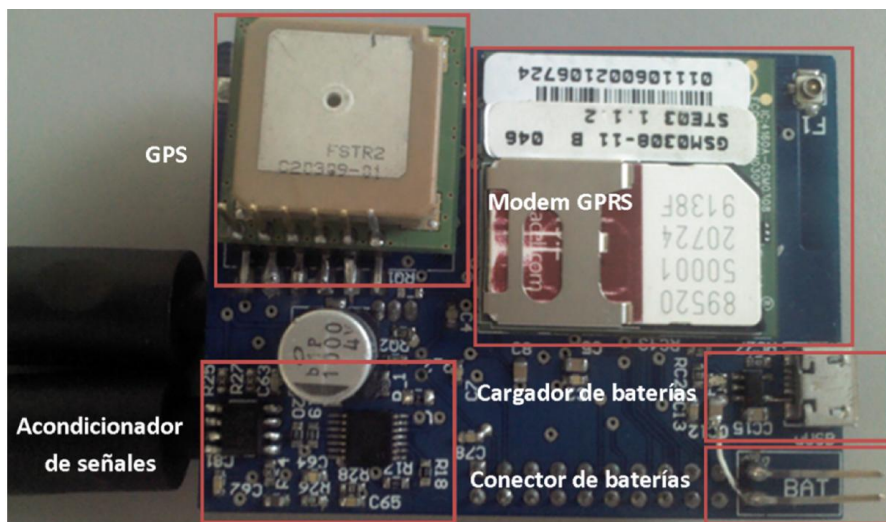


Figura 4.4 Capa inferior del sistema de adquisición de pulso radial

4.1.1. Pruebas al módulo de potencia del sistema de adquisición

Este se encarga de proveer la energía que necesita el sistema de adquisición, está formado por: la batería, el conector para la batería, el interruptor de encendido, el cargador de la batería, los reguladores, así como también las resistencias y capacitores que se necesitan para la operación de estos, además de la resistencia que se encarga de unir la tensión de referencia analógica con la digital.

En este módulo sólo se realizaron mediciones de voltaje a la salida de los reguladores las mediciones se encuentran registradas en la tabla 4.1.

Regulador	Voltaje mínimo permitido [V]	Voltaje máximo permitido [V]	Voltaje medido [V]	Voltaje Nominal [V]
1	3	3.5	3.35	3.3
2	1.6	2	1.79	1.8

Tabla 4.1. Condiciones eléctricas de operación

El cargador de batería se probó cargando baterías lipo (*Lithium Polymer*); para ello se midió el voltaje de inicial de la batería, una hora después su voltaje final. Se realizaron dos mediciones estas quedaron registradas en la tabla 4.2.

Batería	Voltaje inicial [V]	Voltaje Final [V]
1	3.2	4.7
2	3	4.6

Tabla 4.2. Registro de los voltajes en las baterías antes y después de una hora de ser conectarlos al circuito cargador de baterías

4.2. Pruebas de funcionamiento preliminares

Antes de comenzar a probar el sistema de forma completa se realizaron pruebas individuales a cada subsistema, esto, para asegurarnos que este funciona correctamente y realizar correcciones tanto de software como hardware.

Para transferir el programa del entorno de desarrollo se utilizó el programador XDS100V3.

Debido a que la distribución de pines de la tarjeta no es la misma a la del programador se construyó un adaptador para realizar de forma adecuada las conexiones el cual se muestra en la figura 4.5. Un punto importante del diagrama de conexiones es la resistencia **RTCK**, que se encuentra entre las TCK y RTCK esta resistencia es necesaria, ya que el programador verifica con ella que la señal de reloj del protocolo JTAG se encuentra en buen estado; sin embargo el microcontrolador no cuenta con esa señal por lo que es necesario generarla por medio de esta resistencia. El valor de $28[\Omega]$ de la resistencia se determinó a base de pruebas.

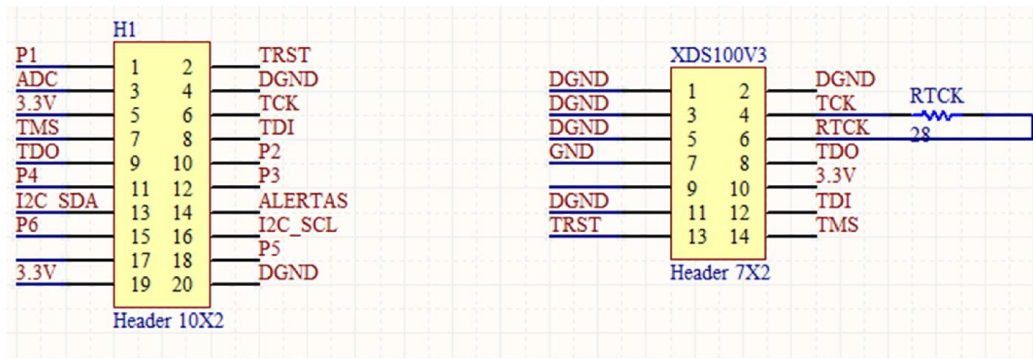


Figura 4.5. Diagrama de conexiones entre la sistema de adquisición (H1) y el programador XDS100V3

En la figura 4.6 se muestra la forma en la que se realizó la conexión entre el programador y el sistema de adquisición.

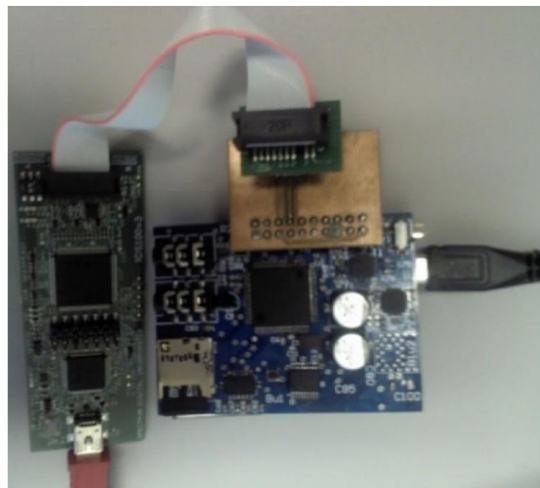


Figura 4.6. Conexión del programador con el sistema de adquisición de datos

4.2.1. Pruebas realizadas al módulo acondicionador de señal

Para realizar la prueba del circuito acondicionador de señal se planeó realizar un barrido de frecuencias por medio de un generador de señales, sin embargo esto no fue posible ya que el generador con el que se cuenta, tiene una amplitud mínima de $50[mV]$ por lo que el voltaje de salida, con la ganancia del amplificador, se estima en $50[V]$ valor que supera la alimentación amplificador operacional.

Para obtener una estimación de las frecuencias de corte se realizó una prueba en la tarjeta protoboard, en este se aumentó el valor del voltaje de alimentación a $10[V]$ y se realizó un barrido de la señal de $0.1[Hz]$ a $15[Hz]$, sin embargo sólo se pudieron obtener frecuencias a las cuales el amplificador no recortaba la señal estas fueron de $0.1[Hz]$ y $13[Hz]$. En la figura 4.7 una imagen de la prueba realizada para la señal de $0.1[Hz]$, la señal de entrada es la señal inferior, sin embargo es tan pequeña que se distorsiona por el ruido, por lo que para ayudar a la visualización

se dibujó una señal dentro de la envoltura, la señal superior representa a la señal de salida si se observan los marcadores puede verse que el valor máximo de la señal es cercano a los 10[V].

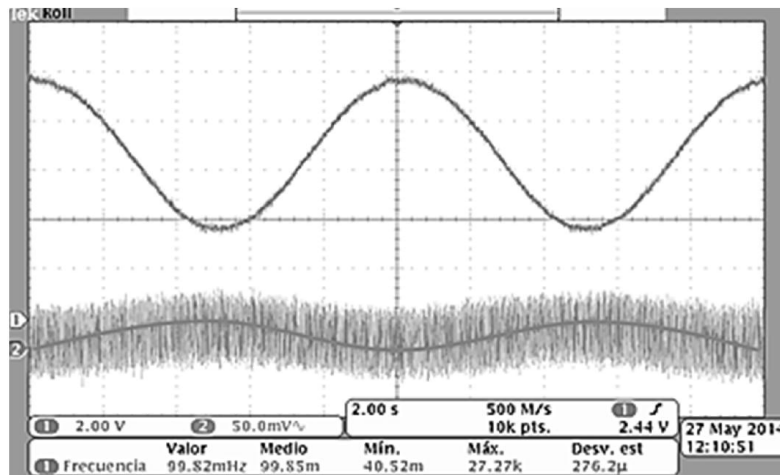


Figura 4.7. Respuesta del circuito acondicionador de señal con una señal de entrada senoidal de 50[mv] y una frecuencia de 0.1[Hz]

Para corroborar el funcionamiento del subsistema se determinó medir la señal de pulso y observar la señal a través del osciloscopio, En la figura 4.8 se muestra la señal obtenida a la salida del filtro del circuito montado en la protoboard al realizar la medición. Si se compara la imagen de la señal obtenida con el fototransistor con la presión arterial promedio de la figura4.9, se observa que estas imágenes son muy similares, y con ello se determinó que el sensor está trabajando de forma adecuada.

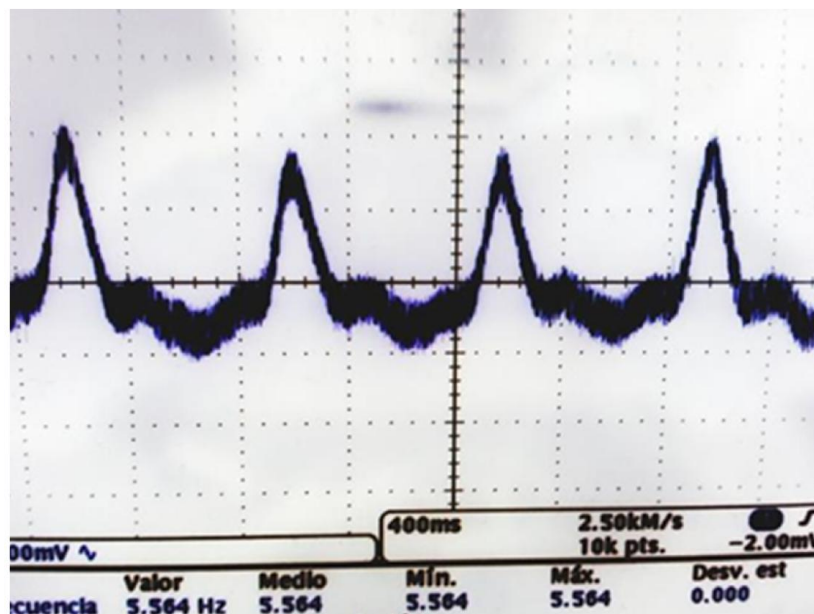


Figura 4.8. Señal obtenida a la salida del filtro

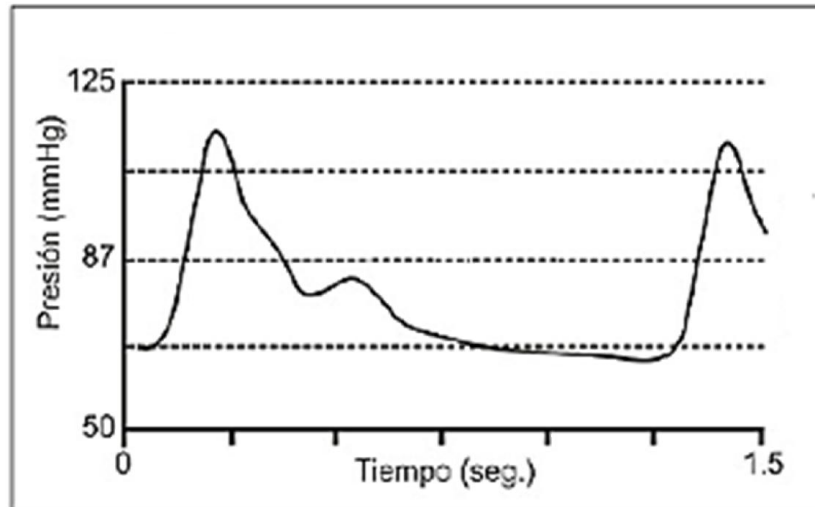


Figura 4.9. Presión arterial promedio.

4.2.2. Pruebas realizadas al módulo GPS

La verificación del GPS se realizó utilizando los dos módulos UART del microcontrolador. La UART1 que es la que interactúa con el GPS y se encarga de recibir los datos que el GPS envía y la UART0 se encarga de retransmitir los datos hacia la computadora por medio del protocolo RS232; para ello se necesitó utilizar el corredor de nivel MAX3232. La comunicación UART1 se configuró con una velocidad de 9600[baudios], un bit de paro, sin paridad y 8 bits de datos; mientras la comunicación UART0 es similar a la UART1 su única diferencia es la velocidad de transmisión que es de 115200[baudios].

Por medio de esto fue posible ver los datos que enviaba el GPS, en la figura 4.10 se muestra un ejemplo de los datos que se recibieron del GPS en un segundo.

```
$GPGGA,184900.000,1919.5306,N,09910.8431,W,1,7,0.96,2294.0,M,-9.0,M,. *64
$GPGSA,A,3,02,24,10,28,12,05,17,,,,,,1.30,0.96,0.87,*0F
$GPGSV,2,1,08,02,61,245,42,10,39,137,21,28,35,114,15,17,33,039,25,*78
$GPGSV,2,2,08,24,31,294,35,05,13,187,34,12,11,322,35,20,03,044,*75
$GPRMC,184900.000,A,1919.5306,N,09910.8431,W,0.20,261.79,180913,, ,A*7C
```

Figura 4.10. Datos recibidos del GPS

Para corroborar que los datos son correctos se utilizó el servidor de mapas de *Google* (*Google Maps*) y se tomaron las coordenadas de la cadena “\$GPRMC,184900.000,A,1919.5306,N,09910.8431,W,0.20,261.79,180913,, ,A*7C” como se indica en la figura 4.11 en esta se observa que las coordenadas obtenidas corresponden al del Instituto de Ciencias Nucleares de la UNAM.

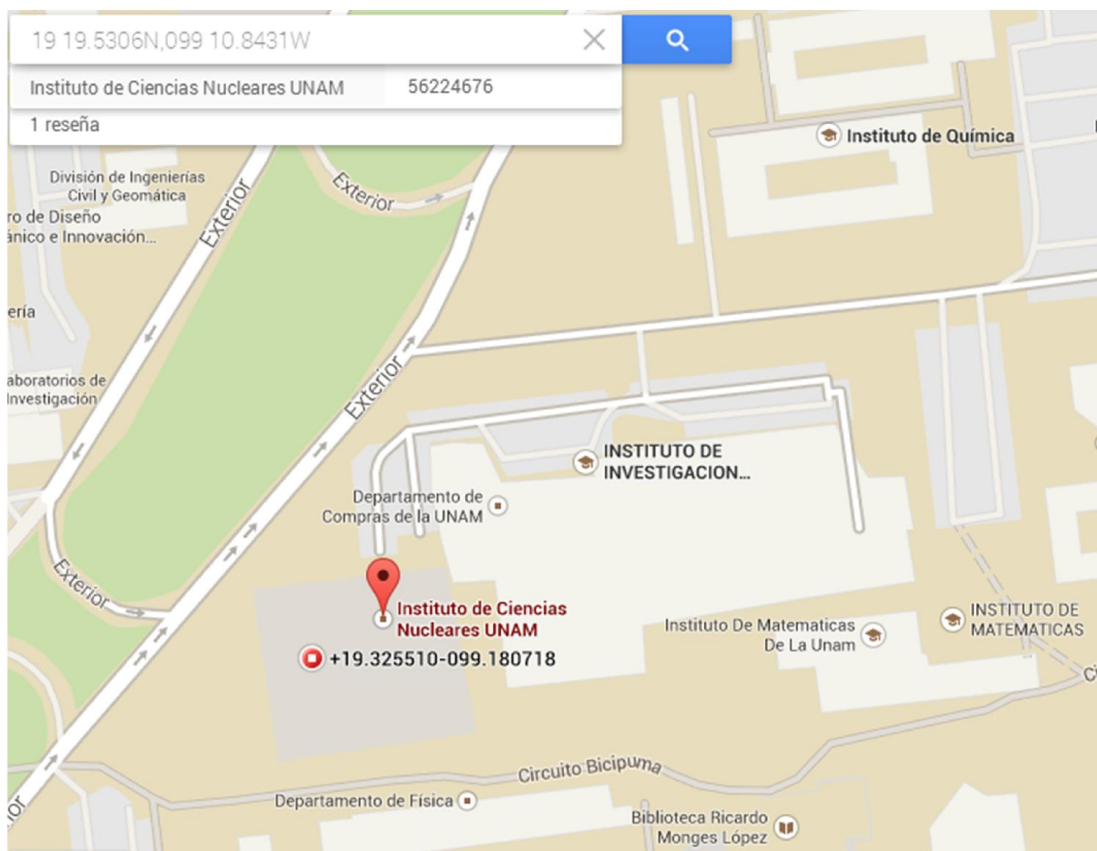


Figura 4.11. Lugar al que corresponden las coordenadas obtenidas el GPS

Al corresponder las coordenadas del GPS con el lugar en el cual se realizaron las pruebas, se da esta prueba por aprobada.

4.2.3. Pruebas realizadas al módulo de almacenamiento de datos

Esta prueba es muy importante ya que servirá como partida para realizar las pruebas finales del sistema de adquisición. En esta prueba consiste en crear dos archivos y que estos guarden la misma información, para verificar que los archivos se guarden en distintos *clusters*. Los archivos llevarán por nombre "TEXTO.TXT" y "TEXTO1.TXT". El microcontrolador deberá escribir en los archivos diez líneas con el mensaje "1, 2, 3."

Para ver los datos que se crean en la microSD se utilizó el programa "winhex", este nos permite ver las regiones que conforman el sistema de archivos, conocer el sector en el que nos encontramos y visualizar los datos que contiene la memoria microSD en hexadecimal o ASCII.

Como puede verse en la figura 4.12 los archivos fueron creados correctamente en el directorio de raíz.

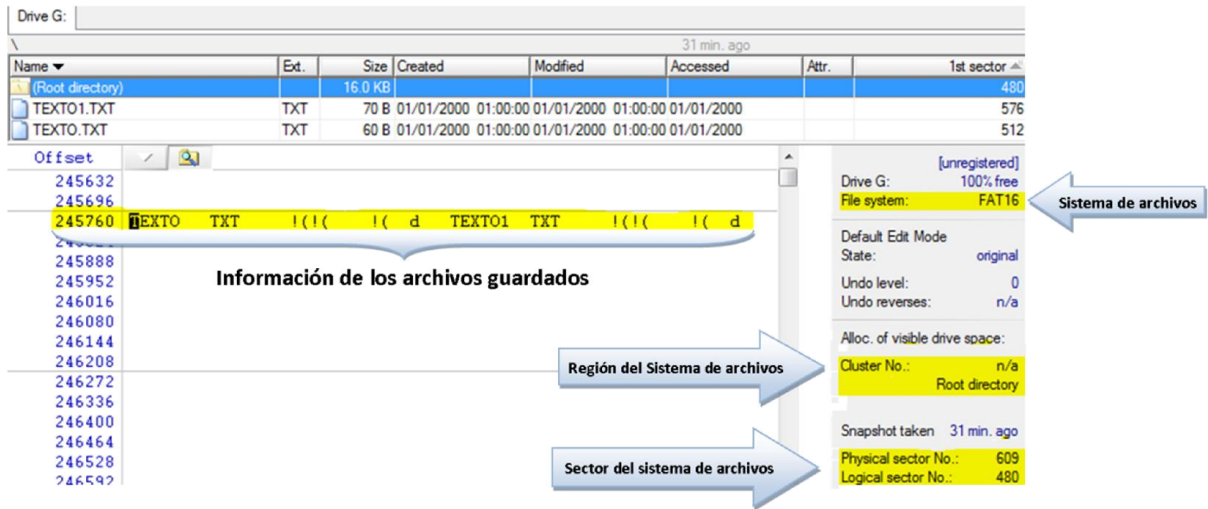


Figura 4.12. Directorio raíz del sistema de archivos FAT

Para realizar un análisis más profundo se activa la visualización de los datos en formato hexadecimal con lo cual se puede observar en la figura 4.13 las intersecciones de la columna 10 con los renglones 245776 y 245808 que los *clusters* de inicio de los archivos son diferentes.

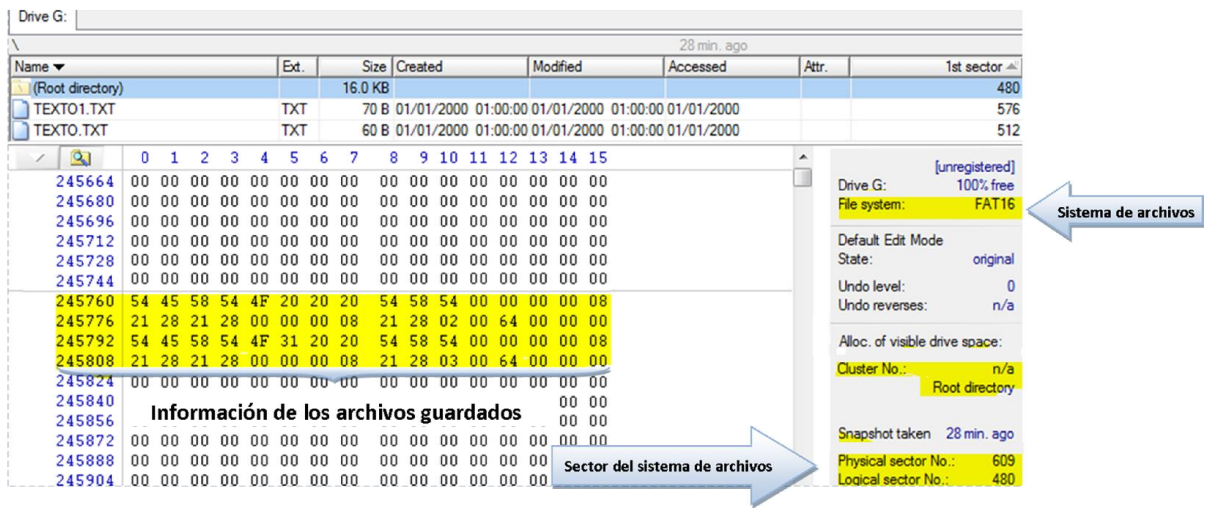


Figura 4.13 Directorio raíz del sistema de archivos FAT en hexadecimal

Se verifica esta información revisando el contenido de los archivos. Como se observa en la figura 4.14 la información del archivo “TEXTO.TXT” efectivamente se encuentra almacenada en el *cluster* 2, mientras que con la figura 4.15 podemos confirmar que la información de “TEXTO1.TXT” se encuentra almacenada en el *cluster* 3.

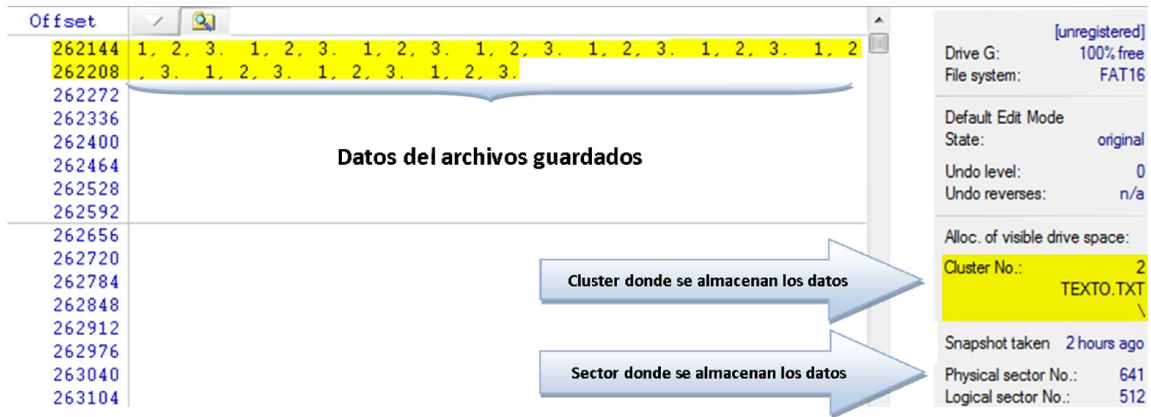


Figura 4.14. Información guardada del archivo TEXT0.TXT

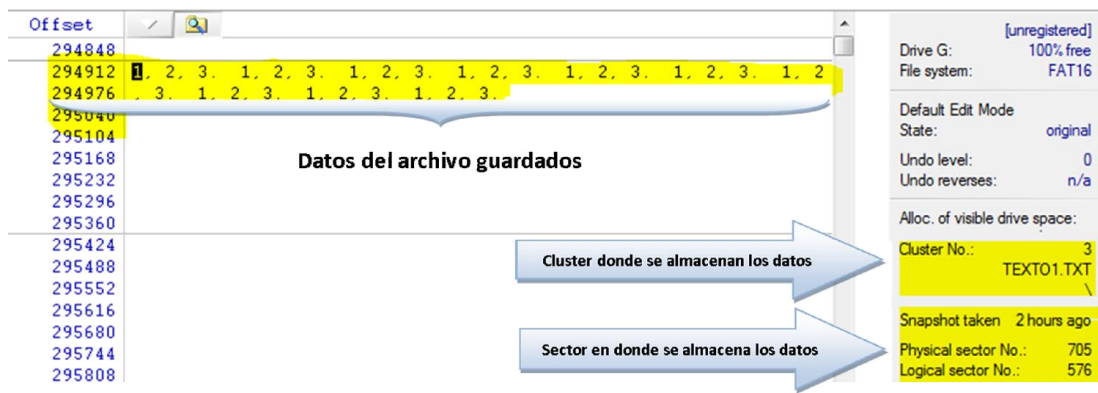


Figura 4.15. Información guardada del archivo TEXT01.TXT

4.2.4. Pruebas realizadas al módulo de envío de datos

Ya que se verificó el funcionamiento de la UART0 en la prueba del GPS, se conectó el modem GPRS junto con el cambiador de nivel para poder interactuar con él y se configuraron los terminales “Activación” y “Reinicio” en colector abierto, la verificación se realiza utilizando desconectando la terminal Tx del modem a un cambiador de nivel el cual nos permitirá ver los datos en la computadora para ello se utiliza el programa X-CTU.

La prueba consiste en enviar comandos desde una terminal en la computadora a la UART1 el microcontrolador tendrá que reenviar los datos a través de la UART0. Sin embargo en esta prueba el modem no responde.

Se analiza el sistema y se encuentra que en la PCB no está conectada la terminal “Activación” por lo que se realiza la conexión y se reinicia la prueba. En este segundo intento el modem siguió sin funcionar.

Después de verificar los niveles de voltaje en las terminales de “Activación” y “Reinicio” se encuentra que en el modem cambian de nivel estas terminales después de un tiempo al ser iniciados, por lo que se revisan las hojas de especificaciones y se prueban las recomendaciones.

Finalmente se opta por acoplar la señal del microcontrolador mediante un transistor en colector abierto como se muestra en la figura 4.16 este operara como interruptor por lo que sólo operara en las regiones de corte y saturación. Al efectuar nuevamente la prueba el modem comenzó a funcionar, por lo que queda solucionado este problema.

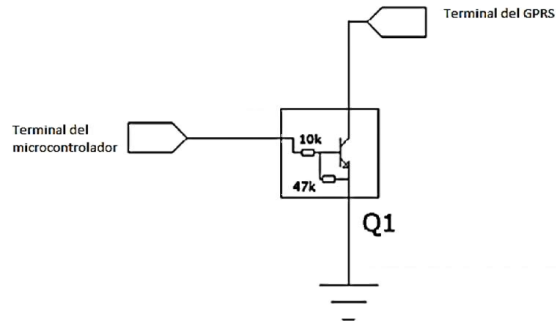


Figura 4.16. Circuito recomendado para la activación de los terminales “Activacion” y “Reinicio”

Se realizaron dos pruebas para verificar el funcionamiento de los algoritmos que se encargan de manejar al modem, estas consistieron en enviar información hacia un servidor y enviar mensajes hacia un celular, en estas pruebas los comandos tendrían que ser enviados por el microcontrolador, la supervisión de estas pruebas se realizó a través de la computadora y a la activación del comando de “eco”.

Para la prueba del servidor, se envió un mensaje hacia un servidor de prueba que proporciona el fabricante (Enfora); el microcontrolador se tiene que conectar al puerto 1721, identificarse con el servidor con el número IMEI del modem y por ultimo mandar el mensaje “Mensaje de prueba generado por VACS 5”.

```
.011106002106872
.
.OK
.AT$MDMID="011106002106872"
.OK
.AT&w
.OK
.AT$AREG=2
.OK
.AT$FRIEND=1,1,"apitest.enfora.com"
.OK
.AT$UDPAPI=,1721
.OK
.AT$WAKEUP=1,1
.OK
.AT&w
.OK
.AT$MSGSEND=2,"Mensaje de prueba generado por
VACS 5"
.OK
.
```

Figura 4.17. Comandos enviados por el microcontrolador al modem para realizar una conexión al servidor de Enfora

El servidor logró conectarse con el servidor de *Enfora* como prueba de ello se muestra la figura 4.18.

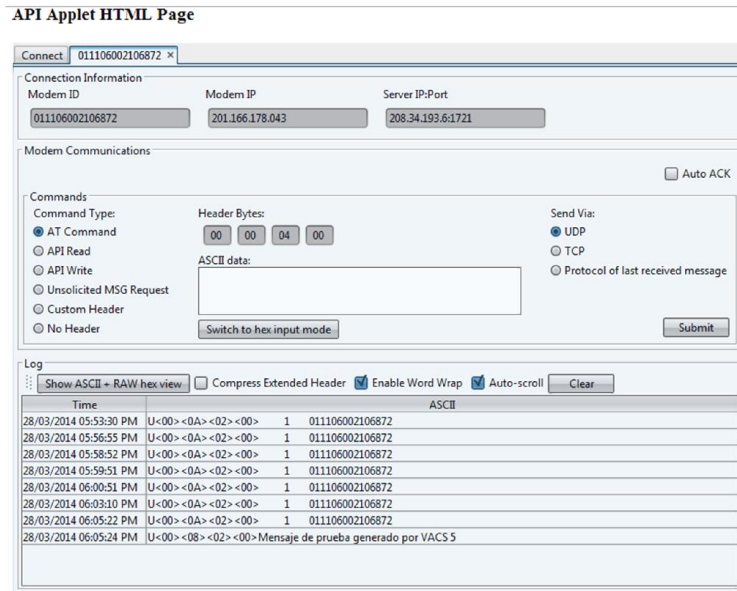


Figura 4.18. Captura de pantalla del servidor de Enfora

Para la prueba de envío de mensajes el microcontrolador debe mandar el mismo mensaje hacia un teléfono celular. Esta prueba también fue exitosa en la figura 4.19 se muestra el mensaje enviado por el microcontrolador.

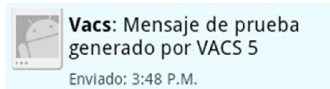


Figura 4.19. Mensaje enviado por el sistema de adquisición recibido por un teléfono celular

Las pruebas de funcionamiento de las bibliotecas programadas se consideran exitosas ya que se efectuaron las rutinas sin error ni bloqueo del microprocesador, por lo que esta prueba se considera aprobada.

4.3. Prueba final

Para corroborar que el sistema de adquisición de datos funcionaba completamente; se decidió medir el pulso de una persona, la cual se encontraba en reposo; con la tarjeta de adquisición lo más paralela posible a una superficie plana, se esperaban obtener señales conocidas de la aceleración del campo gravitatorio terrestre ($1[G]$), la señal de fotoplestismografía y la temperatura directamente en la persona; adicionalmente y a falta de tener un servidor propio al cual se puedan enviar las señales de adquisición, se propuso que el modem envíe un mensaje de texto con la ubicación en la que se encontraba.

Los datos generados de la prueba fueron guardados en la memoria microSD en un archivo llamado "DATOSn", donde "n" represento al número de la prueba realizada.

Los sensores de pulso y temperatura se encontrarán en una pequeña pulsera de velcro la cual se muestra en la figura 4.20. Para colocarla primero se tiene que localizar el pulso radial, para ello se tuvo que palpar el pulso con la yema de los dedos y buscar el lugar en el que se sienta este con mayor intensidad y allí deberá de ser colocado el sensor. El acelerómetro se configuró para que tenga una resolución de 6[G] a 16 bits.



Figura 4.20. Pulsera de velcro con los sensores de pulso y temperatura

Se realizaron 7 pruebas, estas se realizaron como se observa en la figura 4.21 con el brazo inmóvil, la duración de esta fue de aproximadamente 1 minuto.

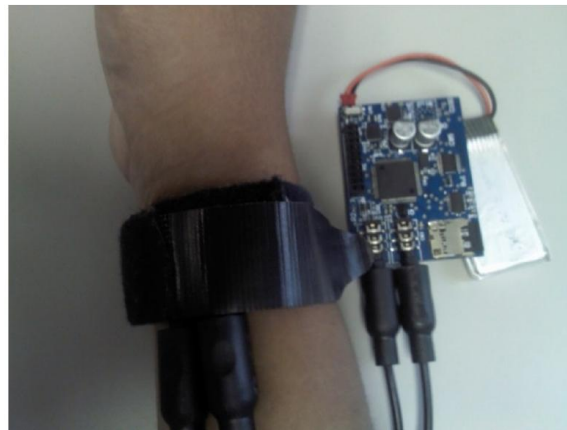


Figura 4.21. Forma en la que se realizó la prueba

Al finalizar las pruebas se revisaron los archivos generados figura 4.22, se muestran los datos almacenados de la séptima prueba. El archivo consiste en una cabecera que indica el valor del canal del ADC leído y los datos de cada uno de los ejes del acelerómetro así como los datos obtenidos. El valor del ADC0 corresponde al canal de la señal de pulso y los valores del ADC1 corresponden al canal la temperatura, cabe señalar que estos datos necesitan de un factor de escala.

	A	B	C	D	E
1	ADC0	ADC1	Eje X	Eje Y	Eje Z
423	593	74	-41	-2	5487
424	569	74	-41	-2	5487
425	547	74	-41	-2	5487
426	529	74	-41	-2	5487
427	514	74	-41	-2	5487
428	502	74	-41	-2	5487
429	498	74	-41	-2	5487
430	496	74	-41	-2	5487
431	498	74	-41	-2	5487
432	504	74	-41	-2	5487
433	511	74	-41	-2	5487
434	516	75	-41	-2	5487
435	523	75	-50	10	5487
436	531	75	-50	10	5515
437	535	75	-50	10	5515
438	541	75	-50	10	5515
439	546	75	-50	10	5515
440	550	75	-50	10	5515
441	549	75	-50	10	5515
442	546	75	-50	10	5515
443	539	75	-50	10	5515
444	530	75	-50	10	5515
445	518	75	-50	10	5515
446	507	75	-50	10	5515

Figura 4.22. Datos obtenidos

Para visualizar el contenido del archivo se utilizó nuestra interfaz creada en el apartado 3.6.2, en la figura 4.23 y la figura 4.24 se puede observar imágenes de la señal de pulso creada por la interfaz , mientras que en la figura 4.25 se observa la señal del acelerómetro.

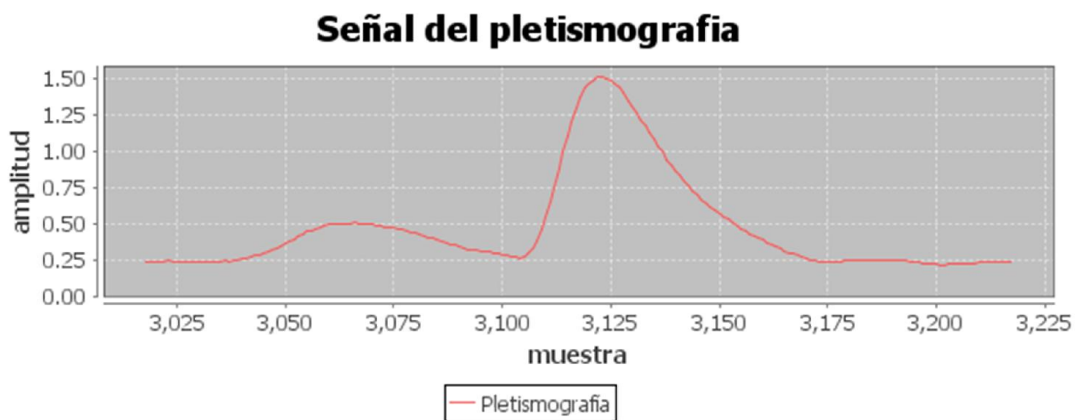


Figura 4.23. Imagen de la señal de pulso generada con 200 muestras

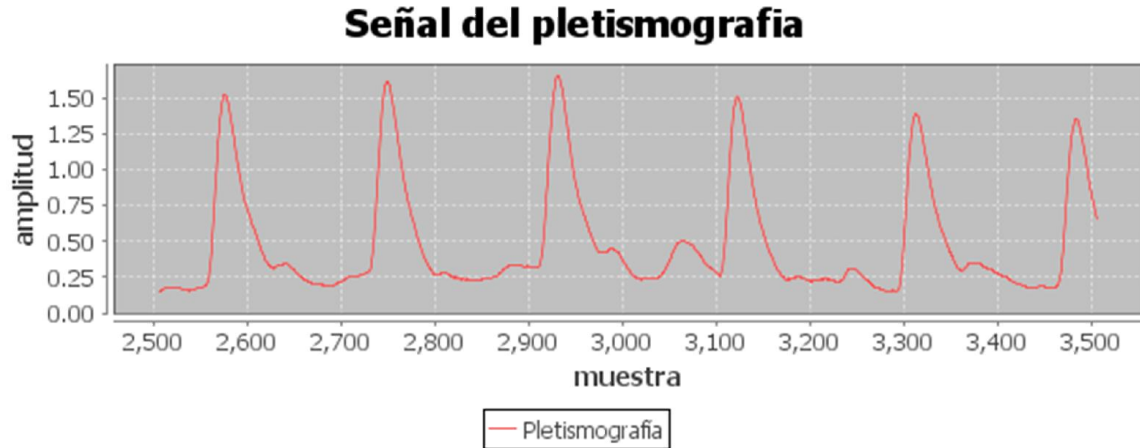


Figura 4.24. Imagen de la señal de pulso generada con 1,000 muestras

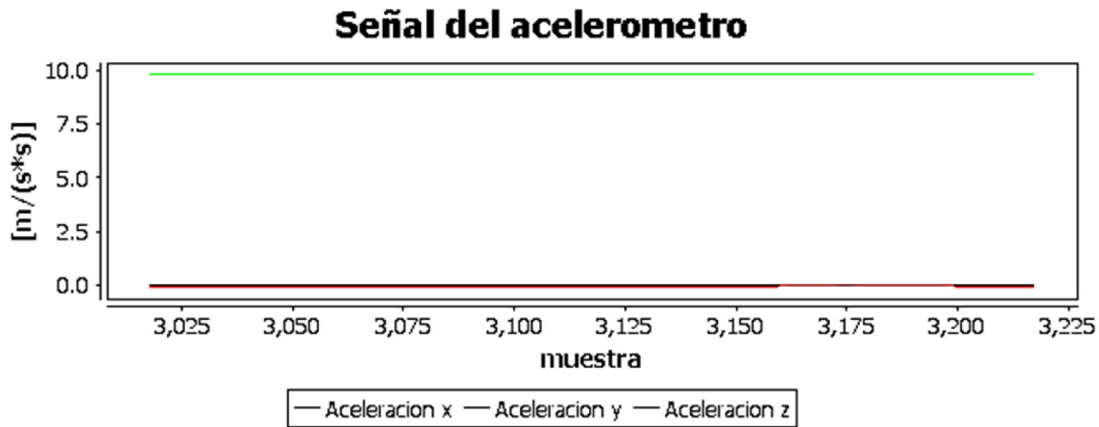


Figura 4.25. Gráfica de la señal del acelerómetro

Para comprobar que los datos que obtuvo el acelerómetro se calcula el módulo del vector de aceleración a partir de los datos obtenidos para este caso se retoman los valores de la columna 434 de la figura 4.22 por lo tanto:

$$\text{Eje X} = -41 \quad \text{Ecuación 4.1}$$

$$\text{Eje Y} = -2 \quad \text{Ecuación 4.2}$$

$$\text{Eje Z} = 5487 \quad \text{Ecuación 4.3}$$

$$a_x = \frac{(-41)6}{32767} 9.81 = -0.074 \left[\frac{m}{s^2} \right] \quad \text{Ecuación 4.4}$$

$$a_y = \frac{(-2)6}{32767} 9.81 = -0.004 \left[\frac{m}{s^2} \right] \quad \text{Ecuación 4.5}$$

$$a_z = \frac{(5487)6}{32767} 9.81 = 9.852 \left[\frac{m}{s^2} \right] \quad \text{Ecuación 4.6}$$

$$|\vec{a}| = \sqrt{a_x^2 + a_y^2 + a_z^2} = \sqrt{(-0.074)^2 + (-0.004)^2 + (9.852)^2} = 9.8566 \left[\frac{m}{s^2} \right] \quad \text{Ecuación 4.7}$$

El valor obtenido en la ecuación 4.7 es cercano a la aceleración en la Tierra que es de $9.81 \frac{m}{s^2}$ Al nivel del mar. El valor anterior indica un error en la calibración ya que la altura de la ciudad de México se estima en $2,421[m]$ por lo que la aceleración gravitatoria debería de ser menor y no mayor como fue el caso.

Por último se revisó el mensaje enviado por el sistema de adquisición este se muestra en la figura 4.26. Al colocar las coordenadas en la herramienta *google maps* como se muestra en la figura 2.27 se comprueba que la ubicación es correcta.

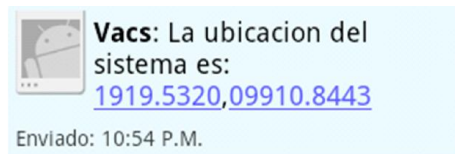


Figura 4.26. Mensaje enviado por el sistema de datos

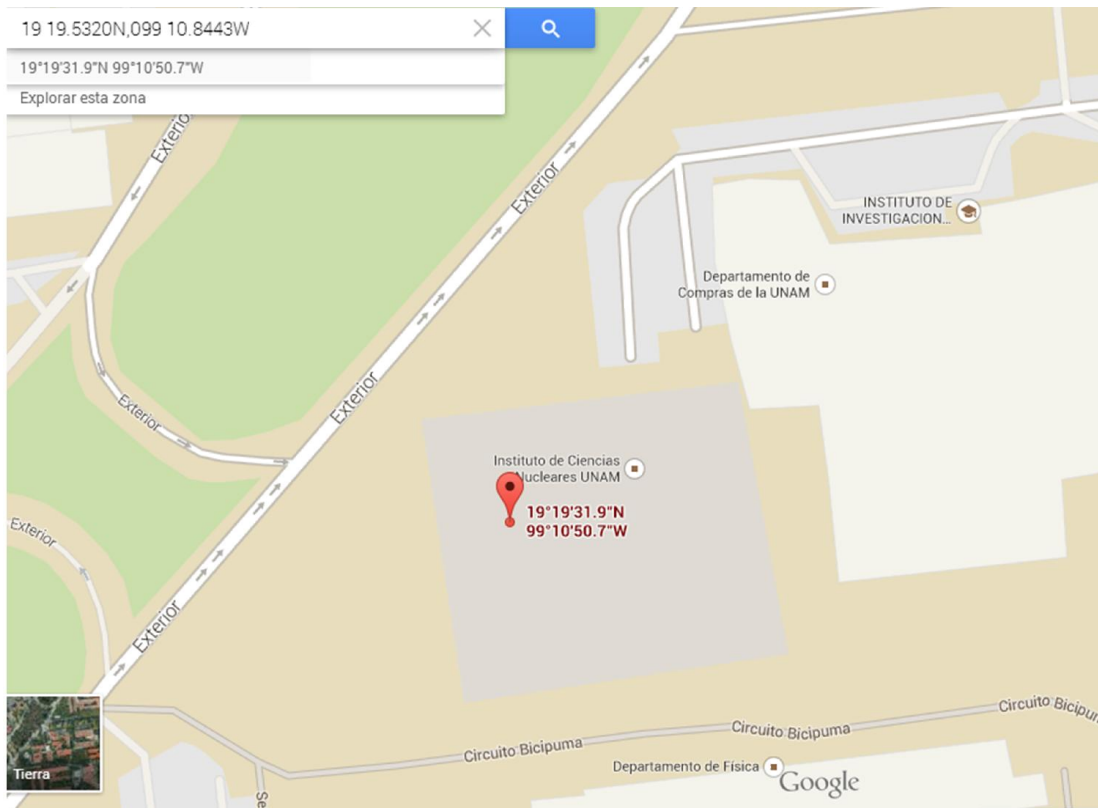


Figura 4.27. Ubicación del sistema de datos

5. Resultados y conclusiones

En este capítulo se abordarán los resultados obtenidos durante el desarrollo y diseño del sistema de adquisición de datos, se presentarán conclusiones y recomendaciones para mejorar el sistema de adquisición en versiones futuras.

5.1. Resultados

Recapitulando, el objetivo inicial de este trabajo fue el diseño de un sistema de adquisición, registro y envío de datos basados en un microcontrolador, para obtener señales del pulso radial, temperatura, aceleración y posición. Estos datos tendrían que ser almacenados en una memoria microSD y enviados por medio de un modem GSM.

Para lograr el objetivo se formularon varias metas logradas satisfactoriamente a través del desarrollo del sistema de adquisición de datos del pulso radial, dichas metas se enlistan a continuación:

- 1) Selección adecuada de los dispositivos de trabajo.
- 2) Diseño y fabricación de los circuitos impresos para el prototipo.
- 3) Desarrollo de software de control.
- 4) Incorporación de un sistema de archivos para el manejo de datos.
- 5) Conexión con servidor para el envío de datos vía GPRS.
- 6) Interfaz de visualización de datos.

Sin embargo, se tienen aspectos que deben mejorarse entre los que destacan:

- 1) El acondicionamiento de la señal de pulso. Si bien se han obtenido buenos comentarios por parte de quienes realizarán el análisis de la señal, ésta aún presenta una considerable cantidad de ruido, es importante destacar que hay filtros que ofrecen mejores resultados, como se explica en el Anexo I de este trabajo, que permitirán un estudio más preciso de la señal de pulso radial.
- 2) Diseño del circuito impreso del prototipo. Como quedó asentado en la sección 4.2.4 correspondiente a las pruebas realizadas al módulo de envío de datos, el circuito impreso debe ser modificado para agregar un par transistores en colector abierto para resolver de forma definitiva el problema de conexión entre el modem y el microcontrolador.
- 3) Consumo de datos. El no utilizar algoritmos de compresión de datos hace que el coste del envío de datos pueda resultar muy oneroso.
- 4) Almacenamiento de datos, el Formato ASCII nos permitió generar archivos que pueden ser operados no solo por la interfaz desarrollada sino también por otros programas como matlab u hojas de cálculo, sin embargo, el uso de éste incrementa la cantidad de memoria utilizada y para lograr la decodificación de binarios a ASCII es necesario realizar divisiones, la cual al no ser una operación propia del

microcontrolador le toma una mayor cantidad de tiempo comparada con el resto de las operaciones; con base en lo anterior, el guardar la información en un formato binario no sólo reduciría la cantidad de memoria utilizada sino que reduciría la cantidad de tiempo que tarda el microprocesador en procesar la información.

- 5) Utilizar parámetros del archivo. El utilizar los campos de fecha y hora de creación así como el de la última modificación del archivo, nos permitiría registrar el momento en el que se comenzaron a registrar los datos así como la duración de las pruebas.
- 6) Interfaz de usuario en el dispositivo. El no tener una interfaz integrada al sistema que nos permita ver la señal que se está adquiriendo dificultó la ubicación óptima de los sensores, por lo que incorporar una interfaz en el dispositivo simplificaría esta tarea además de permitir el despliegue de otros datos útiles para el usuario como la duración de la batería, etc.
- 7) Pruebas. Las pruebas realizadas en el presente trabajo se hicieron en un ambiente controlado, por lo que una vez corregidos los problemas que presenta este prototipo se deberían realizar pruebas en diferentes escenarios realizando actividades distintas.

5.2. Conclusiones

Con base en los resultados obtenidos, se puede concluir que se cumplió con el objetivo principal del proyecto. El sistema de adquisición de datos y movilidad, permitirá obtener nueva información que permitirá desarrollar tratamientos acordes al estilo de vida del paciente y con ello aumentar su calidad de vida; por ser un sistema de adquisición pequeño y más versátil comparado con otros sistemas de medición comerciales.

Se cumplió también con metas adicionales a las que originalmente se plantearon como lo son inclusión de un sistema de archivos, y el desarrollo de una interfaz de usuario. Originalmente no se consideraron estas metas, sin embargo conforme fue avanzando el desarrollo del proyecto se incluyeron para ofrecer al usuario una forma cómoda de manejar y almacenar la información.

Sin embargo, se tiene que mejorar algunos de los elementos del sistema como el acondicionamiento de la señal de pulso. También se tiene que mejorar la interfaz de usuario de forma que este pueda modificar algunos parámetros como la velocidad del muestreo o la posibilidad de poder observar la señal de pulso a través de una pequeña pantalla.

Finalmente y de manera personal, el desarrollo de este proyecto me dio la oportunidad de aplicar los conocimientos adquiridos como estudiante de la carrera de Ingeniería Eléctrica Electrónica. También me permitió integrarme a un equipo de trabajo multidisciplinario así como conocer procesos de fabricación industriales. Todo esto derivó en la realización de este sistema embebido que da solución a un problema real de la ingeniería, en este caso en el campo de la instrumentación médica.

5.3. Trabajo a futuro

Este trabajo es sólo un acercamiento al sistema de adquisición de datos, por lo que quienes se integren al equipo de trabajo deberán desarrollar versiones más robustas del sistema y para ello se deberán establecer condiciones de operación más definidas del sistema.

También se debe definir el rumbo que tomará el proyecto ya que de acuerdo a ello se debe de realizar una migración del dispositivo controlador; si se seguirá sólo con la adquisición de datos se debe migrar a un microcontrolador de ultra bajo consumo de energía; si se opta por seguir con el procesamiento de datos, deberá considerarse en cambiar a un microprocesador y con ello se podrían agregar más utilidades sin comprometer el objetivo principal.

5.4. Recomendaciones

Durante el desarrollo del sistema se hicieron pruebas, recomendaciones, sugerencias y observaciones que permitieron llevar el proyecto hasta este punto. Sin embargo existen muchos aspectos en los que se deben mejorar. Entre estos aspectos se encuentran:

Hardware

- Cambiar el sistema de acondicionamiento.
- Incluir comunicación USB. Además de facilitar el intercambio de datos podría utilizarse para agregarle más funciones al sistema, como establecer el tiempo de monitoreo, agregar datos personales del paciente o agregar propiedades a los sistemas de archivos.
- Incluir una pantalla. Esto mejoraría enormemente la comunicación entre el usuario y el dispositivo, así también como su uso. El incluir una pantalla al sistema nos permitiría observar la señal que estamos adquiriendo en tiempo real y con ello determinar si el lugar en el que se encuentra el sensor es el adecuado o si requiere ser acomodado y permitiría el envío de alertas de una mejor forma.
- Mejorar sensor. Se propone crear una especie de red de sensores en la correa del dispositivo y con ayuda de un algoritmo de reconocimiento de señales poder hacer un barrido en los sensores y así seleccionar aquel que nos proporcione la mejor señal.
- Explorar opciones para enviar datos. Se debe considerar utilizar otras tecnologías como WI-FI, también podría considerarse la posibilidad de enviar la información por medio del teléfono celular del usuario.

Software

- Implementar algoritmos de compresión para disminuir la cantidad de datos que serán enviados y/o almacenados.
- Optimizar las bibliotecas y las funciones creadas para lograr un mejor desempeño del microprocesador.

- Guardar los datos en la microSD en forma hexadecimal, para reducir la cantidad de información que se almacena.
- Incluir los parámetros fecha y hora en las propiedades de los archivos.

ANEXOS

A. Anexo I

Para mejorar el sistema de acondicionamiento de señales se han diseñado nuevos filtros activos; con ello se pretende obtener una mejor señal del pulso radial.

Como se muestra en la figura A.1 existe una gran cantidad de ruido en la señal de pulso radial, por lo que se han realizado pruebas con otro tipo de filtros para disminuir la cantidad de ruido. Estas pruebas se reportan a continuación.

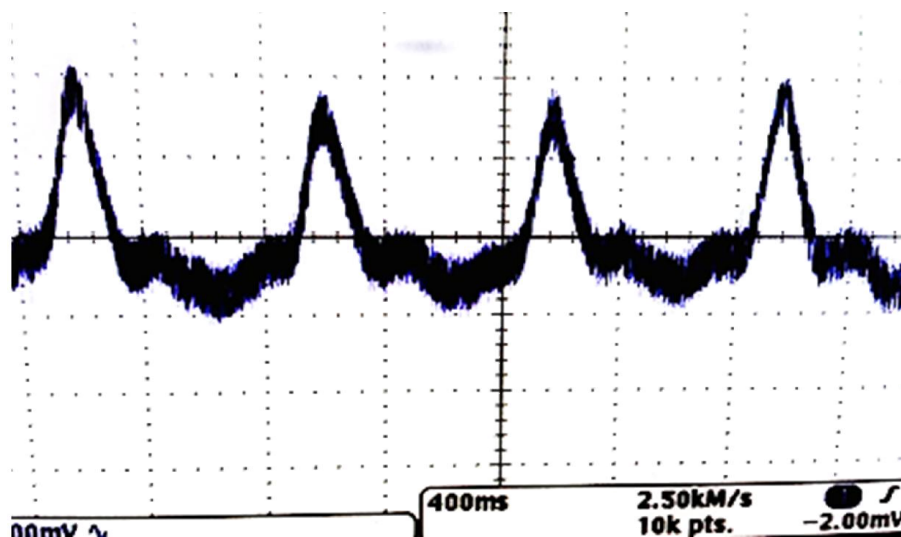


Figura A.1. Señal obtenida con el filtro del sistema.

Diseño de filtros

Se han considerado tres tipos de filtros: Bessel, Butterworth y Chebyshev, sin embargo, el filtro de Chebyshev involucra tener un rizo en la banda de paso o en la banda de corte lo cual podría perturbar la morfología de la señal, razón por la cual se descarta su implementación. Por lo que se implementaron filtros de Bessel y Butterworth para ello se utilizó la topología de Sallen Key.

El cálculo de los valores de las resistencias y capacitores para realizar los filtros se realizó acorde con el apartado 2.2.2 el cual corresponde a la parte de filtros.

A continuación se mostrarán las simulaciones y las señales obtenidas por la implementación de los filtros con los respectivos valores de sus componentes.

Filtro paso bajas de Bessel

En la tabla A.1 se muestran los valores calculados de las resistencias para la implementación del filtro Sallen-Key.

F_{3dB} [Hz]	$f_{(3dB)}/f_c$	F_c [Hz]	C [μF]	α	R [Ω]	R_1 [Ω]	R_2 [Ω]	R_A [Ω]	R_B [Ω]
30	0.785	23.55	0.1	1.732	67,582	67,582	67,582	67,582	39,603
250	0.785	196.3	0.1	1.732	8,110	8,110	8,110	8,110	4,752

Tabla A.1. Cálculos para un filtro Bessel paso bajas.

Se ajustan los valores de resistencias comerciales los cuales se muestran en la tabla A.2.

F_{3dB} [Hz]	C [μF]	R_1 [$k\Omega$]	R_2 [$k\Omega$]	R_A [$k\Omega$]	R_B [$k\Omega$]
30	0.1	68	68	68	39
250	0.1	8.1	8.1	8.1	4.7

Tabla A.2. Componentes ajustados a valores comerciales.

Filtro paso bajas de Bessel a 30 [Hz]

En la figura A.2 se muestra la simulación obtenida del circuito calculado en donde se puede apreciar que la magnitud en la banda de paso es plana, pero no así su fase en la cual puede apreciarse que existe un retardo en la señal, el cual aumenta con la frecuencia y hasta llegar a los 180°.

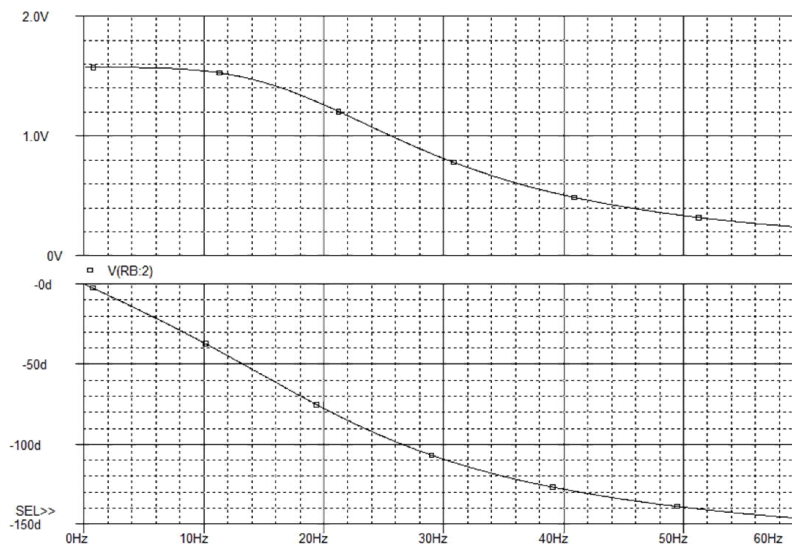


Figura A.2. Simulación Filtro paso bajas Bessel a 30[Hz].

En la figura A.3 se muestra la señal obtenida a través del osciloscopio en el cual se observa claramente que el filtro además de quitar el ruido presente en la señal, la amplifica; se puede observar que la señal resultante es la envolvente de la señal de entrada la cual aún conserva su forma, esto puede ser debido a que las componentes espectrales de la señal de entrada son muy cercanas entre sí lo que ocasiona un pequeño desfase.

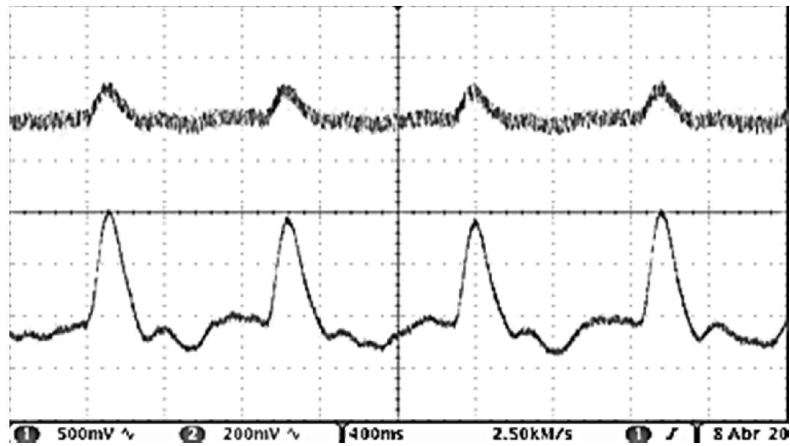


Figura A.3. Implementación filtro paso bajas Bessel a 30[Hz], la señal superior representa la señal de entrada al filtro mientras que la señal inferior la señal de salida.

Simulación Filtro 250 [Hz]

En la figura A.4 se muestra la simulación del circuito calculado en donde se puede observar que este presenta que la magnitud en la banda de paso es plana hasta casi los 40[Hz]. Sin embargo existe un atraso en la fase de la señal muy pronunciado esto repercutirá probablemente en la morfología de la señal.



Figura A.4. Simulación Filtro paso bajas Bessel a 250[Hz].

En la figura A.5 se puede observar que la señal de salida (la señal inferior) se muestra cambios muy abruptos en la señal sobre todo en el intervalo interpulso (comprende el intervalo de la señal entre los lóbulos de mayor amplitud).

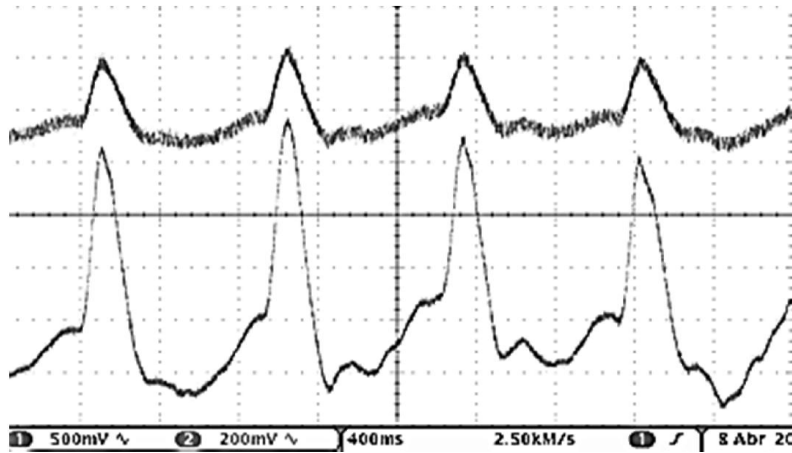


Figura A.5. Implementación Filtro paso bajas Bessel a 250[Hz] , la señal superior representa la señal de entrada al filtro mientras que la señal inferior la señal de salida.

Filtro paso bajas Butterworth

En la tabla A.3 se muestran los valores calculados de las resistencias con ayuda de las ecuaciones del filtro Sallen-Key descritas en el apartado 2.2.

F_{3dB} [Hz]	$f_{(3dB)}/f_c$	F_c [Hz]	C [μF]	α	R [Ω]	R_1 [Ω]	R_2 [Ω]	R_A [Ω]	R_B [Ω]
20	1	20	0.1	1.414	79,577	79,577	79,577	10,000	5,860
250	1	250	0.1	1.414	6,366	6,366	6,366	10,000	5,860

Tabla A.3. Cálculos para un filtro Bessel paso bajas.

Se procede a ajustar los valores de resistencias comerciales los cuales se muestran en la tabla A.4.

F_{3dB} [Hz]	C [μF]	R_1 [k Ω]	R_2 [k Ω]	R_A [k Ω]	R_B [k Ω]
20	0.1	82	82	10	5.6
250	0.1	6.8	6.8	10	5.6

Tabla A.4. Componentes ajustados a valores comerciales.

Filtro paso bajas 20[Hz]

La figura A.6 se muestra la simulación del filtro calculado en el se observan que existe que la ganancia máxima se obtendrá en frecuencias cercanas a los 20[Hz] También se se puede apreciar que su respuesta en frecuencia sobre todo en bajas frecuencias es plana, sin embargo, en frecuencias cercanas a la de corte la señal la señal existe un atraso en la fase muy pronunciado.

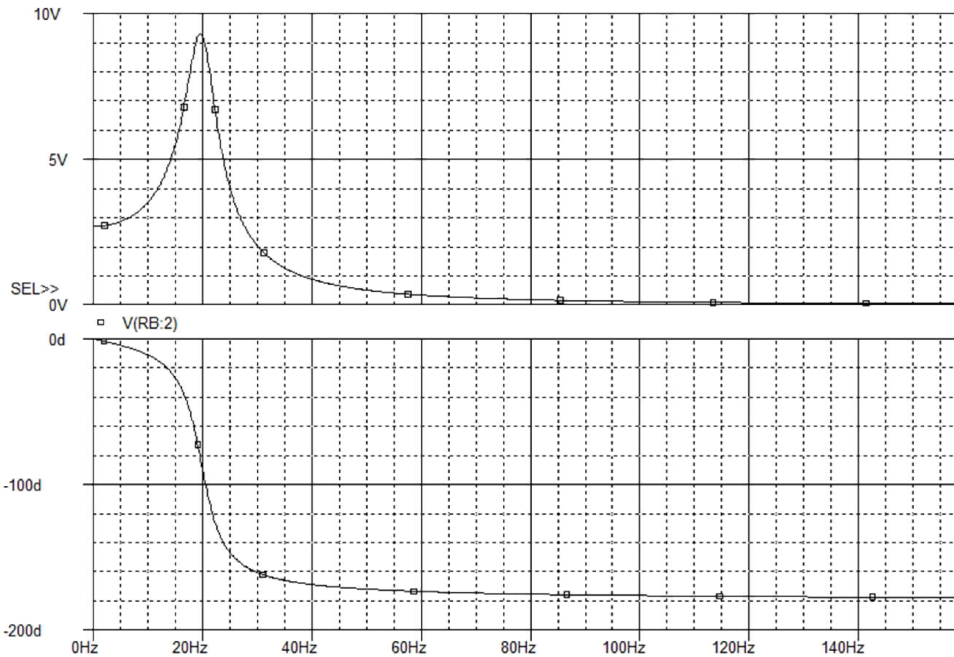


Figura A.6. Simulación filtro paso bajas Butterworth a 20[Hz].

En la figura A.7 se muestra las señales de entrada y salida al filtro en ella se puede apreciar que si bien no elimina completamente el ruido en la señal si reduce la cantidad de ruido y mantiene la forma de una señal de pletismografía.

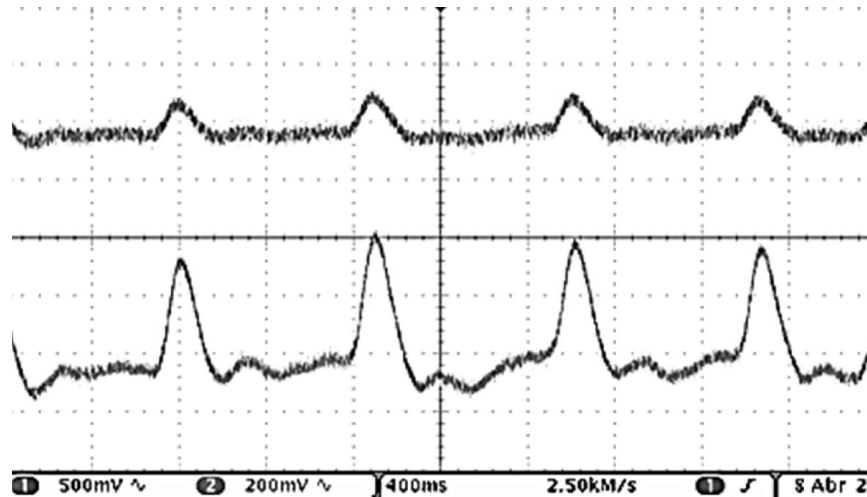


Figura A.7. Implementación filtro paso bajas Butterworth a 20[Hz] , la señal superior representa la señal de entrada al filtro mientras que la señal inferior la señal de salida.

Filtro paso bajas 250[Hz]

En la figura 4.8 se muestra la simulación del circuito calculado en ella se observa que la máxima ganancia del circuito se encuentra cerca de los 220[Hz] Y que en bajas frecuencias se puede considerar que tenemos una respuesta plana.

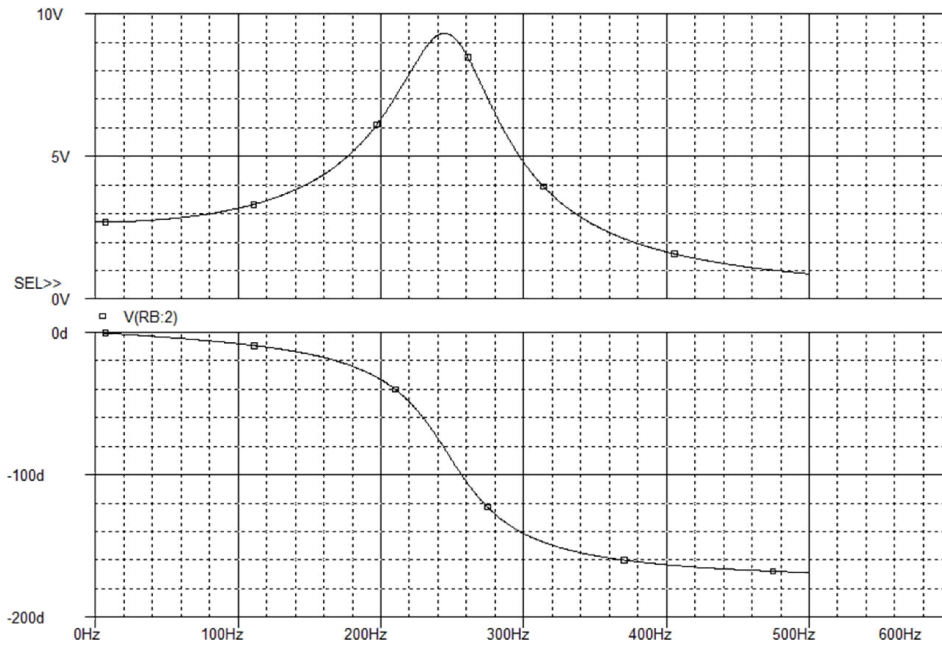


Figura A.8. Simulación filtro paso bajas Butterworth a 250[Hz].

En la figura 4.9 se muestra las señales obtenidas a través del osciloscopio en ella se puede observar que el filtro reduce de forma significativa la cantidad de ruido en la señal y mantiene la forma de onda salvo que a diferencia de los anteriores tres este es que menor ganancia presenta.

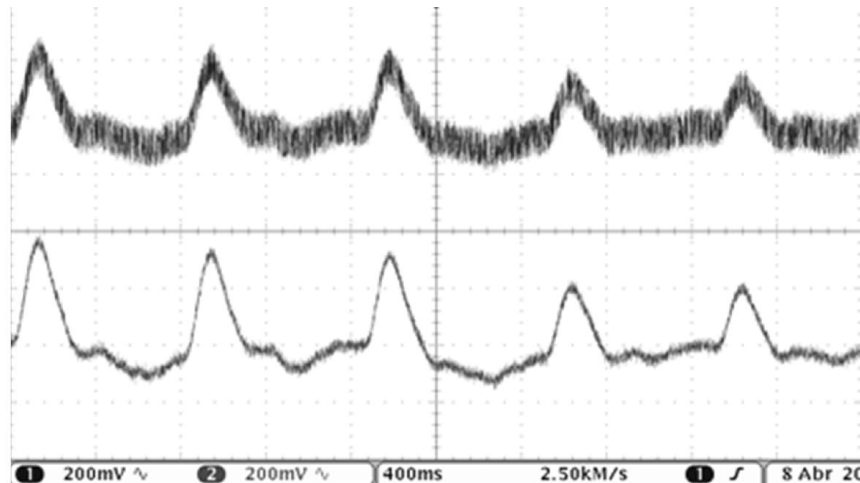


Figura A.9. Implementación filtro paso bajas Butterworth a 250[Hz] , la señal superior representa la señal de entrada al filtro mientras que la señal inferior la señal de salida.

Conclusiones

Los cuatro filtros propuestos cumplen de forma adecuada su tarea, sin embargo, en el filtro de Bessel con frecuencia de corte de 250[Hz] se puede apreciar que la señal es muy afilada por lo que sería el menos recomendable de los 4. Los filtros Butterworth presentan ruido pero conservan la morfología de la señal.

Los más adecuados para ser implementados son el filtro Bessel con frecuencia de corte a 30[Hz] y el filtro Butterworth tipo 1 a 250[Hz].

Como prueba determinante se recomienda construir todos los filtros a excepción del filtro de Bessel con frecuencia de corte de 250[Hz] e implementarlos directamente en el brazalet. Considerando que esto no repercutiría en el PCB final ya que todos los filtros tienen la misma topología solo habría que cambiar el valor de los componentes que constituyen el filtro.

Bibliografía y referencias

Bibliografía

- Boylestad Robert . *Electrónica: Teoría de circuitos y dispositivos electrónicos*, Decima edición, Prentice Hall, México, 2009
- Coughlin Robert F, *Amplificadores operacionales y circuitos integrados lineales*, Prentice Hall, México, 1999
- Microsoft Corporation, *Microsoft Extensible Firmware Initiative FAT32 File System Specification*, United State of America, 2000
- Pallas Areny Roberto, *Sensores y acondicionadores de señal*, Cuarta edición, Marcombo, 2003.
- Tobey, Gene E. et al., **Operational Amplifiers Design and applications**, McGRAW-HILL, United State of America, 1971

Referencias

- [1] Organización mundial de la salud, “Enfermedades cardiovasculares” [en línea] <<http://www.who.int/mediacentre/factsheets/fs317/es/>> [Consulta: 19 de marzo de 2014]
- [2] Organización mundial de la salud, “Caídas” [en línea] <<http://www.who.int/mediacentre/factsheets/fs344/es/>> [Consulta: 19 de septiembre de 2012]
- [3] Organización Panamericana de la salud, “Obesidad” [en línea] <http://www.paho.org/mex/index.php?Option=com_content&view=category§ionid=96&id=824&Itemid=338> [consulta 17 de marzo de 2014]
- [4] Mapfre, canal tu salud, “Pérdida de equilibrio y caídas de personas mayores. Consecuencias y complicaciones” [en línea] <<http://www.mapfre.es/salud/es/cinformativo/consecuencias-caidas-ancianos.shtml>> [Consulta 27 de septiembre 2012]
- [5] MATERSYS GROUP (2011), “*Plataforma de Vigilancia y Análisis Continuo de signos Vitales*”, México.
- [6] Medline Plus Información para usted, servicio de la biblioteca nacional de medicina de los E.E.U.U, “Pulso” [en línea] <<http://www.nlm.nih.gov/medlineplus/spanish/ency/article/003399.htm>> [Consulta 19 de marzo de 2014]

[7] Pallas Areny Ramón, “*Sensores y acondicionadores de señal*”, cuarta edición, Marcombo S.A., España, 2003.

[8] Baird D.C, “Experimentación. Una introducción a la teoría de mediciones y al diseño de experimentos”, segunda edición, Prentice-Hall Hispanoamericana S.A., México, 1991

Mesografía

Faculta de Medicina UNAM, “Prevención y Atención de las Caídas en la Persona Adulta Mayor” [en línea]

<http://www.facmed.unam.mx/deptos/salud/censenanza/spivsa/antol%20%20anciano/2parte2013/V_caidas.pdf> [Consulta 27 de septiembre 2012]

Oxford Journals, “Analysis of Temporal Requirements for Myocardial Tissue Velocity Imaging” [en línea] <<http://ehjcmimaging.oxfordjournals.org/content/3/3/214.full.pdf+html>> [Consulta 19 de marzo de 2014]

Faculta de Contaduría y Administración UNAM, “Apuntes de programación orientada a objetos” [en línea] <<http://fcasua.contad.unam.mx/apuntes/interiores/docs/98/opt/java.pdf>> [Consulta 19 de febrero de 2014]

Mikroelektronika, “El mundo de los microcontroladores” [en línea] <<http://www.mikroe.com/chapters/view/79/capitulo-1-el-mundo-de-los-microcontroladores>> [Consulta 10 Octubre 2012]

División de ciencias Básicas Facultad de Ingeniería UNAM. “Conceptos de Algoritmo y programa” [en línea] <http://dcb.fi-c.unam.mx/users/miguelegc/tutoriales/tutorialcd/cncpt_lgtm_cmpt_4.htm> [Consulta 1 marzo 2014]

División de ciencias Básicas Facultad de Ingeniería UNAM. “Conceptos básicos de programación estructurada” [en línea] <http://dcb.fi-c.unam.mx/users/miguelegc/tutoriales/tutorialcd/cncpts_bscs_prgrmcn_srctrd_1.htm> [Consulta 1 de marzo 2014]

Mobile Manufactures Forum, [En línea] <<http://www.mmfai.org/public/>> [Consulta 17 de febrero 2014]

Revista UNAM, “Sistemas personales de comunicación” <<http://www.revista.unam.mx/vol.2/num2/art3/#1>> [Consulta 10 de marzo 2014]

Elm-chan.org, “How to use MMC/SDC” [en línea] <http://elm-chan.org/docs/mmc/mmc_e.html> [Consulta 16 junio 2013]

Texas Instruments, “Datasheet LM3S8962” [En línea] <<http://www.ti.com/lit/ds/spms001g/spms001g.pdf>> [Consulta 10 enero 2013]

SD Association, “SD Specification Part 1 Physical Layer” [En línea]
<https://www.sdcard.org/downloads/pls/simplified_specs/part1_410.pdf> [Consulta 19 abril 2013]

OLIMEX, “Enfora Enabler IIIG AT Command Reference” [En línea]
<http://www.olimex.cl/pdf/Wireless/GSM0308AT001-Enabler_IIIG_AT_Commands.pdf> [Consulta 7 noviembre 2013]