



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DIGITALIZADOR DE SEÑALES SÍSMICAS
SINCRONIZADO CON UN GPS Y ALMACENAMIENTO
DE DATOS EN UNA SD CARD

T E S I S

QUE PARA OBTENER EL TÍTULO DE :

INGENIERO ELÉCTRICO ELECTRÓNICO

(Á R E A : E L E C T R Ó N I C A)

PRESENTAN:

**Cuchillo Sánchez Héctor
Preciado Linares Erick Antonio**



DIRECTOR DE TESIS: M. I. LAURO SANTIAGO CRUZ

MÉXICO, D.F.

2013

Agradecimientos y Dedicatorias

Agradecimientos (AGS)

A MIS PADRES

*Natividad Cuchillo Corona
Eredina Sánchez Flores*

A mi padre por solventar todos los gastos que implica una formación académica, por brindarme integralmente su apoyo y confianza en todos los ámbitos posibles, aún más importante, por ser el ejemplo para mi persona y enseñarme que con esfuerzo y perseverancia podemos alcanzar nuestras metas.

A mi madre por brindarme su cariño y comprensión, por sus valiosos consejos y enseñanzas, por mostrarme el lado bueno de las cosas. Por todo ello y más, siempre será una motivación para ser una mejor persona día con día.

A MIS HERMANAS

Magdalena, Sandra, Yolanda

Les agradezco su apoyo brindado en todos los momentos difíciles y por compartir todos aquellos momentos de llenos alegría.

A la Universidad Nacional Autónoma de México y al Instituto de Ingeniería de la UNAM, por acogerme dentro de sus instalaciones y brindarme las facilidades durante el desarrollo de esta tesis.

A mi director de tesis por su asesoría en la elaboración de este trabajo.

Agradecimientos y Dedicatorias

Agradecimientos (E.A.D.)

Al Ser Omnipotente, por darme sabiduría, templanza, fortaleza y confianza a lo largo de este camino.

Al la Facultad de Ingeniería, y a todos y cada uno de mis profesores que me formaron a lo largo de esta bella carrera.

Al Instituto de Ingeniería, por darme todas las herramientas para el desarrollo de este proyecto.

Al mi Maestro el M.I. Mauro Santiago Cruz, por haberme dado la oportunidad de formar parte de su equipo de trabajo, y por todo el apoyo brindado a lo largo de todo este tiempo.

Al Ing. Horacio Mijares, por el apoyo brindado y por sus aportaciones para que el D.T.G.I. llegará a su etapa final.

Al mis compañeros de laboratorio, porque sin su compañía el laboratorio habría sido frío y aburrido:

Alonso Valera

Fabio Ruiz

Miguel Ibáñez

Al mi compañero de tesis Héctor Cuchillo, por que sacamos adelante este proyecto.

Al mí, por el esfuerzo, la pasión y la paciencia invertida para culminar este trabajo.

Agradecimientos y Dedicatorias

Dedicatorias (EAPL)

*A lo largo de todo este tiempo, he comprendido que la vida es un juego arduo pero muy apasionante, y que para poder jugarlo no es necesario seguir reglas tan complicadas, ni mucho menos, desarrollar estrategias o algoritmos para encontrarle el sentido a la vida. Así mismo, he entendido que cualquier problema se convierte en una oportunidad, cuando tienes a tu lado a un ser que te da la mano en esos momentos tan difíciles en que crees que todo está perdido. En mi caso, no tengo como agradecer todo el apoyo que me has dado, las fuerzas que me has inyectado, el coraje, la dedicación, tu amistad y sobre todo el amor tan sincero que me has brindado, y que hoy este es el resultado. Me refiero a ti **Ingrid Lizette Abreo Inguanzo**, que estuviste conmigo en todas y cada una de las batallas, desde mi examen de admisión a la **UAM** hasta este momento, gracias por esas desveladas en que me apoyaste para terminar mis trabajos, por esas clases que tomaste cuando tenía que entregar proyectos, por mis horarios que armabas cada inicio de semestre, por ayudarme a inscribir y sentir conmigo el nervio por no encontrar a todos mis maestros, por esos mensajes de texto antes de un examen, por ayudarme a revisar mis "horrores de ortografía" en este trabajo, por hacerme entrar en razón en muchas circunstancias. Simplemente **gracias** por ser **mi pilar y mi brazo derecho** en esta larga, pero tan hermosa carrera de Ingeniería.*

Agradecimientos y Dedicatorias

La fortaleza y confianza que en mí tengo se la debó a mi Papá Juan Antonio Preciado Valsierra, que a lo largo de todo este tiempo siempre me ha apoyado y ha confiado en mí para salir delante de cualquier problema. Gracias por haberme dado la oportunidad de estudiar una carrera profesional, y sobre todo por compartirme sus conocimientos cuando más lo he necesitado. Gracias por hacerme ver desde pequeño, que todo es posible siempre y cuando uno confíe en sí mismo, y que en la vida no hay obstáculos que no se pueda conquistar.

El amor, paciencia y dedicación que tengo, se la debo a mi Mamá Martha Elena Linares Benítez, Gracias "May", por haberme apoyado a lo largo de esta carrera, porque a pesar de que ya estoy grande, siempre me tuviste un desayuno caliente, para no ir atarugado a la Universidad. Gracias por la dedicación que me diste desde niño, que hoy hacen de mí un ser de provecho.

A mi hermana Gisela Abril Preciado Linares, para que esto sea para ti una muestra de que si quieres puedes, y que nada es difícil, siempre y cuando, te esfuerces siempre un poco más.

A la familia de Ingrid Lizette, por haberme acogido como uno más de sus miembros y apoyarme en todo momento... gracias! Pero en especial a Valeria Rivas, por tener esa chispa y recordarme que nunca hay que perder esa alma de niño. Y porque en un futuro sigas el camino del estudio, y seas una mujer de bien.

Contenido

Índice de figuras	IX
Índice de tablas	XIII
Prólogo	XIV
Capítulo I. Introducción	1
1.1. La Sismología	1
1.1.1. Actividad sísmica en México	3
1.2. Tipos de ondas sísmicas	5
1.3. Equipos de medición sísmica	7
1.4. Planteamiento del problema y propuesta de solución	7
Capítulo II. Generalidades	9
2.1. Tipos de sensores sísmicos	9
2.2. Acondicionamiento de señales de entrada.....	18
2.2.1. Tipos de señales.....	18
2.2.2. Operaciones con señales.....	22
2.2.3. Interfaz con los sensores analógicos.....	23
2.2.4. Amplificación de señales.....	23
2.3. Conversión analógica digital	25
2.3.1. Parámetros generales de un convertidor analógico digital	26
2.3.2. Aliasing.....	28
2.3.3. Convertidor analógico digital Sigma – Delta	30
2.4. Memorias de estado sólido	40
2.4.1. Especificaciones de las memorias flash SD Card.....	41
2.4.2. Topologías de comunicación de las memorias SD.....	43
2.4.2.1.Comunicación SD.....	43
2.4.2.2.Comunicación SPI.....	44
2.5. Protocolo de comunicación SPI en la memoria SD Card.....	45
2.5.1. Inicialización en modo SPI.....	49
2.5.2. Operaciones de lectura.....	51
2.5.3. Operaciones de escritura.....	52
2.6. Sistemas de archivos FAT	53
2.6.1. FAT16 y FAT32.....	57

2.7. Referencia de tiempo.....	58
2.7.1. Sistema de posicionamiento global (GPS).....	59
2.7.2. Tipos de receptores GPS.....	65
2.8. Microcontrolador.....	67
2.8.1. Arquitectura Von Newmann.....	70
2.8.2. Arquitectura Harvard.....	71
2.8.3. Familia de microcontroladores.....	72
2.9. Dispositivos periféricos.....	73
2.9.1. Joystick.....	74
2.9.2. Pantalla gráfica de cristal líquido.....	76
2.9.3. Comunicación serial RS-232.....	76
Capítulo III. Diseño y desarrollo del sistema “DIGI”.....	79
3.1. Definición de requerimientos del sistema.....	79
3.2. Desarrollo del Hardware.....	80
3.2.1. Geófono GS – 11D.....	81
3.2.2. Microcontrolador AVR.....	83
3.2.3. Módulo de acondicionamiento de señal.....	85
3.2.4. Módulo para la digitalización de los datos.....	99
3.2.5. Módulo para el almacenamiento de datos.....	102
3.2.6. Interfaz hombre – máquina.....	103
3.2.7. Módulo para la adquisición de datos del GPS.....	107
3.2.8. Integración del Hardware.....	109
3.2.9. PCB del sistema.....	111
3.3. Desarrollo del Software.....	113
3.3.1. Estructura del programa del microcontrolador.....	115
3.3.2. Rutina para la adquisición y conversión de datos.....	117
3.3.3. Rutina para la adquisición de fecha y hora del GPS.....	119
3.3.4. Almacenamiento de datos.....	121
3.3.5. Manejo del joystick.....	128
3.3.6. Manejo del GLCD.....	129
3.3.7. Módulo de comparación y captura.....	135
3.4. Procesamiento de datos.....	139
3.4.1. Entorno de desarrollo de LabVIEW.....	139
3.4.2. Generación de la estampa de tiempo.....	141
3.4.3. Colocación de la estampa en el archivo de datos.....	144
Capítulo IV. Diseño y desarrollo del sistema “DIGI”.....	150
4.1. Integración del sistema en PCB.....	150
4.2. Pruebas realizadas al módulo de acondicionamiento de señal.....	155
4.3. Pruebas realizadas al módulo de adquisición de datos del GPS.....	159

4.4. Pruebas de adquisición y almacenamiento de datos	162
4.5. Pruebas para verificar la estampa de tiempo	164
4.6. Pruebas de reconstrucción de datos y señales	166
Capítulo V. Resultados y conclusiones	174
5.1. Resultados	174
5.2. Conclusiones.....	175
5.3. Recomendaciones	175
Apéndice A: Glosario de términos	178
Bibliografía	180

Índice de Figuras

Capítulo I

Figura 1.1. Principales capas de la Tierra.....	1
Figura 1.2. Epicentro de un terremoto	3
Figura 1.3. Localización de las placas tectónicas	4
Figura 1.4. Ondas P	5
Figura 1.5. Ondas S	6
Figura 1.6. Ondas Rayleigh	6
Figura 1.7. Ondas de Love	7

Capítulo II

Figura 2.1. Rango de frecuencias sísmicas.....	9
Figura 2.2. Sensor mecánico vertical.....	11
Figura 2.3. Respuesta en frecuencia de un sensor mecánico.....	12
Figura 2.4. Sensor electromagnético	13
Figura 2.5. Acelerómetro de fuerza balanceada	14
Figura 2.6. Respuesta en frecuencia de un FBA.....	15
Figura 2.7. Sensor de velocidad de banda ancha.....	16
Figura 2.8. Respuesta en frecuencia de un sensor de velocidad de banda ancha	16
Figura 2.9. Circuito para extender la respuesta de un sensor electromagnético.....	17
Figura 2.10. Señales unipolares.....	19
Figura 2.11. Señales bipolares.....	19
Figura 2.12. Medición de tensión en Z_m	20
Figura 2.13. Medición de corriente en Z_m	20
Figura 2.14. Conexión entre circuito A y una resistencia R_m	21
Figura 2.15. Configuración de amplificador de instrumentación.....	24
Figura 2.16. Diagrama de bloques de un convertidor A/D.....	25
Figura 2.17. Señal sinusoidal muestreada por arriba de la frecuencia de Nyquist.....	28
Figura 2.18. Señal sinusoidal muestreada por debajo de la frecuencia de Nyquist.....	28
Figura 2.19. Espectro limitado en banda	29
Figura 2.20. Señal muestreada sin aliasing.....	29
Figura 2.21. Señal muestreada con aliasing	30
Figura 2.22. Comportamiento de filtro antialiasing	30
Figura 2.23. Diagrama de bloques de un convertidor A/D Sigma – Delta	31
Figura 2.24. Espectro de señal muestreada con el criterio de Nyquist.....	32
Figura 2.25. Espectro de una señal sobremuestreada	32
Figura 2.26. Modulación Delta.....	33
Figura 2.27. Modulador y demodulador delta	33
Figura 2.28. Integrador colocado antes del modulador	34

Figura 2.29. Diagrama de bloques de modulador Sigma – Delta	34
Figura 2.30. Modulador Sigma – Delta en el dominio de Laplace.....	35
Figura 2.31. Modulador Sigma – Delta de tercer orden	36
Figura 2.32. Efecto del modulador Sigma – Delta a la banda base fb.....	37
Figura 2.33. Diagrama de bloques de un convertidor A/D no ideal	39
Figura 2.34. Distribución de terminales de la memoria micro SD	43
Figura 2.35. Esquema del bus SD	43
Figura 2.36. Esquema del bus SPI.....	44
Figura 2.37. Algoritmo para la inicialización de una memoria SD en modo SPI	50
Figura 2.38. Operación de lectura de un solo bloque de datos.....	52
Figura 2.39. Operación de lectura de múltiples bloques de datos	52
Figura 2.40. Operación de escritura de un solo bloque	53
Figura 2.41. Operación de escritura de múltiples bloques de datos	53
Figura 2.42. Organización de sectores y clústers	54
Figura 2.43. Estructura de los sistemas de archivos FAT16 y FAT32.....	55
Figura 2.44. Estructura de las entradas en el directorio raíz.....	57
Figura 2.45. Constelación de satélites NAVSTAR	60
Figura 2.46. Sistemas que integran el GPS	61
Figura 2.47. Principio de funcionamiento del GPS	62
Figura 2.48. Pseudorange de tiempo GPS.....	63
Figura 2.49. Arquitectura Von Neumann	71
Figura 2.50. Arquitectura Harvard	71
Figura 2.51. Imágenes de algunos modelos de Joystick.....	74
Figura 2.52. Comportamiento ideal de la señal de una tecla o botón.....	74
Figura 2.53. Efecto de rebote mecánico al presionar una tecla o botón.....	75
Figura 2.54. Uso de rutinas de retardo para eliminar el efecto de rebote.....	75
Figura 2.55. Pantalla gráfica de cristal líquido (GLCD)	76
Figura 2.56. Conectores DB9 y DB25.....	77

Capítulo III

Figura 3.1. Elementos que forman el hardware del sistema.....	80
Figura 3.2. Estructura del geófono GS – 11D	81
Figura 3.3. Curva de la respuesta del geófono GS – 11D	82
Figura 3.4. Distribución de las terminales del microcontrolador	84
Figura 3.5. Circuito para agregar nivel de offset al geófono.....	87
Figura 3.6. Amplificador de instrumentación.....	88
Figura 3.7. Amplificador recomendado en hoja de especificaciones	89
Figura 3.8. Diagrama de Bode de la función de transferencia 3.7 al variar R2.....	91
Figura 3.9. Filtro RC de primer orden	92
Figura 3.10. Circuito de acondicionamiento de señal para geófonos	93
Figura 3.11. Circuito que modifica la respuesta en frecuencia de un FPB.....	94
Figura 3.12. Respuesta del filtro paso bajas	95
Figura 3.13. Circuito de acondicionamiento de señal para sensores activos.....	96
Figura 3.14. Diodos limitadores de nivel	97
Figura 3.15. Conexión de filtros mediante sensores.....	98

Figura 3.16. Circuito de acondicionamiento de señal	99
Figura 3.17. Distribución de terminales de un convertidor A/D	100
Figura 3.18. Diagrama de conexión entre el microcontrolador y los CAD	101
Figura 3.19. Distribución de las terminales de la memoria micro SD.....	102
Figura 3.20. Conexión entre el microcontrolador y la memoria micro SD.....	103
Figura 3.21. Diagrama de bloques del display gráfico	104
Figura 3.22. Apariencia física del display gráfico.....	104
Figura 3.23. Circuito para ajustar el contraste del display	106
Figura 3.24. Diagrama de conexión entre el joystick y el microcontrolador	106
Figura 3.25. Conexión entre el GPS y el microcontrolador	108
Figura 3.26. Receptor GPS.....	109
Figura 3.27. Hardware del sistema montado en tarjetas protoboard	109
Figura 3.28. Diagrama eléctrico del sistema “DIGI”	110
Figura 3.29. Capa superior del circuito impreso	112
Figura 3.30. Capa inferior del circuito impreso.....	113
Figura 3.31. Entorno de desarrollo de Arduino	114
Figura 3.32. Verificación de puerto y placa en la IDE de Arduino.....	115
Figura 3.33. Trama de bits de salida del convertidor A/D	117
Figura 3.34. Diagrama de flujo para digitalización de datos.....	118
Figura 3.35. Mensaje RMC del protocolo NMEA 0183	119
Figura 3.36. Diagrama de flujo para obtener los datos de hora y fecha del GPS.....	121
Figura 3.37. Estructura de datos almacenados en un archivo.....	122
Figura 3.38. Formato de los datos requeridos por el usuario	124
Figura 3.39. Código para dar formato a los datos de un canal	125
Figura 3.40. Diagrama de flujo del bloque de escritura de datos del archivo	127
Figura 3.41. Diagrama de flujo para el manejo del joystick.....	129
Figura 3.42. Diagrama de flujo de los menús mostrados en el display gráfico.....	130
Figura 3.43. Logotipo del Instituto de Ingeniería.....	131
Figura 3.44. Menú principal	131
Figura 3.45. Menú de selección de frecuencia de muestreo.....	132
Figura 3.46. Menú opciones en el modo de prueba.....	132
Figura 3.47. Gráfica de un canal.....	133
Figura 3.48. Estampa de tiempo obtenida del GPS	133
Figura 3.49. Encabezado de archivos	134
Figura 3.50. Selección entre días y horas	134
Figura 3.51. Selección del periodo de prueba	135
Figura 3.52. Diagrama de bloques del módulo de comparación y captura	135
Figura 3.53. Registro para configurar el módulo de comparación y captura	136
Figura 3.54. Configuración de los bits WGMn3:0 para entrar en modo CTC	137
Figura 3.55. Configuración de los bits CSnn2:0 para preescalar la fosc/64.....	137
Figura 3.56. Diagrama de flujo del módulo de comparación y captura “CTC”	138
Figura 3.57. Distribución del contenido del archivo	141
Figura 3.58. Generación de la estampa de tiempo.....	142
Figura 3.59. VI para generar la estampa de tiempo.....	143
Figura 3.60. Interfaz para mostrar la estampa de tiempo	144
Figura 3.61. Algoritmo para convertir la estampa de tiempo a formato binario	145

Figura 3.62. VI para convertir la estampa de tiempo a formato binario.....	146
Figura 3.63. Algoritmo para etiquetar cada bloque de datos en un archivo nuevo	147
Figura 3.64. VI para copiar un bloque de datos de un archivo a otro.....	148
Figura 3.65. Interfaz para copiar los datos de un archivo a otro	148
Figura 3.66. Estructura del nuevo archivo de datos	149

Capítulo IV

Figura 4.1. Capa superior de la PCB del sistema	150
Figura 4.2. Capa inferior de la PCB del sistema	151
Figura 4.3. Ensamblado de los componentes en la PCB	151
Figura 4.4. Prototipo final del sistema DIGI	152
Figura 4.5. Corriente demandada por el sistema DIGI.....	152
Figura 4.6. Programador AVRISP mkII.....	153
Figura 4.7. Conexión entre el programador, la PC y el DIGI.....	153
Figura 4.8. Características a verificar antes de programar el microcontrolador.....	154
Figura 4.9. Opciones del menú “memories” para programar al microcontrolador	155
Figura 4.10. Módulo de acondicionamiento de señal en PCB	156
Figura 4.11. Prueba realizada al amplificador.....	156
Figura 4.12. Prueba realizada a los limitadores de nivel	157
Figura 4.13. Respuesta del filtro pasa bajas para sensores pasivos	158
Figura 4.14. Respuesta del filtro pasa bajas para sensores activos.....	158
Figura 4.15. Verificación de funcionamiento del GPS.....	160
Figura 4.16. Comprobación de datos de fecha y hora	162
Figura 4.17. Archivo generado por el DIGI para pruebas cortas	163
Figura 4.18. Extracto de datos del archivo generado por el sistema DIGI.....	163
Figura 4.19. Tamaño del archivo final que ya contiene las etiquetas de tiempo.....	165
Figura 4.20. Archivos de prueba de un día.....	166
Figura 4.21. Gráficas a partir del archivo de datos de prueba de un día	167
Figura 4.22. Encabezados de tiempo de archivo de un día.....	168
Figura 4.23. Archivos generados por el sistema DIGI para pruebas largas	169
Figura 4.24. Datos del archivo generado por el DIGI para pruebas largas	169
Figura 4.25. Tamaño del nuevo archivo	170
Figura 4.26. Archivos de prueba de siete días.....	171
Figura 4.27. Gráficas a partir del archivo de datos de prueba de siete días	172
Figura 4.28. Encabezados de tiempo de archivo de siete días.....	173

Índice de Tablas

Capítulo II

Tabla 2.1. Características de la interfaz de comunicación de las memorias SD	42
Tabla 2.2. Clasificación de las memorias SD de acuerdo a su tasa de transferencia	42
Tabla 2.3. Estructura de comandos.....	46
Tabla 2.4. Descripción de comandos para operar con memorias SD	46
Tabla 2.5. Estructura de la respuesta R1	47
Tabla 2.6. Estructura de la respuesta R2	48
Tabla 2.7. Estructura de la respuesta R3	48
Tabla 2.8. Estructura de la respuesta R7	49
Tabla 2.9. Voltaje aceptado por la respuesta R7	49
Tabla 2.10. Definición del registro OCR.....	51
Tabla 2.11. Tamaño de particiones en sistemas de archivos FAT16 y FAT32.....	58
Tabla 2.12. Características de comunicación serial para receptor GPS	64
Tabla 2.13. Mensajes de salida del protocolo NMEA 0183.....	65
Tabla 2.14. Microcontroladores más comunes en el mercado	72
Tabla 2.15. Niveles de voltaje para los valores lógicos en el estándar RS-232	77
Tabla 2.16. Descripción de las terminales del conector DB9.....	78

Capítulo III

Figura 3.1. Características generales del microcontrolador AVR.....	83
Figura 3.2. Características eléctricas del amplificador y el convertidor A/D.....	86
Figura 3.3. Condiciones para activar D1 y D2	97
Figura 3.4. Combinación de selectores dependiendo el tipo de sensor	98
Figura 3.5. Descripción de terminales del convertidor A/D.....	100
Figura 3.6. Descripción de las terminales de la memoria micro SD	102
Figura 3.7. Descripción de las terminales del display	105
Figura 3.8. Características del receptor GPS.....	107
Figura 3.9. Descripción de los mensajes de salida del receptor GPS.....	119
Figura 3.10. Valores del registro OCR3A para diferentes tasas de muestreo	138
Figura 3.11. Términos comunes de LabVIEW y sus equivalencias.....	140
Figura 3.12. Número de bytes de cada bloque de información	141

Capítulo IV

Figura 4.1. Valores de capacitores para filtro antialiasing	159
---	-----

Prólogo

El Instituto de Ingeniería de la Universidad Nacional Autónoma de México tiene como principal objetivo contribuir al desarrollo del país por medio de la generación de recursos humanos y la generación de investigación y tecnología nacional. En los campos de la sismología e instrumentación se cuenta con una amplia experiencia, que han permitido el desarrollo de nuevas tecnologías en dichos campos, como son los sistemas de adquisición, registro y almacenamiento de datos con aplicaciones en actividad sísmica.

En el presente trabajo se describe el proceso de diseño y desarrollo de un sistema de adquisición y almacenamiento de datos de señales sísmicas, con lo que se pretende tener un equipo totalmente autónomo, sin requerir el uso de un equipo de cómputo, que sirva como base para la adquisición y almacenamiento de fenómenos sísmicos por periodos prolongados de tiempo.

Este sistema contará con tres canales analógicos, en los cuales se pueden conectar diversos sensores de acuerdo a las características del fenómeno en prueba; las señales provenientes de los sensores pasan a través de un etapa de acondicionamiento, que tiene como función principal eliminar el ruido de la señal y poner a ésta dentro de un rango que sea admitido por los convertidores analógicos digitales, ya que estos se encargan de digitalizar las señales analógicas a señales digitales, para que puedan ser procesadas por un microcontrolador y posteriormente almacenadas en una memoria SD card. Adicionalmente, nos da la posibilidad de seleccionar la frecuencia de muestreo a la que se desee trabajar. Se cuenta también con una pantalla gráfica y un *joystick* conformando una interfaz usuario – máquina, que le permite al operador configurar el sistema para su operación, así como verificar que cada canal del dispositivo esté funcionando; puesto que en el modo de prueba se puede observar la gráfica de la señal de entrada, además de que en esta pantalla se pueden observar los datos de fecha y tiempo UTC (*Universal Time Coordinated*) provenientes del GPS (*Global Positioning System*).

Con base en lo mencionado anteriormente, la estructura de este trabajo se dividirá en cinco capítulos, los cuales son desarrollados de la siguiente manera:

Capítulo 1. En este capítulo se definen los conceptos básicos necesarios en el estudio de la sismología, mencionando el tipo de ondas que implican a este tipo de fenómenos. Además de hacer una breve descripción de la actividad sísmica en nuestro país, se presenta de manera general el problema que se desea resolver, así como la propuesta de solución.

Capítulo 2. Se presentan los temas básicos y fundamentales que permiten comprender el proceso de desarrollo del sistema; asimismo, se abordan tópicos relacionados con un

sistema de adquisición de señales sísmicas, como es el caso de los transductores utilizados en el área sísmica, el módulo de acondicionamiento y digitalización de la señal, la sincronización de tiempo con los datos, el almacenamiento de los datos en una SD Card, y la interfaz usuario - máquina

Capítulo 3. Aquí se plantea formalmente las necesidades que debe cubrir el proyecto, presentando el diseño y desarrollo del sistema. Se tienen dos partes fundamentales, a saber: el hardware integrado por todos los dispositivos y componentes interconectados y el software constituido por las funciones y rutinas programadas dentro del microcontrolador.

Capítulo 4. Se describen las pruebas que se realizaron a los distintos bloques de hardware y software que conforman al sistema, llegando a su etapa final, es decir, funcionando todo en conjunto.

Capítulo 5. Finalmente, se presentan los resultados obtenidos en el desarrollo de este trabajo, las conclusiones a las que se llegaron y algunas recomendaciones para mejorar el sistema desarrollado.

También se incluyen unos anexos importantes para una mejor comprensión del trabajo, así como la bibliografía utilizada.

Capítulo I

Introducción

En este capítulo se mencionarán los fundamentos teóricos del estudio de la sismología, abarcando el tipo de ondas sísmicas que interesan a los investigadores de esta rama y la actividad sísmica en nuestro país. Por otra parte, se presentará una breve introducción al contenido de la presente tesis, definiendo el problema que se desea resolver, así como la propuesta de solución ante dicho problema.

1.1. La Sismología

El término sismología viene del griego *seísmos*: *sismo* y *logos*: *estudio*. La sismología es una rama de la geofísica que se encarga del estudio de los sismos o terremotos e implica la observación de las vibraciones naturales del terreno y de las señales sísmicas generadas de forma artificial, con muchas ramificaciones teóricas y prácticas. Asimismo estudia los maremotos y las marejadas asociadas a otros fenómenos, así como vibraciones previas a erupciones volcánicas.

Como es sabido, la Tierra está formada por tres capas concéntricas: corteza, manto y núcleo, con propiedades físicas distintas. Estas capas han podido ser detectadas y definidas a partir del estudio de los registros del movimiento de su superficie y más concretamente por los estudios de los terremotos. En la figura 1.1 se muestra un esquema de las capas principales que forman nuestro planeta.

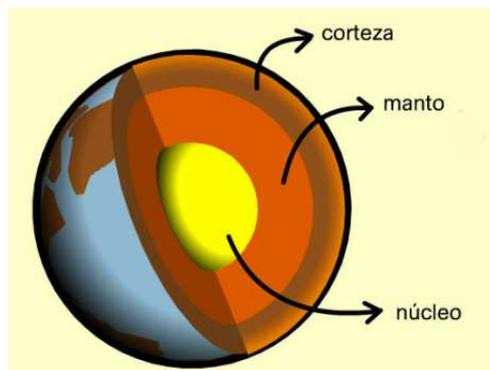


Figura 1.1. Principales capas de la Tierra.

Los objetos de estudio más importantes de la sismología son los siguientes:

- Los sismos: detección, localización, magnitud, energía y movimiento a lo largo de las fallas.
- Estudio de la estructura interna de la Tierra.
- La delineación de la geología de las cuencas sedimentales en la búsqueda de petróleo, gas y carbón.
- Reconocimiento de la corteza terrestre superior en la hidrología y la exploración de agua subterránea.
- La sismología teórica o matemática y el procesamiento de los datos.

La magnitud de un temblor es un valor único y es una medida cuantitativa del sismo relacionada con la energía sísmica liberada; el cálculo de ésta se realiza a partir de los registros de instrumentos sísmicos y se basan en su contenido energético, el cual puede calcularse con la información de las amplitudes máximas que presenta el registro sísmico y la duración del registro. Para calcular de forma confiable la magnitud de un temblor es necesario que el registro sísmico se encuentre completo, lo que generalmente toma varios minutos dependiendo del tamaño del temblor.

Para medir la magnitud de la actividad sísmica se cuenta con dos escalas empleadas a nivel mundial:

- Escala de Richter: Expresa y cuantifica la magnitud de un terremoto a partir del registro de las amplitudes máximas de las ondas sísmicas por medio de un sismógrafo.
- Escala de Mercalli y de Rossi – Forel: Cuantifican la intensidad de un terremoto a partir de la observación y clasificación de los daños causados por el.

Las causas de un terremoto son:

- Movimiento de las placas tectónicas.
- La ruptura repentina de las rocas, que han sido distorsionadas más allá de su límite de resistencia.
- La explosión de un volcán.

El punto inicial de un terremoto se denomina foco o hipocentro. Su proyección hacia la superficie terrestre se denomina epicentro. La mayoría de la energía sísmica (85%) se libera en profundidades entre 0 – 70 km; en una profundidad de 70 – 300 km se libera el 12% de la energía, a una profundidad entre 300 – 700 km se libera sólo el 3% de la energía sísmica y a profundidades mayores de 720 km ya no se detecta.

Para determinar a qué distancia se encuentra una estación sísmica S1 del epicentro, se obtiene la diferencia del tiempo de llegada entre las ondas P y S en esa estación y el

resultado se multiplica por la velocidad teórica del viaje de las ondas P que es de 8 km/s. Si se traza un círculo alrededor de la estación sísmica, cuyo radio es el valor obtenido anteriormente (figura 1.2), se obtendrá una circunferencia cuyo límite sería la región probable donde ocurrió el evento. Al realizar este mismo procedimiento para otras dos estaciones (S2 y S3) se verá que las tres circunferencias cortan entre sí en un mismo punto, este es el lugar es donde se encontrará el epicentro del sismo.

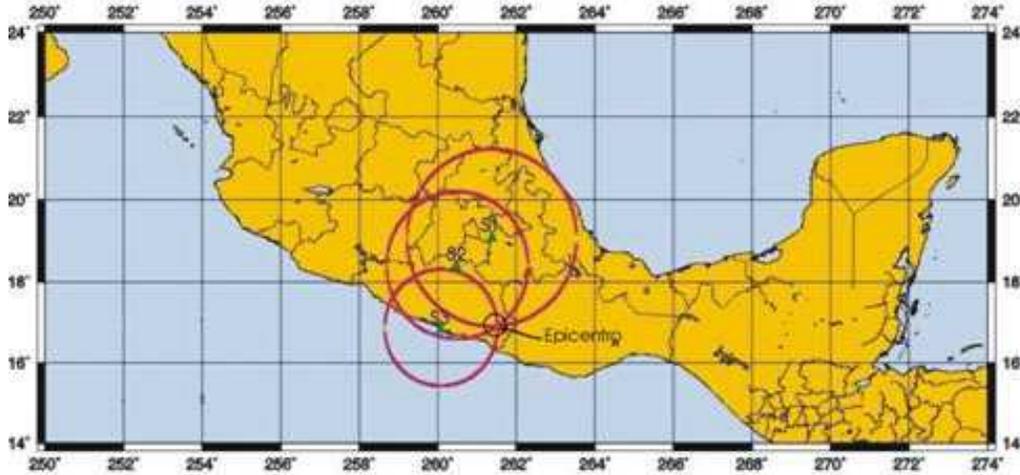


Figura 1.2. Epicentro de un Terremoto.

El procedimiento descrito es ideal, ya que se considera que el material del que está compuesta la Tierra es uniforme, sin embargo, en la realidad esto es muy diferente. Como se mencionó anteriormente, la Tierra está compuesta de diferentes capas, por tal motivo las ondas se comportan de diferente manera al atravesar de un medio a otro. El cálculo de la distancia epicentral se vuelve entonces más complejo y son necesarios sistemas más especializados que requieren de mayor tiempo para efectuar una localización de gran precisión.

1.1.1. Actividad sísmica en México

Las placas tectónicas son estructuras que forman nuestro planeta. En términos geológicos, una placa es una plancha de roca sólida que conforma la superficie de la Tierra (litosfera), flotando sobre roca ígnea y fundida que forma el centro de la Tierra (astenosfera). La litósfera tiene un grosor que varía entre los 15 y los 200 km, siendo más gruesa en los continentes que en el fondo marino. En la figura 1.3 se ilustran las principales placas tectónicas que conforman a nuestro planeta.

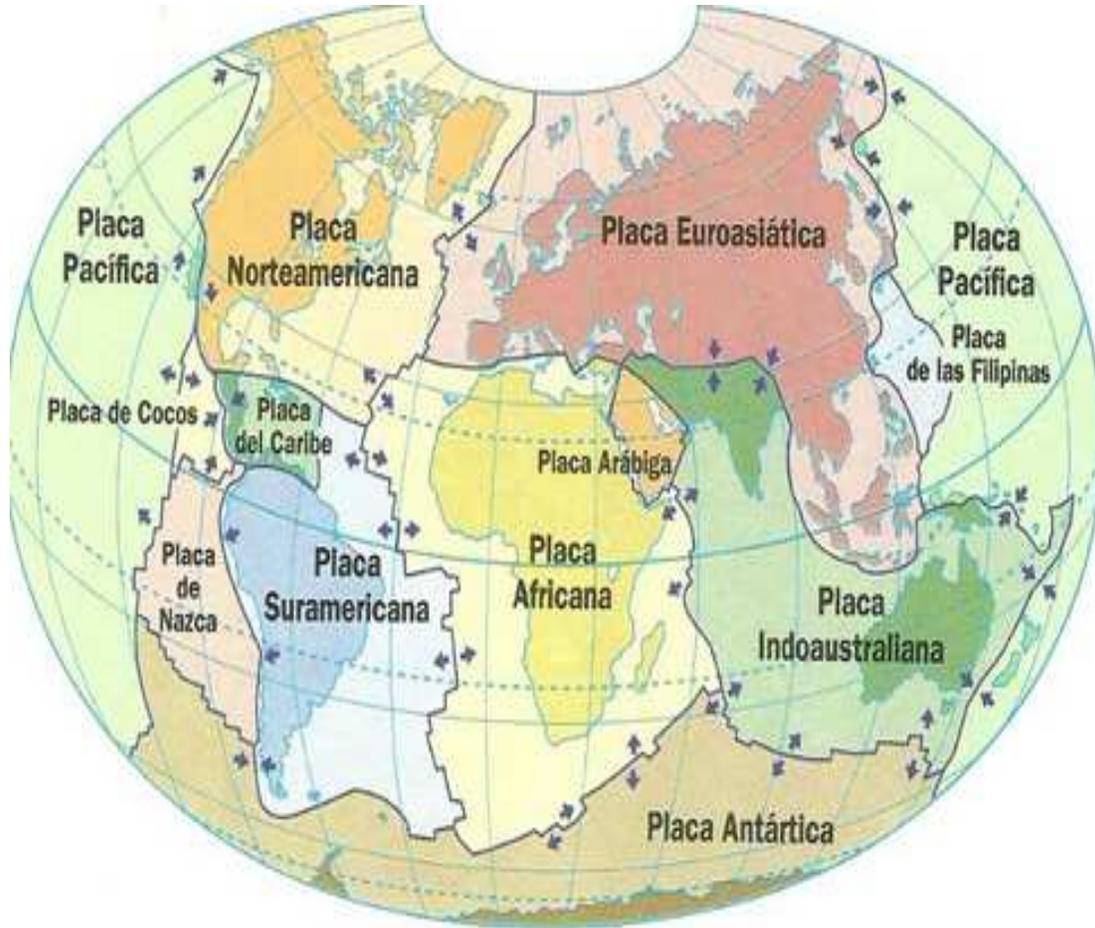


Figura 1.3. Localización de las placas tectónicas.

El territorio Mexicano tiene un alto nivel de sismicidad, debido a la interacción de la Placa de Cocos, la del Caribe y la norteamericana, donde la Placa Norteamericana se desliza lentamente por debajo de la de Cocos. Este movimiento es constante, pero cuando llega haber fricción entre las Placas y terminan por atorarse, se acumula energía en ellas, debido a que las corrientes de convección del manto terrestre las siguen empujando, lo que ocasiona que se acumule energía y al cabo de un tiempo, cuando se llega a liberar, ocasiona fuertes movimientos sísmicos, tales como los acontecidos en 1957 y el de 1985.

Al norte del territorio Mexicano se encuentra la Falla de San Andrés, que a diferencia de la zona de subducción formada por la Placa Norteamericana y la de Cocos, esta falla es del tipo transformante, es decir que se desplaza hacia los lados, siendo de las fallas más activas en el mundo, por lo que la actividad sísmica en esa zona es muy común; aunque rara vez ha ocasionado un sismo de gran magnitud. El último sismo sumamente destructivo y de magnitudes muy grandes fue el de San Francisco a inicios del siglo XX.

1.2. Tipos de ondas sísmicas

Un sismo consiste precisamente en la liberación repentina de los esfuerzos impuestos al terreno, que permite la generación de ondas, las cuales se transmiten por el interior de la Tierra siguiendo caminos curvos, debido a la variada densidad y composición interior de la Tierra. A este tipo de ondas se les conoce como ondas internas, centrales o de cuerpo, las cuales transmiten los temblores preliminares de un terremoto pero poseen poco poder destructivo.

Las ondas de cuerpo son divididas en dos grupos: Ondas Primarias (P) y Ondas Secundarias (S).

Ondas primarias (P)

Son ondas longitudinales, lo cual significa que el suelo es alternadamente comprimido y dilatado en la dirección de la propagación, tal como se muestra en el esquema de la figura 1.4, se muestra el modelo de las ondas P. Estas ondas generalmente viajan a una velocidad 1.73 veces mayor que las ondas S y lo hacen a través de cualquier tipo de material.

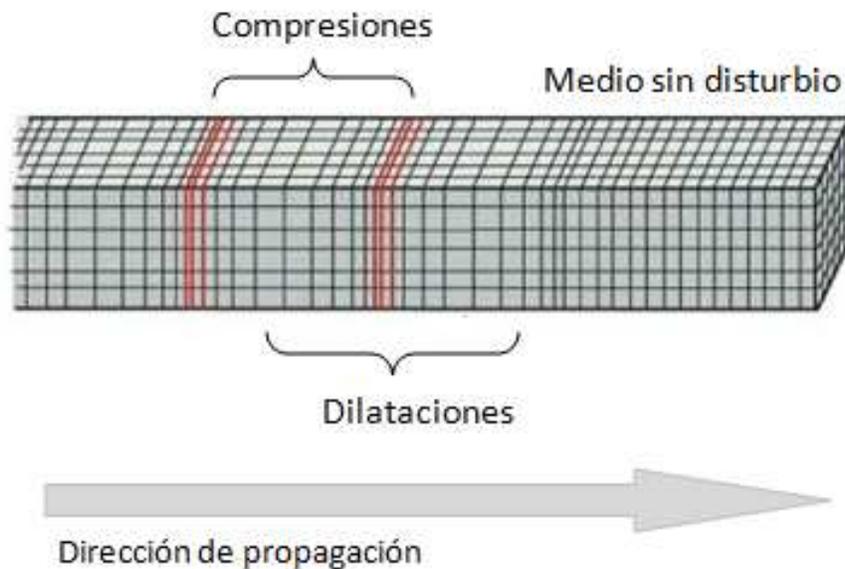


Figura 1.4. Ondas P.

Ondas secundarias (S)

Son ondas transversales o de corte, lo cual significa que el suelo es desplazado perpendicularmente a la dirección de propagación, alternadamente hacia un lado y hacia el otro; en la figura 1.5 se muestra este tipo de ondas. Las ondas S pueden viajar únicamente a través de sólidos, debido a que los líquidos no pueden soportar esfuerzos de corte. Usualmente, la onda S tiene mayor amplitud que la onda P y se siente más fuerte que ésta.

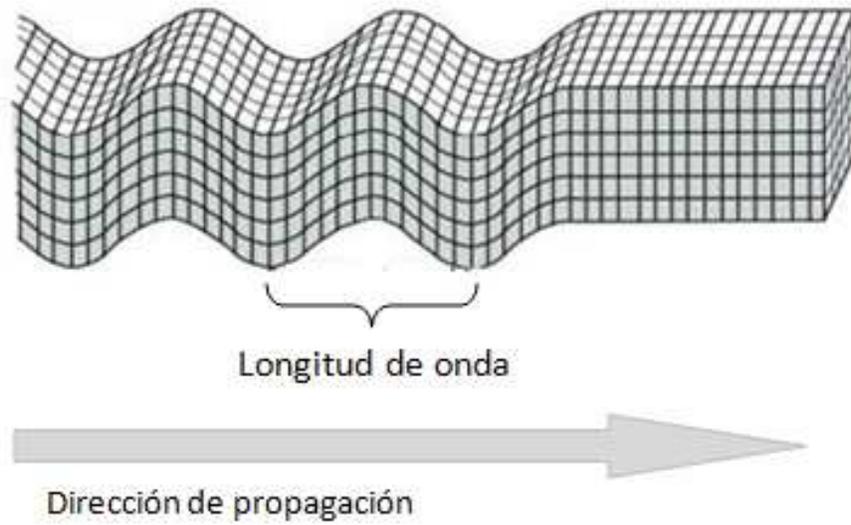


Figura 1.5. Ondas S.

Por otra parte, también se propagan ondas por la superficie, que son las que más tardan en llegar, debido a su baja frecuencia, provocan resonancia en edificios con mayor facilidad que las ondas de cuerpo causando los efectos más devastadores; existen de dos tipos: de Rayleigh (R) y de Love (L).

Ondas de Rayleigh (R)

Cuando un sólido posee una superficie libre, como la superficie de la Tierra, pueden generarse ondas que viajan a lo largo de la superficie. Estas tienen su máxima amplitud en la superficie libre, la cual decrece exponencialmente con la profundidad. La trayectoria que describen las partículas del medio, al propagarse la onda, es elíptica, retrógrada y ocurre en el plano de propagación de la onda, como se muestra en la figura 1.6.

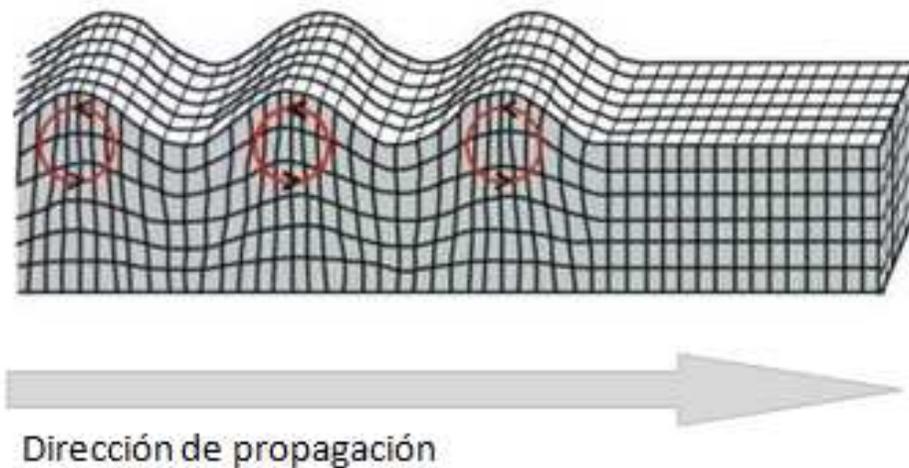


Figura 1.6. Ondas de Rayleigh.

Ondas de Love (L)

Este tipo de ondas se generan sólo cuando un medio elástico se encuentra estratificado, situación que se cumple en nuestro planeta pues se encuentra formado por capas de diferentes características físicas y químicas. Las ondas de Love se propagan con un movimiento de las partículas perpendicular a la dirección de propagación, como las ondas S, sólo que polarizadas en el plano de la superficie de la Tierra; es decir, sólo poseen componentes horizontales a superficie, en la figura 1.7 se muestra este tipo de ondas.

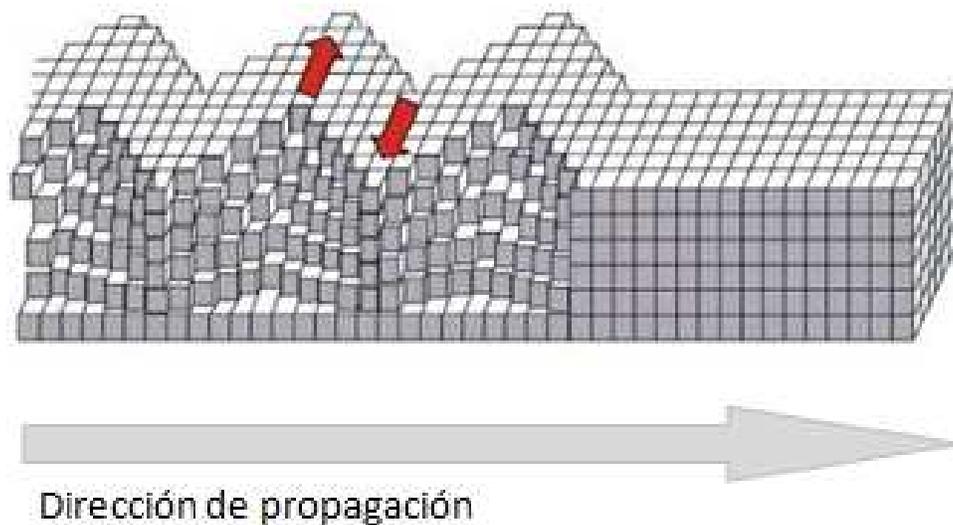


Figura 1.7. Ondas de Love.

1.3. Equipos de medición sísmica

Un sismógrafo o sismómetro es un instrumento utilizado para censar la actividad sísmica. Las versiones preliminares de este instrumento de medición consistían en un péndulo, que por su masa permanecía inmóvil debido a la inercia, mientras todo a su alrededor se movía; dicho péndulo llevaba un punzón que iba escribiendo sobre un rodillo de papel pautado en tiempo, de modo que se registraba dicho movimiento en el papel. A esta representación gráfica se le conoce como sismograma.

En la actualidad, los instrumentos modernos usados en la medición sísmica son electrónicos y brindan grandes prestaciones, ya que se puede almacenar la información en pequeñas memorias. Debido a los componentes electrónicos, se tienen equipos con un gran ancho de banda que hace muy versátiles estos tipos de instrumentos.

1.4. Planteamiento del problema y propuesta de solución

En el mercado internacional existen equipos de gran costo y alto nivel de integración que permiten llevar a cabo el registro y almacenamiento de señales sísmicas, cabe comentar que el mantenimiento de estos equipos es realizado por mano de obra extranjera, cuyo costo es elevado y la mayoría de las veces tardado. En el Instituto de Ingeniería de la UNAM, se han tenido experiencias con el uso de este tipo de instrumentos.

Dicho lo anterior, la presente tesis de Licenciatura tiene como finalidad el desarrollo de un sistema embebido de bajo costo y alta confiabilidad, que permita la digitalización de las señales sísmicas para posteriormente ser almacenadas en una memoria SD Card. El sistema podrá admitir tanto sensores pasivos, como es el caso de los geófonos, como sensores activos, como son los acelerómetros de fuerza balanceada.

Con la finalidad de garantizar el funcionamiento del dispositivo, éste será sometido a pruebas en las cuales se tenga que digitalizar una señal conocida en amplitud y frecuencia, pasando por la etapa de acondicionamiento y observándola ya en su forma digitalizada.

Es importante mencionar que este tipo de sistemas no pueden perder información en ningún momento, debido a que si un sismo se presenta, se requiere tener completo el registro del mismo, desde que inicia el fenómeno hasta que termina. Dicho lo anterior, la referencia de tiempo será sincronizada con los valores obtenidos del GPS, para posteriormente hacer un procesamiento de la información, que consiste en etiquetar la hora y fecha de las muestras obtenidas cada segundo.

En el sistema a diseñar, denominado “DIGI”, se podrá configurar su tasa de muestreo por medio de un *joystick* y un GLCD, que hacen una interfaz más amigable con el usuario, además, de que al momento de ser probado en el campo, no sera necesario llevar consigo una PC para realizar el almacenamiento de los datos sísmicos, o para la verificación de que el sistema se encuentre en operación, ya que por medio de esta pantalla se podrán observar los datos digitalizados por cada canal, así como también, los datos referentes a la fecha y hora UTC. Por otra parte, el sistema de almacenamiento aceptará memorias de hasta 8 GB, lo que significa que el sistema podrá configurarse para trabajar por más de un mes ininterrumpido.

En el siguiente capítulo se describirá de manera general los fundamentos teóricos que permitirán la comprensión de las partes que componen el sistema propuesto.

Capítulo II

Generalidades

En este capítulo se describen los conceptos básicos que se emplearán en el desarrollo de esta tesis, haciendo referencia a los componentes electrónicos que integran el sistema para la digitalización y almacenamiento de señales sísmicas, así como los dispositivos para la interfaz usuario – máquina. Esto con la finalidad de comprender mejor el sistema.

2.1. Tipos de Sensores Sísmicos

Un sensor sísmico es un dispositivo diseñado para registrar el movimiento del suelo, cuando éste es sacudido por una perturbación artificial o natural. A nivel cualitativo el sensor se puede considerar como un sistema en donde la entrada corresponde a una variable cinemática, como la aceleración, velocidad o desplazamiento y la salida de este sistema corresponde a una variable eléctrica (voltaje o corriente). Dependiendo de la variable medida los sensores sísmicos se pueden clasificar como de desplazamiento, velocidad, aceleración o deformación.

El rango de frecuencias de las señales sísmicas es muy amplio, comenzando en valores tan bajos como 0.00001 Hz extendiéndose hasta 1000 Hz. En la figura 2.1 se ilustra el rango de frecuencias sísmicas.

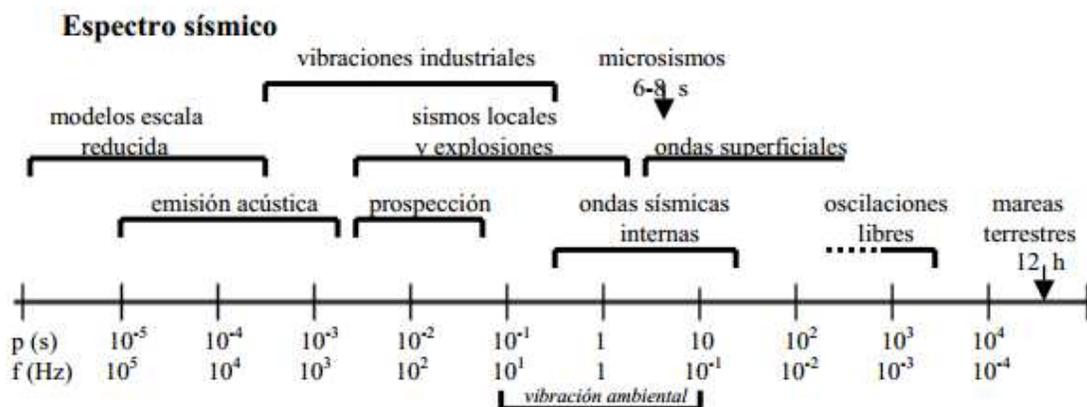


Figura 2.1. Rango de frecuencias sísmicas.

Por tal motivo, los sensores pueden clasificarse por su ancho de banda como:

- **Periodo Corto (SP: Short Period).** Son capaces de detectar señales de 0.1 a 100 Hz, con una frecuencia natural entorno a 1Hz (valores típicos). Este tipo de sensores son utilizados para el estudio de sismos de baja magnitud o experimentos de refracción y tienen la ventaja de fácil instalación y operación, además de que no requieren de una fuente de alimentación.
- **Banda ancha (BB: Broad Band).** Detectan señales sísmicas en el rango de frecuencia de 0.01 a 50 Hz, y requieren una fuente de alimentación con polaridad simple o doble. Este tipo de sensores son utilizados en estaciones sísmicas, debido a que pueden registrar señales de muy baja frecuencia.
- **Banda muy ancha (VBB: VeryBroad Band).** Detectan señales sísmicas comprendidas en el rango de frecuencia de 0.001 a 10 Hz y tienen como principal aplicación la investigación de la profundidad de la tierra.

La elección del sensor depende directamente de la aplicación de interés, ya que en aplicaciones donde se requiere detectar señales de muy baja frecuencia, el sensor llega a ser el componente más caro de todo el sistema.

Los sensores sísmicos también se clasifican en pasivos y activos y pueden medir las componentes de movimiento horizontal o vertical.

Sensores Pasivos

Son sensores que no requieren de una fuente de alimentación para operar y comprenden sensores mecánicos y electromecánicos.

- *El sensor mecánico*

Debido a que las mediciones se hacen en un marco de referencia en movimiento (la superficie de la tierra), casi todos los sensores sísmicos están basados en la inercia de una masa suspendida, la cual tenderá a permanecer estacionaria en respuesta a un movimiento externo.

El movimiento relativo entre la masa suspendida y la tierra será entonces una función del movimiento de la tierra.

Como se observa en la figura 2.2, un sensor mecánico vertical consiste básicamente de una masa, un resorte y un amortiguador.

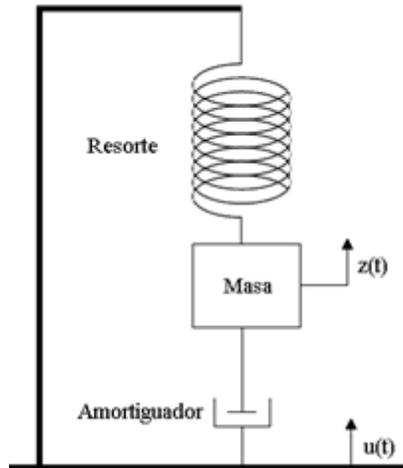


Figura 2.2. Sensor mecánico vertical.

Considerando $u(t)$ como el movimiento vertical de la tierra y a $z(t)$ como el desplazamiento de la masa m con respecto a la tierra, ambos positivos hacia arriba, hay dos fuerzas que se oponen al movimiento de la masa: la fuerza del resorte $-kz$, donde k es la constante del resorte, y el amortiguamiento $-d\dot{z}$, donde d es la constante de fricción y \dot{z} es la velocidad. La aceleración de la masa con respecto al marco de referencia inercial es la suma de la aceleración (\ddot{z}) con respecto a la tierra y la aceleración de la tierra (\ddot{u}). Aplicando la segunda ley de Newton, que dice que la suma de las fuerzas debe ser igual a la suma de las aceleraciones por la masa, se puede plantear la siguiente ecuación:

$$-kz - d\dot{z} = m\ddot{z} + m\ddot{u} \quad (2.1)$$

Para un sistema de segundo orden como el descrito por la ecuación anterior, se define la frecuencia natural como $\omega_0 = \sqrt{k/m}$ y la constante de amortiguamiento como $h = d/(2m\omega_0)$. De modo que la ecuación 2.1 puede reescribirse como:

$$\ddot{z} + 2h\omega_0\dot{z} + \omega_0^2 z = -\ddot{u} \quad (2.2)$$

Resolviendo la ecuación 2.2 en el dominio de la frecuencia, se puede calcular la relación entre la salida y la entrada (también denominada la función de la respuesta en frecuencia del desplazamiento), $T_d(\omega)$.

$$T_d(\omega) = \frac{Z(\omega)}{U(\omega)} = \frac{\omega^2}{\omega_0^2 - \omega^2 + i2\omega\omega_0 h} \quad (2.3)$$

De la expresión anterior se puede calcular la respuesta en amplitud $A_d(\omega)$ y la respuesta en fase $\varphi_d(\omega)$ del desplazamiento:

$$A_d(\omega) = |T_d(\omega)| = \frac{\omega^2}{\sqrt{(\omega_0^2 - \omega^2)^2 + 4h^2\omega^2\omega_0^2}} \quad (2.4)$$

$$\varphi_d(\omega) = \tan^{-1} \left(\frac{-2h\omega\omega_0}{\omega_0^2 - \omega^2} \right) \quad (2.5)$$

A manera de ejemplo, del análisis anterior, la figura 2.3 muestra la respuesta en fase y en amplitud de un sensor con una frecuencia natural de 1Hz y una constante de amortiguamiento con valores de 0.25 a 4. Como se observa, una constante de amortiguamiento baja ($h < 1$) da como resultado un pico en la respuesta. Si $h=1$, la masa del sensor retorna a su posición de reposo en el menor tiempo posible, sin sobrepaso, la curva no tiene pico y se dice que el sensor está críticamente amortiguado. De la curva y la ecuación 2.3, se puede ver que el sensor se comporta como un filtro paso alto para el desplazamiento de la tierra. El valor más comúnmente usado para la constante de amortiguamiento es $h=0.707$, debido a que se puede describir la respuesta como un filtro paso altas de segundo orden con frecuencia de corte en ω_0 . Por otro lado, cuando $h > 1$ la sensibilidad del sensor disminuye.

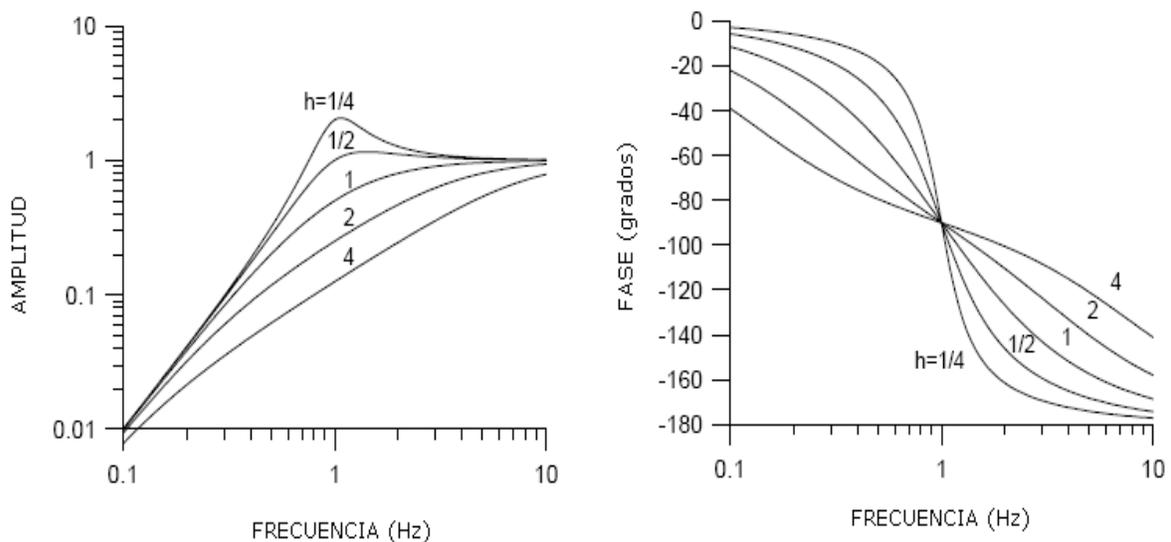


Figura 2.3. Respuesta en frecuencia de un sensor mecánico.

- *El sensor electromagnético*

Los sensores electromagnéticos tienen como principio de funcionamiento el movimiento de una bobina dentro de un campo magnético, esto se puede implementar por medio de un imán fijo y una bobina que se mueve con la masa, produciéndose a la salida de la bobina un voltaje proporcional a la velocidad de la masa (figura 2.4).

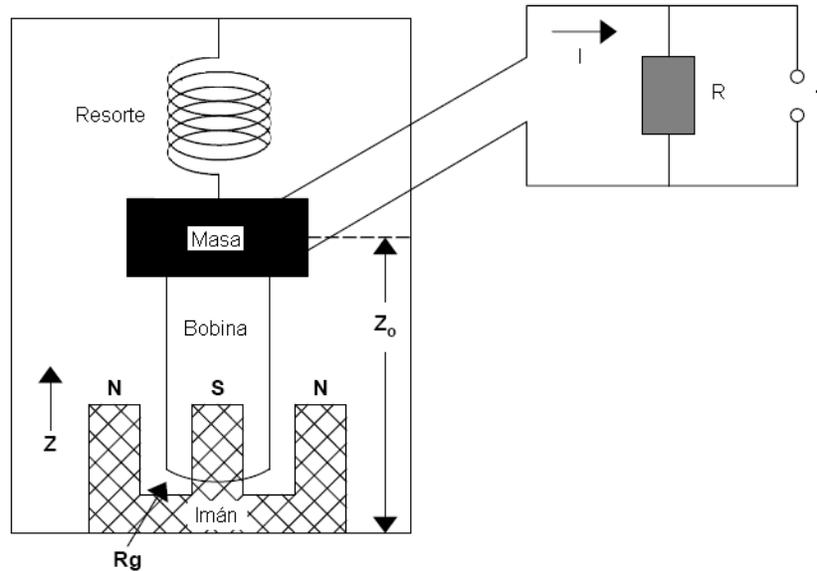


Figura 2.4. Sensor electromagnético.

A comparación del sistema puramente mecánico, en este tipo de sistemas se agregan dos constantes, las cuales son:

La constante del generador G, la cual relaciona la velocidad de la tierra con la salida de la bobina, y se mide en V/ms^{-1} . Los valores típicos para esta constante se encuentran en el rango de $30 - 500 V/ms^{-1}$. Y la resistencia de la bobina del generador R_g , la cual se expresa en ohms.

La resistencia R_g permite amortiguar el sistema con el simple hecho de agregarle una carga resistiva (R), por lo que al variar el valor de la carga (R) se tendrán diferentes respuestas del sistema de segundo orden, controlando de esta manera su factor de amortiguamiento. Este tipo de sensores presentan un funcionamiento óptimo cuando el valor del amortiguamiento es cercano a 1, es decir cuando el sistema es críticamente amortiguado.

Como ya se mencionó, en este sistema la señal de salida es un voltaje cuyo valor es proporcional a la velocidad de la masa, $\dot{Z}(\omega)$, y a la constante del generador G, en lugar de $Z(\omega)$, como lo era para los sensores mecánicos. De aquí que la respuesta del desplazamiento para el sensor de velocidad se expresa en la ecuación 2.6.

$$T_d^v(\omega) = \frac{\dot{Z}(\omega)}{U(\omega)} G = \frac{i\omega^3 G}{\omega_0^2 - \omega^2 + i2\omega\omega_0 h} \quad (2.6)$$

Sensores Activos

Los sensores activos son aquellos que requieren de una fuente de alimentación para su funcionamiento, ya que contienen circuitos electrónicos que requieren ser polarizados. Dentro de esta clasificación se encuentran los acelerómetros, los sensores de velocidad de banda ancha y los sensores de realimentación negativa.

- Los acelerómetros

Los acelerómetros, o sensores de movimiento fuerte, por definición miden aceleraciones grandes, típicamente de 1 a 2 g, donde g representa la aceleración de la gravedad (9.81 m/s²). Los acelerómetros antiguos se basaban en un sensor mecánico con una frecuencia natural alta, para asegurar que la salida fuera proporcional a la aceleración y se grababa ópticamente en una película; los acelerómetros actuales son de fuerza balanceada (FBA: *Force Balanced Accelerometer*) y han tenido un mejoramiento importante en su desempeño.

El FBA está compuesto por una masa, un resorte, una bobina y un transductor de desplazamiento de tipo capacitivo (figura 2.5a). En este tipo de sensor, el transductor de desplazamiento genera una corriente en respuesta al movimiento de la masa. A través del resistor se envía dicha corriente a la bobina, donde se genera una fuerza para compensar la fuerza externa ejercida sobre la masa e impedir que se mueva del todo. Debido a que la corriente es proporcional a la aceleración de la tierra, ésta se puede estimar a partir del voltaje en el resistor.

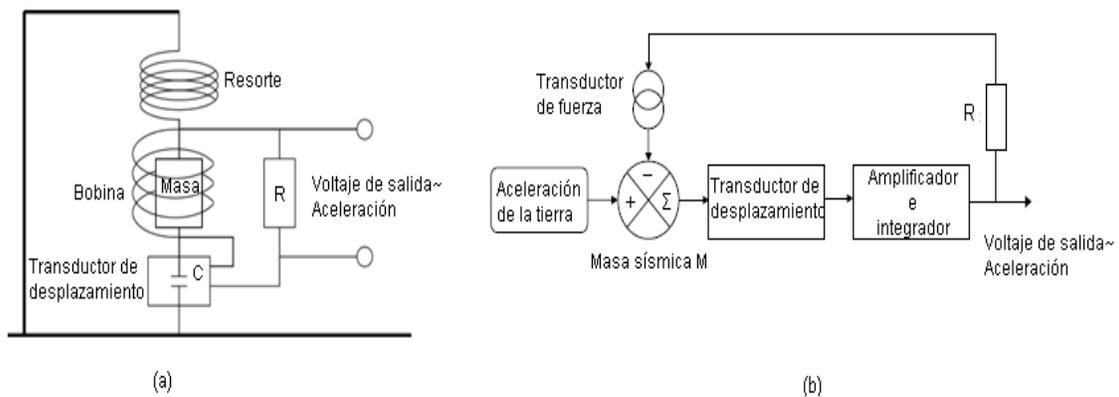


Figura 2.5. Acelerómetro de fuerza balanceada.

Como se observa en la figura 2.5b, este tipo de sensor se representa como un sistema con realimentación, cuya entrada es la aceleración de la tierra y cuya salida es un

voltaje proporcional a dicha aceleración. La función de transferencia que caracteriza este tipo de sensor se define en la ecuación 2.7.

$$T(\omega) = \frac{V(\omega)}{\ddot{U}(\omega)} = \frac{T_1(\omega)}{\omega_0^2 + \frac{\sigma T_1(\omega)}{mR} - \omega^2 + 2ih\omega\omega_0} \quad (2.7)$$

En donde m es la masa sísmica (kg), R es la resistencia de realimentación (Ω), h es la constante de amortiguamiento, ω_0 es la frecuencia natural en (hz), σ es la constante del transductor de fuerza en (N/A) y $T_1(\omega)$ es la respuesta de un integrador (filtro paso bajas) caracterizada por la siguiente ecuación:

$$T_1(\omega) = \frac{D}{i\omega\tau_1 + 1} \quad (2.8)$$

En donde D es la constante del transductor de desplazamiento (V/m) y τ_1 es la constante de tiempo del filtro paso bajas con frecuencia de corte en $f=1/(2\pi\tau_1)$.

El FBA tiene la ventaja de ofrecer una respuesta plana a frecuencias tan bajas como DC, pero con la limitante de que tiene una frecuencia superior límite, como se muestra en la figura 2.6.

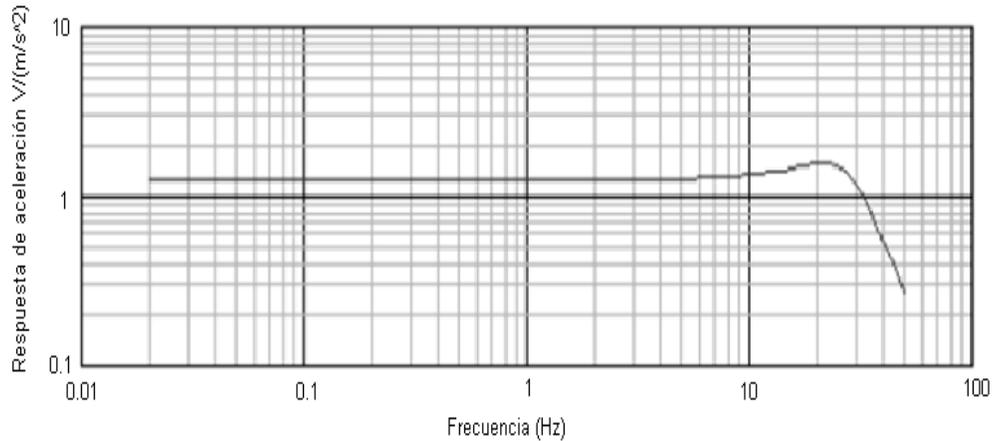


Figura 2.6. Respuesta en frecuencia de un FBA.

- *Sensores de velocidad de banda ancha*

Los sensores de velocidad de banda ancha se basan en el principio de operación de un FBA, pero con la salvedad de que utilizan una realimentación seleccionada cuidadosamente para obtener un resultado eficiente en términos de rango dinámico.

Dicha realimentación consiste en un capacitor grande, que contribuye a que la corriente de realimentación sea proporcional a la aceleración y a que la salida a través del capacitor sea la integral de dicha corriente y por tanto proporcional a la velocidad de la tierra (ver figura 2.7).

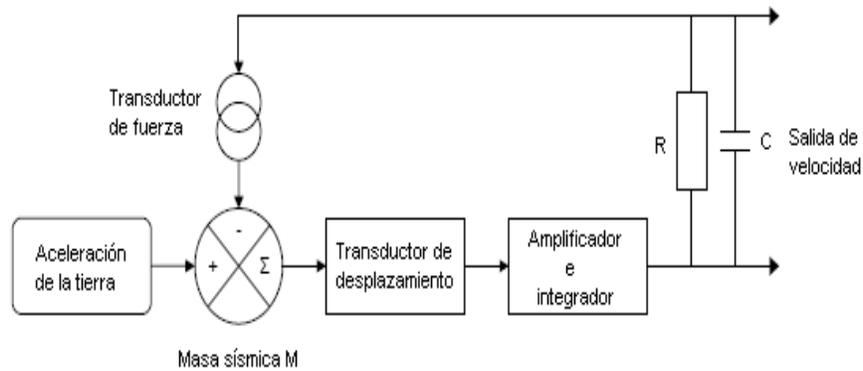


Figura 2.7. Sensor de velocidad de banda ancha.

En la práctica se suele agregar otro nodo de realimentación, que aporta una corriente proporcional a la integral de la salida de velocidad. Esto último se hace con la intención de modelar la respuesta en frecuencia, como si fuera un sensor electromagnético de periodo largo, extendiendo la respuesta a bajas frecuencias y con una frecuencia máxima menor (ver figura 2.8).

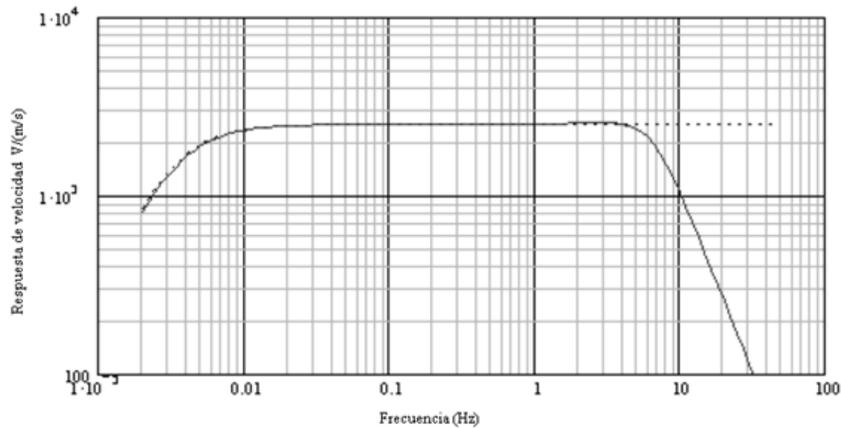


Figura 2.8. Respuesta en frecuencia de un sensor de velocidad de banda ancha.

La ecuación de lazo cerrado que caracteriza a este tipo de sensor se muestra en la expresión 2.9.

$$T(\omega) = \frac{-T_1(\omega)i\omega}{(\omega_0^2 - \omega^2 + 2ih\omega\omega_0) + \frac{\sigma}{m} \left(\frac{RCi\omega + 1}{R} + \frac{T_2(\omega)}{R_2} \right) T_1(\omega)} \quad (2.9)$$

Las variables de la ecuación 2.9 son las mismas que se explicaron para las ecuaciones 2.7 y 2.8. Solamente se tienen tres variables nuevas: C que es el valor del capacitor de salida en Faradios, R_2 que es la resistencia del segundo nodo de realimentación en Ω y $T_2(\omega)$ que es un segundo integrador del mismo tipo que $T_1(\omega)$, pero con una constante de tiempo más grande.

La respuesta de velocidad de este tipo de sensor está determinada principalmente por 4 componentes pasivos (R , C , la masa y la bobina de realimentación). Además, se puede considerar como si se tratara de un sensor de velocidad convencional, descrito por sus parámetros comunes: período libre, amortiguamiento y constante del generador.

Estos sensores están disponibles en un extenso rango de características y precios; generalmente, entre mejor sea la relación señal a ruido y más baja sea la frecuencia inferior, el costo es mayor.

- *Sensores con realimentación negativa*

Los sensores con realimentación negativa se construyen a partir de sensores electromagnéticos convencionales, pero a la salida de éstos se coloca un circuito de realimentación como el mostrado en la figura 2.9, con el que se logra extender la respuesta en frecuencia.

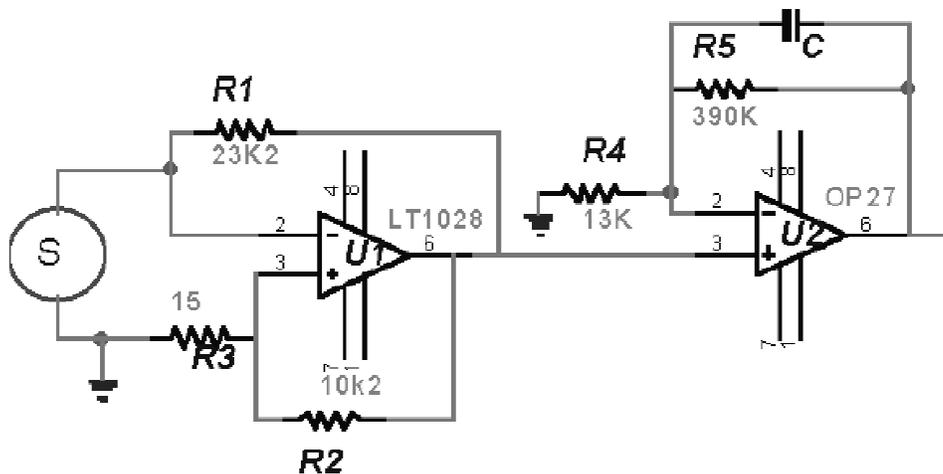


Figura 2.9. Circuito para extender la respuesta de un sensor electromagnético.

La ventaja de este diseño es que se pueden construir sensores con respuesta en frecuencia extendida a partir de sensores baratos, como los geófonos de 4.5 Hz. Sin embargo, debido a las limitaciones de los parámetros eléctricos y mecánicos, la frecuencia inferior no se puede extender más de un factor de 5.

2.2. Acondicionamiento de señales de entrada

Los elementos que conectan un sistema electrónico con su entorno, como son los sensores y actuadores, no están diseñados en general para ser conectados directamente con un microprocesador. En realidad el acondicionamiento de una señal depende de cada aplicación, sin embargo y en general se tienen bloques bien identificados antes de digitalizar la señal, como son el filtrado de la señal, la amplificación y en ocasiones la adición de una componente que proporciona un voltaje corriente directa (CD), denominado “*offset*”. Esto con la finalidad de que al momento de digitalizar la señal, se encuentre en el rango de voltaje de entrada del convertidor analógico digital. Con estas etapas se garantiza que la señal digitalizada sea una réplica de la señal analógica de interés.

En las siguientes secciones se mencionarán algunos de los conceptos básicos para poder acondicionar una señal proveniente de un transductor.

2.2.1. Tipos de Señales

Una señal es toda magnitud eléctrica cuyas variaciones llevan información sobre un proceso o magnitud física. Las señales se pueden clasificar en su forma más básica en señales analógicas, digitales y discretas.

- **Señales analógicas:** Este tipo de señales, tienen como principal característica que su amplitud varía de forma continua en el tiempo.
- **Señal discreta:** Este tipo de señales se representan por una secuencia de números denominados muestras. Por lo que se puede tener una señal discreta en el tiempo pero continua en amplitud, o viceversa discreta en amplitud y continua en el tiempo.
- **Señal digital:** Son señales que sólo toman valores concretos en instantes determinados. Su amplitud viene dada por un código que se representa físicamente mediante señales con sólo dos niveles de tensión (1 y 0).

Por otra parte, las señales también se pueden clasificar dependiendo de la disposición física de las terminales en las que se presentan, por lo que las señales pueden ser unipolares o bipolares

- **Señales unipolares:** Este tipo de señales se miden entre una terminal y otra de referencia. Se denominan señales unipolares puestas a tierra aquellas cuya terminal de referencia está conectada a tierra. Si la terminal de referencia es independiente de tierra, se tiene una señal flotante; las terminales se pueden invertir, o una de ellas se puede conectar a tierra sin que se altere el circuito. Si entre la terminal de referencia y tierra existe una tensión, se dice que tiene una tensión en modo común y no se

podrá conectar a tierra ninguna de las terminales. En la figura 2.10 se ilustran las señales unipolares descritas.

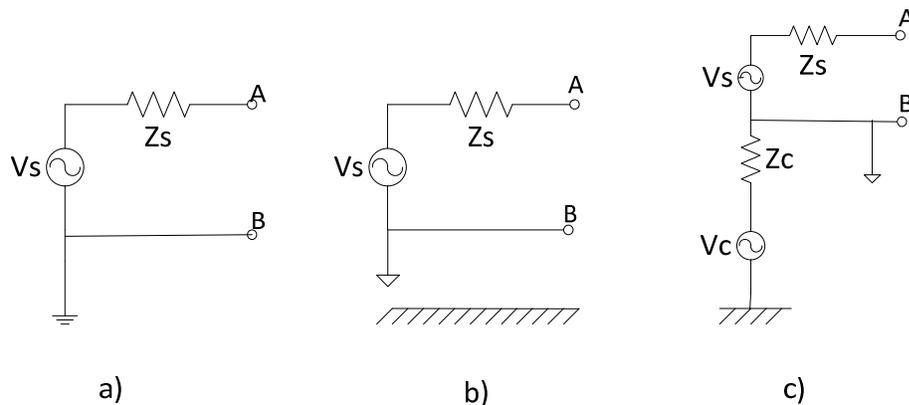


Figura 2.10. Señales unipolares: a) Unipolar puesta a tierra, b) Unipolar flotante, c) Unipolar con tensión en modo común.

- Señales bipolares o diferenciales:** La medición entre este tipo de señales se realiza entre dos terminales independientes de la terminal de referencia, que a su vez puede estar o no conectada a tierra, la polaridad con que se tome la señal es irrelevante, ya que sólo cambia el signo. Existen también tres posibilidades, las cuales son: señal diferencial puesta a tierra, flotante o con tensión en modo común. El punto de referencia para las señales flotantes puede conectarse a tierra; para las señales con tensión en modo común, no se puede conectar a tierra ninguna terminal, ni si quiera la de referencia. En la figura 2.11 se ilustran las señales bipolares descritas.

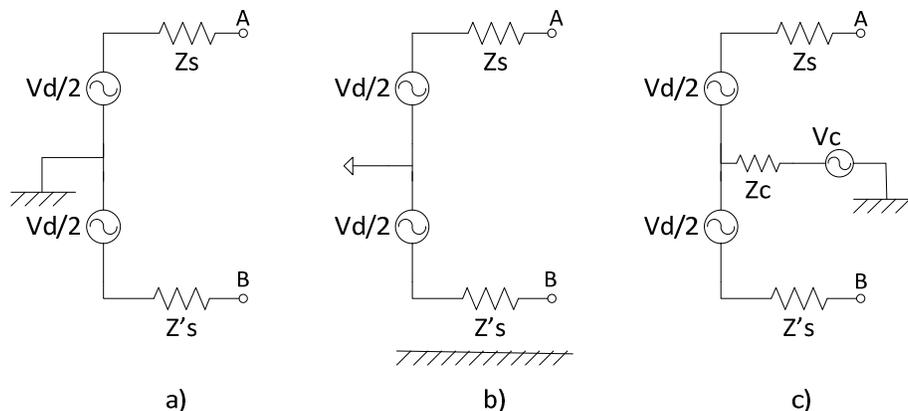


Figura 2.11. Señales bipolares. a) Bipolar puesta a tierra, b) Bipolar flotante, c) Bipolar con tensión en modo común.

- Señales de alta y baja impedancia:** Al conectar dos elementos de un sistema electrónico hay dos posibles situaciones: La primera sería, que la tensión o corriente de entrada de un elemento coincida con la correspondiente a la salida del elemento

precedente, y la segunda, que se transfiera la máxima potencia de un elemento a otro. En ambos casos es necesario adaptar la impedancia de entrada del circuito a conectar.

En el primer caso y para evitar una atenuación en la tensión de la señal, se debe tomar en cuenta la figura 2.12, la cual ilustra la conexión entre un circuito A y un circuito B, en donde la tensión V_o debe ser la misma en Z_m .

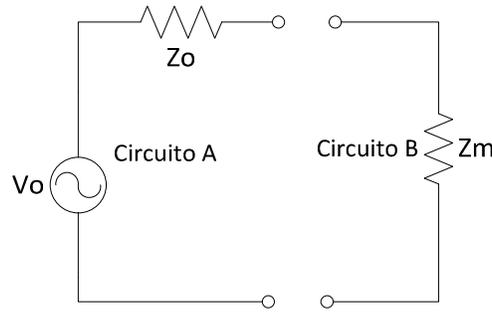


Figura 2.12. Medición de tensión en Z_m .

Del análisis del circuito anterior se obtiene que la tensión en la impedancia de entrada del circuito B esté dada por la ecuación 2.10.

$$V_m = \frac{Z_m}{Z_m + Z_o} V_o \quad (2.10)$$

Por lo que para que V_o tenga aproximadamente el mismo valor que V_m , el valor de la impedancia de entrada, Z_m del circuito B, tiene que ser mucho mayor que la impedancia de salida Z_o del circuito A.

Por otra parte, en el caso de que se desee evitar una atenuación en la corriente de la señal, se debe tomar en cuenta la figura 2.13, que ilustra la conexión entre un circuito A y un circuito B, en donde la corriente i_o tiene que ser la misma en la impedancia Z_m .

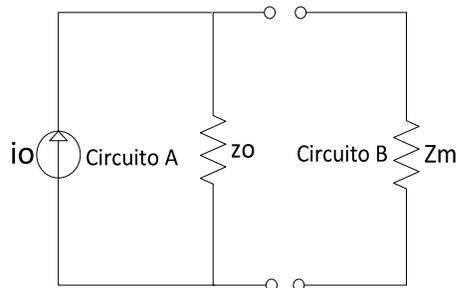


Figura 2.13. Medición de corriente en Z_m .

Del análisis del circuito anterior, se obtiene que la corriente en la impedancia de entrada del circuito B, esté dada por la ecuación 2.11.

$$i_m = \frac{Z_o}{Z_m + Z_o} i_o \quad (2.11)$$

Por lo que para que i_o tenga aproximadamente el mismo valor que i_m , el valor de la impedancia de entrada, Z_m del circuito B, tiene que ser mucho menor que la impedancia de salida Z_o del circuito A.

En el segundo caso, en donde se desea transferir la máxima potencia de un elemento a otro, es necesario considerar la figura 2.14, en donde se ilustra la conexión entre un circuito A y una resistencia R_m para transferirle la máxima potencia.

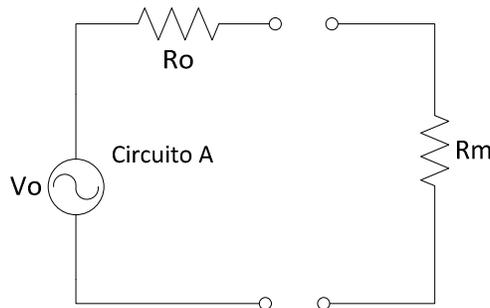


Figura 2.14. Conexión entre un circuito A y una resistencia R_m .

Para obtener el valor de la resistencia R_m que haga máxima la potencia que se le entrega, es necesario considerar la corriente que pasa por esta misma, la cual se muestra en la ecuación 2.12. Por otra parte, la potencia suministrada en la resistencia R_m se obtiene mediante la expresión 2.13.

$$i = \frac{V_o}{R_m + R_o} \quad (2.12)$$

$$p = R_m * \left(\frac{V_o}{R_m + R_o} \right)^2 \quad (2.13)$$

La ecuación 2.13 tiene que ser derivada e igualada a cero para obtener el valor de R_m que maximiza la potencia, en donde se debe considerar que los valores de V_o y R_o serán constantes con la finalidad de que la expresión quede en términos de R_m , lo cual se muestra en la ecuación 2.14.

$$\frac{dp}{dRm} = V_o^2 \frac{(Ro + Rm)^2 - 2(Ro + Rm)Rm}{(Rm + Ro)^4} \quad (2.14)$$

Al igualar 2.14 con cero se obtiene la expresión 2.15

$$(Ro + Rm)^2 - 2(Ro + Rm)Rm = 0 \quad (2.15)$$

Al resolver la ecuación 2.15, se obtiene que $Rm = Ro$. Por lo tanto, para lograr una máxima transferencia de potencia a la carga, el valor de Rm tendrá que ser el mismo valor que Ro .

2.2.2. Operaciones con señales

La información de una señal está en el valor de alguno de sus parámetros, definidos en el dominio de las amplitudes o en el de las frecuencias. En los sistemas de adquisición de señales lo habitual es que la información esté en la amplitud, pero además, cuando las señales se consideran en su contexto surge la necesidad de otras operaciones. Por ejemplo: si se desea medir la temperatura media en el aire, se puede partir de la señal de varios sensores de temperatura y calcular su media; o bien, para calcular la velocidad de un avión basta con integrar la señal de un sensor que mida su aceleración.

Con base en lo anterior, se pueden realizar operaciones con señales analógicas y digitales.

- Operaciones con señales analógicas

Se clasifican en lineales y no lineales, es decir si cumplen o no con el principio de superposición, algunos ejemplos de estas operaciones son: suma, producto, división, potencia, integración, derivación, comparación, recorte, valor absoluto, valor máximo, valor medio.

- Operaciones con señales digitales

Para este tipo de señales se distinguen dos tipos de operaciones:

- ✓ Sobre la información representada por el código (aritméticas, lógicas, registros, conversiones). Se pueden realizar por medio de un procesador o un elemento específico.
- ✓ Sobre los parámetros físicos (frecuencia, fase, tiempo, amplitud). En la mayoría de estas operaciones sólo podemos utilizar componentes específicos.

2.2.3. Interfaz con los sensores analógicos

Como se mencionó en la sección 2.1, existen diversos tipos de sensores que se clasifican en pasivos y activos. La interfaz con un sensor pasivo y un convertidor analógico digital requiere de una etapa de amplificación y adaptación de impedancias, mientras que la salida de algunos sensores activos basados en una variación de impedancia es en forma de señal alterna modulada en amplitud, su interfaz debe incluir un CAD (Convertidor Analógico Digital) especial.

La mayoría de los sensores, sin embargo, ofrecen simplemente una variación de impedancia del tipo $Z = Z_0(1+x)$, por lo que para obtener una señal eléctrica hacen falta circuitos previos al amplificador (divisores de tensión, puentes de impedancias), o bien se puede incorporar en un oscilador cuya frecuencia dependa de Z . La primera solución es más habitual a frecuencias menores de 1MHz.

La generación y amplificación de tensiones es normalmente más simple que la de corrientes, y por ello el acondicionamiento de señal tiende a orientarse en esta dirección.

2.2.4. Amplificación de señales

La señal de salida de un sensor puede ser tan pequeña, como la de un geófono, o moderadamente grande, como la generada por un sensor de banda ancha. Si, por ejemplo, la velocidad de la tierra es una onda sinusoidal con una frecuencia de 5 Hz y una amplitud de 1nm/s; un geófono estándar de 4.5 Hz con una constante de generador de 30 V/ms^{-1} , daría una salida de 30 nV y un sensor de banda ancha con una constante de generador de 1500 V/ms^{-1} , daría una salida de 1500 nV. Estos voltajes deben ser amplificados tomando en cuenta que el margen de entrada de un CAD es de 0 a 5 V o de 0 a 10 V. En consecuencia, las señales procedentes de una gran cantidad de sensores, de los divisores de voltaje y los puentes de impedancias, que manejan señales pequeñas requieren de una amplificación; sin embargo, el tema de amplificación no es tan simple, ya que se requiere considerar un amplificador de bajo ruido, el cual además abarque el extenso rango de frecuencias utilizado en sismología.

Un amplificador en su operación produce ruido, no obstante, no es la única fuente de ruido, ya que el acoplamiento entre el sensor y el amplificador introduce ruido a la señal. Si se usan cables largos, el ruido inducido por un ambiente eléctricamente ruidoso puede también producir señales no deseadas en la entrada de los amplificadores. Con la finalidad de eliminar este ruido, se emplean amplificadores de instrumentación.

El amplificador de instrumentación es un circuito que se encarga de amplificar la diferencia entre dos señales de entrada, y rechazar cualquier señal que sea común a ambas

señales. Estos circuitos son utilizados principalmente para amplificar señales diferenciales muy pequeñas en muchos procesos industriales, medición, adquisición de datos y aplicaciones médicas. Cabe mencionar que este tipo de amplificadores se pueden construir a partir con elementos discretos, o bien se pueden encontrar en un solo encapsulado.

Las principales características de un amplificador de instrumentación son las siguientes:

- Ganancia: seleccionable, estable y lineal.
- Entrada diferencial: con un nivel alto de rechazo en modo común (*CMRR: Common Mode Rejection Ratio*).
- Impedancia de entrada alta.
- Impedancia de salida baja.

La configuración más utilizada como amplificador de instrumentación está formada por tres amplificadores operacionales (figura 2.15), en donde los amplificadores A1 y A2 constituyen una etapa de pre-amplificación, y el amplificador A3 forma una etapa diferencial.

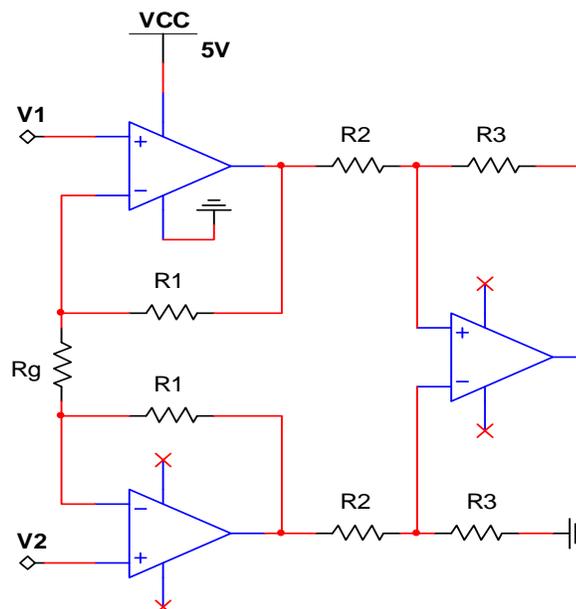


Figura 2.15. Configuración de amplificador de instrumentación.

Del análisis del amplificador de instrumentación mostrado en la figura 2.15, se obtiene que el voltaje en la salida está definido por la siguiente expresión.

$$V_{out} = \frac{R3}{R2} \left(1 + \frac{2R1}{Rg} \right) (V2 - V1) \quad (2.16)$$

2.3. Conversión Analógica Digital

En la actualidad, el procesamiento de señales se hace por medio de microcontroladores, que permite el tratamiento de los datos a grandes velocidades, así como la ejecución de otras tareas.

En el medio en el que vivimos, las variables físicas como temperatura, presión y velocidad, por citar algunos ejemplos, son continuas en el tiempo y para poder medirlas mediante un dispositivo electrónico digital se necesita realizar una conversión del dominio analógico al digital.

En la figura 2.16, se presenta el diagrama de bloques que ilustra el proceso de conversión analógica-digital.

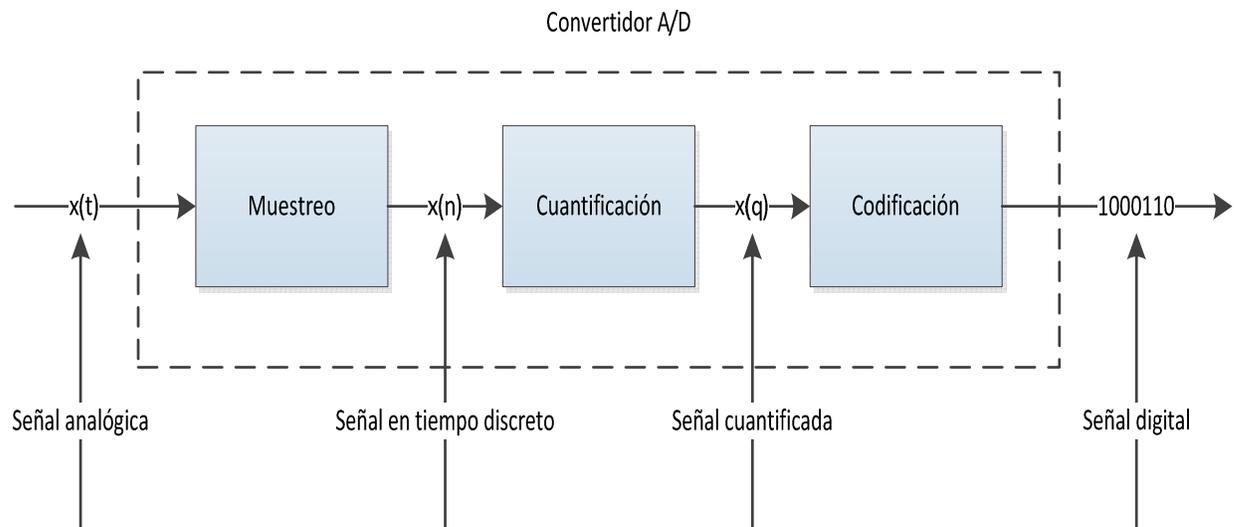


Figura 2.16. Diagrama de bloques del Convertidor A/D.

El **muestreo** consiste en tomar muestras periódicas de la amplitud de la onda. 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 y está establecida por el teorema de Nyquist¹, que indica que la frecuencia de muestreo F_s debe ser por lo menos el doble de la frecuencia máxima F_m de la señal a muestrear.

$$F_s \geq 2F_m \quad (2.17)$$

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

¹ PROAKIS, John G. et. al., **Digital Signal Processing**, 3a. ed., Prentice-Hall, United State of America, p. 29

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, que 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.

2.3.1. Parámetros generales de un convertidor analógico digital

Existen algunas características que deben de ser consideradas para la correcta elección de un convertidor analógico digital, tales como: la resolución, la frecuencia de muestreo, la máxima entrada de escala completa, el rango dinámico, la monotonicidad, la impedancia de entrada, el offset, el tiempo de conversión, el tiempo de adquisición, el tiempo de asentamiento, el *slew rate*, la precisión y la linealidad.

- *Resolución*, indica el número de valores discretos que el convertidor puede producir sobre el intervalo de valores analógicos. Es usualmente expresada en *bits*, por lo tanto el número de valores o niveles discretos es usualmente una potencia de dos. Por ejemplo, un convertidor analógico digital de 8 *bits* puede codificar una señal en 256 niveles, ya que $2^8 = 256$.
- *Frecuencia de muestreo*, es el número de muestras que se pueden tomar por segundo. Este valor se expresa en Hertz o en muestras por segundo (mps).
- *Máxima entrada de escala completa*, se refiere a la máxima entrada de voltaje que el dispositivo puede convertir sin que éste se sature.

- *Rango dinámico*, se define como la relación entre el valor más grande y el valor más pequeño que el CAD (Convertidor Analógico Digital) puede aceptar. Por ejemplo, si el valor más grande es 4V y el más chico 1V, el rango dinámico es $4/1 = 4$; en decibeles, $20 \log (4/1) = 12 \text{ dB}$. En algunos convertidores, los *bits* menos significativos contienen ruido, por lo que el rango dinámico se define como la relación entre la entrada más grande de voltaje y el nivel de ruido del convertidor.
- *Monotonicidad*, se considera que un convertidor es monotónico cuando a un incremento de tensión en la entrada le corresponda un incremento en la salida, y para una disminución de la entrada, se tiene el correspondiente descenso a la salida.
- *Impedancia de entrada*, este parámetro idealmente debe ser tan grande como sea posible para no alterar la señal proveniente del sensor. Un valor típico es $1\text{M}\Omega$.
- *Offset*, es el nivel de voltaje de DC presente en la salida del convertidor analógico digital debido a la desigualdad de los componentes a la entrada, esto para cuando el voltaje en la entrada es cero.
- *Tiempo de conversión*, es el tiempo mínimo requerido para efectuar una conversión. Típicamente se expresa como la máxima frecuencia de muestreo.
- *Tiempo de adquisición*, es el tiempo durante el cual el sistema de muestreo y retención (sample & hold) debe permanecer en estado de muestreo (*sample*). Esto para asegurar que el consiguiente estado de retención (*hold*) esté dentro de la banda de error especificada para la señal de entrada.
- *Tiempo de asentamiento*, es el intervalo de tiempo entre la retención de la señal y el definitivo asentamiento de la señal.
- *Slew rate*, es la velocidad a la cual el valor de la salida del sistema *sample & hold*, converge al valor muestreado.
- *Precisión*, se refiere a la dispersión del conjunto de valores obtenidos de mediciones repetidas de una magnitud por medio del convertidor analógico digital. En donde cuanto menor sea la dispersión de los datos, mayor será la precisión de la lectura.
- *Linealidad*, se refiere a que la señal de salida del dispositivo conservará las características (amplitud y frecuencia) de la señal de entrada.

2.3.2. Aliasing

De acuerdo con el teorema de Nyquist, descrito por la ecuación 2.16, se establece que la frecuencia con la que se debe muestrear una señal continua, para poder recuperarla sin distorsión, debe ser al menos dos veces la frecuencia de la señal. Por ejemplo, en la figura 2.17 se ilustra una señal sinusoidal continua de 1kHz, la cual es muestreada a una frecuencia de 3kHz, representada por la señal punteada sobre la señal original.

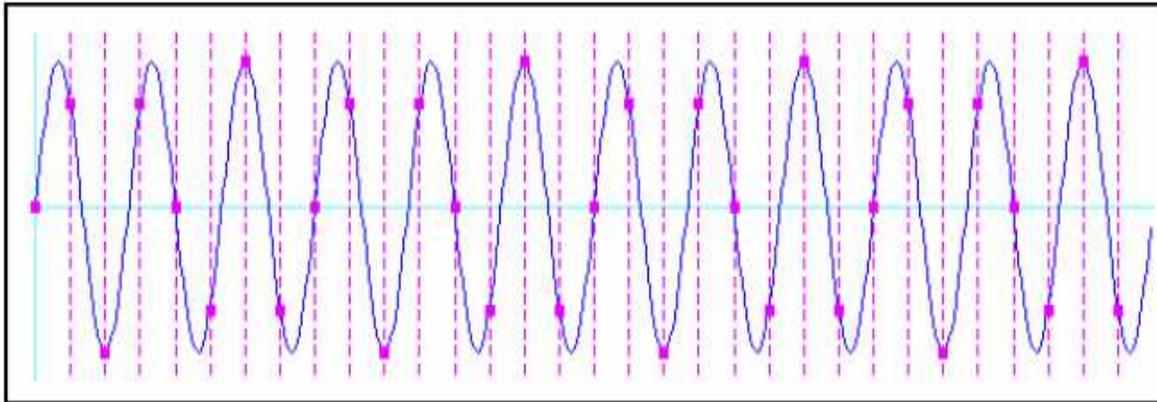


Figura 2.17. Señal sinusoidal muestreada por arriba de la frecuencia de Nyquist.

De la figura 2.17, lo que se puede observar es que la señal de 1kHz, al ser muestreada a una frecuencia mayor que la de Nyquist, se puede reconstruir a partir de sus muestras sin perder información. Por el contrario, si se muestrea a una frecuencia menor que la de Nyquist, más de una señal sinusoidal puede ser representada mediante las muestras obtenidas y existirá una ambigüedad en cuanto a la reconstrucción de la señal. En la figura 2.18 se ilustra una señal sinusoidal continua de 1kHz, muestreada a una frecuencia de 500 Hz y representada, esta última, por los puntos unidos mediante la línea a baja frecuencia.

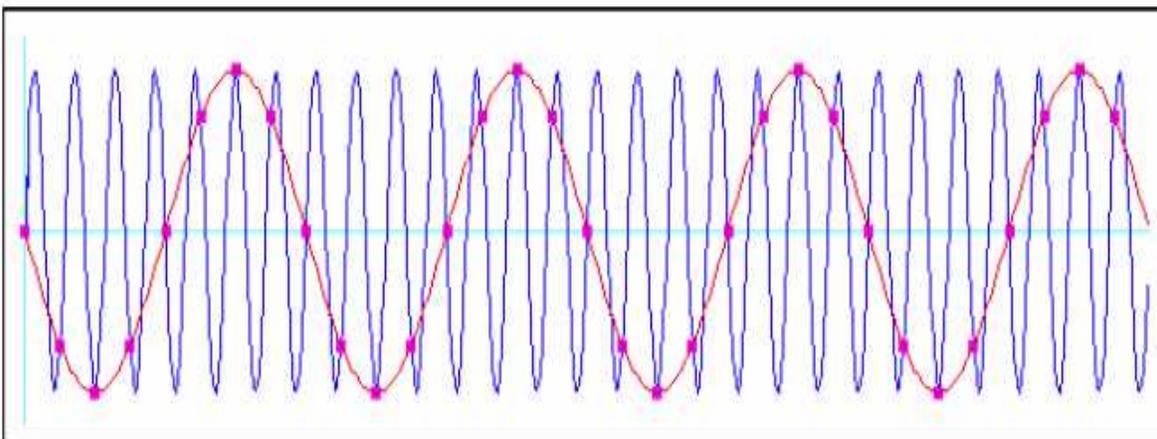


Figura 2.18. Señal sinusoidal muestreada por debajo de la frecuencia de Nyquist.

El análisis anterior se ha realizado en el dominio del tiempo, sin embargo, en el dominio de la frecuencia también se presenta el fenómeno de *aliasing*. En la figura 2.19 se ilustra el espectro de una señal de frecuencia f_m limitada en banda.

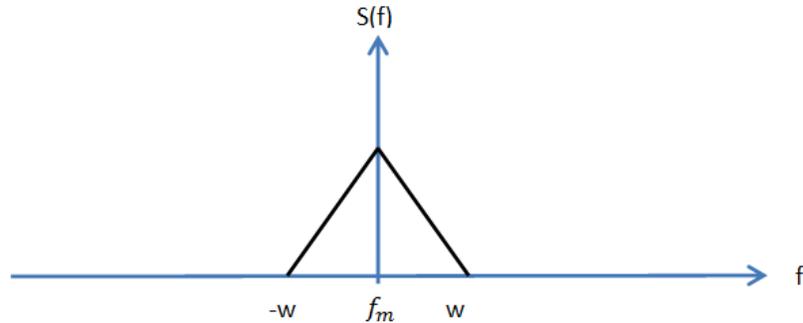


Figura 2.19. Espectro limitado en banda.

Cuando se muestrea dicha señal a una frecuencia mayor que la de Nyquist, su espectro aparece repetido y centrado en múltiplos de la frecuencia de muestreo f_s , como se ilustra en la figura 2.20. Además, podemos observar que no se presenta traslape en los espectros.

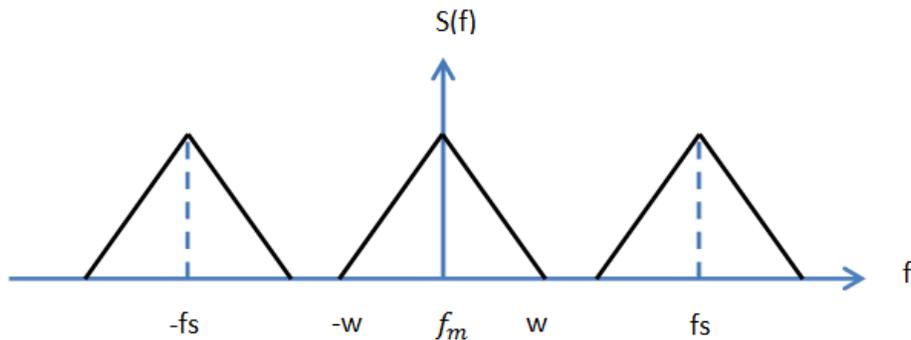


Figura 2.20. Señal muestreada sin aliasing.

En la figura 2.21 se ilustra una señal muestreada con una frecuencia menor a la definida por Nyquist, por lo que podemos observar que las réplicas del espectro se traslapan, dando origen a la pérdida de información. Al pasar al dominio del tiempo, la señal no corresponde con la original, como se vio en la figura 2.18.

Para evitar el fenómeno de *aliasing* en la digitalización de señales analógicas, se debe aumentar la frecuencia de muestreo, como lo indica el criterio de Nyquist, pero además, en la práctica, la señal de interés generalmente contiene componentes de frecuencias mayores a la frecuencia deseada, que contribuyen a presentar *aliasing*; para evitar este problema se debe diseñar un *filtro* anti *aliasing* (filtro pasa bajas), el cual permite el paso de las componentes de baja frecuencia y atenúa las de alta.

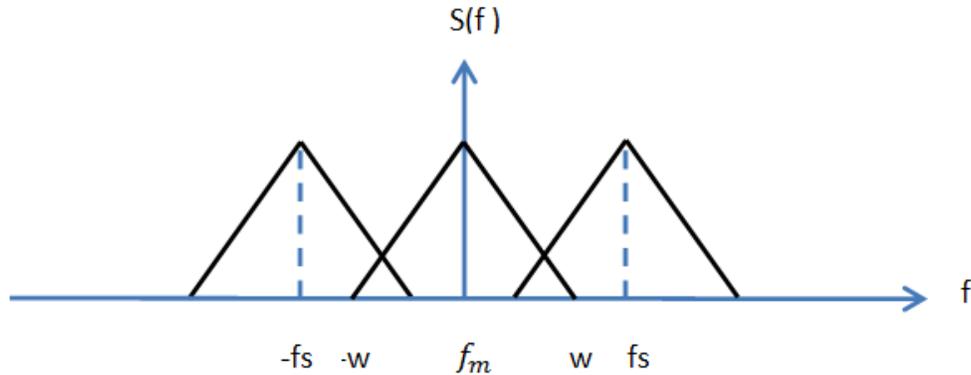


Figura 2.21. Señal muestreada con aliasing.

En la figura 2.22 se muestra el comportamiento de este filtro, en donde se dejan pasar todas las frecuencias menores a f_1 y se eliminan las frecuencias mayores a f_2 . La región comprendida entre f_1 y f_2 se denomina banda de transición, y el comportamiento real de esta banda es de una atenuación gradual de las frecuencias de entrada. Las frecuencias de interés son las menores a f_1 , pero las señales de la banda de transición podrían causar *aliasing*; por tal motivo, en la práctica se utiliza una frecuencia de muestreo mucho mayor a dos veces la máxima frecuencia en la banda de transición.

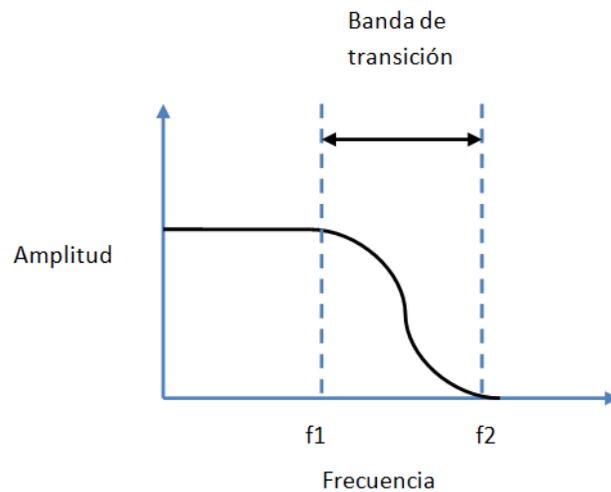


Figura 2.22. Comportamiento de filtro antialiasing.

2.3.3. Convertidor analógico digital Sigma – Delta

El convertidor analógico digital con arquitectura sigma – delta es un dispositivo que opera por sobremuestreo, entendiendo como tal, el uso de una frecuencia de muestreo mayor que la establecida por Nyquist, además, de que se integra un modulador sigma-delta, que proviene de una derivación de la modulación delta, y un bloque puramente digital

que se encarga de filtrar el ruido que la señal contenga a la salida del modulador y reducir la tasa de muestreo mediante un proceso denominado diezmo.

La ventaja de la arquitectura sigma – delta, a comparación de otras arquitecturas disponibles para la conversión analógica digital, recae en que se reducen los requerimientos de circuitería analógica; como caso particular, el uso de un filtro limitador de banda a la entrada del convertidor, también llamado filtro anti *aliasing*, el cual se reduce a un filtro pasivo de primer orden. Además de que no se necesitan circuitos de muestreo y retención (*sample & hold*), ya que este proceso es inherente en la modulación sigma delta. Estas ventajas hacen que los convertidores A/D, con este tipo de tecnología, sean de gran interés para la realización de interfaces analógico digitales de altas prestaciones, en tecnologías micrométricas con bajas tensiones de polarización.

En la figura 2.23 se muestra de forma general el diagrama de bloques que conforman a un convertidor sigma – delta.

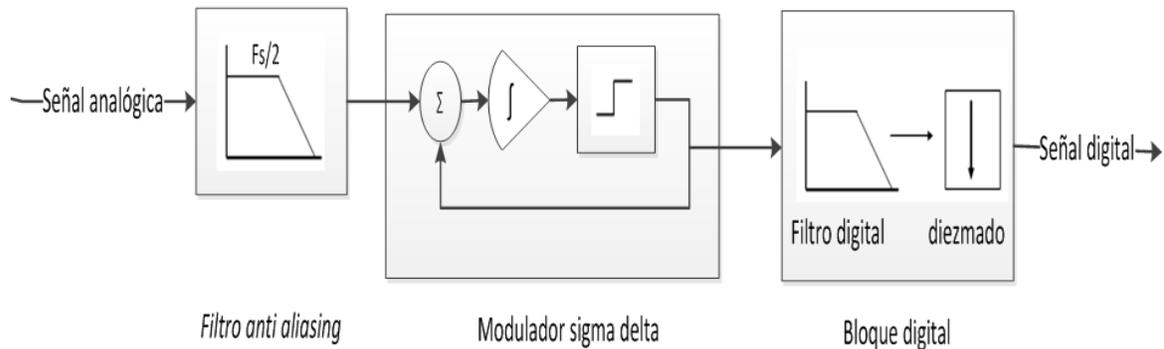


Figura 2.23. Diagrama de bloques de un convertidor A/D sigma – delta.

A continuación se describirá el funcionamiento de cada uno de los bloques que integran al convertidor analógico digital sigma – delta.

Filtro anti – aliasing. Para cualquier tipo de convertidor analógico digital, es necesario colocar en su entrada un filtro, que limite la banda de la señal analógica proveniente de los transductores, eliminando las componentes espectrales que se encuentren por encima de la frecuencia de muestreo.

Cuando una señal analógica es muestreada con el criterio de Nyquist, se debe diseñar un filtro anti – aliasing que tenga una banda de transición lo más pequeña posible, con la finalidad de no perder información; sin embargo, en la práctica, el diseñar este tipo de filtros trae como consecuencia utilizar elementos activos y pasivos que permitan la respuesta deseada, haciendo más compleja la circuitería del filtro.

En la figura 2.24 se presenta el espectro de una señal muestreada con el criterio de Nyquist; en donde la banda de transición del filtro a diseñar está limitada entre las

frecuencias f_m y f_1 , siendo f_m la frecuencia máxima, f_1 es la frecuencia de interés y f_s es la frecuencia de muestreo.

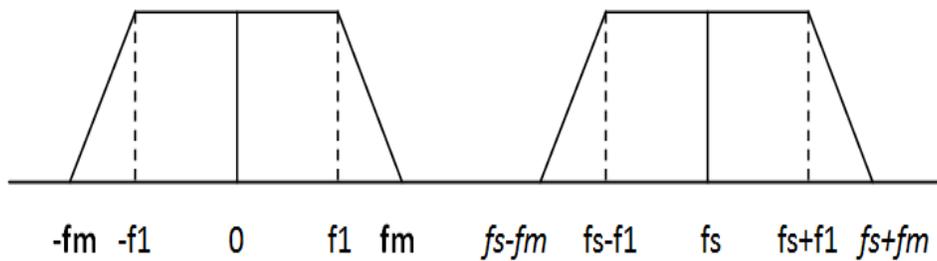


Figura 2.24. Espectro de señal muestreada con el criterio de Nyquist.

Como se puede observar de la figura anterior, la respuesta del filtro debe de ser lo más rápida posible, ya que los espectros son muy cercanos, y si el filtro no tiene esta respuesta se pueden traslapar los espectros, dando como resultado la pérdida de información. Para evitar esta pérdida, el sobremuestreo de una señal hace que los espectros estén más separados, por lo que permite que la respuesta del filtro sea más lenta, lo que involucra que se pueda construir por medio de un filtro RC. En la figura 2.25 se ilustra una señal sobremuestreada, en donde se presenta la banda de transición para el filtro a diseñar.

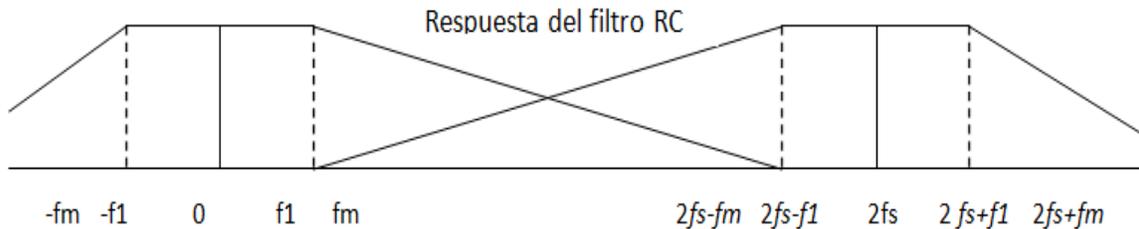


Figura 2.25. Espectro de una señal sobremuestreada.

El sobremuestreo de la señal no sólo se restringe al diseño del filtro anti – *aliasing*, sino que es uno de los principios de operación del modulador sigma – delta, por lo que a continuación se explicará el funcionamiento de dicho modulador.

Modulador Sigma – Delta. Como se mencionó anteriormente, el modulador sigma – delta es una variación de la modulación delta.

La modulación Delta consiste en comparar la señal analógica con una sucesión de pulsos de amplitud, los cuales son crecientes mientras la amplitud de esta sucesión se encuentra por debajo de la amplitud de la señal analógica, y es decreciente cuando la amplitud de los pulsos supera la amplitud de la señal analógica, tal como se muestra en la figura 2.26.

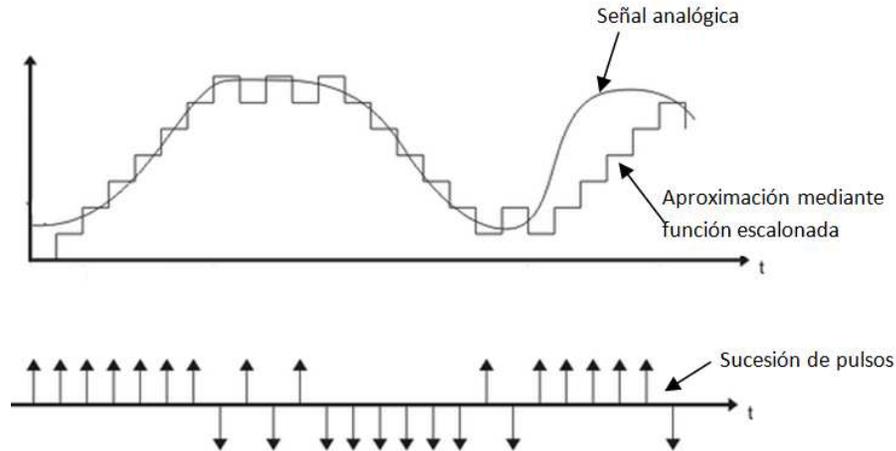


Figura 2.26. Modulación Delta.

Como se observa de la figura 2.26, la modulación delta aproxima a la señal analógica mediante una función escalonada, por lo que la señal analógica y la aproximación, es cuantificada en solo dos niveles, por medio de un comparador. El cambio de la señal debe ser relativamente lento en comparación con la tasa de muestreo. Este requerimiento implica que la señal debe ser sobremuestreada.

Con base en lo anterior, se obtiene de esta manera una secuencia de unos y ceros que representan a la señal modulada. Por otra parte, para reconstruir la señal analógica de la señal modulada, es necesario un demodulador delta, el cual está formado por un integrador que actúa como un decodificador, y para quitar el ruido de cuantificación es necesario que la señal pase por un filtro pasa bajas. En la figura 2.27 se muestra el diagrama de bloques del modulador y demodulador delta.

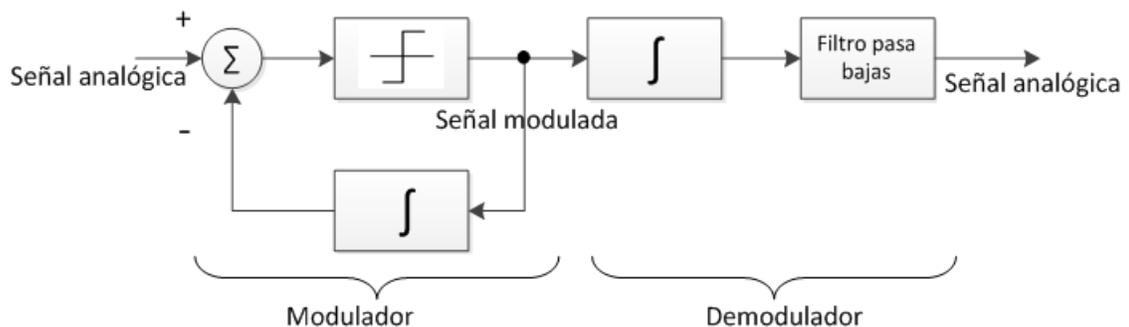


Figura 2.27. Modulador y demodulador delta.

Como es sabido, la integración es una operación lineal, que cumple con la propiedad de aditividad y homogeneidad, por lo que el bloque integrador del demodulador delta se puede poner antes del modulador sin alterar las características generales de entrada y salida

(figura 2.28). El hecho de poner el integrador antes del modulador delta trae como consecuencia el término de modulador sigma – delta.

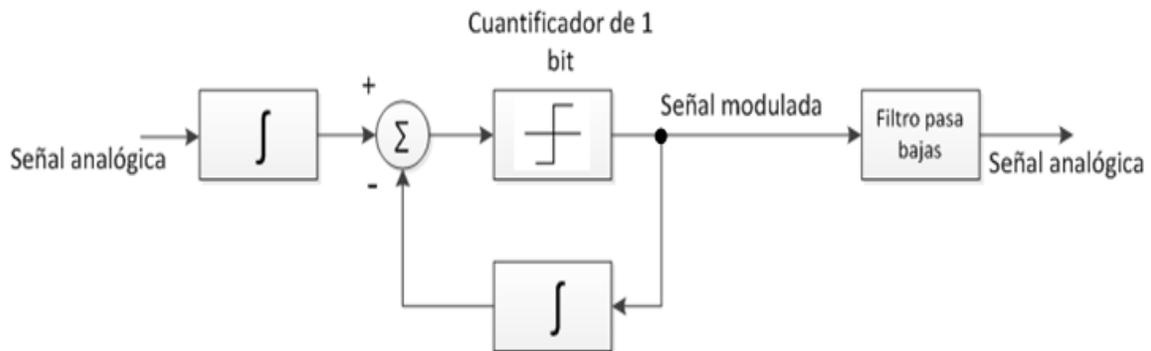


Figura 2.28. Integrador colocado antes del modulador.

De la figura 2.28 se puede observar que existen dos bloques integradores antes del punto de comparación, por lo que al ser operaciones lineales, se pueden reducir estos bloques tal como se ilustra en la figura 2.29, dando origen al diagrama de bloques del modulador sigma – delta.

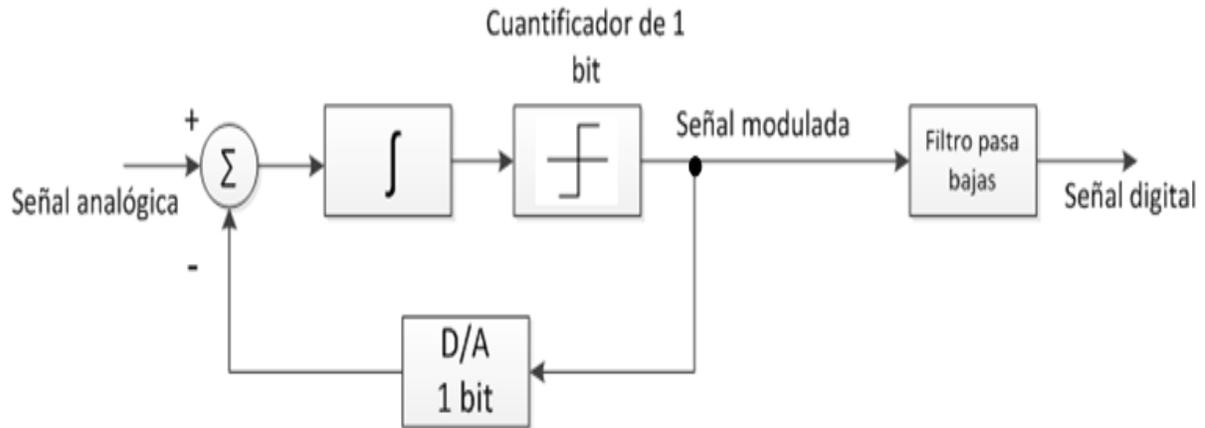


Figura 2.29. Diagrama de bloques del modulador sigma – delta.

Con la finalidad de entender el funcionamiento del modulador sigma – delta, mostrado en la figura 2.29, se realizará el análisis en el dominio de Laplace, en donde el comparador que realiza la función de cuantificación será substituido por un punto suma, al que se le agregará el ruido de cuantificación $N(s)$. El integrador será substituido por su equivalente en el dominio de Laplace, y el bloque del convertidor digital – analógico de 1 bit puede omitirse en este análisis, ya que éste sólo se emplea para convertir la señal cuantificada a una señal analógica, para poder ser comparada con la señal analógica de entrada. En la figura 2.30 se ilustra el diagrama de bloques del modulador sigma – delta en el dominio de Laplace.

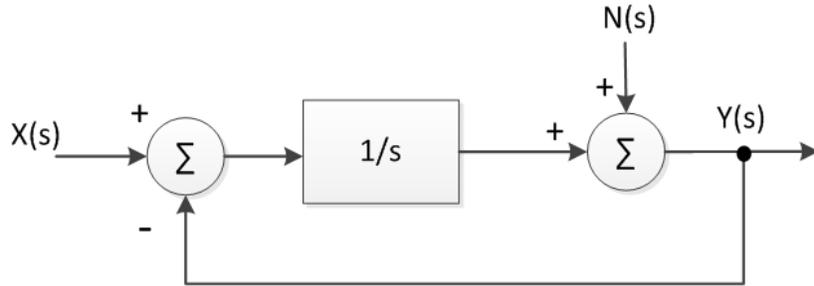


Figura 2.30. Modulador sigma delta en el dominio de Laplace.

De la figura 2.30, se puede observar que al hacer $N(s) = 0$ y $X(s) \neq 0$ se obtiene la salida $Y(s)$ en función de la señal de entrada $X(s)$, como se indica en la ecuación (2.18).

$$Y(s) = \frac{1}{s} [X(s) - Y(s)] \quad (2.18)$$

De la ecuación 2.18 se obtiene la función de transferencia en función de la señal de entrada, expresada en la ecuación 2.19.

$$\frac{Y(s)}{X(s)} = \frac{\frac{1}{s}}{1 + \frac{1}{s}} = \frac{1}{s + 1} \quad (2.19)$$

Como se puede observar, la función de transferencia obtenida en la expresión 2.19 corresponde a la función de transferencia de un filtro pasa bajas.

Por otra parte, al hacer $X(s) = 0$ y $N(s) \neq 0$, de la figura 2.30, se obtiene la salida $Y(s)$ en función del ruido de cuantificación $N(s)$, tal y como se muestra en la ecuación 2.20.

$$Y(s) = -Y(s) \frac{1}{s} + N(s) \quad (2.20)$$

Por lo que de la ecuación 2.20 se puede obtener la función de transferencia en función del ruido de cuantificación, expresada en la ecuación 2.21.

$$\frac{Y(s)}{N(s)} = \frac{1}{1 + \frac{1}{s}} = \frac{s}{s + 1} \quad (2.21)$$

La función de transferencia obtenida en la expresión 2.21 corresponde a la función de transferencia de un filtro pasa altas.

Del análisis anterior se puede concluir que a medida que el bucle integra el error entre la señal muestreada y la señal de entrada, el modulador sigma – delta se comporta como un filtro pasa bajas para la señal de entrada y como un filtro pasa altas para el ruido

de cuantificación. En otras palabras, la señal de entrada se deja sin cambios siempre y cuando su frecuencia no exceda la frecuencia de corte del filtro pasa bajas. Por otra parte, el lazo de realimentación empujará al ruido en una banda de frecuencia más alta, lo cual se conoce como “*noise shaping*”. Por lo que el sobremuestreo de la señal de entrada hace que el ruido de cuantificación sea distribuido en un ancho de banda más amplio, trayendo como consecuencia que la densidad de ruido en la banda de interés (banda base) disminuya significativamente. Cabe mencionar que la salida del modulador sigma – delta es una secuencia de datos de 1 *bit* de alta frecuencia.

Es importante indicar que el modulador descrito corresponde a un modulador de primer orden, por lo que para obtener moduladores de tercer orden (que es lo utilizado en los convertidores con este tipo de tecnología), bastará con conectar en cascada el número de moduladores de primer orden que se desee, sin embargo, en las hojas de especificaciones de este tipo de convertidores es difícil identificar los moduladores de primer orden, por lo que otra forma para saber el orden del modulador es contando el número de bloques integradores que este tenga. En la figura 2.31 se ilustra el diagrama de bloques de un modulador sigma – delta de tercer orden.

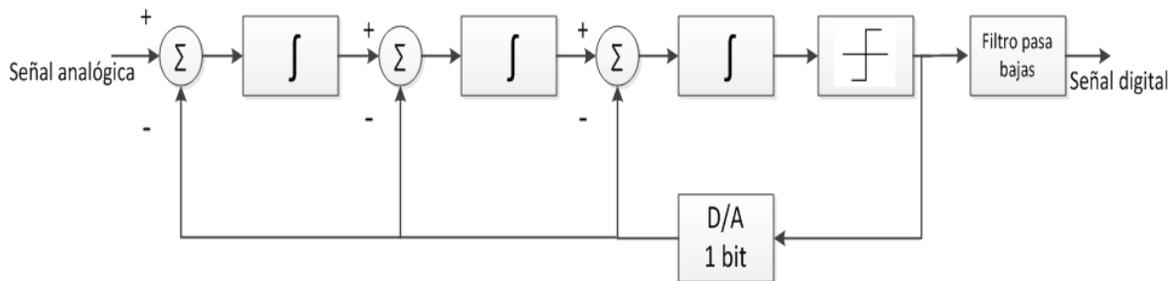


Figura 2.31. Modulador sigma – delta de tercer orden.

El funcionamiento de los moduladores de orden superior es exactamente el mismo que el descrito para el modulador de primer orden, sólo que la señal que pasa al siguiente bucle será el término de error del bucle actual, siendo este error la diferencia entre la salida del integrador y la salida del cuantificador. Por lo tanto, los moduladores de orden superior, llevan a cabo un mayor número de operaciones de diferencia del error producido por el cuantificador, y como consecuencia de esto se tendrá una mayor atenuación a bajas frecuencias del ruido de cuantificación, eliminando más ruido en la banda base.

En la figura 2.32 se muestra el efecto que produce un modulador de primer, segundo y tercer orden a la densidad de ruido en la banda de interés (banda base). Como se observa, para un modulador de tercer orden se elimina mayor densidad de ruido en comparación de un modulador de primer orden.

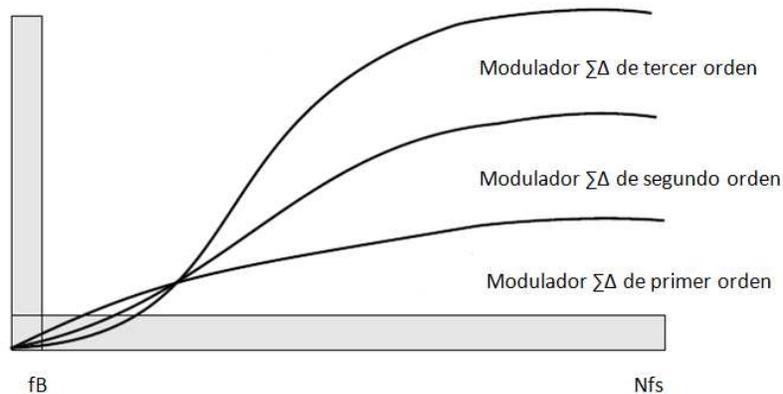


Figura 2.32. Efecto del modulador sigma-delta a la banda base f_B .

Una vez que se ha explicado el funcionamiento del modulador sigma – delta, es conveniente explicar el funcionamiento del último bloque que conforma al convertidor analógico digital sigma – delta.

Bloque digital: Como se mencionó anteriormente, el modulador sigma – delta tiene como principal objetivo suprimir el ruido de cuantificación en la banda base, por lo que este ruido es colocado en una banda de alta frecuencia. Teniendo en cuenta lo anterior, el último bloque que forma al convertidor A/D está integrado por circuitería puramente digital, la cual filtra nuevamente la señal proveniente del modulador, eliminando los posibles residuos de ruido en la banda base, lo cual trae como consecuencia que la densidad de ruido en la banda base sea casi nula y se aumente la resolución efectiva (*ENOB: Effective number of bits*) de la salida digital.

Una vez que el ruido de cuantificación ha sido totalmente filtrado, se tiene una redundancia de datos por efectos del sobremuestreo, por lo que es necesario reducir la tasa de muestreo (proceso de diezmo) a la tasa de muestreo de Nyquist. Con esto último se obtiene una reducción significativa de los datos, facilitando la posterior transmisión, almacenamiento, o procesamiento digital. Es importante mencionar que en este proceso de diezmo no se pierde información, sólo se elimina redundancia de datos. Por último, cuando se ha reducido la frecuencia de muestreo a la de Nyquist, es necesario proporcionar un rechazo adicional para el fenómeno de *aliasing* de la señal de entrada, esto se hace porque en ocasiones no se logra limitar totalmente en banda la señal analógica de entrada, debido a las variaciones que puedan tener los valores de los capacitores o resistores que se empleen para el filtro RC de la entrada del convertidor A/D.

En resumen, el último bloque del convertidor analógico digital sigma – delta desarrolla las funciones de remover el posible ruido de cuantificación en la banda base, reducir la tasa de muestreo a la de Nyquist, y reforzar el filtro anti *aliasing* para no perder información.

En la sección 2.3.1 se mencionaron algunos de los principales parámetros que deben ser considerados para la elección de un convertidor A/D, sin embargo, en el caso de los convertidores de sobremuestreo, estos presentan dos parámetros que caracterizan el desempeño de dichos dispositivos, estos parámetros son el ENOB y SINAD (*Signal to Noise and Distortion Ratio*).

En un sistema de medición, como en el caso de un sismógrafo, el ruido de fondo físico se puede establecer por ruido ambiental, en donde se puede incluir el tráfico vehicular en un camino cercano. El ruidos de fondo limita la medida más pequeña que se puede tomar con seguridad, cualquier amplitud medida en promedio puede ser no menos que el ruido de fondo.

El ruido de fondo es una medida de la señal creada por la suma de todas las fuentes de ruido, es decir señales no deseadas en el sistema de medición, en donde el ruido es definido como cualquier otra señal ajena a la de interés.

Dicho lo anterior, el parámetro ENOB es una forma de cuantificar la calidad de un convertidor A/D, un valor alto de este parámetro significa que los niveles de voltaje que registra el convertidor analógico digital son más precisos. El estándar IEEE-1241, el cual define la manera en como los convertidores A/D deben especificarse y probarse, utiliza el concepto de ENOB para medir el desempeño del convertidor. Por lo que el ENOB es un factor de calidad que se calcula a partir de la medición de la relación señal a ruido más distorsión (SINAD) usando la transformada rápida de Fourier (FFT).

Un convertidor analógico digital ideal es completamente lineal y simplemente cuantifica la señal de entrada. Sin embargo, en la realidad el proceso de cuantificación introduce ruido a la señal. Por lo que utilizando la potencia de la señal y la potencia de ruido, es posible obtener la relación señal a ruido (SNR: *Signal to Noise Ratio*) de la señal después de la conversión analógica digital. Con base en lo anterior, si una onda sinusoidal es utilizada como entrada, la relación señal a ruido es calculada como se indica en la expresión 2.22².

$$SNR = 1.5 \times 2^{2B} \quad (2.22)$$

En donde B es el número de bits del convertidor analógico digital. Expresando la ecuación 2.22 en dB, se obtiene la siguiente expresión.

$$SNR_{dB} \approx 1.76 \text{ dB} + 6.02B \text{ dB} \quad (2.23)$$

² Rohde & Schwarz Technical Paper, The effective number of bits (ENOB) of my R&S Digital Oscilloscope.

Resolviendo la ecuación para B, se obtiene:

$$B \approx \frac{SNR_{dB} - 1.76 \text{ dB}}{6.02 \text{ dB}} \quad (2.24)$$

La ecuación 2.24 muestra como el número de bits se puede derivar de la relación señal a ruido y proporciona la base para el cálculo del ENOB. Para el caso de los convertidores A/D ideales, el ENOB será igual a un número entero positivo real, mientras que para los convertidores A/D no ideales el ENOB puede ser cualquier número real positivo.

Los convertidores A/D ideales no existen, ya que cada convertidor incluye algunas distorsiones en la señal de entrada, como es el caso de ruido y las no linealidades que resultan de armónicos que reducen el parámetro SNR.

En la figura 2.33 se ilustra el diagrama de bloques de un convertidor A/D no ideal.

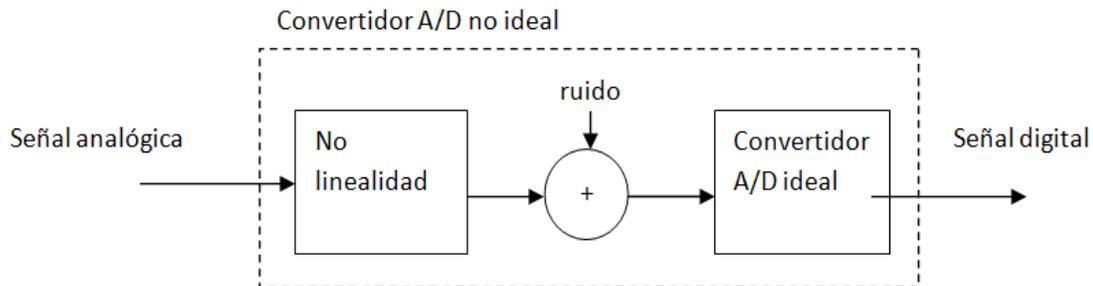


Figura 2.33. Diagrama de Bloques de un convertidor A/D no ideal.

El ruido contribuye directamente con la degradación del parámetro (SNR). Por lo tanto, y a manera de ejemplo, si un convertidor A/D de 12 bits es especificado con un ENOB de 10.5 bits, significa que a pesar de que las salidas del convertidor A/D serán de 12 bits, la SNR alcanzada corresponderá a la de un ADC ideal con 10.5 bits.

Dependiendo de cómo el ADC esté diseñado, el ENOB también puede ser dependiente de la frecuencia de entrada, por lo que las frecuencias altas puede resultar en peores no linealidades dentro del circuito y por lo tanto la degradación de la ENOB. Por tal motivo, una especificación detallada de un ADC por lo general incluye el comportamiento de la ENOB con respecto a la frecuencia de entrada.

Ahora bien, el parámetro denominado SINAD (*Signal to Noise and Distortion Ratio*) se refiere a la relación de la amplitud rms de la señal y el valor de la sumatoria de la raíz cuadrada de todas las otras componentes espectrales, incluyendo armónicos y excluyendo niveles de DC. Por lo que el parámetro SINAD es un buen indicador de rendimiento dinámico global del convertidor A/D, ya que incluye todas las componentes que pueden introducir ruido y distorsión, para una amplitud y frecuencia dada, en la

expresión (2.25) se muestra la ecuación para calcular dicho parámetro, en donde interviene la Distorsión Total Armónica (*THD: Total Harmonic Distortion*) y el ruido (*N: Noise*).

$$\text{SINAD} = \text{THD} + N \quad (2.25)$$

Con base en lo anterior, si se tiene un valor pequeño de SINAD significa que el rendimiento del sistema es deficiente.

2.4. Memorias de estado sólido

Una memoria de estado sólido es un dispositivo de almacenamiento secundario, hecho con componentes electrónicos en estado sólido, pensado para usarse en equipos informáticos en lugar de disco duro convencional. Pueden estar constituidas por unidades volátiles como la memoria SDRAM (*Synchronous Dynamic Random – Access Memory*), o por unidades no volátiles como la memoria *flash* o ROM (*Read Only Memory*).

Las memorias CMOS (*complementary Metal-Oxide Semiconductor*) pueden dividirse en dos categorías principales:

- Memoria de acceso aleatorio (RAM: *Random Access Memory*) es del tipo volátil, es decir, pierde la información cuando su fuente de alimentación es apagada.
- Memorias de sólo lectura (ROM), son del tipo no volátil, conserva la información aun cuando no están conectadas a la fuente de alimentación. Las memorias EEPROM (*Electrically-Erasable Programmable Read-Only Memory*) son un tipo de memoria ROM, que puede ser programada, borrada y reprogramada eléctricamente de 100,000 a 1, 000,000 de veces.

La memoria *Flash* es un tipo de memoria basada en celdas microscópicas de semiconductores del tipo no volátil que al igual que las memorias EEPROM se programan, borran y reprograman eléctricamente. Sus datos no se eliminan al quitar la fuente de energía eléctrica. Permiten la lectura y escritura de múltiples posiciones de memoria en la misma operación, ofreciendo la posibilidad de trabajar con sectores, en vez de un solo *bit* (como es el caso de las memorias EEPROM), lo cual permite alcanzar velocidades de funcionamiento muy superiores frente a la tecnología EEPROM. Debido a su alta velocidad, durabilidad y bajo consumo de energía, la memoria *flash* resulta ideal para aplicaciones donde se requiere almacenamiento de grandes cantidades de información, a un costo razonable y donde el tamaño es un factor crítico, como es el caso de aplicaciones portátiles. Además, este tipo de memorias no tiene partes móviles, lo que la hace más resistente a posibles golpes, evitando la pérdida de información a causa de ellos.

2.4.1. Especificaciones de las memorias flash SD Card

Uno de los formatos de tarjeta de memoria *flash* más usado es el SD (*Secure Digital*), basado en tecnología *flash-NAND*, creada por Panasonic en colaboración con SanDisk Corporation y Toshiba Corporation. Estas tres empresas conforman actualmente la *SD Association*.

Las especificaciones técnicas describen la interfaz física y el protocolo de comandos usados por las tarjetas de memoria *SD Card*, éstas se encuentran en la versión simplificada de las especificaciones técnicas (*Physical Layer Simplified Specification Version*³). Su propósito es definir a las tarjetas de memoria SD en cuanto a su estructura y manipulación.

Algunas de las características principales de estas memorias son:

- Capacidad de almacenamiento de la memoria
 - Memorias SD de Capacidad Estándar (SDSC): Hasta 2 GB.
 - Memorias SD de Alta Capacidad (SDHC): Mayor a 2 GB con un tamaño máximo de 32 GB.
 - Memorias SD de Capacidad Extendida (SDXC): Mayor a 32 GB con un tamaño máximo de 2 TB.

- Rango de voltaje
 - Memorias SD de alto voltaje: Voltaje de operación de 2.7-3.6 V.
 - Memorias SD UHS-II: Voltaje de operación (de acuerdo a la velocidad de del *bus*) VDD1 de 2.7-3.6 V, VDD2 de 1.70 a 1.95 V.

- Diseño de tarjeta de sólo lectura o de lectura-escritura

- Velocidad del Bus (usando 4 líneas de datos paralelos)
 - Modo default: señales de 3.3 V, frecuencia de operación hasta de 25 MHz, con una tasa de transferencia hasta de 12.5 MB/s.
 - Modo alta velocidad: señales de 3.3 V, frecuencia de operación hasta de 50 MHz, con una tasa de transferencia hasta de 25 MB/s.
 - SDR12: UHS-I: señales de 1.8 V, frecuencia de operación hasta de 25 MHz, con una tasa de transferencia hasta de 12.5 MB/s.
 - SDR25: UHS-I: señales de 1.8 V, frecuencia de operación hasta de 50 MHz, con una tasa de transferencia hasta de 25 MB/s.
 - SDR50: UHS-I: señales de 1.8 V, frecuencia de operación hasta de 100 MHz, con una tasa de transferencia hasta de 50 MB/s.

³ SD Specifications part 1, **Physical Layer Simplified Specification**. Version 4.10, January 22, 2013.

- SDR104: UHS-I: señales de 1.8 V, frecuencia de operación hasta de 208 MHz, con una tasa de transferencia hasta de 104 MB/s.
 - DDR50: UHS-I: señales de 1.8 V, frecuencia de operación hasta de 50 MHz, muestreando en ambos flancos de reloj se alcanza una transferencia hasta de 50 MB/s.
 - UHS156: UHS-II: Rango de frecuencia de operación 26 MHz - 52 MHz, con una tasa de transferencia hasta de 1.56 Gbps.
- Corrección de errores en campos de memoria
 - Remover la memoria durante la operación de lectura no afectará su contenido.
 - Mecanismo de protección de contenido
 - Mecanismo de protección contra escritura
 - Detección de tarjeta (inserción y detección)
 - Se define el protocolo para la interfaz de comunicación con los atributos mencionados en la tabla 2.1
 - Comandos de aplicación específica
 - Forma de la memoria SD
 - Tarjeta de memoria SD de tamaño estándar
 - Tarjeta de memoria miniSD
 - Tarjeta de memoria microSD

Interfaz de comunicación de las memorias SD Card
Canal de comunicación de 6 vías (Reloj, comandos, 4 líneas de datos)
Transferencia de datos con protección de errores
Transferencia de datos por bloque o múltiples bloques

Tabla 2.1. Características de la interfaz de comunicación de las memorias SD.

Existe una clasificación de acuerdo a la velocidad de transmisión mínima garantizada; se realiza por clases y se especifica la tasa de transferencia de la memoria SD, independientemente que tenga la misma capacidad entre sí, tal como se muestra en la tabla 2.2.

Clase de memoria SD	Tasa de transferencia
Clase 0	No especifica su desempeño
Clase 2	2 MB/s
Clase 4	4 MB/s
Clase 6	6 MB/s
Clase 10	10 MB/s
SDXC	15 MB/s

Tabla 2.2. Clasificación de las memorias SD de acuerdo a su tasa de transferencia.

2.4.2. Topologías de comunicación de las memorias SD

La mayoría de los tipos de memoria SD Card soportan los protocolos de comunicación SD y SPI (*Serial Peripheral Interface*), el modo de comunicación es seleccionado de acuerdo a las características de la aplicación, y sobre todo, a que el uso del protocolo SD requiere licencia. En la figura 2.34 se muestra la asignación de terminales de la memoria micro SD, de acuerdo al protocolo de comunicación empleado.



Figura 2.34. Distribución de terminales de la memoria micro SD.

2.4.2.1. Comunicación SD

El modo SD es el modo de comunicación nativo desarrollado para las memorias SD Card. La comunicación está basada en el flujo de comandos y datos que son inicializados por un bit de inicio y finalizados por un bit de paro; está diseñado para trabajar a una frecuencia máxima de 50 MHz a través de una interfaz de 9 terminales (reloj, comandos, 4 terminales de datos y 3 de alimentación). En la figura 2.35 se muestra el esquema de conexión de la interfaz microcontrolador – memoria SD para este modo de comunicación.

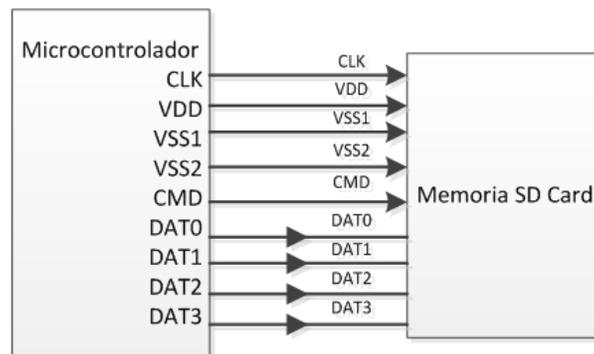


Figura 2.35. Esquema del bus SD.

Para implementar una comunicación entre los dispositivos (microcontrolador – memoria SD), es necesario que exista un dispositivo maestro, el cual controla todas las

acciones que se ejecutarán sobre un dispositivo denominado esclavo. En el modo de comunicación SD, el maestro tiene la posibilidad de configurar el número de líneas de datos activas, ya que después de ser energizada, la tarjeta SD utiliza únicamente una línea para la transferencia de datos, en este caso la línea usada es la línea DAT0.

2.4.2.2. Comunicación SPI

El modo de comunicación SPI consiste en un protocolo de comunicación síncrono secundario, que es soportado por las tarjetas de memoria SD flash. Éste es un subconjunto del protocolo de comunicación SD, diseñado para comunicarse con canales o módulos SPI, que se encuentra disponible en la mayoría de los microcontroladores que actualmente están en el mercado.

El modo de comunicación es seleccionado durante el primer comando de reset (CMD0), después de energizar a la memoria SD. El estándar SPI sólo define la conexión física y no el protocolo completo de transferencia de datos, ya que sólo usa una línea para la transferencia de datos.

La principal ventaja en el modo SPI es la facilidad de implementar un maestro a partir de un microcontrolador de propósito general, haciendo más fácil el diseño. Con base a las especificaciones del fabricante, la principal desventaja es la pérdida de rendimiento, que se puede observar en la reducción de velocidad de transferencia y que sólo puede utilizarse una línea de datos. En la figura 2.36 se muestra el esquema de este protocolo de comunicación.

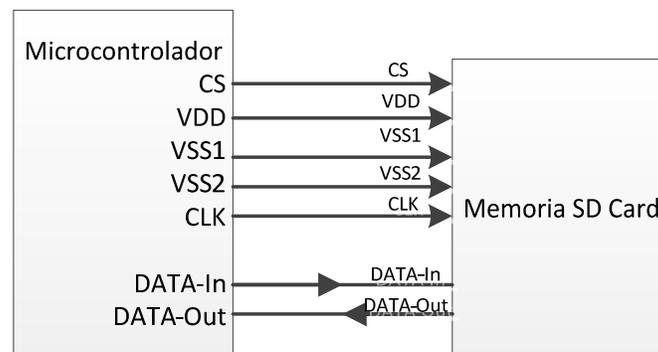


Figura 2.36. Esquema del bus SPI.

La interfaz de comunicación SPI comprende las siguientes señales:

CS: Señal de selección del dispositivo (*Chip Select*). Del host a la memoria.

CLK: Señal de reloj. Del host a la memoria.

DATA IN: Señal de datos. Del host a la memoria.

DATA OUT: Señal de datos. De la memoria al host

2.5. Protocolo de comunicación SPI en la memoria SD Card

Como ya se ha mencionado, el protocolo de comunicación SPI sólo es un subconjunto del protocolo de comunicación en memorias SD Card basadas en tecnología *flash*. A continuación se presentan las características que definen la comunicación en modo SPI⁴:

- Todos los comandos y bloques de datos están conformados por bytes (conjuntos de 8 bits).
- Existen operaciones de comandos, respuestas y transferencia de datos.
- Los comandos de protección contra escritura no son soportados.
- El código de detección de errores CRC está deshabilitado y por lo tanto es ignorado por el maestro.
- El host trata a todas las memorias como si fueran de clase 0, en cuanto a la velocidad de transmisión de información.
- Si la tarjeta se encuentra con un problema de datos en la operación de lectura, ésta responderá con un error que reemplazará el bloque de datos esperado.
- Todas las operaciones entre el maestro y la memoria son controladas por el maestro, es decir, el maestro inicia toda operación en la memoria a través de la activación de la señal CS.
- Las tarjetas siempre responden a los comandos.
- Cuando se trabaja con una tarjeta de capacidad estándar (*Standard Capacity Memory Card*), el bloque de datos puede ser tan grande como un bloque de escritura (512 bytes) y tan pequeño como un byte.
- Para el caso de las tarjetas de memoria de alta capacidad SDHC (*High Capacity SD Memory Card*) y SDXC (*Secure Digital Extended Capacity*), el tamaño de los bloques de datos es única (512 bytes).

Cabe mencionar que para el manejo de la memoria SD Card se hace uso de una librería implementada por Arduino. Dicho lo anterior, en los siguientes apartados relacionados con el uso de la memoria, sólo se hace una descripción de la estructura de los comandos y respuestas básicas, con el objetivo de comprender el funcionamiento y comunicación entre el maestro (microcontrolador) y la memoria SD Card.

⁴SD Association, **Physical Layer Simplified Specification Version 4.10**, January 22, 2013

Formato de comandos

En el modo SPI, todos los comandos de la memoria SD están conformados por 6 bytes (48 *bits*), como se observa en la tabla 2.3, y la trasmisión de los comandos inicia con el bit más significativo (MSB).

Posición de los <i>bits</i>	47	46	[45:40]	[39:8]	[7:1]	0
Cantidad de <i>bits</i>	1	1	6	32	7	1
Valor	'0'	'1'	X	x	x	'1'
Descripción	Bit de inicio	Bit de transmisión	Índice de comando	Argumento	CRC7	Bit de fin

Tabla 2.3. Estructura de comandos.

Un comando siempre inicia con un *bit* de inicio (siempre con valor cero), seguido por el *bit* que indica la dirección de transmisión (maestro=1), los siguientes 6 *bits* indican el índice o número del comando, el cual es interpretado como un número en código binario y puede tener un valor entre 0 y 63; los siguientes 32 *bits* contienen el argumento del comando, los siguientes 7 *bits* corresponde al CRC7 (*Cyclic Redundancy Code*), y finalmente se encuentra el bit de fin (siempre con valor uno).

Es importante mencionar que no todos los comandos contienen un argumento. En la tabla de arriba, un valor denotado por 'x' indica que esta variable depende del comando. En la tabla 2.4 se muestra la descripción de algunos de los comandos más importantes, así como sus correspondientes respuestas.

Comando	Respuesta	Descripción
CMD0	-	Resetea todas las tarjetas a estado de espera (<i>idle state</i>)
CMD8	R7	Envía a la memoria SD las condiciones de operación de la interfaz, incluye voltaje soportado por el maestro y pregunta si la tarjeta soporta este voltaje.
CMD17	R1	En el caso de memorias SDSC este comando lee un bloque de datos de la memoria, del tamaño seleccionado por el comando <i>SET-BLOCKLEN</i> (CMD16). Para memorias SDXC lee un bloque de datos de 512 bytes.
CMD18	R1	Transfiere continuamente bloques de datos de la memoria al maestro hasta ser interrumpido por el comando <i>STOP_TRANSMISSION</i> .
CMD24	R1	En el caso de memorias SDSC este comando escribe un bloque de datos a la memoria, del tamaño seleccionado por el comando <i>SET-BLOCKLEN</i> (CMD16). Para memorias SDXC escribe un bloque de datos de 512 bytes.

Tabla 2.4. Descripción de comandos para operar con memorias SD. (Continúa)

Comando	Respuesta	Descripción
CMD25	R1	Continuamente escribe bloques de datos a la memoria, hasta ser interrumpido por el comando <i>STOP_TRANSMISSION</i> .
CMD58	R3	Lee el OCR (Operation Condition Register), es usado por el host como un mecanismo para identificar a las tarjetas que cumplen con el rango de voltaje de alimentación suministrado por el host. Si existe una incompatibilidad con el rango de voltaje no se continuará el proceso de inicialización.
ACM41	R3	Envía al maestro a través de R3, información acerca de la capacidad soportada (HCS), además, envía el contenido del registro OCR (Operating Condition Register). Para verificar si el proceso de inicialización ha terminado

Tabla 2.4. Descripción de comandos para operar con memorias SD.

Respuestas

En general, la transmisión de la respuesta siempre comienza con el *bit* de la izquierda, su longitud depende del tipo de respuesta. Una respuesta siempre comienza con un *bit* de inicio (siempre cero), seguido por el *bit* que indica la dirección de la transmisión (memoria = 0), un valor denotado por ‘x’, que indica que es variable y depende del tipo de respuesta, todas las respuestas excepto la R3 estas protegidas por un CRC, finalmente cada respuesta termina con un *bit* de fin (siempre uno).

Existen cinco tipos de respuestas para la memoria SD Card. A continuación se presenta las que se usan en el protocolo SPI.

R1 (Respuesta normal de comando)

R1 tiene una longitud de 48 *bits*; los *bits* 45:40 indican el índice del comando al cual va dirigida la respuesta, este valor es interpretado como un número en código binario y puede tener un valor entre 0 y 63; el estado de la tarjeta se encuentra en los siguientes 32 *bits*; En la tabla 2.5 se muestra la estructura de la respuesta R1.

Posición de los <i>bits</i>	47	46	[45:40]	[39:8]	[7:1]	0
Cantidad de <i>bits</i>	1	1	6	32	7	1
Valor	‘0’	‘0’	x	x	X	‘1’
Descripción	Bit de inicio	Bit de transmisión	Índice de comando	Estado de la tarjeta	CRC7	Bit de fin

Tabla 2.5. Estructura de la respuesta R1.

Si una transferencia de datos está presente en la tarjeta, una bandera de *busy* puede aparecer en la línea de datos después de la transmisión de cada bloque. El maestro verificará esta bandera después de cada bloque de transmisión.

R2 (Estado de los registros CID, CSD)

R2 tiene una longitud de 136 *bits*. El contenido del registro CID es enviado como una respuesta a los comandos CMD2 y CMD10. El contenido del registro CSD es enviado como una respuesta al comando CMD9, sólo los *bits* 127:1 de los registros CID y CSD son transferidos, el *bit* reservado (0) de estos registros es reemplazado por el *bit* de fin de la respuesta. En la tabla 2.6 se muestra la estructura de la respuesta R2.

Posición de los bits	135	134	[133:128]	[127:1]	0
Cantidad de bits	1	1	6	127	1
Valor	'0'	'0'	'111111'	X	'1'
Descripción	Bit de inicio	Bit de transmisión	Reservados	Contenido de los registros CID y CSD incluyendo le CRC7 interno	Bit de fin

Tabla 2.6. Estructura de la respuesta R2.

R3 (Estado del registro OCR)

R3 tiene una longitud de 48 *bits*. El contenido del registro OCR es enviado como una respuesta al comando ACMD41. En la tabla 2.7 se muestra la estructura de la respuesta R3.

Posición de los bits	47	46	[45:40]	[39:8]	[7:1]	0
Cantidad de bits	1	1	6	32	7	1
Valor	'0'	'0'	'111111'	x	'1111111'	'1'
Descripción	Bit de inicio	Bit de transmisión	Reservados	Contenido del OCR	Reservados	Bit de fin

Tabla 2.7. Estructura de la respuesta R3.

R7 (Condiciones de la interfaz entre la tarjeta y el maestro)

R7 tiene una longitud de 48 *bits*, envía la información sobre el rango de voltaje de alimentación (3.3 V) de la tarjeta como respuesta al comando CMD8, los *bits* 19:16 indican el rango de voltaje que soporta la tarjeta. En la respuesta, la tarjeta resalta el rango de voltaje soportado, así como el valor puesto en el argumento del comando enviado por el maestro. En la tabla 2.8 se muestra la estructura de la respuesta R7.

Posición de los bits	47	46	[45:40]	[39:20]	[19:16]	[15:8]	[7:1]	0
Cantidad de bits	1	1	6	20	4	8	7	1
Valor	'0'	'0'	'001000'	'00000h'	x	x	x	'1'
Descripción	Bit de inicio	Bit de transmisión	Índice de comando	Reservados	Voltaje aceptado	Patrón de verificación	CRC7	Bit de fin

Tabla 2.8. Estructura de la respuesta R7.

En la tabla 2.9 se describe el campo de voltaje aceptado.

Bits del campo de voltaje aceptado	Valor
0000b	No definido
0001b	2.7 – 3.6 V
0010b	Reservado para rango de bajo voltaje
0100b	Reservado
1000b	Reservado
Otros	No definido

Tabla 2.9. Voltaje aceptado en respuesta R7.

2.5.1. Inicialización en modo SPI

Cuando la memoria SD es energizada, automáticamente entra en modo SD. Para activarla en modo SPI, la señal CS se debe mantener en un nivel lógico bajo; mientras el microcontrolador envía el comando de *reset* (CMD0) a la memoria SD *Card*, ésta enviará la respuesta R1 al maestro, si es que ha entrado en modo SPI.

En la figura 2.37 se muestra el diagrama de flujo de la secuencia de comandos necesarios para inicializar las memorias SD *Card* en modo SPI.

La memoria SD analizará el argumento de los comandos y responderá según corresponda, al final del proceso el maestro tendrá información suficiente para determinar la versión de la memoria SD y si puede o no trabajar con ella.

Las funciones de los comandos son:

El comando CMD8 es usado para verificar las condiciones de operación de la interfaz. La memoria SD comprueba la validez de estas condiciones, analizando el argumento del comando enviado por el maestro; el maestro hace lo propio analizando la respuesta R7 de la memoria SD, en la que se envía el voltaje aceptado por la tarjeta.

El comando CMD58 lee el OCR, es usado por el maestro como un mecanismo para identificar a las tarjetas (memorias SD) que cumplen con el rango de voltaje de

alimentación suministrado por el maestro. Si existe una incompatibilidad con el rango de voltaje no se continuará el proceso de inicialización. La respuesta R3 está asociada a este comando. En la tabla 2.10 se muestran los *bits* que componen al registro OCR de la memoria, que es enviado como respuesta (R3) al comando CMD58. Con la información de este registro se puede conocer el margen de voltaje soportado por la tarjeta, la capacidad de la tarjeta y si ésta ha terminado el ciclo de encendido.

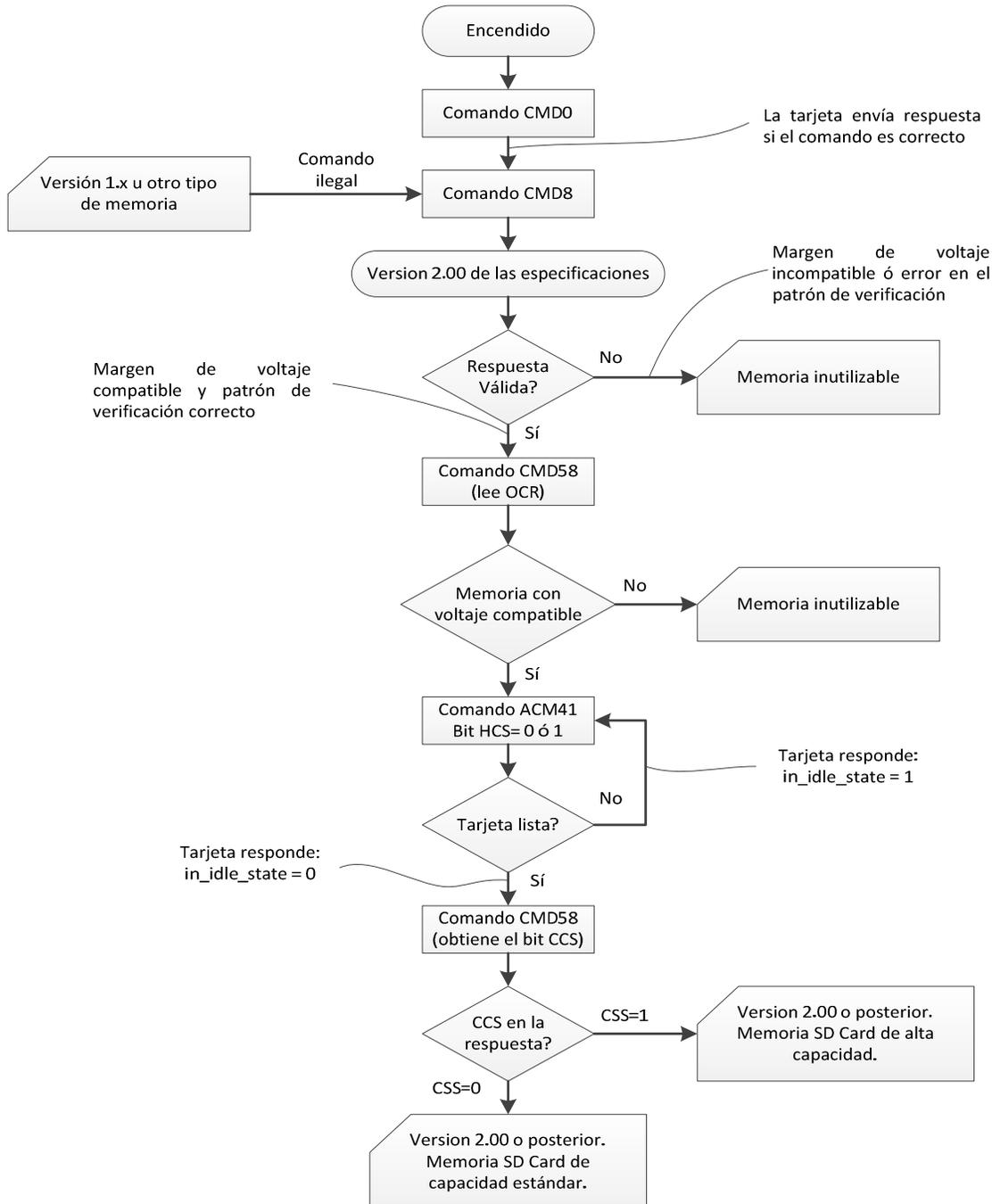


Figura 2.37. Algoritmo para la inicialización de una memoria SD en modo SPI.

Posición de <i>bits</i>	Campo del OCR
0 – 6	Reservado
7	Reservado para rango de bajo voltaje
8 – 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	
25 – 28	Reservado
29	Estado de la memoria tipo UHS-II
30	Capacidad de memoria (CCS)
31	Estado de encendido de la tarjeta (busy)

Tabla 2.10. Definición del registro OCR.

En el caso del valor del bit, un valor de ‘0’ (*bits* 15:23), indica que ese rango de voltaje no es soportado; un valor de ‘1’ (*bit* 31), indica que el estado de encendido de la tarjeta ha finalizado. El valor de CCS (*Card Capacity Status*) (*bit* 31) fue descrito en el diagrama de flujo anterior.

El comando ACMD41 es usado para comenzar la inicialización y para verificar, a través del *bit* “*in_idle_state*”, si la tarjeta ha completado este proceso. Es obligatorio que el maestro envíe el comando CMD58 antes de enviar por primera vez el comando ACMD41.

2.5.2. Operaciones de lectura

El modo SPI soporta operaciones de lectura de un solo bloque (CMD17) y de múltiples bloques (CMD18). Para el caso de lectura de un solo bloque, el maestro envía el comando (CMD17), si el comando de lectura es válido, la tarjeta enviará la respuesta R1, seguida por el bloque de datos. La figura 2.38 muestra el diagrama de la operación de lectura de un solo bloque.

Cada bloque de datos está acompañado de un bloque de 16 *bits*, que es un código de detección de errores del tipo CRC. En general, en el modo SPI los *bits* del CRC son ignorados.

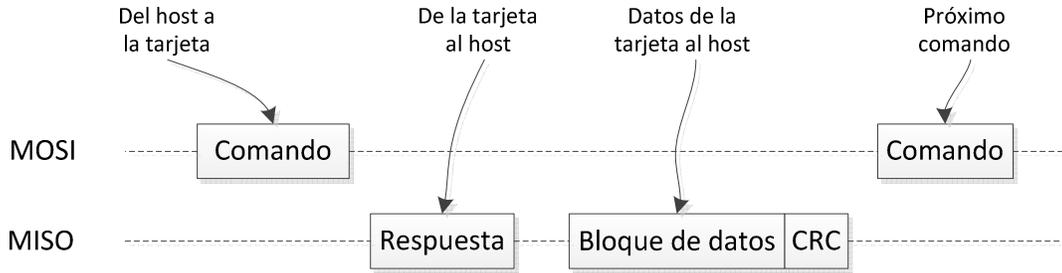


Figura 2.38. Operación de lectura de un solo bloque de datos.

En el caso de la operación de lectura de múltiples bloques (al igual que en la lectura de un bloque individual), cada bloque de datos transferido está acompañado de 16 bits del CRC, con la diferencia de que los bloques de datos serán enviados continuamente por la tarjeta hasta que se reciba el comando de fin de transmisión (CMD12). Esta operación se muestra en la figura 2.39.

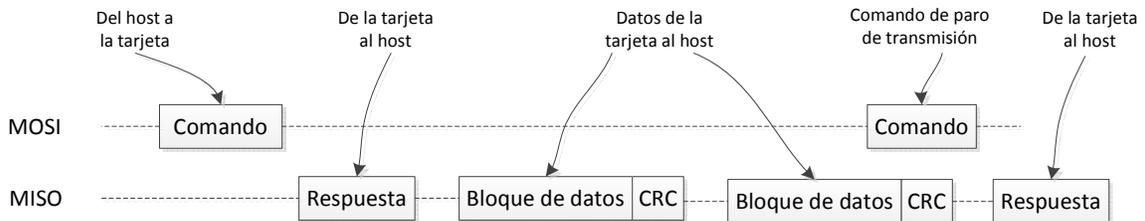


Figura 2.39. Operación de lectura de múltiples bloques de datos.

2.5.3. Operaciones de escritura

El modo SPI soporta operaciones de escritura de un solo bloque (CMD24) y también de múltiples bloques (CMD25). Para la escritura de un solo bloque, el maestro envía el comando (CMD24), si el comando de escritura es válido, la tarjeta enviará la respuesta R1 y esperará por un bloque de datos enviados por el maestro. Por el contrario si el comando no es válido, la memoria no responderá y el comando no será ejecutado, por lo tanto, la memoria no cambiará de estado. Al existir un comando no válido se activará el bit de error (*Illegal – Command*) en el registro de estatus.

Cada bloque de datos enviado por el maestro debe ser precedido por un *bit* de inicio (*Start Block Token*); después de que el bloque de datos es recibido por la tarjeta, ésta enviará una respuesta de datos (*Data Response*). Mientras el bloque de datos se esté escribiendo en la memoria SD, la tarjeta enviará la señal de *busy*, indicando que se está

realizando la operación antes mencionada. En la figura 2.40 se muestra el diagrama para la realización de esta operación.

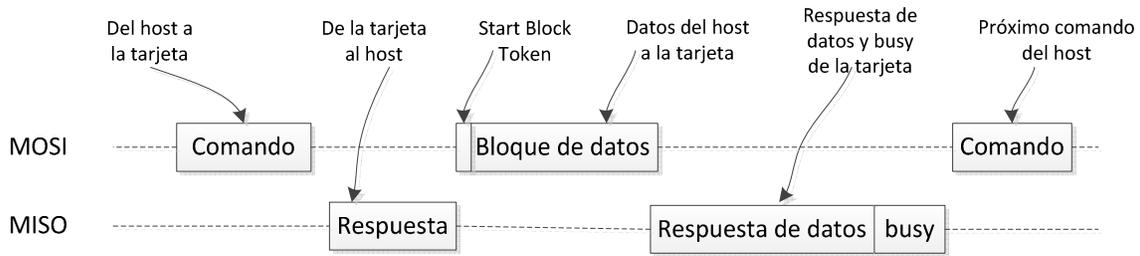


Figura 2.40. Operación de escritura de un solo bloque.

Para el caso de escritura de múltiples bloques, el procedimiento es similar al de escritura de un solo bloque, con la diferencia de que la transmisión de bloque de datos termina al enviar el byte de fin de transmisión “*Stop Tran*”, al final del último bloque transmitido. En la figura 2.41 se muestra el diagrama para la realización de esta operación de escritura.

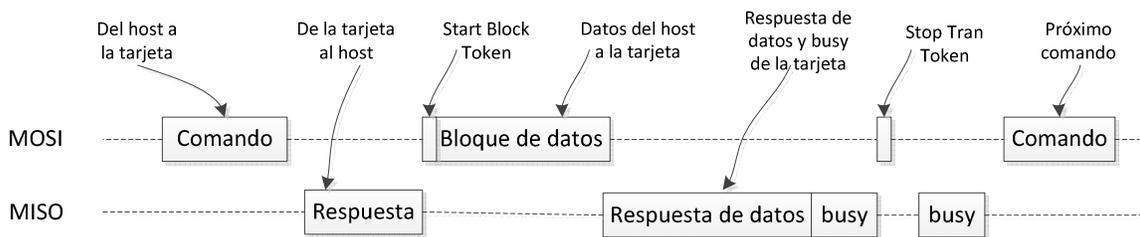


Figura 2.41. Operación de escritura de múltiples bloques.

2.6. Sistemas de archivos FAT

Todos los dispositivos de almacenamiento masivo requieren que la información guardada se encuentre organizada, de manera que se pueda acceder a ella de una manera eficiente. Un sistema de archivos manipula, organiza y le da una estructura a los datos almacenados con ciertas especificaciones, para que puedan ser leídos por equipos que sean compatibles con estas reglas, es decir que soporten el sistema de archivos.

El sistema de archivos FAT (*File Allocation Table*) fue desarrollado por IBM PC *Machine Architecture* y tuvo sus orígenes entre finales de 1970 y principios de 1980; fue el sistema de archivos soportado por MS-DOS de Microsoft. Originalmente fue desarrollado con un sistema de archivos simple, pensado para disco flexibles con capacidad menor a 500 kB, pero con el paso del tiempo ha sido mejorado para soportar volúmenes de mayor

capacidad. Una característica importante de este sistema de archivos, es que todas las estructuras de datos en el disco tienen el formato de “*Little endian*”, es decir, el byte menos significativo se almacena en la primera localidad de memoria y el byte más significativo se almacena en la última localidad de memoria.

Actualmente existen tres sistemas de archivos FAT: FAT12, FAT16 y FAT32. La diferencia básica y la razón de sus nombres, radica en el tamaño de *bits* de las entradas en la región FAT de la estructura del disco. Hay 12 *bits* en una entrada FAT para FAT12, 16 *bit* para FAT16 y 32 *bits* para FAT32.

Un aspecto importante que se debe de considerar en estos sistemas de archivos, es que, el espacio para datos de una unidad de almacenamiento se divide en pequeñas unidades de 512 bytes (tamaño mínimo) llamadas sectores, esto es a nivel físico. Sin embargo, a nivel lógico, el sistema de archivos divide la unidad de almacenamiento en unidades llamadas *clúster*; el cual agrupa una cierta cantidad de sectores contiguos, que es determinada por el sistema operativo que formatea a la unidad y por la capacidad de almacenamiento de la misma, es decir, cuanto más grande sea la unidad de almacenamiento mayor será el tamaño del *clúster*. El número de sectores dentro de un *clúster* siempre es potencia de 2 (1, 2, 4, 8, etc). En la figura 2.42 se muestra la organización de sectores y un clúster. Para el sistema de archivos, el *clúster* es la unidad mínima de almacenamiento que se puede asignar a un archivo y direccionar en la tabla FAT.

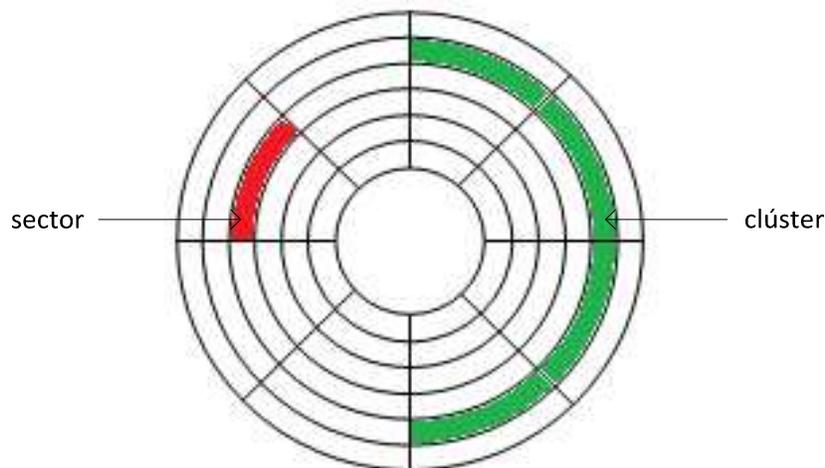


Figura 2.42. Organización de sectores y clústers.

La FAT es una tabla formada por elementos que se corresponden con cada uno de los *clústers* de la unidad, es decir, el elemento situado en la posición 40 de la FAT controla el *clúster* 40 del disco (que a su vez corresponderá a unos determinados sectores del disco). Cada elemento de la FAT puede tener uno de los tres valores siguientes:

- Una marca especial (el valor 0), para indicar que se trata de un *clúster* libre; es decir, que los sectores gobernados por ese *clúster* no tienen datos.
- Una marca especial (valores entre FFF8 - FFFF para FAT16 y FFFFFFFF8 – FFFFFFFF para FAT32) para indicar que se trata del último *clúster* de un archivo, es decir, que los sectores de ese *clúster* almacenan la parte final del archivo.
- Cualquier otro valor numérico se interpreta como el *clúster* siguiente del archivo. Por ejemplo, si el elemento 135 de la FAT tiene el valor 227, quiere decir que detrás de los sectores del *clúster* 135, el archivo continúa con los sectores del *clúster* 227.

El sistema de archivos FAT está compuesto de cuatro regiones básicas (figura 2.43), que se encuentran en el siguiente orden dentro del volumen de almacenamiento:

- R0. Región reservada
- R1. Región FAT
- R3. Región del directorio raíz (no existe en volúmenes FAT32)
- R4. Región de datos, archivos y directorios

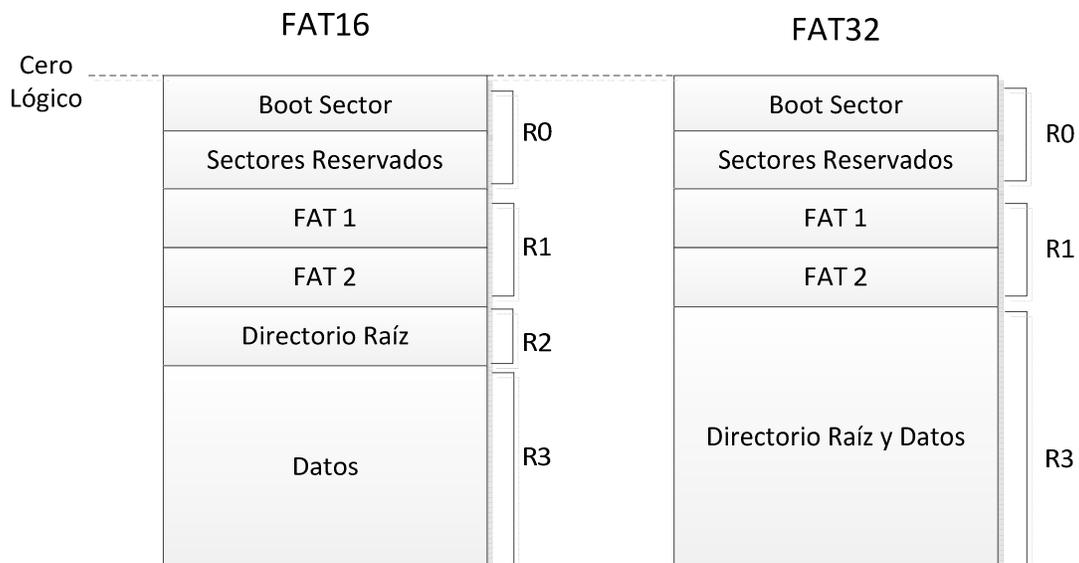


Figura 2.43. Estructura de los sistemas de archivos FAT16 y FAT32.

Boot Sector and BPB

La primera estructura de datos importante en un volumen FAT es el BPB (*BIOS Parameter Block*), localizado en el primer sector del volumen, dentro de la región reservada; este sector es llamado algunas veces “boot sector” o “sector 0”, pero su importancia radica en que es el primer sector del volumen. EL BPB contiene información sobre el sistema de archivos, la cual es útil para tener acceso al volumen y realizar operaciones de lectura y escritura dentro de él.

Estructura de datos FAT

La región FAT, también llamada tabla de asignación de archivos, es una lista ligada de los *clusters* que componen a los distintos archivos y sirve como un mapa de la región de datos de la partición. Cada clúster es identificado por un número con el que se crea una entrada en la región FAT.

Las entradas en la región FAT se crean en una posición que corresponde con el número del clúster; el contenido de estas entradas indica el número del siguiente clúster de un archivo dado y con una marca especial el fin del archivo. De esta forma, se obtiene una cadena FAT, la cual es una lista vinculada de referencias que apunta a los *clusters* sucesivos hasta el final del archivo.

La región FAT contiene dos tablas FAT, la segunda es una copia exacta de la primera, sirviendo como copia de seguridad si la primera tabla se corrompe.

Directorio raíz

La región del directorio raíz fue definida solo para el sistema de archivos FAT16 y es de tamaño fijo. En sistemas FAT32 el directorio raíz es de tamaño variable, dependiendo de la cantidad de directorios y archivos que se tengan creados. Por lo general, el directorio raíz en FAT32 se encuentra en el primer clúster de la región de datos.

En el directorio raíz se encuentra la disposición de los datos contenidos en la unidad de almacenamiento, los cuales son almacenados en una estructura de carpetas y archivos. Un directorio es un fichero (archivo) especial que tiene activado el atributo de directorio y cuyo contenido son las entradas correspondientes a cada elemento que se almacena dentro de dicho directorio

El tamaño de la entrada para un archivo o carpeta es de 32 bytes, de los cuales: ocho bytes son para el nombre del archivo, tres para la extensión, un byte de atributos, ocho bytes reservados, dos bytes para la parte alta del número del primer clúster, cuatro bytes reservados, dos bytes para la parte baja del número del primer clúster y finalmente cuatro bytes para el tamaño del archivo⁵. Esta información es utilizada por el sistema operativo para localizar el archivo deseado; obteniendo el primer número del primer *clúster* de un archivo encontrara el resto entrando a la tabla FAT.

En la figura 2.44. Se muestra la estructura de las entradas en el directorio raíz, la ‘x’ indica que el valor de ese byte es variable.

⁵Microsoft Corporation, **Microsoft Extensible Firmware Initiative FAT32 File System Specification Version 1.03**, December 6, 2000

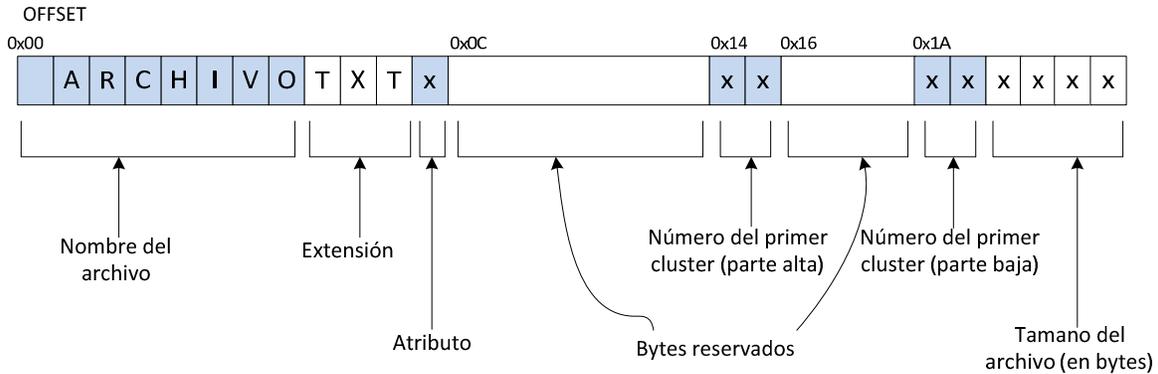


Figura 2.44. Estructura de las entradas en el directorio raíz.

Región de datos

En la región de datos se encuentran registrados los archivos y carpetas del usuario, por lo general, la región de datos comienza en el clúster 2.

2.6.1. FAT16 y FAT32

FAT16

El sistema de archivos FAT16 es un sistema de 16 *bits*, esto implica que las direcciones de clúster no pueden ser mayores a 16 *bits*. El número máximo de clústers al que se puede hacer referencia con el sistema FAT es, por consiguiente, 65 536 ($2^{16} - 1$) clústers; ya que un *clúster* se compone de un número fijo de sectores de 512 bytes continuos. El tamaño máximo de la partición FAT se puede determinar multiplicando el número de *clústers* por el tamaño de un clúster. Con *clústers* de 32 kbytes, el tamaño máximo de una partición dentro de la unidad es de 2 GB.

Además, un archivo sólo puede ocupar un número entero de *clústers*, esto significa que si un archivo ocupa varios *clústers*, el último solamente estará ocupado en forma parcial y no se podrá utilizar todo el espacio disponible de la unidad. Como resultado, cuanto menor sea el tamaño del clúster, menor será el espacio desperdiciado.

FAT32

FAT32 utiliza valores de 32 *bits* para las entradas FAT. De hecho, sólo se utilizan 28 *bits*, ya que 4 *bits* se reservan para su uso en el futuro.

Cuando surgió el sistema de archivos FAT32, el máximo número de *clústers* por partición aumentó de 65 535 a 268 435 455 ($2^{28} - 1$). Por lo tanto, FAT32 permite particiones

mucho más grandes; aunque en teoría, el tamaño máximo de una partición FAT32 es de 8 TB, Microsoft lo redujo a 32 GB. Ya que una partición FAT32 puede contener muchos *clústers*, más que una partición FAT16, es posible reducir significativamente el tamaño de los *clústers*, y así, limitar también el espacio desperdiciado en el disco.

En la tabla 2.11 se muestra el tamaño máximo de una partición para los sistemas de archivos FAT16 y FAT 32, de acuerdo al tamaño del clúster que estén manejando.

Tamaño del clúster	FAT16	FAT32 (en teoría)
512 bytes	32 MB	64 MB
1 kbytes	64 MB	128 MB
2 kbytes	128 MB	256 MB
4 kbytes	256 MB	8 GB (1 TB)
8 kbytes	512 MB	16 GB (2 TB)
16 kbytes	1 GB	32 GB (4 TB)
32 kbytes	2 GB	Más de 32 GB (8 TB)

Tabla 2.11. Tamaño de particiones en sistemas de archivos FAT16 y FAT32.

Podemos observar que el sistema de archivos FAT32 usa el espacio del volumen de manera más eficiente y además soporta volúmenes de almacenamiento mayores con respecto al sistema de archivos FAT16.

2.7. Referencia de tiempo

Una de las necesidades más importantes en el diseño de sistemas de adquisición de señales sísmicas, es el de contar con una referencia de tiempo lo más exacta posible, debido a que estos dispositivos registran el tiempo en el que ocurre un fenómeno sísmico.

Para contar con una referencia lo más exacta posible, se puede recurrir a dos opciones: La primera es utilizar dispositivos con un alto nivel de exactitud y bajo costo, como es el caso de los receptores GPS (*Global Positioning System*). Algunos modelos de estos dispositivos brindan un pulso por segundo (1 PPS) con una precisión en el orden de los micro segundos, por lo que dicha señal puede ser sincronizada con un reloj de tiempo real, obteniendo de esta manera un sistema más preciso. La segunda opción se enfoca para los modelos que no cuentan con el pulso por segundo, por lo que bastará con leer una de las cadenas del protocolo NMEA 0183, que contengan los datos de fecha y hora, para posteriormente procesar la información por medio de un software, que permita estampar cada segundo los datos digitalizados.

En relación a la referencia de tiempo se debe considerar que las señales del receptor GPS pueden desaparecer repentinamente, esto depende de la localización de la antena del GPS, o de las condiciones climatológicas que se encuentren cuando se realice la prueba. Sin embargo, con ambas propuestas no se ve afectada la sincronización del sistema, ya que en

la primera opción, al utilizar un reloj en tiempo real este seguirá funcionando aún sin la señal de (1 PPS). Y con la segunda opción basta con tener una sola referencia de tiempo para poder sincronizar los datos.

En las dos opciones mencionadas anteriormente se hace uso del tiempo universal coordinado (UTC: Universal Time Coordinated). Este dato es un estándar de tiempo calculado a partir de una medida ponderada de las señales de los relojes atómicos, localizados en cerca de 70 laboratorios nacionales de todo el mundo. Además de que se toma como referencia para determinar las diferentes zonas horarias, en aplicaciones sísmicas se utiliza para tener un estándar en cuanto al registro de sismos.

2.7.1. Sistema de Posicionamiento Global (GPS)

El GPS es un sistema compuesto por una red de satélites, radio bases terrestres y receptores GPS que permiten casi todas las posibilidades de navegación y posicionamiento en cualquier parte del mundo. Este sistema, está compuesto por tres subsistemas principales:

- ✓ Sistema satelital
- ✓ Sistema de control terrestre
- ✓ Sistema del usuario

El sistema satelital está formado por una constelación de 24 satélites, denominada NAVSTAR (*Navigation System Using Timing and Ranging*), que gira alrededor de la Tierra en seis planos orbitales a unos 60° entre sí, con cuatro satélites en cada plano. Hay 21 satélites activos y 3 de reserva. En caso de que falle un satélite, uno de los de reserva puede ocupar su lugar. En la figura 2.45 se muestran las órbitas de los 21 satélites funcionales de la red NAVSTAR.

Los satélites NAVSTAR no son geosincrónicos, debido a que giran en torno a la Tierra en órbitas circulares inclinadas. El ángulo de elevación en el modo ascendente es 55° con respecto al plano ecuatorial y la elevación promedio de un satélite NAVSTAR es de unos 20,200 km sobre la Tierra.

Esta red de satélites fue desarrollada por el Departamento de Defensa de los Estados Unidos de América y declarada totalmente operativa desde el 27 de abril de 1995. Al principio fue creada exclusivamente para fines militares y, posteriormente, dadas las aplicaciones tecnológicas, se puso al servicio de la comunidad civil.

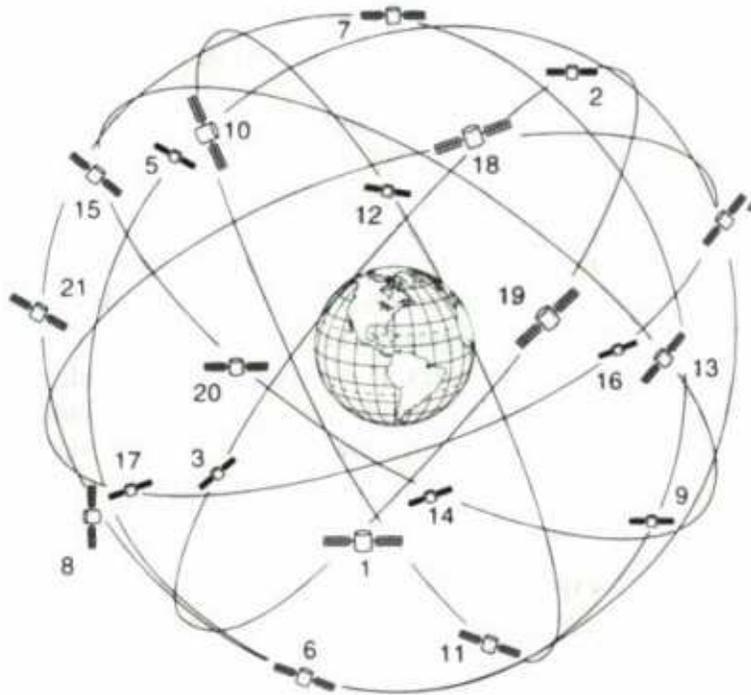


Figura 2.45. Constelación de satélites NAVSTAR.

El sistema de control terrestre también denominado “sistema de control de operación”, incluye todas las estaciones monitoras terrestres fijas que se encuentran ubicadas en todo el mundo. Las estaciones monitoras no son más que receptores GPS que rastrean los satélites cuando pasan sobre ellas y acumulan datos de telemetría y efemérides de los mismos. Esta información se transmite a una *Estación de Control Maestro* donde se procesa y determina si la posición real del satélite es igual a la calculada por el GPS. La estación de control – maestro recibe datos de las estaciones monitoras en tiempo real, con esa información se determina si los satélites sufren cambios de reloj o de información y detecta el mal funcionamiento de los equipos, por lo que se calcula nuevamente la información de navegación a partir de las señales monitoreadas y se envía a los satélites, junto con órdenes de mantenimiento rutinario.

El sistema del usuario de GPS consiste en todos los receptores GPS y la comunidad de usuarios. Los receptores GPS convierten las señales recibidas de los vehículos espaciales en estimaciones de posición, velocidad y tiempo.

En la Figura 2.46, se ilustran los tres sistemas ya descritos.

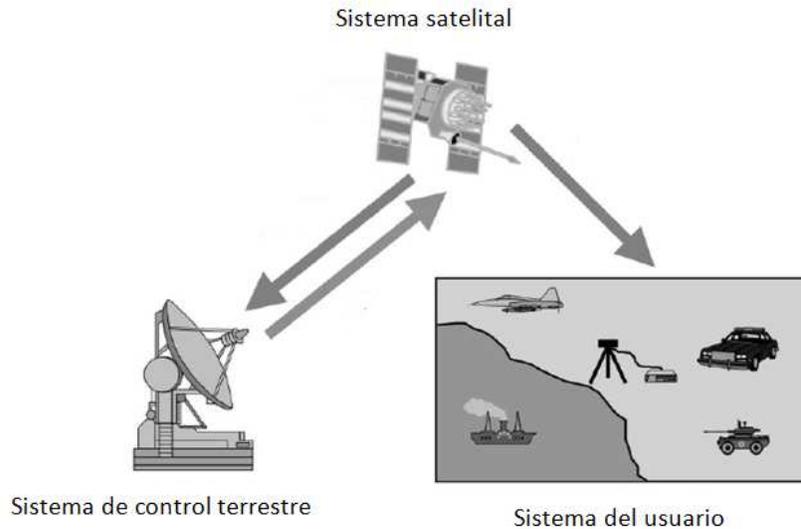


Figura 2.46. Sistemas que integran el GPS.

Funcionamiento del GPS

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.

Los dos códigos de modulación se llaman Adquisición/Gruesa (C/A: *Coarse Acquisition*) y Precisión (P: *Precisión*); cada código consiste de un flujo de dígitos binarios (unos y ceros), conocidos como *bits* o *chips*. Esta modulación se llama 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*) ya que parecen códigos aleatorios, aunque en realidad son generados con algoritmos complejos. En la actualidad, el código C/A está modulado únicamente en el portador L1, mientras que el código P está modulado en ambos portadores L1 y L2.

El código C/A es una cadena de 1023 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.

Por otra parte, el mensaje de navegación es un flujo de datos añadido a los portadores L1 y L2 como una modulación binaria de doble fase con una tasa de 50 kbps. Ésta consiste de 25 “frames” de 1,500 bits cada uno (37,500 bits en total), los cuales son transmitidos en 750 segundos. El mensaje de navegación contiene coordenadas de los satélites GPS en función del tiempo, condición del satélite, corrección del reloj del satélite, almanaque del satélite, datos atmosféricos, entre otros.

La idea detrás del GPS es muy simple: si se conoce la distancia de un punto de la Tierra (receptor GPS) a tres satélites y si se conoce la posición de los satélites, entonces la posición del punto (receptor) puede ser determinada por medio de la triangulación. Teóricamente, sólo tres distancias a tres satélites rastreados simultáneamente son necesarias. En este caso, el receptor se encuentra en la intersección de tres esferas; cada una con un radio equivalente a la distancia del receptor a un satélite y centrada en ese satélite en particular, como se ilustra en la Figura 2.47. Desde el punto de vista práctico es necesario un cuarto satélite para compensar el reloj del receptor.

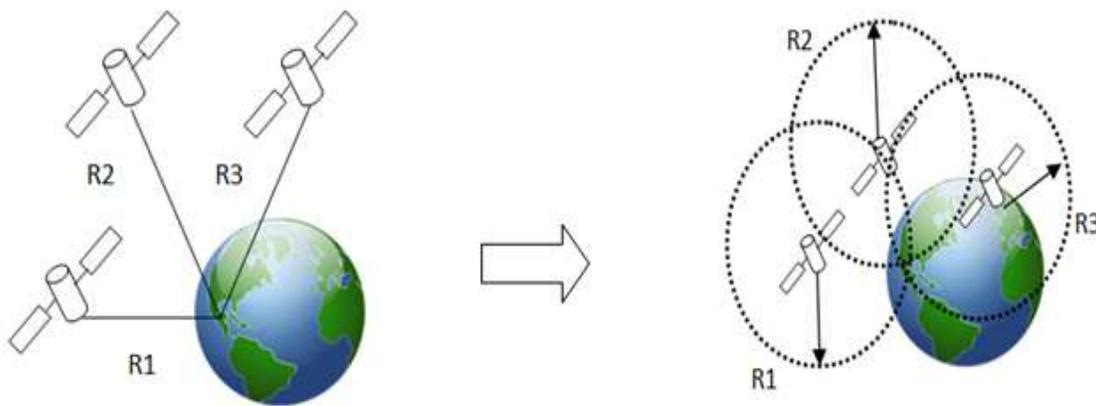


Figura 2.47. Principio de Funcionamiento del GPS.

La distancia entre el receptor GPS y el satélite se mide a través de un pseudorange, que no es más que la distancia entre la antena del receptor y la antena del satélite, dicho procedimiento es el siguiente:

Suponiendo que el reloj del satélite y el reloj del receptor están sincronizados perfectamente entre sí, cuando el código PRN (C/A o P) es transmitido desde el satélite, y el receptor genera una réplica exacta del mismo (Figura 2.48.) después de un tiempo, equivalente al tiempo de recorrido de la señal en el espacio Δt , el código transmitido será registrado por el receptor. Comparando el código transmitido con su réplica, el receptor puede calcular el tiempo de recorrido de la señal multiplicándolo por la velocidad de la luz (aproximadamente 3×10^8 m/s) y se obtiene la distancia entre el satélite y el receptor.

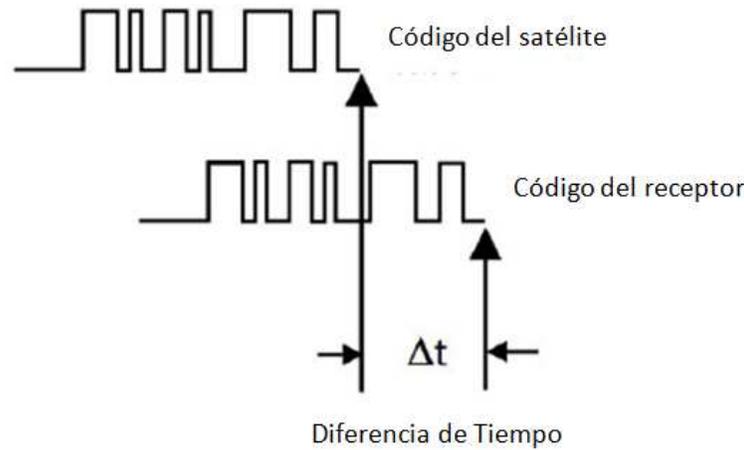


Figura 2.48. Pseudorango de tiempo GPS.

A manera de ejemplo, para ilustrar mejor los párrafos anteriores, si de la figura 2.48, se sabe que Δt es igual a 4 milisegundos, entonces, la distancia entre el receptor y el satélite se obtiene como se indica en la expresión 2.26.

$$distancia = 3 \times 10^8 \frac{m}{s} * 4ms \quad (2.26)$$

Como resultado de la expresión 2.26, se tendrá una distancia de 1200 km entre el receptor y el satélite.

Desafortunadamente, ya que en realidad los relojes no están perfectamente sincronizados, el rango medido está contaminado con errores y tendencias, por esta razón se le denomina pseudorango y no rango.

Cada fabricante de GPS tiene sus propios formatos para guardar las mediciones del mismo, es difícil combinar datos de diferentes receptores. Un problema similar ocurre al tratar de comunicar varios dispositivos de navegación, incluyendo el GPS, por lo que para evitar estas limitaciones, varios grupos de investigación se han dado a la tarea de desarrollar estándares de usuario. Algunos de los protocolos estándares son: RINEX, NGS-SP3, RTCM SC-104 y NMEA 0183. En nuestro caso se describirá el protocolo NMEA 0183, debido a que es el más utilizado y es el que se utilizó en este proyecto.

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 0183 utiliza un protocolo de comunicaciones serie ASCII, el cual define como se transmiten los datos en una cadena de un “hablador” a varios “oyentes” a la vez, los cuales son transmitidos mediante una comunicación serial.

Para obtener los datos del receptor GPS es necesario enlazar el receptor GPS con un microcontrolador que procese la información de dicho dispositivo, por tal motivo, en la tabla 2.12 se muestra la configuración del puerto serial para poder transmitir los datos del receptor GPS al microcontrolador.

<i>Baud rate</i> (valor típico)	4800
<i>Data bits</i>	8
<i>Parity</i>	ninguno
<i>Stop bits</i>	1
<i>Handshake</i>	ninguno

Tabla 2.12. Características de comunicación serial para receptor GPS.

La cadena de datos del NMEA 0183 puede incluir información sobre la posición, fecha, profundidad y otras variables. Las características globales a considerar son las siguientes:

- Los datos son enviados en forma de cadenas.
- Cada cadena comienza con el signo “\$” y termina con un retorno de carro/siguiente línea <CR><LF>.
- El signo “\$” está seguido por un campo de 5 caracteres, generalmente en mayúsculas, que identifica al hablante (los primeros dos caracteres), al tipo de dato y el formato de los campos sucesivos (los últimos tres caracteres).
- El último campo de cualquier cadena es una suma de control, precedido por un carácter delimitador “*”.

La siguiente cadena del protocolo NMEA 0183, ilustra un ejemplo de las cadenas proporcionadas por el GPS:

\$GPRMC,044235.000,A,4322.0289,N,00824.5210,W,0.39,65.46,020911,,A*44

De acuerdo con la información del protocolo NMEA, la cadena anterior significa:

- ✓ \$GPRMC, es una de los mensajes de salida del protocolo NMEA el cual significa *Recommended Minimum Specific Data*.
- ✓ 044235.000, es la hora UTC (04:42:35).

- ✓ A es la indicación de que el dato de posición está fijado y es correcto.
- ✓ 4322.0289, es la longitud ($43^{\circ}22.0289'$).
- ✓ N, norte.
- ✓ 00824.5210, es la latitud ($8^{\circ}24.5210'$).
- ✓ W, oeste.
- ✓ 0.39, velocidad en nudos.
- ✓ 65.46, orientación en grados.
- ✓ 020911, fecha (2 de Septiembre del 2011).
- ✓ *44, se refiere a la suma verificadora (*check sum*), la cual está formada por la operación XOR de todos los datos de la cadena incluyendo las separaciones “;” pero no los limitadores \$ y *.

Como se mencionó en el ejemplo anterior, el mensaje RMC es uno de los mensajes de salida del protocolo NMEA 0183, en la tabla 2.13 se muestran todos los mensajes de salida de dicho protocolo

Mensaje	Significado	Descripción
GGA	<i>Global positioning system fixed data</i>	Contiene datos referente a tiempo posición y tipo de datos fijos.
GLL	<i>Geographic Position</i>	Contiene datos de latitud, longitud, hora UTC de fijación de la posición y el estado.
GSA	<i>Active satellites</i>	Contiene el modo de funcionamiento del receptor GPS y los satélites utilizados.
GSV	<i>Satellites in View</i>	Contiene el número de satélites GPS en vista, número de identificación de los satélites, elevación y valores SNR (<i>Signal to Noise Ratio</i>).
MSS	<i>MSK receiver signal</i>	Contiene información referente a la relación señal a ruido, intensidad de la señal, frecuencia
RMC	<i>Recommended Minimum Specific data</i>	Contiene información referente a la hora UTC, fecha, rumbo y datos de posición.
VTG	<i>Course over ground and ground speed</i>	Contiene información del curso y velocidad con respecto al suelo
ZDA	<i>SiRF Timing Message</i>	Mensaje de sincronización PPS.

Tabla 2.13. Mensajes de salida del protocolo NMEA 0183.

2.7.2. Tipos de receptores GPS

Existen diferentes tipos de receptores GPS, dependiendo de sus características se pueden generalizar en receptores geodésicos de doble frecuencia, receptores geodésicos con

medición de fase sobre L1, receptores de código avanzado y los navegadores convencionales.

Receptores geodésicos de doble frecuencia

Trabajan con la portadora L1 y L2, lo cual permite disminuir los errores derivados de la propagación desigual de la señal a través de las distintas capas atmosféricas (sobre todo en la ionosfera) y resolver la ambigüedad del número de ciclos de la portadora.

Con este tipo de equipos se puede llegar a obtener precisiones por debajo del centímetro, con post – procesamiento, para distancias de hasta 10 km, y por debajo del metro para distancias de hasta 500 km.

Receptores geodésicos con medición de fase sobre L1

Son receptores que trabajan con la onda portadora L1, acumulando información que con un post procesamiento en gabinete, permite obtener precisiones relativas centimétricas, en el mejor de los casos para distancias de hasta 25 o 30 km y submétricas para distancias de hasta 50 km; además de que permiten el cálculo de vectores con su elevación estadística y son aptos para el ajuste de redes.

Receptores de código avanzado

Este tipo de receptores además de analizar el código C/A disponen de lectura (con ciertas limitaciones) de la fase portadora L1, permiten el uso de metodologías diferenciales, en ocasiones bajo la forma de suscripciones a servicios vía satélite, consiguiendo con esta tecnología precisiones en torno a un metro en tiempo real.

Este tipo de servicio tiene la enorme ventaja de que se dispone de corrección instantánea sin necesidad de montar ninguna estación de referencia, y para casi para cualquier parte del mundo en tiempo real.

Navegadores convencionales

Los navegadores son los tipos de receptores GPS más extendidos, dados su bajo costo y multiplicidad de aplicaciones, estos dispositivos consisten en receptores capaces de leer el código C/A, que pueden tener incluso capacidad para leer señales diferenciales vía radio y también capacidad para representar cartografía sencilla en una pantalla de cristal líquido.

Permiten conocer las coordenadas en varios formatos, además de tener navegación asistida con indicación de rumbos, direcciones y señales audibles de llegada en rutas definidas por el usuario a través de puntos de referencia.

2.8. Microcontrolador

El microcontrolador es un dispositivo que se emplea para el manejo de uno o varios procesos, por ejemplo, realizar tareas de procesamiento de datos provenientes de convertidores analógico digitales, realizar rutinas para almacenamiento de datos, adquirir los datos provenientes de un receptor GPS y realizar la interacción humano – máquina, a través de las instrucciones que se den con un *joystick* y mostrarlas en una pantalla de cristal líquido.

Un microcontrolador es un circuito de alta escala de integración que incorpora en un solo encapsulado los siguientes bloques:

- ✓ *Comparador analógico*: Algunos modelos de microcontroladores disponen internamente de un amplificador operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por un puerto de entrada. La salida del comparador proporciona un nivel lógico 1 o 0 según una señal sea mayor o menor que la otra.
- ✓ *Convertidor A/D*: Los microcontroladores que incorporan un convertidor A/D, pueden procesar señales analógicas, tan abundantes en las aplicaciones; suelen disponer de un multiplexor que permite aplicar a la entrada del convertidor diversas señales analógicas.
- ✓ *Convertidor D/A*: Transforma los datos digitales obtenidos del procesamiento del computador en su correspondiente señal analógica que saca al exterior por los puertos de salida del dispositivo.
- ✓ *Memoria RAM*: Sirve para contener los datos.
- ✓ *Memoria para el programa* tipo ROM/PROM/EPROM.
- ✓ *Modulador de ancho de pulso (PWM)*: Son circuitos que proporcionan en su salida impulsos de anchura variable, que se ofrecen al exterior a través los puertos de salida.
- ✓ *Perro guardián (watch dog)*: Este modulo consiste en un temporizador, que cuando se desborda y pasa por cero, provoca un *reset* automáticamente en el sistema. Se debe diseñar el programa de trabajo que controla la tarea de forma que refresque o inicialice el perro guardián antes de que provoque el *reset*.
- ✓ *Protección ante falla de alimentación (Brownout)*: Se trata de un circuito que reinicia al microcontrolador cuando el voltaje de alimentación es inferior a un voltaje mínimo, comenzando a funcionar normalmente cuando sobre pasa dicho valor.
- ✓ *Puertos de comunicación*: Estos elementos dotan al microcontrolador de la posibilidad de comunicarse con otros dispositivos externos, puertos de microcontroladores, puertos de sistemas y poder adaptarlos con otros elementos bajo otras normas y protocolos.

Algunos modelos disponen de recursos que permiten directamente esta tarea, entre los que destacan:

- UART, adaptador de comunicación serie asíncrona.
 - USART, adaptador de comunicación serie síncrona y asíncrona.
 - Puerto paralelo esclavo para poder conectarse con las líneas de datos y direcciones de otros microprocesadores.
 - USB (*Universal Serial Bus*), este es un protocolo de comunicación serial que se usa para comunicar a los microcontroladores con las PCs.
 - Bus I^2C , que es una interfaz de comunicación de datos serial de dos hilos, desarrollado por Philips.
 - CAN (*Controller Area Network*), para permitir la adaptación con redes de conexión multiplexado, desarrollado conjuntamente por Bosch e Intel para el cableado de los dispositivos en automóviles.
-
- ✓ *Puertos de Entrada/Salida*: La principal función de estos puestos es proveer terminales que permitan controlar y recibir información de dispositivos periféricos, como teclados, pantallas de cristal líquido, relevadores, motores, entre otros.
 - ✓ *Reloj de Sistema*: Todos los microcontroladores disponen de un circuito oscilador que genera una onda de alta frecuencia, que configura los impulsos de reloj usados en la sincronización de todas las operaciones del sistema. Generalmente, el circuito de reloj está incorporado en el microcontrolador y sólo se necesitan unos pocos componentes exteriores para seleccionar y estabilizar la frecuencia de trabajo; dichos componentes suelen consistir en un cristal de cuarzo junto a elementos pasivos o bien un resonador cerámico o una red R-C.
 - ✓ *Temporizadores (Timers)*: Se emplean para controlar periodos de tiempo y para llevar la cuenta de acontecimientos que suceden en el exterior (contadores). Para la medida de tiempos se carga un registro con el valor adecuado y a continuación dicho valor se va incrementando o decrementando al ritmo de los pulsos de reloj o algún múltiplo hasta que se desborde y llegue a cero, momento en el que se produce un aviso.

Los sistemas que para su funcionamiento emplean microcontroladores disponen de las siguientes ventajas:

- ✓ *Aumento de la fiabilidad*: Al reemplazar el microcontrolador por un elevado número de elementos, disminuye el riesgo de averías y se precisan menos ajustes.
- ✓ *Reducción del tamaño en el producto acabado*: La integración del microcontrolador en un circuito integrado disminuye el volumen del sistema final.

- ✓ *Mayor flexibilidad:* Debido a que las características de control están programadas, la modificación de las funciones a realizar sólo necesita cambios en el programa de instrucciones.

Existe una gran diversidad de microcontroladores, quizá la clasificación más importante sea entre microcontroladores de 4, 8, 16 o 32 *bits*; aunque las prestaciones de los microcontroladores de 16 y 32 *bits* son superiores a los de 4 y 8 *bits*, la realidad es que los microcontroladores de 8 *bits* dominan el mercado y los de 4 *bits* se resisten a desaparecer.

La razón de esta tendencia es que los microcontroladores de 4 y 8 *bits* son apropiados para la gran mayoría de las aplicaciones, por lo que no tiene caso emplear microcontroladores de 16 o 32 *bits* ya que sería el elemento más caro de todo el sistema.

En cuanto a las técnicas de fabricación, cabe decir que prácticamente la totalidad de los microcontroladores actuales se fabrican con tecnología CMOS (*Complementary Metal Oxyde Semiconductor*), esta tecnología supera a las técnicas anteriores de fabricación debido a su bajo consumo de energía y alta inmunidad al ruido.

También los modernos microcontroladores de 32 *bits* van afianzando posiciones en el mercado, siendo las áreas de más interés el procesamiento de imágenes, las comunicaciones, las aplicaciones militares, los procesos industriales y el control de los dispositivos de almacenamiento de datos.

Antes de seleccionar un microcontrolador es indispensable analizar los requisitos de aplicación:

Procesamiento de datos: Puede ser necesario que el microcontrolador realice cálculos críticos en un tiempo limitado, en ese caso debemos asegurarnos de seleccionar un dispositivo lo suficientemente rápido para ello. Por otro lado, habrá que tener en cuenta la precisión de los datos a manejar, ya que si no es suficiente con un microcontrolador de 8 *bits*, puede ser necesario acudir a microcontroladores de 16 o 32 *bits*, o incluso se puede pensar en hardware de punto flotante. Una alternativa más barata es usar librerías para manejar datos de alta precisión.

- *Entrada Salida:* Para determinar las necesidades de Entrada/Salida del sistema, es conveniente dibujar un diagrama de bloques del mismo, de tal forma que sea sencillo identificar la cantidad y tipo de señales a controlar. Una vez realizado este análisis puede ser necesario añadir periféricos, hardware externos, o cambiar a otro microcontrolador que se adecue más al sistema.
- *Consumo:* Algunos productos que incorporan microcontroladores están alimentados con baterías y su funcionamiento puede ser tan vital como el hecho de que el almacenamiento de datos no se vea interrumpido, ya que puede perderse

información del fenómeno de interés; lo más conveniente es que el microcontrolador esté en estado de bajo consumo pero que despierte ante la activación de una señal (una interrupción) y ejecute el programa adecuado para procesarla.

- *Memoria:* Para detectar las necesidades de memoria de muestra aplicación, se debe separar en memoria volátil (RAM), memoria no volátil (ROM, EPROM, etc.) y memoria no volátil modificable (EEPROM). Este último tipo de memoria puede ser útil para incluir información específica de la aplicación como un número de serie o parámetros de calibración.
- *Ancho de palabra:* El criterio de diseño debe ser seleccionar un microcontrolador de menor ancho de palabra que satisfaga los requerimientos de la aplicación. Usar un microcontrolador de 4 *bits* supondrá una reducción importante en los costos, mientras que uno de 8 *bits* puede ser el más adecuado si el ancho de los datos es de un byte. Los microcontroladores de 16 y 32 *bits*, debido a su elevado costo, deben reservarse para aplicaciones que requieran sus altas prestaciones (Entrada/Salida potente o espacio de direccionamiento muy elevado).

Los microcontroladores pueden ser clasificados de acuerdo al conjunto de instrucciones que soportan:

- ✓ CISC (*Complex Instruction Set Computer*), disponen de más de 80 instrucciones máquina, algunas de las cuales requieren muchos ciclos para su ejecución.
- ✓ RISC (*Reduced Instruction Set Computer*), el repertorio de instrucciones máquina es muy reducido y las instrucciones son simples y, generalmente, se ejecutan en un ciclo.

Otra forma en la que son clasificados los microcontroladores es por el tipo de arquitectura por la que están formados, es decir la manera en que el microcontrolador accede a los recursos con que dispone mencionados anteriormente.

2.8.1. Arquitectura Von Newmann

Algunos sistemas con microprocesadores basan su estructura interna según la arquitectura Von Newmann; en la cual la unidad central de proceso (CPU) está conectada a una memoria principal única (la cual puede ser EEPROM), 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 2.49 se ilustra dicha arquitectura.

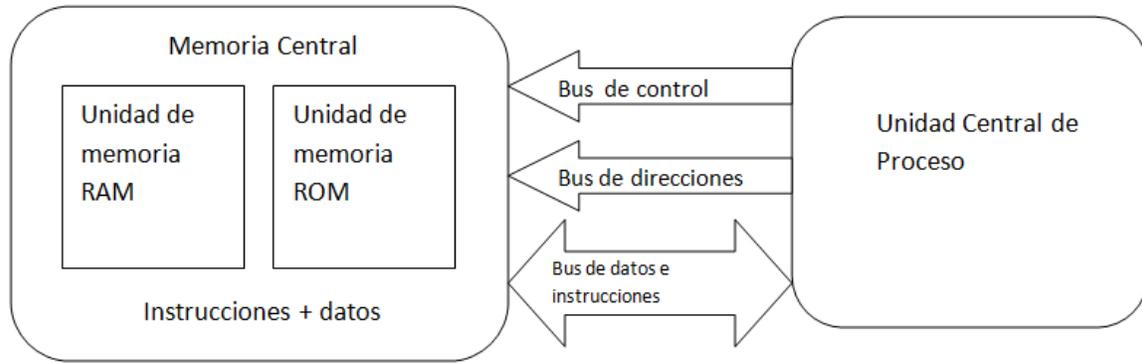


Figura 2.49. Arquitectura Von Neumann.

En un sistema con arquitectura Von Newman, el tamaño de la unidad de datos o instrucciones está fijado por el ancho de bus de datos y direcciones que comunica la memoria con la CPU. Así un microcontrolador de 8 bits con un bus de 8 bits, tendrá que manejar datos e instrucciones de una o más unidades de 8 bits 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 es su respuesta, ya que no puede buscar en memoria una nueva instrucción mientras no finalicen las transferencias de datos de la instrucción anterior.

2.8.2. Arquitectura Harvard

En este tipo de arquitectura se tiene la CPU conectada a dos memorias (una con las instrucciones y otra con los datos) por medio de dos buses diferentes. En la figura 2.50 se ilustra este tipo de arquitectura.

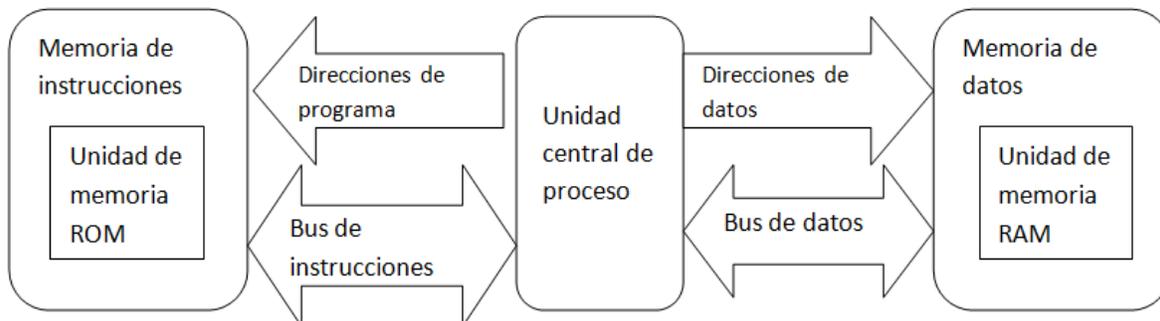


Figura 2.50. Arquitectura Harvard.

Una de las memorias contiene solamente las instrucciones del programa (Memoria de Programa) y los otros sólo almacenan datos (Memoria de Datos).

Ambos *buses* son totalmente independientes, lo que permite que la CPU pueda acceder de forma simultánea a la memoria de datos y a la de instrucciones. Como los *buses* son independientes, éstos pueden tener distintos contenidos en la misma dirección y también distinta longitud. Así mismo, la longitud de los datos y las instrucciones puede ser distinta, lo que optimiza el uso de la memoria general.

Al ser los *buses* independientes, la CPU puede acceder a los datos para completar la ejecución de una instrucción y al mismo tiempo leer la siguiente instrucción a ejecutar.

2.8.3. Familia de microcontroladores

Actualmente existe una gran variedad de marcas y modelos de microcontroladores. En donde, dependiendo de la aplicación que se desee desarrollar se escogerá el microcontrolador que más se adecue a las necesidades del diseño, ya que como se ha mencionado, la mayoría de microcontroladores están compuestos por los módulos ya descritos, lo único importante para la elección de un microcontrolador es conocer las especificaciones de cada dispositivo, con la finalidad de que se adapte mejor a nuestro sistema. Algunos de estos microcontroladores tienen una forma más accesible en cuanto a su programación, ya que se pueden programar por lenguaje ensamblador o lenguaje C, lo más importante es que el usuario se familiarice con el entorno de programación de cada marca de microcontrolador, ya que en general todos se comportan igual al momento de desarrollar una tarea.

En la tabla 2.14. Se hace referencia a los microcontroladores más comunes en uso, ya que el hablar de cada familia de cada marca haría muy extenso este apartado.

Empresa	8 bits	16 bits	32 bits
Atmel	AVR (mega y tiny), 89Sxxx familia similar 8051	-----	SAM7(ARM7TDMI), SAM3(ARM Cortex-M3), SAM9 (ARM926), AVR32
Freescle (antes motorola)	68HC05, 68HC08, 68HC11, HCS08	68HC12,68HCS12,68H CX12,68HC16	683xx,Power PC,ColdFire
Fujitsu	$F^2MC - 8FX$	$F^2MC - 16FX$ $F^2MC - 16FXS$ $F^2MC - 16LX$	FR60Lite FR60/70 FR80/81
Intel	MCS-48 (familia 8048) MCS51(familia 8051) 8xC251	MCS96, MXS296	-----

Tabla 2.14. Microcontroladores más comunes en el mercado. (Continúa)

Empresa	8 bits	16 bits	32 bits
INFINEON	C500 XC800	C100 XC100 XE100	T1130 TC116x TC1167x
Microchip	Familia 10f2xx, 12Cxx,12Fxx, 16Cxx, 16Fxx, 18Cxx, 18Fxx	PIC24F, PIC24H, y DSPIC30FXX, DSPIC33F	Pic32
NXP Semiconductors (antes Philips)	80C51 80C552 80C31	PXAC37 XA-G30 XA-S3	Cortex M3, Cortex M0,ARM7,ARM9 LPC1300
Renesas (antes Mitsubishi, Hitachi y NEC)	78K, H8	H8S,78K0R,R8C,R32C /M32C/M16C	RX,V850,SuperH,SH-Mobile,H8SX
ST Microelectronics	STM8L, STM8S, STM8AF, STM8T		STM32 (ARM7)
Texas Instruments	TMS370	MSP430	C2000, CortexM3(ARM),TMS570(ARM)
Zilog	Z51F0410 Z51F0811 Z51F3220 Z51F6412	Z16F2810 Z16F2811 Z16F3211 Z16F6411	-----
ARM	-----	-----	Cortex –A series Cortex –R series Cortex –M series

Tabla 2.14. Microcontroladores más comunes en el mercado.

2.9. Dispositivos periféricos

En un microcontrolador existen terminales designadas para la entrada y salida de datos. Dichas terminales son usadas para el intercambio de información entre el microcontrolador y el mundo exterior.

Los dispositivos de entrada son aquellos dispositivos auxiliares de un sistema que permiten detectar información del mundo exterior, mientras que los dispositivos de salida son aquellos indicadores y dispositivos auxiliares que nos permiten ver resultados de los

procesos que se están realizando dentro del sistema. A este conjunto de dispositivos se les llama periféricos.

2.9.1. Joystick

Un joystick es un dispositivo de entrada digital usado para el control de dos o tres ejes. En su composición más elemental está formado por un conjunto de interruptores (botones o teclas) en cruzeta, situada en la base, más un botón de acción. Estos botones son de contacto mecánico que generan un estado lógico ('1' o '0') al cerrar o abrir dicho interruptor. En la figura 2.51 se muestran algunos de los tipos de *joystick* que existen en el mercado



Figura 2.51. Imágenes de algunos modelos de Joystick.

La forma de onda que entrega un interruptor en forma ideal es la mostrada en la figura 2.52.

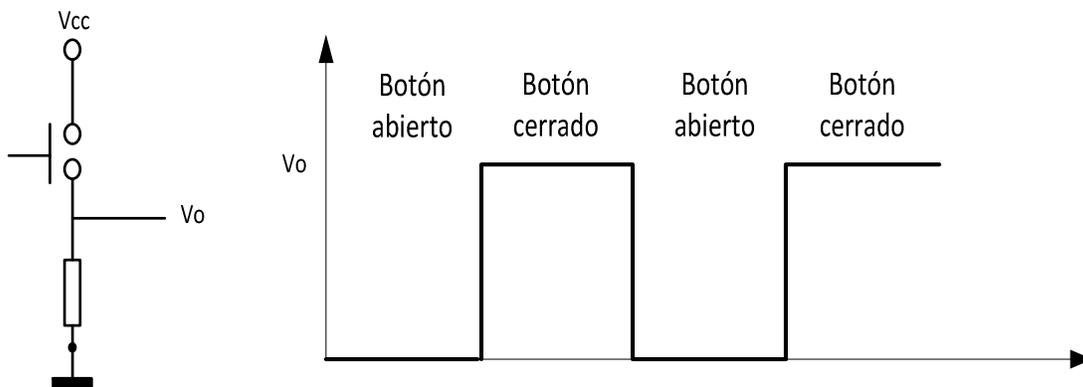


Figura 2.52. Comportamiento ideal de la señal de una tecla o botón.

Sin embargo, debido a las características mecánicas de los elementos que constituyen a un botón, se presenta una vibración (efecto rebote) en la señal cuando se cierra o abre el botón; este efecto se muestra en la figura 2.53.

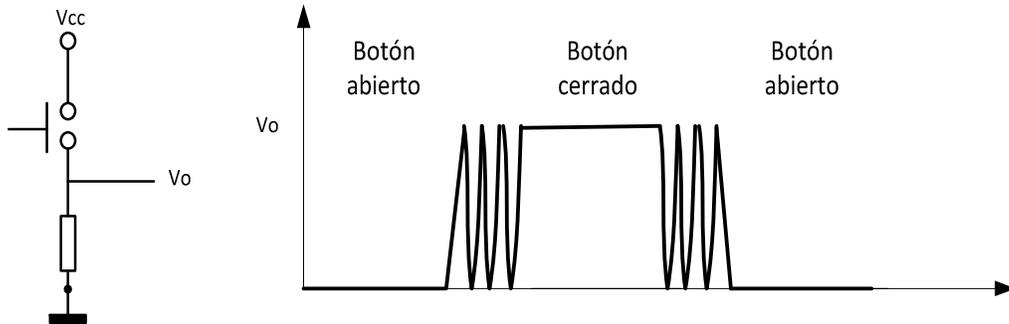


Figura 2.53. Efecto de rebote mecánico al presionar una tecla o botón.

Este efecto rebote es indeseado, ya que puede causar problemas en la lectura del estado lógico del botón. El microcontrolador puede interpretar este efecto rebote como si se pulsara varias veces, provocando errores en las secuencias o rutinas de programación cuando se esté leyendo los estados o niveles lógicos de estos botones.

Existen varias formas de eliminar este efecto. Una de ellas hace uso de componentes adicionales conectados al botón, entre los cuales sobresalen, capacitores (filtros) o compuertas lógicas. Una alternativa es agregar retardos en las rutinas encargadas de leer el estado lógico del botón, este nos permite esperar que se establezca la señal proporcionada por el botón al ser presionado y una vez estable la señal se procede a leer el estado lógico del botón, como se muestra en la figura 2.54.

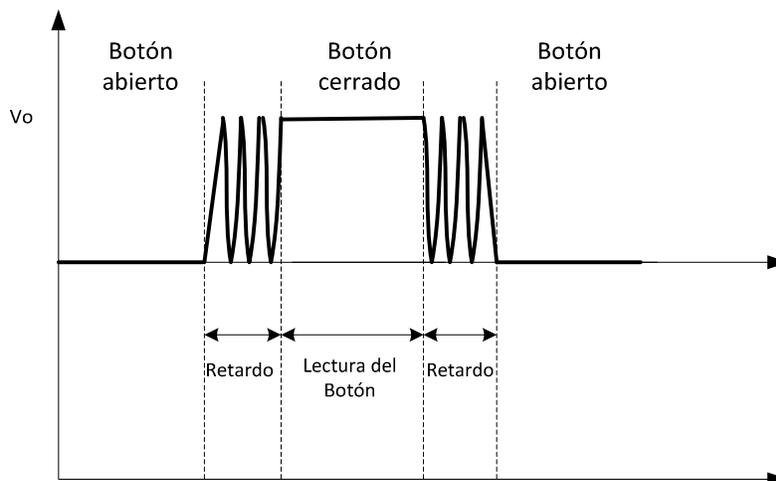


Figura 2.54. Uso de rutinas de retardo para eliminar el efecto de rebote.

2.9.2. Pantalla gráfica de cristal líquido

En general, las pantallas de cristal líquido (LCD, Liquid Crystal Display) son visualizadores pasivos, con un bajo consumo de energía. Están formadas por un cristal líquido entre dos placas paralelas de vidrio a una cierta separación. Estas placas de vidrio tienen unos electrodos especiales, que definen con su forma los caracteres o símbolos que se visualizarán. La superficie del vidrio que hace contacto con el líquido es tratada de manera que induzca la alineación de los cristales en dirección paralela a las placas. Esta alineación permite el paso de la luz incidente sin ninguna alteración. Cuando se aplica la polarización adecuada entre los electrodos, se forma un campo eléctrico perpendicular entre ellos, el cual causa que las moléculas del líquido se agrupen en sentido paralelo al campo eléctrico y aparezca una zona oscura sobre un fondo claro.

Para el caso de una pantalla grafica GLCD (Graphic Liquid Crystal Display), el funcionamiento es exactamente el mismo, con la diferencia que no se tiene un conjunto de caracteres o símbolos, en cambio, se cuenta con una matriz de pixeles que pueden dar forma a caracteres, símbolos, líneas y figuras definidas por el usuario, al encender o apagar un pixel. En la figura 2.55 se muestran algunos tipos de GLCD.

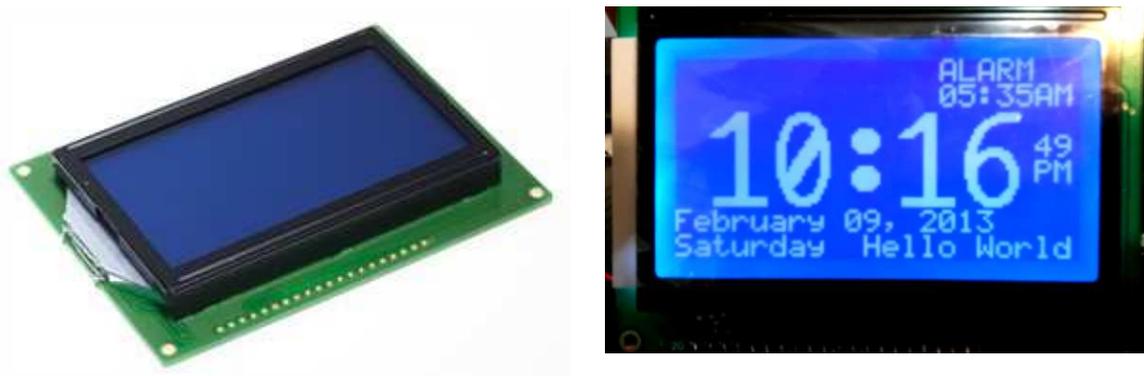


Figura 2.55. Pantalla gráfica de cristal líquido (GLCD).

2.9.3. Comunicación Serial RS-232

El estándar de comunicación RS-232 (*Recommended Standard 232*) es una norma mundial que rige los parámetros de uno de los modos de comunicación serial. Por medio de éste se rigen las velocidades de transferencia de datos, el control que usa dicha transferencia, los niveles de voltajes aceptados, el tipo de cable permitido y la distancia entre equipo y conectores.

Los márgenes de voltaje aceptado por este estándar se describen en la tabla 2.15.

Valor lógico	Margen de voltaje transmisor	Margen de voltaje receptor
1	-15 a -5 Volts	-15 a -3 Volts
0	5 a 15 Volts	3 a 15 Volts

Tabla 2.15. Niveles de voltaje para los valores lógicos en el estándar RS-232.

La interfaz RS-232 fue diseñada para realizar la transmisión de datos con velocidades de hasta 20 kbps, a una distancia máxima de 15 metros entre transmisor y receptor. Esta interfaz puede trabajar de manera síncrona o asíncrona y con tipos de canales simplex, half dúplex y full dúplex. Los conectores usados para esta comunicación son DB25 (25 terminales) y DB9 (9 terminales), como se muestran en la figura 2.56.

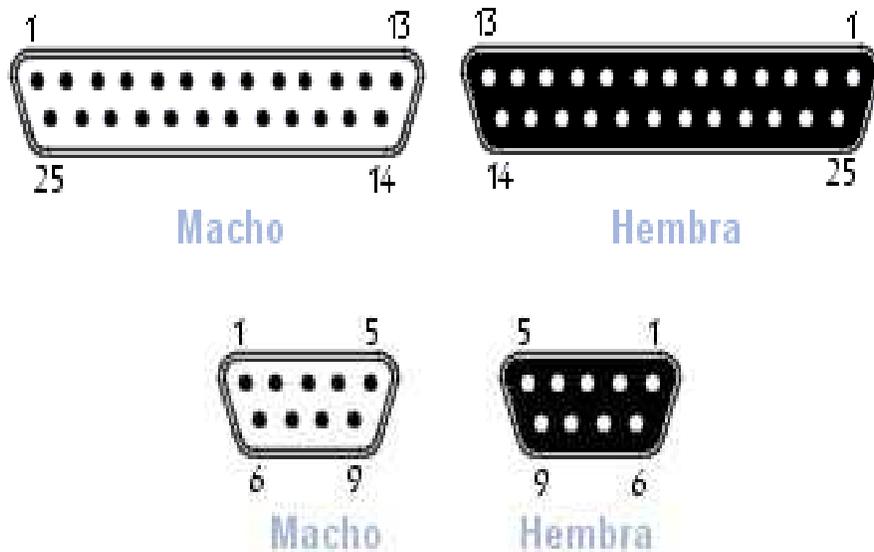


Figura 2.56. Conectores DB25 y DB9.

En la tabla 2.16 se describen las funciones de cada terminal del conector DB9, que es el usado en este proyecto.

Terminal	Nombre	Función
1	CD	Detector de transmisión (Carrier Detect)
2	RXD	Recepción de datos (Receive Data)
3	TXD	Transmisión de datos (Transmit Data)

Tabla 2.16. Descripción de las terminales del conector DB9. (Continúa)

Terminal	Nombre	Función
4	DTR	Terminal de datos lista (Data Terminal Ready)
5	GND	Tierra
6	DSR	Conjunto de datos lista (Data Set Ready)
7	RTS	Permiso para transmitir (Request to Send)
8	CTS	Listo para enviar (Clear to Send)
9	RI	Indicador de llamada (Ring Indicator)

Tabla 2.16. Descripción de las terminales del conector DB9.

Uno de los transceptores más usados para este estándar es el MAX232, el cual permite adaptar los niveles de voltaje TTL de un microcontrolador a los niveles de voltaje del estándar RS-232 o viceversa.

En este capítulo se ha descrito de manera general los temas necesarios para poder llevar a cabo el desarrollo y diseño del sistema “DIGI”. En el siguiente capítulo se describirá ampliamente el diseño del dispositivo.

Capítulo III

Diseño y desarrollo del sistema

“DIGI”

En este capítulo se presentan las necesidades que debe cubrir el proyecto, por lo que se describe ampliamente el diseño y desarrollo del sistema “DIGI”; el cual y de forma general se compone por un hardware, integrado por todos los elementos electrónicos conectados entre sí, y un software, que permite realizar las tareas de digitalización, sincronización y almacenamiento de las señales sísmicas.

3.1. Definición de requerimientos del sistema

En la actualidad existen diferentes sistemas que permiten el registro y almacenamiento de señales que involucran fenómenos geológicos, sin embargo, el costo de estos sistemas es elevado, debido a la gran complejidad de su circuitería y de los mantenimientos preventivos que se les tiene que dar para su correcta operación. Además, de que se debe contar con el software del fabricante de cada dispositivo, con la finalidad de poder configurarlo y procesar los datos, haciendo que el sistema dependa de una PC tanto para su configuración como para la verificación de su funcionamiento. Por otra parte, estos dispositivos son utilizados para realizar pruebas fuera del laboratorio, por lo que su tamaño y peso son lo más compacto posible para poder trasladarlos sin mayor complicación.

El propósito principal del diseño y desarrollo del sistema “DIGI”, es crear un prototipo que permita digitalizar las señales provenientes de los sensores utilizados para la detección de sismos, en donde primeramente se deberá sincronizar con un GPS para obtener los datos de fecha y hora, y posteriormente se almacenarán los datos en una memoria micro SD, con la finalidad de que una vez terminada la prueba se procesen los datos y se tenga un historial de cuando se presentó dicho fenómeno sísmico. Además, a dicho equipo se le integrará una interfaz usuario – máquina, formado por un GLCD y un *joystick*, los cuales permiten realizar la configuración del sistema, brindándole una cierta autonomía, ya que muchos de los equipos para este fin requieren de una PC, como se menciono anteriormente. Por lo que en el sistema “DIGI” se estará verificando en el GLCD que los datos provenientes del GPS y de los tres canales de digitalización sean correctos.

Teniendo en cuenta lo anterior, el prototipo “DIGI” deberá contar con los siguientes requerimientos:

- ✓ Configuración de la tasa de muestreo (50, 100 y 200 muestras por segundo).
- ✓ Verificar que los datos del GPS sean correctos, es decir, deberá desplegar la fecha y hora antes de realizar la prueba.
- ✓ Verificar que los tres canales operen adecuadamente, a través de un modo de prueba, en donde se graficarán los datos de cada canal en el GLCD.
- ✓ Asignarle un nombre a la unidad por medio de la interfaz usuario – máquina, esto con la finalidad de identificar de que unidad provienen los datos, al momento de realizar el procesamiento de los mismos.
- ✓ Fijar el tiempo de duración de la prueba, que puede ser desde una hora hasta un mes.

3.2. Desarrollo del Hardware

Con la finalidad de cumplir el propósito descrito en la sección anterior, se requiere integrar una serie de componentes electrónicos, como es el caso de los sensores sísmicos, (geófonos), un módulo de acondicionamiento de las señales provenientes de los geófonos, un módulo para la digitalización de las señales analógicas, un módulo de almacenamiento de datos en la memoria micro SD, un módulo para la interfaz usuario - máquina, y un GPS que permite la sincronización de los datos.

En la figura 3.1. Se ilustra de forma general el diagrama de bloques con los elementos ya mencionados.

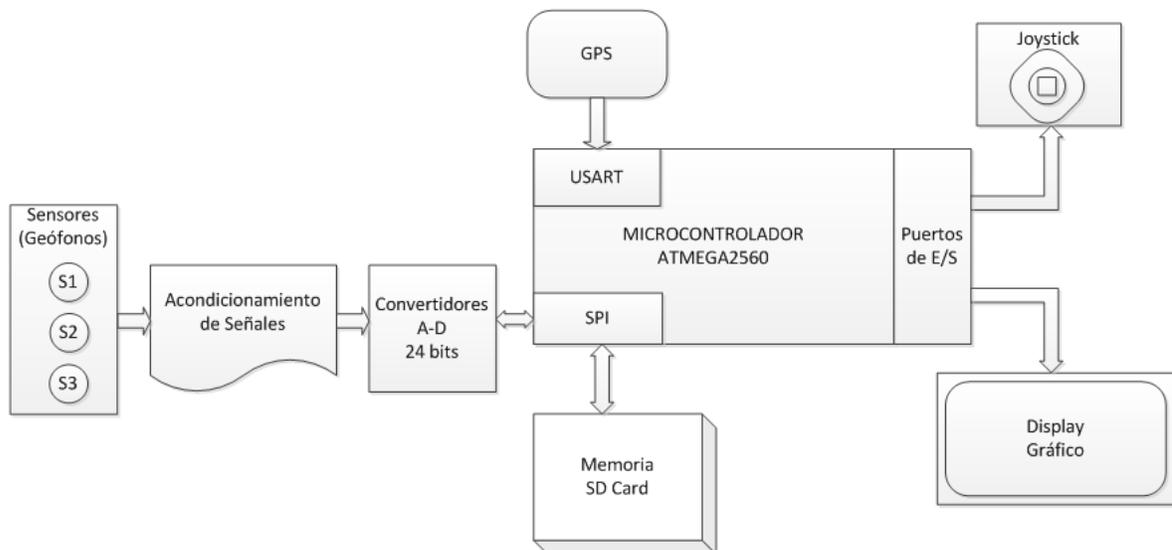


Figura 3.1. Elementos que forman el hardware del sistema.

En los siguientes puntos se hará una descripción más detallada de los módulos que integran el sistema.

3.2.1. Geófono GS-11D

El geófono GS-11D es un sensor de tipo electromagnético, fabricado por Geo Space. La estructura de este geófono es de un cilindro metálico, en donde se aloja un imán y una bobina sostenida por un resorte; este dispositivo está diseñado y construido para resistir choques de manejo brusco. En la figura 3.2 se ilustra la estructura interna del geófono.

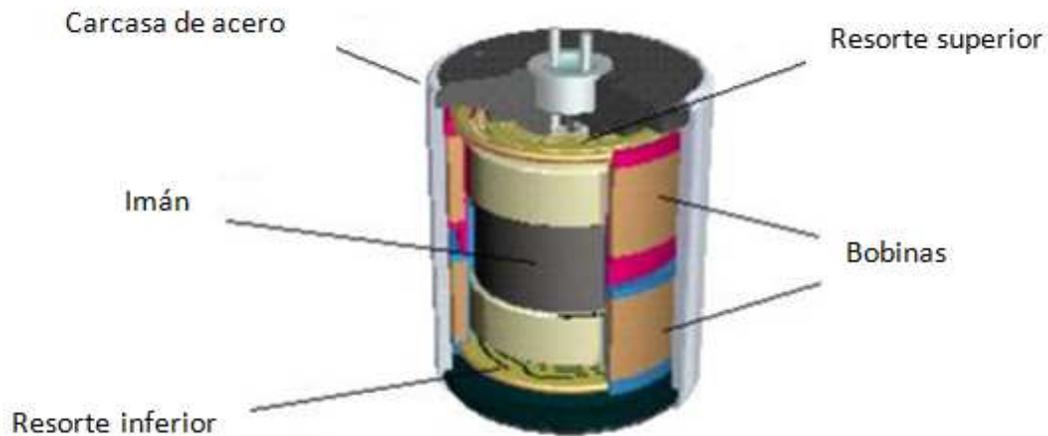


Figura 3.2. Estructura del geófono GS-11D.

Por otra parte, el geófono cuenta con dos contactos en la parte superior, que es en donde se toma la señal de salida para ser conectada a la entrada del sistema. Como se menciona en la sección 2.1, del capítulo 2, entre estas terminales se coloca una resistencia para modificar el amortiguamiento del sistema.

En la figura 3.3 se muestra la curva de respuesta de velocidad proporcionada por el fabricante, en donde según las hojas de especificaciones, se indica que para obtener un amortiguamiento del 35% (gráfica A), se deben dejar libres los contactos de salida; pero para un amortiguamiento del 50% (gráfica B) y 70% (gráfica C), se debe conectar un resistor entre las terminales de los geófonos de $4420\ \Omega$ y $1740\ \Omega$ respectivamente.

En nuestro caso se colocó una resistencia de $1.8\ \text{k}\Omega$, ya que es el valor comercial más cercano a $1740\ \Omega$ que sugiere el fabricante, para obtener un amortiguamiento del 70%, y como se mencionó anteriormente, este tipo de sensores tienen una mejor respuesta cuando se tiene un sistema críticamente amortiguado.

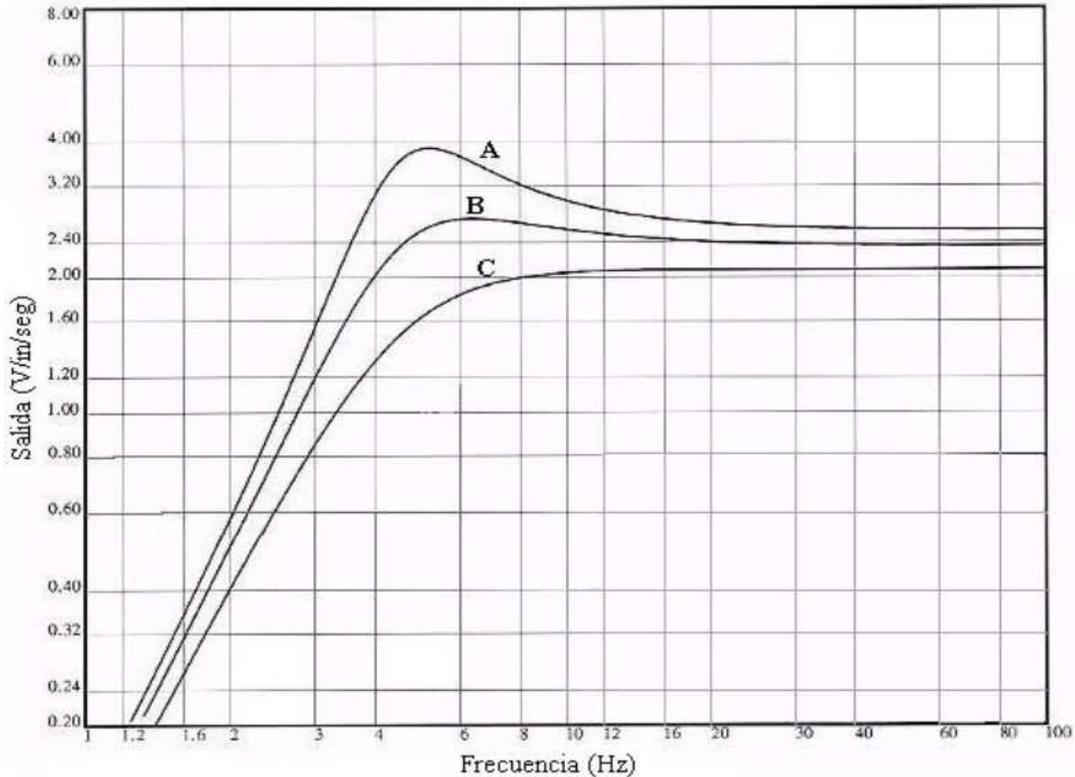


Figura 3.3. Curva de la respuesta del geófono GS-11D.

En el proceso de fabricación de los geófonos, de la compañía Geo Space, se incluye la comprobación de todos los parámetros de funcionamiento de los mismos; sin embargo, y en particular este modelo de geófono, debe utilizarse de acuerdo a la componente que se desee medir, ya sea vertical u horizontal, ya que las hojas de especificaciones muestran una tolerancia en grados para que los sensores operen adecuadamente.

Las especificaciones generales del geófono GS-11D para componente vertical y horizontal son las siguientes:

- ✓ Frecuencia natural: 4.5 ± 0.75 Hz.
- ✓ Resistencia de la bobina: 380Ω a $25^\circ\text{C} \pm 5\%$.
- ✓ Sensibilidad de voltaje: 0.81 V/in/seg (0.32 V/cm/seg).
- ✓ Masa móvil: $23.6\text{g} \pm 5\%$.
- ✓ Rango de operación vertical: posición vertical hasta 7° .
- ✓ Rango de operación horizontal: posición horizontal $\pm 0.5^\circ$.
- ✓ Dimensiones
 - Altura: 1.32 in (3.35 cm).
 - Diámetro: 1.25 in (3.18 cm).
 - Peso: 3.9 oz (111gr).

3.2.2. Microcontrolador AVR

El microcontrolador es el elemento más importante para el desarrollo de este sistema, ya que a través de un programa previamente diseñado se pueden controlar los dispositivos electrónicos que componen a todo el sistema, así como poder procesar la información realizando operaciones lógicas o aritméticas.

Para el desarrollo del sistema “DIGI” se utilizó un microcontrolador de la familia AVR, fabricado por la empresa norteamericana ATMEL. La razón por la cual se utilizó este microcontrolador en el diseño del proyecto fue, porque en un principio se pretendía utilizar la tarjeta de desarrollo “Arduino UNO”; sin embargo, al compilar el programa desarrollado para el sistema DIGI en la interfaz de programación de Arduino, resultó que dicho programa excedía la capacidad de la memoria flash de la tarjeta “Arduino UNO” (32 *kbytes*). Por tal motivo se decidió utilizar la tarjeta de desarrollo “Arduino MEGA”, en donde el microcontrolador en el que se basa dicha tarjeta tiene una capacidad de memoria flash de 256 *kbytes*, por lo que esta capacidad de memoria es suficiente para almacenar el programa del sistema “DIGI”.

Es importante mencionar que se utilizaron las tarjetas de desarrollo de Arduino, debido a que se contaba con la infraestructura y experiencia necesaria para desarrollar el prototipo del sistema con dichas tecnologías.

En la tabla 3.1 se muestran algunas de las características más importantes del microcontrolador utilizado.

Parámetro	Especificaciones
Memoria Flash	256 <i>kbytes</i>
Memoria EPROM	4 <i>kbytes</i>
Memoria interna SRAM	8 <i>kbytes</i>
Número de terminales	100
Frecuencia máxima de operación	16 MHz
CPU	8 <i>bits</i>
Interrupciones externas	32
Módulo maestro – esclavo	SPI
<i>Timers</i>	2 de 8 <i>bits</i> y 4 de 16 <i>bits</i>
PWM	4 canales PWM de 8 <i>bits</i>
Comunicación serial USART	4

Tabla 3.1. Características generales del microcontrolador AVR. (Continúa)

Diseño y Desarrollo del sistema “DIGI”

Parámetro	Especificaciones
Convertidor A/D	16 canales
Comparadores analógicos	1
<i>Timer Watchdog</i>	Sí
Voltaje de alimentación	4.5 – 5.5 V
Detección de bajo voltaje programable	Sí
Tipo de encapsulado	TQFP

Tabla 3.1. Características generales del microcontrolador.

En la figura 3.4 se muestra la distribución de terminales del microcontrolador AVR utilizado.

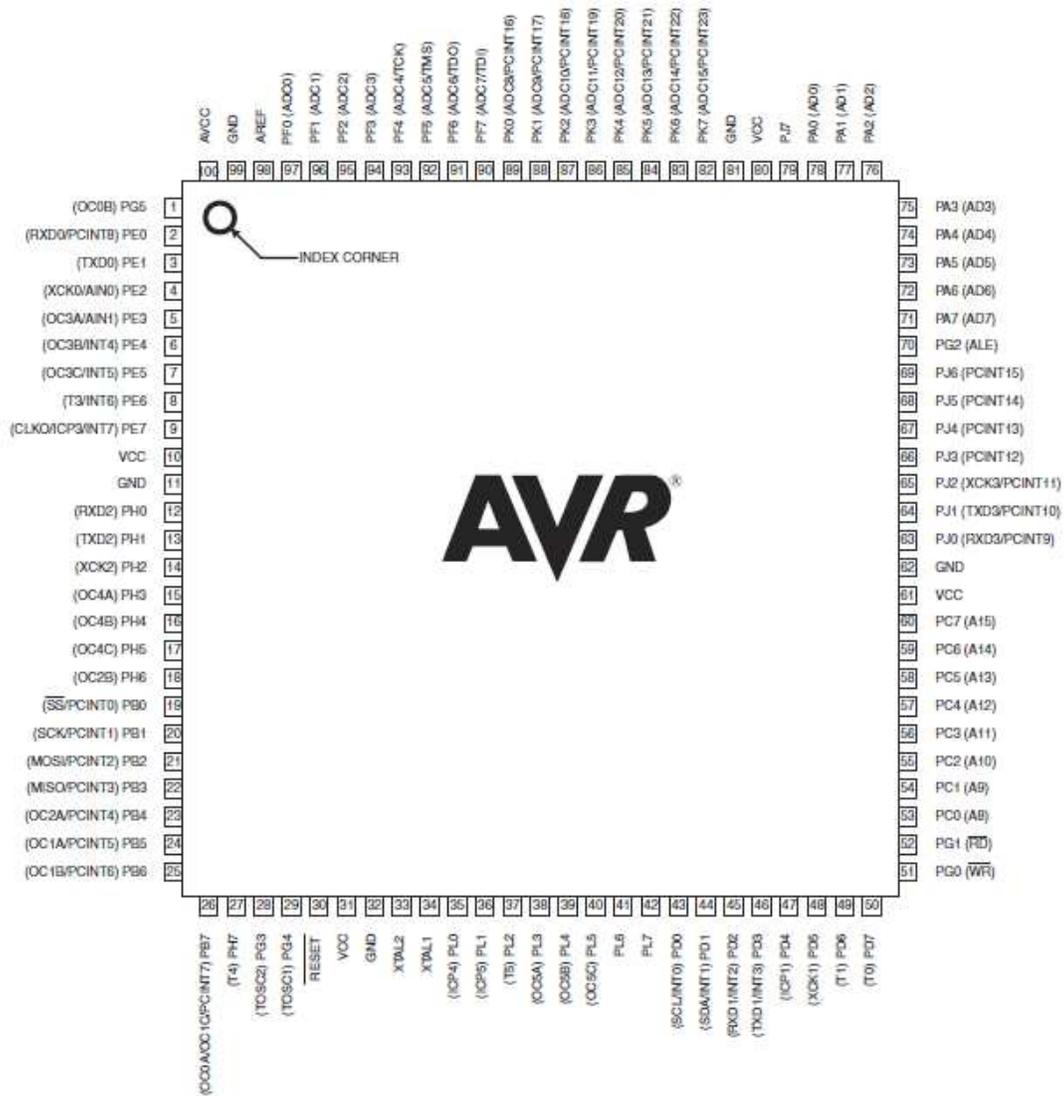


Figura 3.4. Distribución de terminales del microcontrolador.

3.2.3. Módulo de acondicionamiento de señal

Como se menciona en el apartado 2.2, las señales provenientes del medio físico, como son presión, velocidad, aceleración, temperatura, etc., son adquiridas mediante sensores, los cuales convierten dichas variables en una señal eléctrica. Con base en las características de la señal de entrada tendríamos que definir el acondicionamiento que se le va a dar, en términos generales la señal tendrá que pasar por un bloque de filtrado para limitar la banda de la señal de interés, eliminando de esta manera componentes que contaminen a la señal. Posteriormente, la señal tendrá que pasar por un bloque de amplificación para brindarle una ganancia, esto último con la intención de que la señal se encuentre dentro de los parámetros eléctricos de entrada del convertidor analógico digital, el cual dará como resultado la digitalización de la señal analógica proveniente de los sensores.

Para el desarrollo del sistema “DIGI”, se diseñó un circuito para el acondicionamiento de las señales provenientes de los sensores sísmicos. En donde atendiendo a las necesidades del usuario, el circuito diseñado permite la conexión de sensores pasivos, como es el caso de los geófonos, como también sensores activos, como el caso de los acelerómetros de fuerza balanceada.

En las siguientes secciones se explicará el diseño del circuito de acondicionamiento de señal, tanto para sensores pasivos como para sensores activos.

- ***Acondicionamiento de señales para sensores pasivos***

Para el caso de los sensores pasivos, como se explicó en la sección 2.1, no se requiere de una fuente de alimentación externa para su funcionamiento, además de que en la estructura interna de estos (caso particular el geófono), se encuentra un imán que se mueve en función del movimiento relativo de la superficie a estudiar, por lo que el movimiento de este imán induce un voltaje a una bobina, siendo esta la señal eléctrica correspondiente a la velocidad con que se ha movido la superficie terrestre. El tipo de señal que se obtiene en las terminales de los geófonos es del tipo unipolar flotante, esto quiere decir que no se conecta a tierra alguna de las terminales del sensor, ya que si alguna de las terminales se conecta a tierra implicaría que sólo se estaría obteniendo señal de salida, únicamente cuando el imán se moviera hacia una sola dirección.

Con base en lo descrito anteriormente y para darle una cierta ganancia a la señal proveniente de los geófonos, se pensó en diseñar un amplificador en configuración de instrumentación, con la finalidad de eliminar las componentes en modo común, y posteriormente la salida de la señal del amplificador de instrumentación fuera digitalizada por medio de un convertidor A/D de 24 *bits*. Este último dato se debe a que los sistemas diseñados para registro de señales sísmicas, que se encuentran en el mercado, tienen esa resolución en la conversión de sus datos.

Diseño y Desarrollo del sistema “DIGI”

Al buscar un convertidor de 24 *bits* de resolución se encontró un dispositivo que ayudaría mucho al diseño del circuito, ya que el convertidor elegido en su estructura interna cuenta con una entrada diferencial, lo cual implica que la idea original se reduciría a eliminar el amplificador diferencial del amplificador de instrumentación. Aunado a esto, el convertidor analógico digital a utilizar tiene una arquitectura sigma – delta, lo cual y como se explicó en la sección 2.3.3, no se requiere de un circuito *sampling and hold* y el filtro anti *aliasing* se reduce a un filtro pasivo pasa bajas de primer orden.

En la tabla 3.2 se muestran las principales características eléctricas del amplificador operacional y del convertidor analógico digital, que fueron consideradas en el diseño del circuito de acondicionamiento.

Dispositivo	Amplificador operacional	Convertidor A/D sigma - delta
Voltaje de alimentación	-18 V a 18V	4.5 a 5.5 V
Corriente de alimentación	2mA por amplificador	8mA en proceso de conversión y 8uA en estado de <i>sleeping</i>
Impedancia de entrada	2 GΩ	110 kΩ
Impedancia de salida	1 Ω	Alta impedancia al desactivar convertidor y baja impedancia al activarlo ya que es una salida digital
Voltaje de entrada	30 mV	-0.5 Vref a 0.5 Vref
Slew rate	6 V/us	-----
Rechazo en modo común	120 dB	120 dB en la entrada diferencial

Tabla 3.2. Características eléctricas del amplificador y del convertidor A/D.

De la tabla 3.2 se puede observar que el voltaje de alimentación crítico es el correspondiente al convertidor analógico digital, ya que una de las especificaciones del diseño del equipo, dadas por el usuario, es utilizar una fuente de 12V. El convertidor A/D no puede ser polarizado con ese voltaje, ya que de ser así resultaría dañado, por lo que se utilizó un regulador de voltaje de 5V para alimentar tanto al convertidor A/D como también al amplificador operacional, este último se debe alimentar con dicha tensión, con la finalidad de que la señal amplificada se encuentre dentro del rango del voltaje de entrada del convertidor A/D.

Debido a las características eléctricas de la señal de salida del geófono, las cuales ya han sido mencionadas, es necesario agregarle a esta una componente de corriente directa denominada comúnmente como “*offset*”, ya que al energizar el amplificador con una fuente unipolar de 5V es necesario mover la referencia de voltaje, con la finalidad de que la señal proveniente del geófono se amplifique tanto para amplitudes positivas como

negativas. Por tal motivo, el *offset* agregado a la señal es de 2.5V, y con esta nueva referencia la señal amplificada estará dentro del rango de amplificación del amplificador operacional (0V – 5V). Para agregar dicho nivel de *offset* se utilizó un divisor de tensión entre la salida del regulador de tensión de 5V y la tierra del sistema, además de que a la salida del divisor de voltaje se le agregó una resistencia con un valor de 100 kΩ, esta última con la finalidad de limitar la corriente suministrada por el divisor de tensión.

En la figura 3.5 se muestra la configuración descrita para agregar un nivel de *offset* a la señal proveniente del geófono.

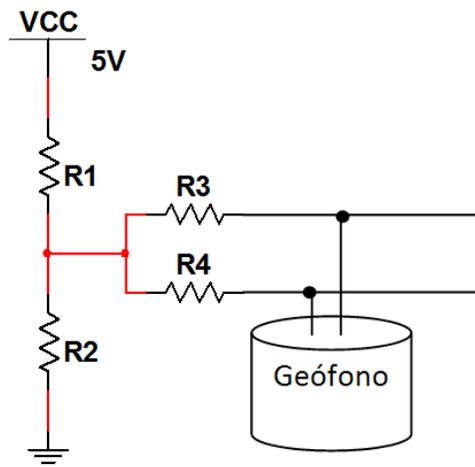


Figura 3.5. Circuito para agregar nivel de *offset* al geófono.

Es importante comentar que el nivel de *offset* agregado a la señal proveniente del geófono, no se verá reflejada al momento de la digitalización de la señal, ya que la entrada diferencial del convertidor A/D eliminara dicha componente, digitalizando única y exclusivamente la señal amplificada del geófono.

Una vez que se han descrito los niveles de tensión de alimentación de los dispositivos y el *offset* que se debe agregar a la señal de salida de los geófonos, es necesario definir la ganancia que se le dará al amplificador, para que la señal de salida de este mismo pueda ser procesada por el convertidor A/D. En la figura 3.6 se muestra la configuración del amplificador a utilizar.

Del análisis realizado al circuito mostrado en la figura 3.6, se determinaron las expresiones 3.1 y 3.2, las salidas VA y VB en función de las entradas V1 y V2, respectivamente.

$$V_A = V_1 \left(1 + \frac{R_1}{R_g} \right) - V_2 \left(\frac{R_1}{R_g} \right) \quad (3.1)$$

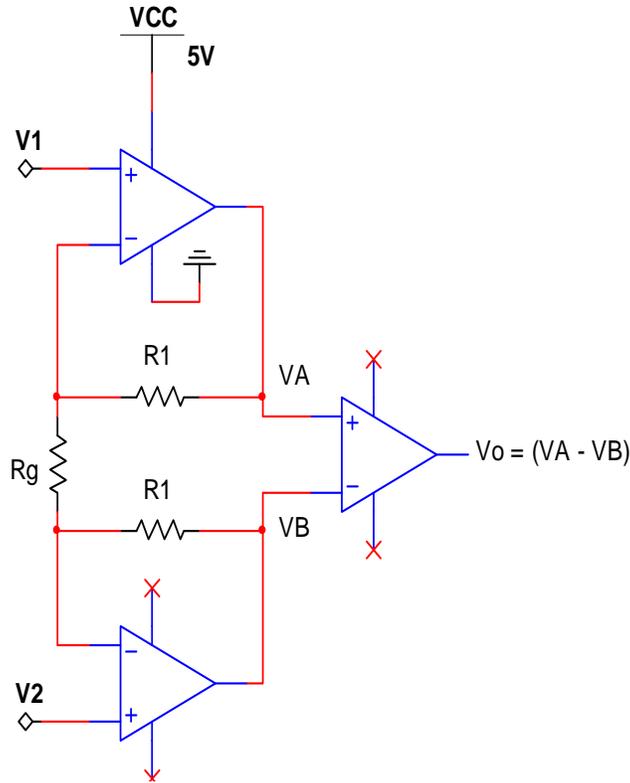


Figura 3.6. Amplificador de Instrumentación.

$$V_B = -V_1 \left(\frac{R_1}{R_g} \right) + V_2 \left(1 + \frac{R_1}{R_g} \right) \quad (3.2)$$

Efectuando la diferencia entre la salida VA y VB se obtiene la siguiente expresión.

$$V_o = \left(1 + 2 \frac{R_1}{R_g} \right) (V_1 - V_2) \quad (3.3)$$

De la ecuación 3.3 se puede obtener la función de transferencia que describe el comportamiento del circuito mostrado en la figura 3.6.

$$\frac{V_o}{V_1 - V_2} = \left(1 + 2 \frac{R_1}{R_g} \right) \quad (3.4)$$

De la función de transferencia mostrada en la expresión 3.4, se puede deducir que la ganancia está en función de la resistencia Rg, de esta manera si el valor de Rg es menor a R1 entonces se tendrá una ganancia mayor a 1.

Diseño y Desarrollo del sistema “DIGI”

Con base en el análisis anterior, para determinar el valor de la resistencia R_g que definirá la ganancia del amplificador, se tomó en cuenta que para no saturarlo, el voltaje de salida del amplificador tendría que ser igual a $4V_{pp}$, además de que se fijó un valor para R_1 en el orden de los $k\Omega$, y por último se tomó en cuenta el voltaje de la señal de salida de los geófonos que es igual a 0.32 V . Dicho lo anterior y con base a la expresión 3.4, se obtuvo que el amplificador requería de una ganancia igual a 12 dB para no saturar su salida.

Por otra parte, dentro de la hoja de especificaciones del convertidor A/D utilizado en el desarrollo del sistema “DIGI”, se recomienda agregar una resistencia y un capacitor en la red de realimentación del amplificador de instrumentación tal como se muestra en la figura 3.7.

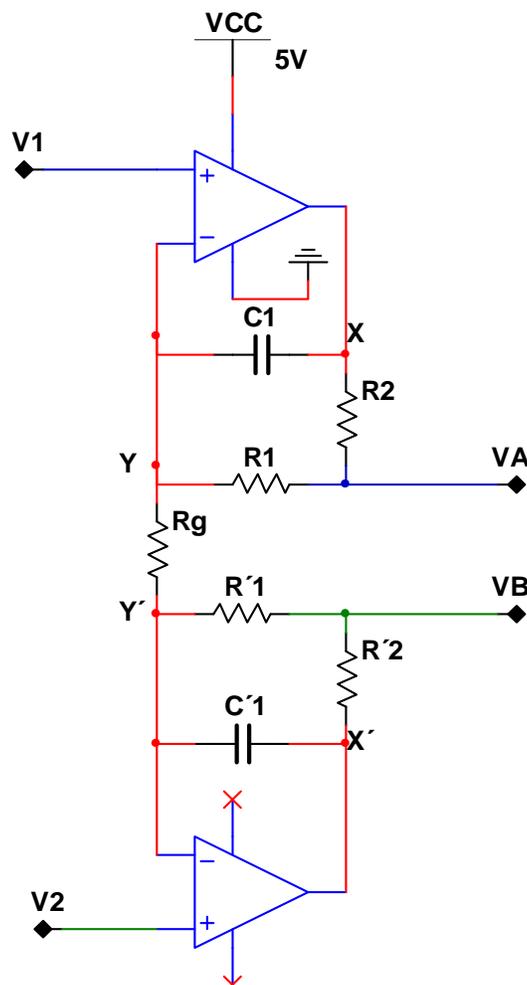


Figura 3.7. Amplificador recomendado en hoja de especificaciones.

La configuración mostrada en la figura 3.7 fue analizada, obteniendo de este modo su función de transferencia, esto con la finalidad de determinar el funcionamiento de dicha configuración que el fabricante del convertidor A/D recomienda.

En las ecuaciones 3.5 y 3.6 se muestran las salidas VA y VB en función de las entradas V1 y V2; donde se consideró que $R1 = R1'$, $R2 = R2'$, $C1 = C1'$.

$$V_A = \left(\frac{R_1 + R_2}{R_1 R_2} \right) \frac{\left(S + \frac{R_1 + R_g}{C_1 R_g (R_1 + R_2)} \right)}{\left(S^2 + S \left(\frac{R_1 + R_2}{R_1 R_2} \right) + \frac{1}{C_1 R_1 R_2} \right)} V_1 - \left(\frac{1}{C_1 R_2 R_g} \right) \frac{1}{S^2 + S \left(\frac{R_1 + R_2}{R_1 R_2} \right) + \frac{1}{C_1 R_1 R_2}} V_2 \quad (3.5)$$

$$V_B = - \left(\frac{1}{C_1 R_2 R_g} \right) \frac{1}{S^2 + S \left(\frac{R_1 + R_2}{R_1 R_2} \right) + \frac{1}{C_1 R_1 R_2}} V_1 + \left(\frac{R_1 + R_2}{R_1 R_2} \right) \frac{\left(S + \frac{R_1 + R_g}{C_1 R_g (R_1 + R_2)} \right)}{\left(S^2 + S \left(\frac{R_1 + R_2}{R_1 R_2} \right) + \frac{1}{C_1 R_1 R_2} \right)} V_2 \quad (3.6)$$

Como ya se ha mencionado, la entrada del convertidor A/D es diferencial, por lo que al realizar la diferencia entre 3.5 y 3.6 se obtiene la siguiente expresión.

$$\frac{V_0}{V_1 - V_2} = \frac{(R_1 + R_2)}{R_1 R_2} \left(\frac{S + \left(\frac{2R_1 + R_g}{C_1 R_g (R_1 + R_2)} \right)}{\left(S^2 + S \left(\frac{R_1 + R_2}{R_1 R_2} \right) + \frac{1}{C_1 R_1 R_2} \right)} \right) \quad (3.7)$$

La función de transferencia mostrada en la expresión 3.7 fue simulada en *Matlab*, con la finalidad de observar el comportamiento de la configuración del amplificador propuesta por el fabricante del convertidor A/D; sin embargo, para simularla se le asignaron valores a los elementos que intervienen en dicha función de transferencia, en donde los valores de R1, R1' y Rg, fueron los mismos que se calcularon para obtener una ganancia de 12 dB, el valor del capacitor C1 se fijo a 100nF, tal y como lo indica la hoja de especificaciones. De esta manera sólo queda como incógnita el valor de R2, en donde al variar el valor de esta resistencia, se observó que la frecuencia de corte de la función de transferencia mostrada en 3.7 se modificaba. En la figura 3.8 se muestra el diagrama de Bode obtenido tras la simulación de dicha función de transferencia con dos valores diferentes de R2.

De la figura 3.8 se puede observar que la frecuencia de corte del sistema corresponde a 22.8 Hz con un valor de R2 igual a 100 kΩ y 1.51 kHz para un valor de R2 igual a 100Ω, además, la ganancia del sistema sigue siendo la que se calculó para no saturar la salida del amplificador (12 dB). Debido a que la configuración del amplificador propuesta en la hoja de especificaciones del convertidor A/D no afecta la ganancia del sistema, se puede concluir que el beneficio que se obtiene con el arreglo que propone el fabricante del convertidor A/D para el amplificador de instrumentación, radica en que se puede controlar el ancho de banda del amplificador, a comparación de que si no se le agrega al amplificador la resistencia y el capacitor (R2 y C1) en su red de realimentación.

Diseño y Desarrollo del sistema "DIGI"

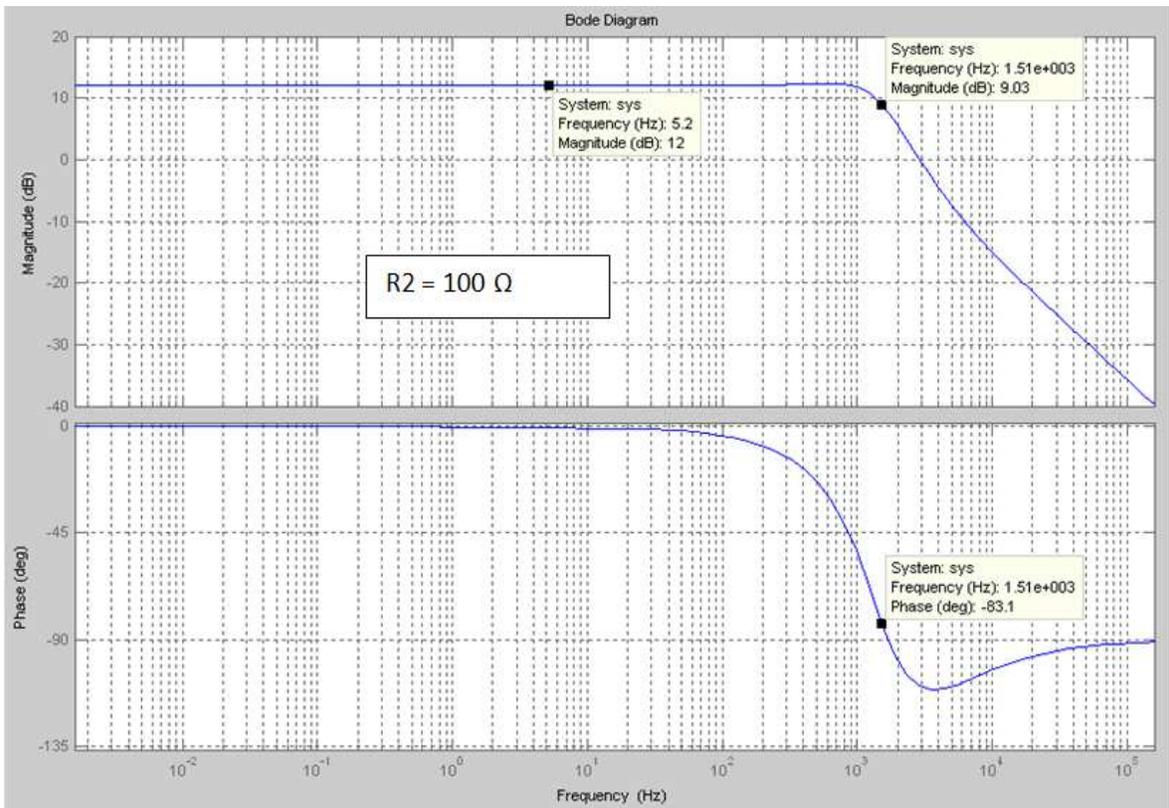
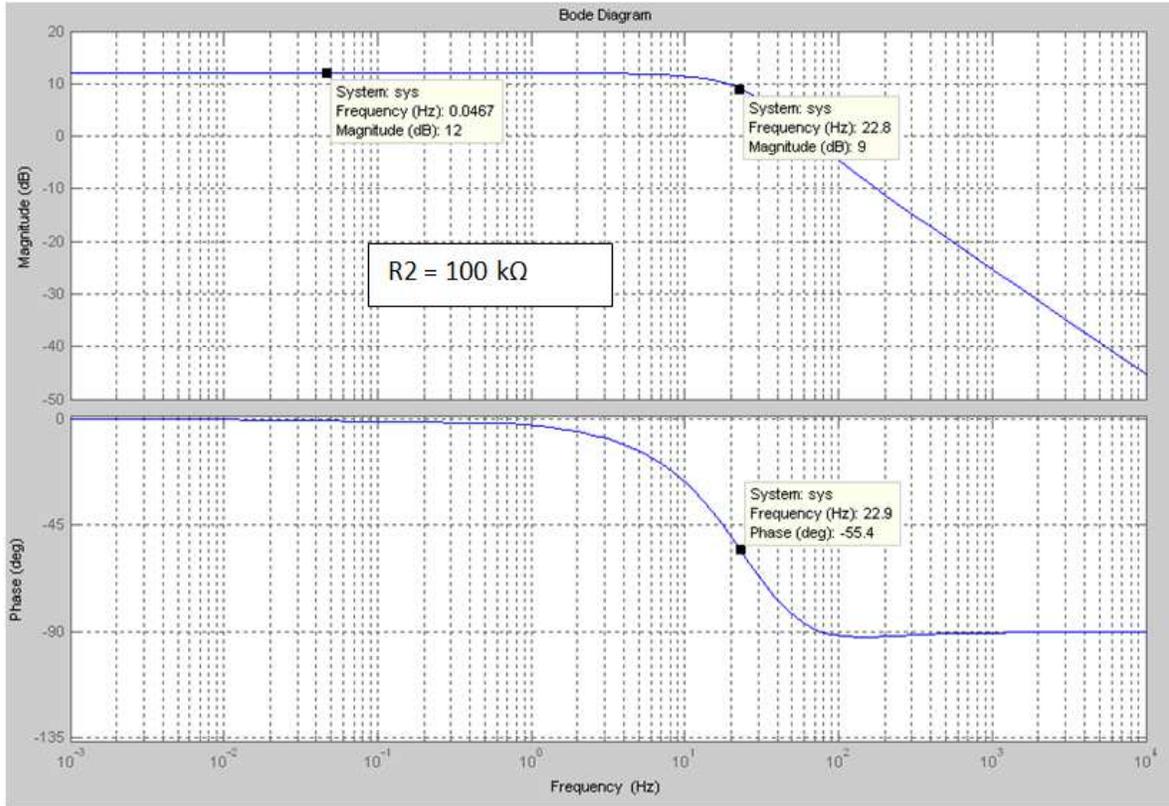


Figura 3.8. Diagrama de Bode de la función de transferencia al variar $R2$.

Para el caso del sistema DIGI lo anterior resultó de gran importancia en su diseño, ya que como se mencionó en la sección 2.1, el espectro sísmico abarca frecuencias que se encuentran desde 0.001Hz hasta los 1000 Hz. Por lo que para ajustar el ancho de banda del amplificador para altas frecuencias, se calculó el valor de la resistencia R2. De esta manera se obtuvo el valor de R2 que hace posible extender el ancho de banda del amplificador cubriendo el rango sísmico ya mencionado y con una misma ganancia de amplificación de la señal.

Finalmente para terminar el acondicionamiento de la señal para sensores pasivos, es necesario agregar un filtro anti *aliasing*, que limite la banda de la señal de entrada. La señal de los geófonos tiene una frecuencia natural de 4.5Hz, por lo que para poder digitalizar dicha señal es necesario diseñar un filtro pasa bajas con frecuencia de corte de 9 Hz. Esto último ya se ha justificado en la sección 2.3.2. La figura 3.9 muestra la configuración de elementos pasivos que forman un filtro paso bajas de primer orden.

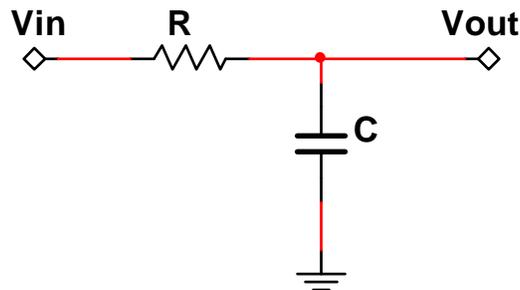


Figura 3.9. Filtro RC de primer orden.

Al analizar el circuito mostrado en la figura 3.9, se obtuvo su función de transferencia, la cual se muestra en la siguiente expresión.

$$\frac{V_{out}}{V_{in}} = \frac{1}{s + \frac{1}{RC}} ; w = \frac{1}{RC} ; f = \frac{1}{2\pi RC} \quad (3.8)$$

De la expresión 3.8 se sabe que la frecuencia a la que debe cortar el filtro tendrá que ser de 9Hz, quedando entonces dos incógnitas en la ecuación mencionada (el valor del capacitor C y el resistor R), por lo que para facilitar los cálculos se propuso un valor a la resistencia R igual a 8.2 kΩ, de esta manera se redujo el problema a sólo calcular el valor del capacitor, el cual dio un valor teórico de 1.96uF. El valor calculado no es un valor comercial de capacitancia, por lo que se aproximó el valor teórico con uno comercial, encontrando que el valor más próximo correspondía a 2.2uF.

Una vez que se ha explicado el diseño de cada uno de los elementos que integran el circuito de acondicionamiento para sensores pasivos, en la figura 3.10 se ilustra de manera general el circuito de acondicionamiento para este tipo de sensores.

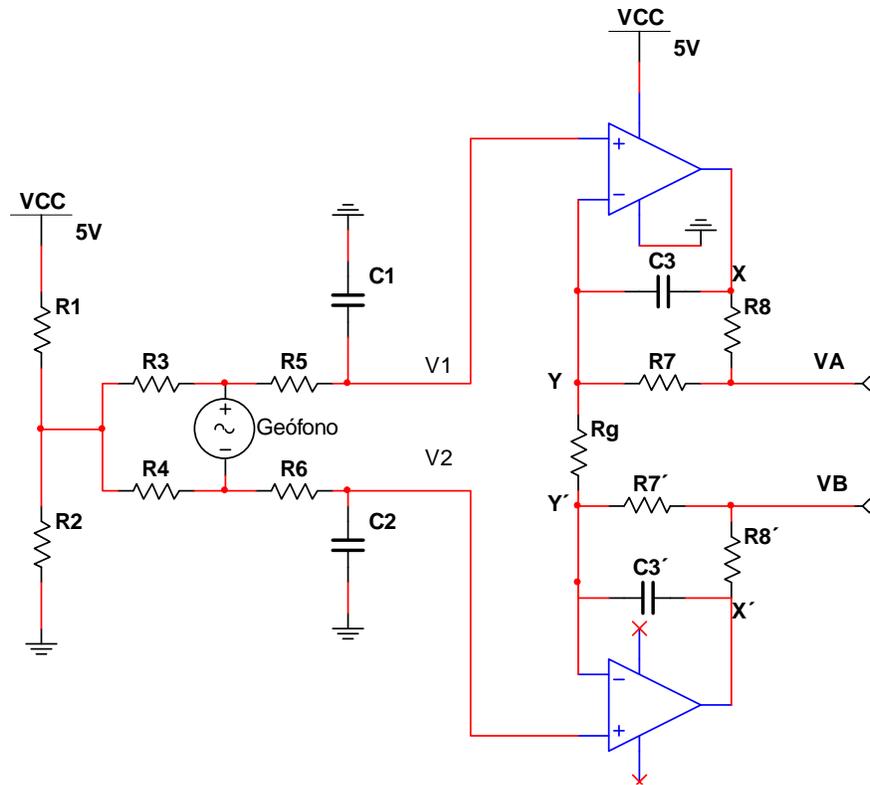


Figura 3.10. Circuito de acondicionamiento de señal para geófonos.

- **Acondicionamiento de señales para sensores activos**

Los sensores activos a diferencia de los pasivos, requieren una fuente externa de alimentación para que puedan funcionar, ya que internamente contienen circuitos electrónicos que deben estar energizados para su operación.

Debido a que en el laboratorio de instrumentación del Instituto de Ingeniería no se contaba con un sensor activo para su caracterización, se investigaron de forma general las características eléctricas de la señal de salida de este tipo de sensores. Encontrando que la señal de salida de estos sensores es del tipo bipolar puesta a tierra, su frecuencia de operación se encuentra en el rango de 0.1Hz – 10Hz, y la amplitud en la salida de estos oscila en un rango de 0.30V – 0.80V. Cabe mencionar que este tipo de sensores son de aceleración.

Con base en las características eléctricas ya mencionadas de este tipo de sensores, se tomó la decisión de que el amplificador a utilizar sería el mismo que fue diseñado para el caso de los sensores pasivos, debido a que la amplitud de la señal es similar a la

proporcionada por los geófonos, de esta manera se redujo el trabajo de diseño del circuito de acondicionamiento para sensores activos, ya que el problema radicaría en rediseñar la respuesta en frecuencia del filtro anti *aliasing*, debido a que para el caso de los sensores pasivos este último tiene una frecuencia de corte de 9 Hz, y para el caso de los sensores activos se requiere una frecuencia de corte de 20 Hz.

En un principio y para poder modificar la frecuencia de corte del filtro, se consideró recalcular el valor del capacitor C de la expresión 3.8, con la finalidad de obtener una frecuencia de corte de 20 Hz. Como consecuencia del cálculo mencionado, el valor del capacitor se encontraba en el orden de los nF, sin embargo, esta solución traía como consecuencia aumentar la densidad de componentes que conformarían al circuito de acondicionamiento, ya que se implementaría un filtro pasa bajas para los sensores pasivos y otro para los sensores activos.

Con base en la problemática que se presentó, se investigó en diferentes fuentes la manera de modificar la respuesta en frecuencia de un filtro paso bajas sin tener que cambiar el valor de alguno de los componentes. Como resultado de la investigación, en la figura 3.11 se muestra la configuración encontrada para modificar la frecuencia de corte de un filtro pasa bajas.

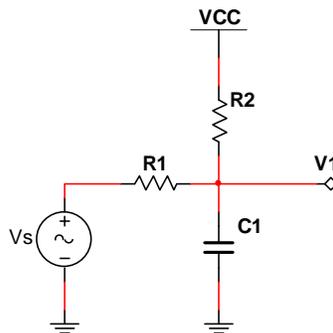


Figura 3.11. Circuito que modifica la respuesta en frecuencia de un FPB.

El circuito mostrado en la figura 3.11 fue analizado con la finalidad de conocer su comportamiento, y posteriormente saber que elementos debían de modificarse para cumplir con la condición deseada. En la ecuación 3.9 se muestra el resultado de dicho análisis.

$$V1 = \frac{1}{C1R1} \left(\frac{1}{S + \frac{R1+R2}{R1C1R2}} \right) Vs + \frac{R1}{R1 + R2} VCC \quad (3.9)$$

Como se puede observar, de la ecuación mostrada en la expresión 3.9, la salida del circuito está formada por una componente de corriente alterna (CA) y una componente de

Diseño y Desarrollo del sistema "DIGI"

corriente directa (DC). En donde de la componente de corriente alterna se puede obtener la frecuencia de corte del filtro mediante la expresión 3.10.

$$\omega = \frac{R1 + R2}{C1 R1 R2} \quad ; \quad \omega = 2\pi f \quad (3.10)$$

Con base en la expresión 3.10, se consideró que R1 y C1 tuvieran los mismos valores que fueron utilizados para el filtro anti *aliasing* de los sensores pasivos, por lo que la única incógnita correspondía al valor de R2. En la figura 3.12a se muestra una simulación del circuito mostrado en la figura 3.11 con R2 = 10Ω y en la figura 3.12b con R2 = 100kΩ.



a)



b)

Figura 3.12. Respuesta del filtro paso bajas.

Con base en la figura 3.12, se corroboró que la configuración mostrada en la figura 3.11 permite modificar la respuesta del filtro pasa bajas, por lo que para obtener una frecuencia de corte de 20Hz, de la ecuación 3.10, se despeja a R2 quedando la siguiente expresión

$$R2 = \frac{R1}{(\omega R1 C1) - 1} \quad (3.11)$$

Por lo que a partir de la expresión 3.11 se obtuvo el valor de la resistencia R2 para tener la frecuencia de corte deseada. Por otra parte, y como ya se comentó, la expresión 3.9 está formada por una componente de corriente directa, la cual y como ya se ha justificado en el diseño del circuito de acondicionamiento para sensores pasivos, es necesaria para mover la referencia de la señal para ser amplificada dentro del rango de los niveles de alimentación del amplificador. Con el valor de R2 calculado se obtiene un nivel de *offset* igual a 2.5V, el cual coincide con el nivel requerido para poder amplificar la señal.

En la figura 3.13 se muestra de manera general el circuito diseñado para el acondicionamiento de señal de salida de los sensores activos.

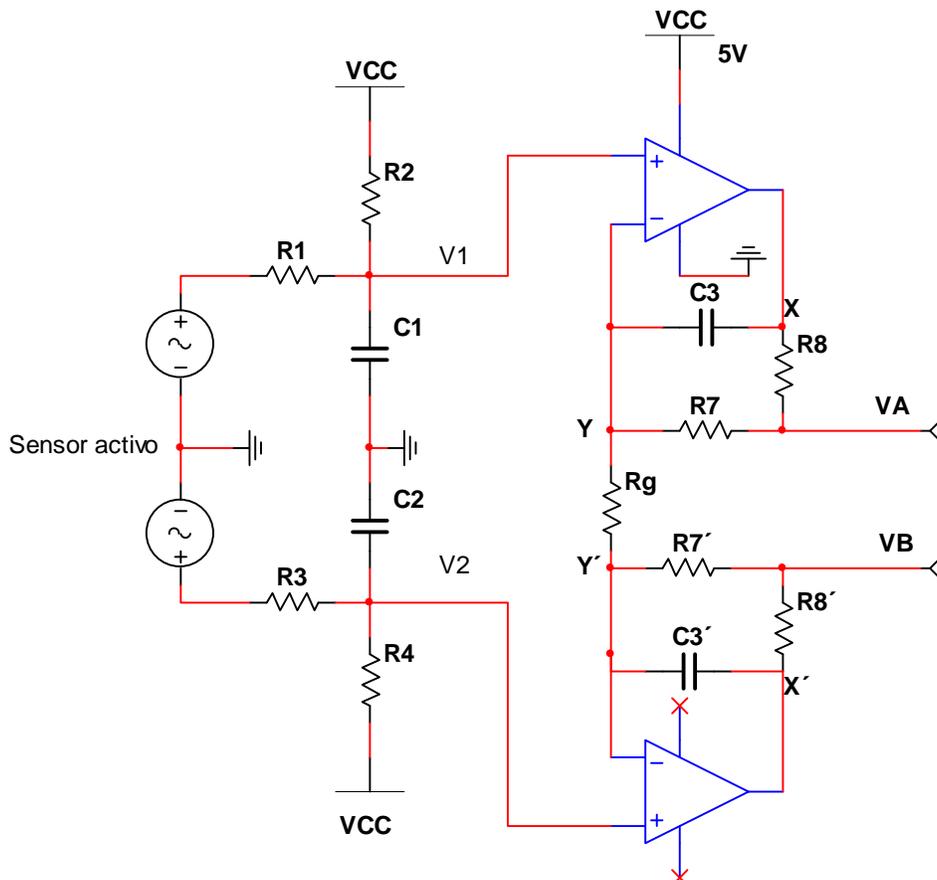


Figura 3.13. Circuito de acondicionamiento de señal para sensores activos.

Una vez que fueron diseñados los circuitos de acondicionamiento para los sensores pasivos y activos, se simularon ambos circuitos con la finalidad de verificar su comportamiento. Con la intención de proteger las entradas contra variaciones de voltaje, al circuito de acondicionamiento se le agregaron limitadores de nivel, los cuales fueron implementados con un arreglo de diodos. Con esta medida preventiva al diseño, se garantizaría que aunque el usuario no introdujera un nivel de voltaje adecuado, los diodos contrarrestarían esta entrada y sólo dejarían pasar un nivel de voltaje entre 0V y 5V, teóricamente. En la figura 3.14 se muestra la configuración de los diodos limitadores para la protección de la entrada del amplificador.

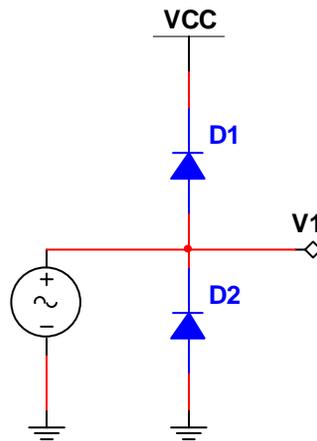


Figura 3.14. Diodos limitadores de nivel.

De la figura 3.14 se observa que el diodo D1 conducirá siempre y cuando el voltaje de entrada sea mayor en 0.7 que el voltaje de polarización. Para el caso del diodo D2, conducirá siempre y cuando el voltaje de entrada sea menor en 0.7 que la referencia, para este caso "tierra". Las condiciones mencionadas se resumen en la tabla 3.3.

	D1	D2
Conduce	$V_s > V_{d1} + V_{cc}$	$V_s < V_{d2}$
No conduce	$V_s < V_{d1} + V_{cc}$	$V_s > V_{d2}$

Tabla 3.3. Condiciones para activar D1 y D2.

Con base en lo descrito anteriormente, referente al diseño del circuito de acondicionamiento de señal para sensores pasivos y activos, se observa que del diseño realizado para ambos sensores se utiliza el mismo amplificador, y lo único que hace diferentes a estos circuitos es la respuesta en frecuencia del filtro anti *aliasing*. Dicho lo anterior y con la finalidad de reducir ambos circuitos a uno solo, este último se implementó con unos selectores (*jumpers*) la manera de elegir el filtro en función del tipo de sensor a conectar. Es decir, el usuario simplemente tendrá que abrir o cerrar un par de *jumpers*

dependiendo el tipo de sensor que utilizará. En la figura 3.15 se muestra la forma en que se han conectado los filtros con los selectores.

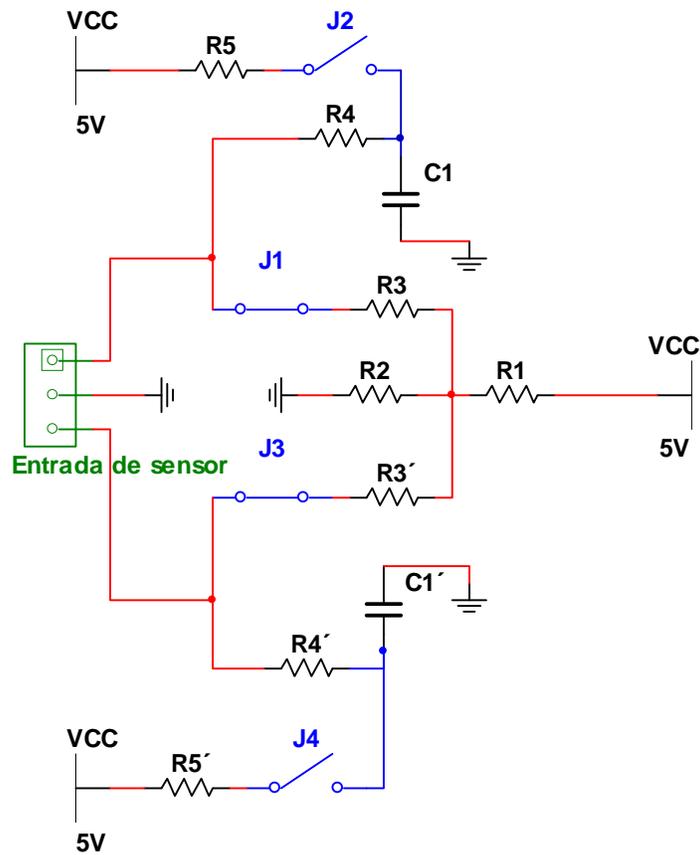


Figura 3.15. Conexión de filtros mediante selectores.

En la tabla 3.4 se muestra la forma de conmutación de los *jumpers* para la elección del filtro anti *aliasing* dependiendo el tipo de sensor a conectar.

	Geófonos	Sensores activos
J1	Cerrar	Abrir
J2	Abrir	Cerrar
J3	Cerrar	Abrir
J4	Abrir	Cerrar

Tabla 3.4. Combinación de selectores dependiendo el tipo de sensor.

Es importante mencionar que si se realiza otra configuración diferente a la mostrada en la tabla 3.4, los datos no se digitalizarán adecuadamente, ya que el circuito está diseñado única y exclusivamente para conectar sensores pasivos o activos. En la figura 3.16 se ilustra de manera general el circuito de acondicionamiento de señal para sensores pasivos y activos.

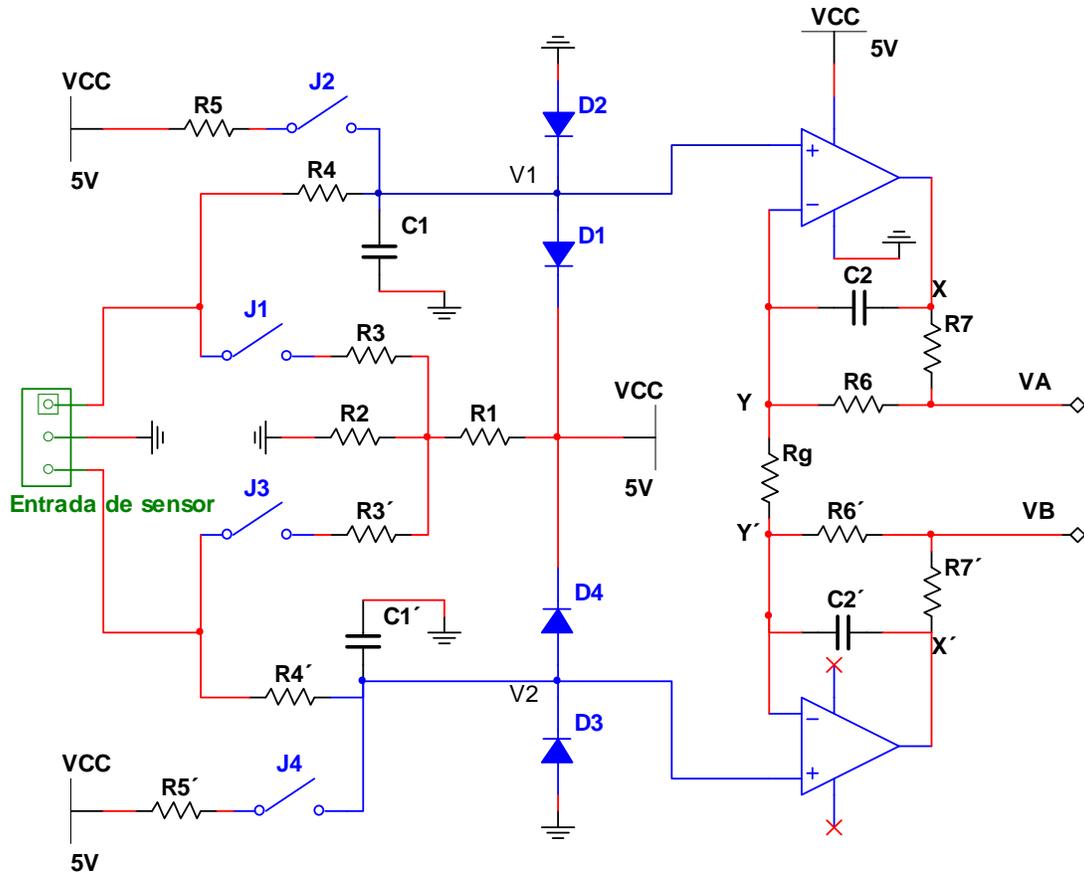


Figura 3.16. Circuito de acondicionamiento de señal.

El circuito mostrado en la figura 3.16, corresponde al circuito de acondicionamiento de señal para un solo canal, por lo que en el caso del sistema DIGI se necesita implementar físicamente tres veces este circuito de acondicionamiento, para tener los tres canales analógicos con los que debe contar el sistema.

3.2.4. Módulo para la digitalización de los datos

Una vez que la señal de interés ha pasado por el módulo de acondicionamiento de señal, se encuentra lista para ser digitalizada. El convertidor analógico digital empleado es de la marca “Linear Technology”, la arquitectura de este dispositivo es $\Sigma\Delta$, por lo que su principio de operación se basa en sobre muestrear la señal analógica, modular la señal por medio de técnicas sigma – delta, y posteriormente pasar por un proceso de diezmado que consta en bajar la frecuencia de muestreo a la frecuencia de Nyquist. Este último proceso se realiza por medio de bloques puramente digitales, la ventaja de este tipo de arquitecturas es que se reduce la complejidad de la circuitería analógica para el filtro pasa bajas, que limita a la señal de entrada.

Diseño y Desarrollo del sistema “DIGI”

En el diseño del sistema “DIGI” se ha conectado el microcontrolador al convertidor analógico – digital a través de la interfaz SPI, de esta manera es posible agregar varios convertidores a esta interfaz. Para el diseño del DIGI, en un principio se pensó en implementar con un solo convertidor A/D multiplexado el módulo para la digitalización de los datos. Sin embargo, al investigar acerca de las características de digitalización de los sistemas diseñados para el registro de señales sísmicas, se encontró que el estándar de diseño de estos sistemas requieren de un convertidor A/D por canal, esto se debe a que el convertidor analógico digital multiplexado se lleva un tiempo para cambiar de un canal a otro, y aunque el multiplexado sea lo suficientemente rápido, para este tipo de aplicaciones no se puede perder tiempo al digitalizar la señal. Por lo que en nuestro caso se emplearon tres convertidores A/D (uno para cada canal), ya que la idea es digitalizar las señales provenientes de los sensores sísmicos en los tres ejes (X, Y y Z) al mismo tiempo.

En la figura 3.17, se ilustra la distribución de terminales del convertidor analógico digital utilizado, y en la tabla 3.5 se muestra la descripción de las terminales.

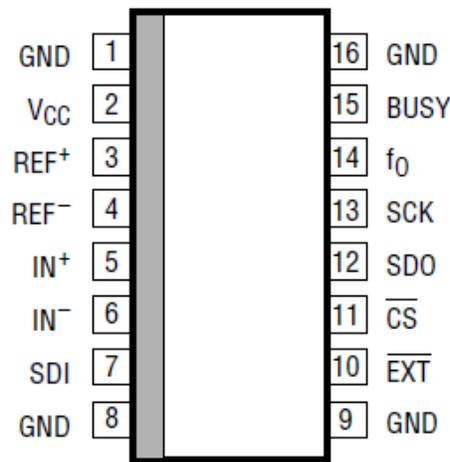


Figura 3.17. Distribución de terminales del convertidor A/D.

Terminal	Nombre	Función
1	GND	Tierra
2	VCC	Fuente de alimentación (5 volts)
3	Ref +	Referencia de voltaje positiva (0.1 – 5 volts)
4	Ref -	Referencia de voltaje negativa (0 – 4.9 volts)
5	In +	Tensión de entrada positiva (0.5 Vref)
6	In -	Tensión de entrada negativa (-0.5 Vref)

Tabla 3.5. Descripción de terminales del convertidor A/D. (Continúa)

Diseño y Desarrollo del sistema "DIGI"

Terminal	Nombre	Función
7	SDI	Terminal utilizada para seleccionar la velocidad y resolución del CAD (MOSI)
8	GND	Tierra
9	GND	Tierra
10	EXT	Terminal utilizada para seleccionar entre el oscilador interno o externo del CAD
11	CS	Habilitador del dispositivo
12	SDO	Terminal de salida de los datos digitalizados (MISO)
13	SCK	Reloj para controlar salida de datos de SPI
14	Fo	Terminal para ingresar señal de reloj externo
15	Busy	Indicador cuando la conversión esta en progreso
16	GND	Tierra

Tabla 3.5. Descripción de terminales del convertidor A/D.

Los convertidores tienen como dispositivo maestro a un microcontrolador, siendo ellos los esclavos. En la figura 3.18 se muestra la conexión entre el microcontrolador y los convertidores analógicos digitales.

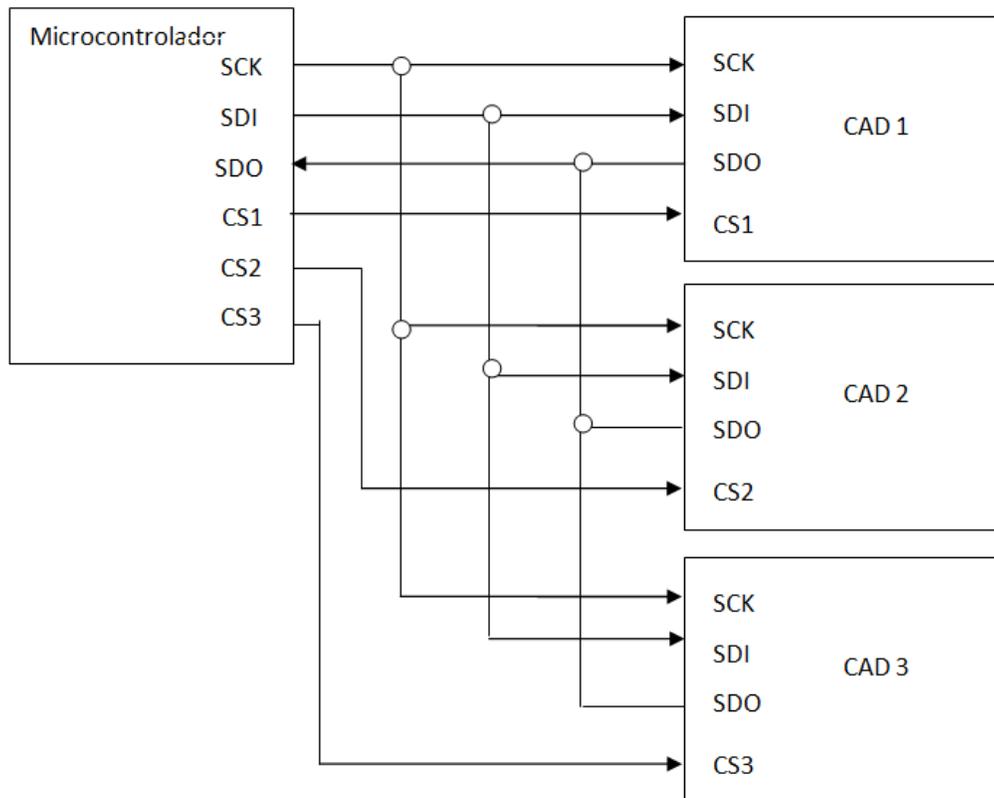


Figura 3.18. Diagrama de conexión entre el microcontrolador y los CAD.

3.2.5. Módulo para el almacenamiento de datos

El registro de los datos de los sensores sísmicos se hará en una memoria SD, la cual será controlada desde el microcontrolador. Para efectuar la comunicación entre éstos, es necesario conocer las características principales de este tipo de memorias, entre ellas podemos mencionar el voltaje de alimentación, distribución de terminales y protocolos de comunicación; estas características fueron presentadas a detalle en los apartados 2.4 y 2.5.

En la figura 3.19 se muestra la distribución de terminales, la forma y las dimensiones de la memoria micro SD. Además, en la tabla 3.6 se describe la función que tiene asignada cada terminal.

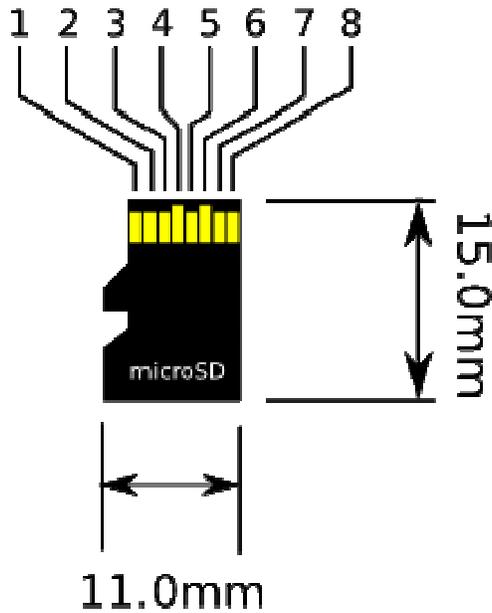


Figura 3.19. Distribución de las terminales de la memoria micro SD.

Terminal	Nombre	Función
1	X	Reservado
2	CS	Señal de control (chip select)
3	DI	Comandos y datos hacia la tarjeta (Data In)
4	VDD	Fuente de alimentación (3.3 volts)
5	CLK	Señal de reloj
6	VSS	Tierra
7	DO	Datos y estado desde la tarjeta (Data Out)
8	X	Reservado

Tabla 3.6. Descripción de las terminales de la memoria micro SD.

Como se mencionó anteriormente, las memorias micro SD operan en un rango de voltaje de 2.6 a 3.7 volts, mientras que el microcontrolador opera a 5 volts. Por ello, para asegurar la compatibilidad de voltajes se empleó un cambiador de nivel, el cual al alimentarlo con un voltaje de 3.3 volts, nos permite obtener señales de 3.3 volts, a partir de las señales TTL que proporciona el microcontrolador, y así, poder conectarlas a la memoria micro SD sin problema alguno.

El microcontrolador tiene implementado un módulo SPI por hardware, con terminales específicas para esta comunicación. En la figura 3.20 se muestra el diagrama de conexión entre la memoria micro SD y el microcontrolador, ayudándonos del cambiador de nivel mencionado anteriormente.

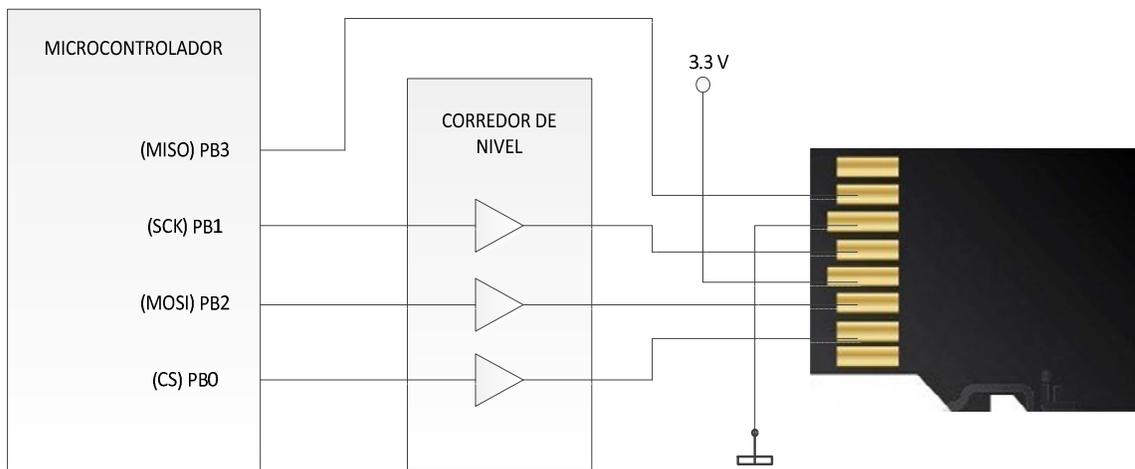


Figura 3.20. Conexión entre el microcontrolador y la memoria micro SD.

3.2.6. Interfaz hombre – máquina

La interfaz ente el equipo y el usuario está formada por un *display* gráfico y un *joystick* (conjunto de *push bottons*), los cuales permiten configurar el equipo y visualizar los datos obtenidos de los sensores.

Display gráfico

El *display* gráfico consta de 128x64 puntos y está basado en un controlador, el cual nos permite acceder a cada punto, a través del control de instrucciones y datos enviados por el microcontrolador. En la figura 3.21 se muestra el diagrama de bloques del *display* en cuestión y las señales de control y datos que se requieren.

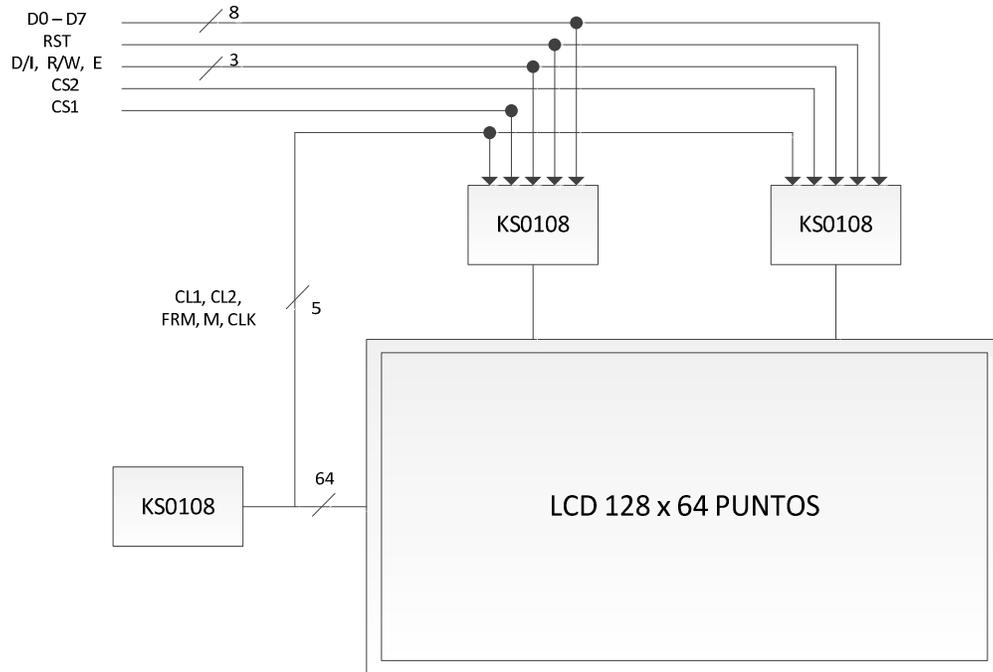


Figura 3.21. Diagrama de bloques del display gráfico.

El *display* está dividido en dos secciones, cada una de ellas administrada por dos controladores. Cada sección está integrada por un bloque de 64x64 puntos, que en conjunto forman la matriz de 128x64 puntos.

En la figura 3.22 se muestra la apariencia física del *display*, y en la tabla 3.7 se muestra la distribución de las terminales, su descripción y la terminal del microcontrolador a la cual se conecta.

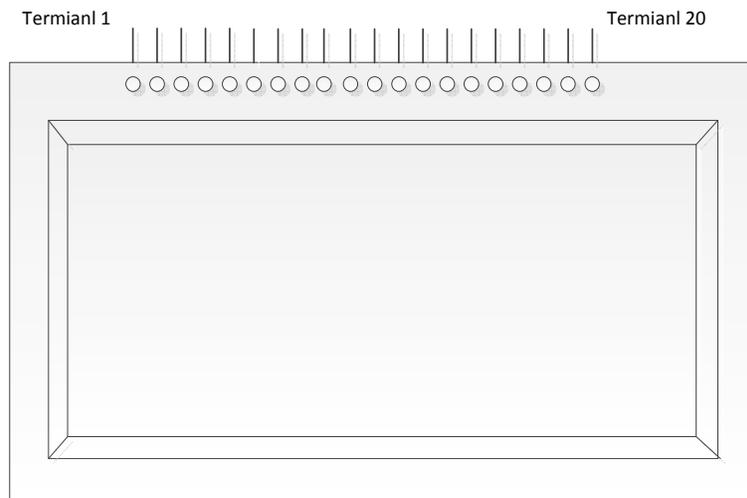


Figura 3.22. Apariencia física del display gráfico.

<i>Display JHD-63512864E</i>			ATMEGA2560
Terminal	Nombre	Descripción	Terminal
1	CSA	Habilitación parte izquierda del <i>display</i>	PC3 (56)
2	CSB	Habilitación parte derecha del <i>display</i>	PC4 (57)
3	VSS	Tierra	-----
4	VDD	Alimentación (5 volts)	-----
5	Vo	Ajuste de contraste	-----
6	D/I	Control para comandos/datos	PC1 (54)
7	R/W	Lectura o escritura de comandos/datos	PC2 (55)
8	E	Señal de habilitación del <i>display</i>	PC0 (53)
9-16	A0 - A7	Líneas de datos	Puerto A
17	RST	Señal de reset del <i>display</i>	-----
18	VEE	Salida interna de 10 volts	-----
19	LED (+)	Ánodo del led interno del <i>display</i>	PC5 (58)
20	LED (-)	Cátodo del led interno del <i>display</i>	-----

Tabla 3.7. Descripción de las terminales del display.

Como podemos observar en la tabla 3.7, el *display* se conecta al microcontrolador a través de los puertos C y A, el primero se usa para la parte de control, mientras que el segundo es usado para la transmisión de datos.

La polarización del *display* se realiza a través de las terminales VDD y VSS. El *display* cuenta con un circuito interno que genera un voltaje VEE de 10 volts en la terminal 18. Este voltaje es usado para ajustar el contraste del *display*, con ayuda de un potenciómetro, como se muestra en la figura 3.23.

La terminal de control D/I permite indicarle al *display* si se le está enviando datos o instrucciones; si la terminal tiene un estado lógico bajo, indica envió de instrucciones, por el contrario, si la terminal tiene un estado lógico alto, indica que se le envía datos.

La terminal de control R/W indica al *display* si se presenta una operación de lectura o escritura de datos. Si en esta terminal se presenta un estado lógico bajo, se trata de una operación de escritura, por el contrario, si en esta terminal tienen un estado lógico alto, se tiene una operación de lectura.

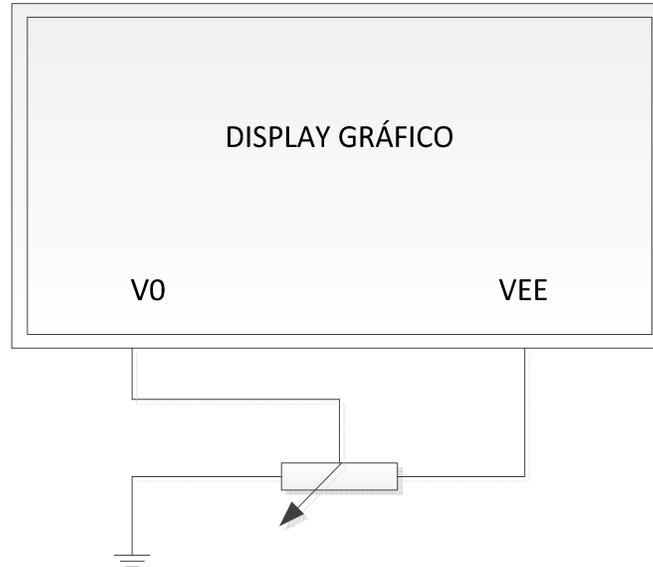


Figura 3.23. Circuito para ajustar el contraste del display.

Joystick

El joystick mini es un interruptor de cinco vías, que se encuentra dentro de un encapsulado pequeño. Este dispositivo nos permite desplazarnos a través de los diferentes menús, que se han programado en el microcontrolador y que son mostrados a través del *display* gráfico, con la finalidad de configurar al equipo y puesta en marcha de la adquisición de datos. En la figura 3.24 se muestra las terminales del microcontrolador que están conectadas a las terminales del joystick.

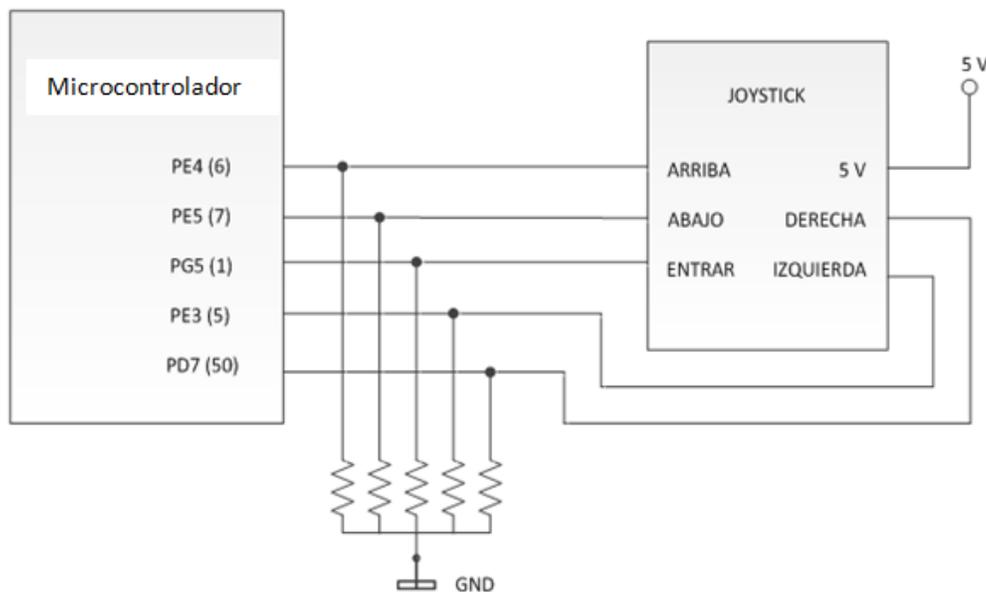


Figura 3.24. Diagrama de conexión entre el joystick y el microcontrolador.

Para esta conexión se cuenta con resistencias de *pull-down*, con la finalidad de detectar un estado lógico alto cuando alguna de las cinco vías sea oprimida, y así, efectuar la tarea programada de acuerdo a la siguiente correspondencia:

- **Tecla arriba.** Permite desplazar el cursor hacia arriba para seleccionar una opción
- **Tecla abajo.** Permite desplazar el cursor hacia abajo para seleccionar una opción
- **Tecla derecha.** Reservada para futuras implementaciones
- **Tecla izquierda.** Regresa al menú principal
- **Tecla centro.** Permite ejecutar la opción seleccionada

3.2.7. Módulo para la adquisición de datos del GPS

Una de las principales funciones que debe tener un sistema diseñado para la adquisición de señales sísmicas, es la de contar con una referencia de tiempo, lo más exacta posible. Esto con la finalidad de contar con la fecha y hora en que ocurrió el fenómeno de interés. En el diseño del sistema “DIGI” se integró un GPS, el cual a través del protocolo NMEA 0183, se obtienen los datos de fecha y hora en formato UTC; cabe mencionar que estos datos no son convertidos a tiempo local, ya que los sistemas de digitalización de señales sísmicas se basan en este formato para tener la misma referencia en todo el mundo.

Existe una gran variedad de receptores GPS en el mercado, sin embargo, se utilizó un modelo de bajo costo, las principales características del receptor utilizado se presentan en la tabla 3.8.

Características del receptor GPS	
Terminal	Nombre
Exactitud	Posición: 10 metros. Velocidad: 0.1 m/s. Tiempo: 1us sincronizado con el tiempo GPS.
Tiempo de adquisición	Readquisición de datos: 0.1 segundos en promedio. Arranque en frío: 42 segundos en promedio. Arranque en tibio: 38 segundos en promedio. Arranque en caliente: 1 segundo en promedio.
Condiciones dinámicas	Altitud: máximo 18 km. Velocidad: máximo 515 m/s (1000 nudos náuticos). Aceleración: menor a 4g.
Alimentación	Voltaje de alimentación: 4.5V – 6.5V. Corriente demanda: 70mA.

Tabla 3.8. Características del receptor GPS. (Continúa)

Características del receptor GPS EM-406	
Terminal	Nombre
protocolo	Niveles eléctricos: Nivel TTL. Baudaje: 4800 bps. Mensajes de salida: NMEA 0183, GGA, GSA, GSV, RMC, VTG, GLL.
General	Sensibilidad: -159 dBm. Chipset: SiRF Star II. Canales: 20 canales.

Tabla 3.8. Características de receptor GPS.

Es importante mencionar que para este tipo de receptores basta con suministrar el voltaje de alimentación para poder obtener las cadenas de datos del protocolo NMEA 0183; sin embargo, para poder visualizar los datos provenientes del GPS, como son: la fecha, la hora, la altitud, la latitud y la velocidad, es necesario conectar el microcontrolador con el receptor GPS. Esta conexión se realiza a través de la UART, en donde la velocidad de comunicación del puerto se tiene que fijar a 4800 bps, ya que si no se configura a esta velocidad se estarán registrando datos totalmente ajenos a los establecidos por el protocolo NMEA 0183.

En la figura 3.25 se ilustra la conexión entre el receptor GPS y el microcontrolador empleado para el desarrollo del sistema “DIGI”.

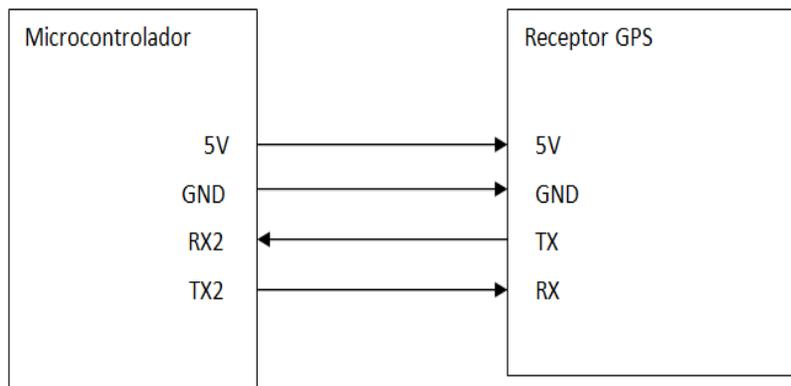


Figura 3.25. Conexión entre el GPS y el microcontrolador.

Por otra parte, este receptor posee un LED (*Light Emitting Diode*) que se activa de manera intermitente cuando se ha adquirido correctamente la posición, es en este estado cuando es seguro realizar el registro de los datos de fecha y hora, ya que si el LED permanece apagado, indicará que no se tiene una buena recepción de la señal proveniente de los satélites que forman el sistema GPS y por consecuencia los datos serán erróneos.

En la figura 3.26 se ilustra el aspecto del receptor utilizado, además de que en el círculo se indica el LED mencionado anteriormente.

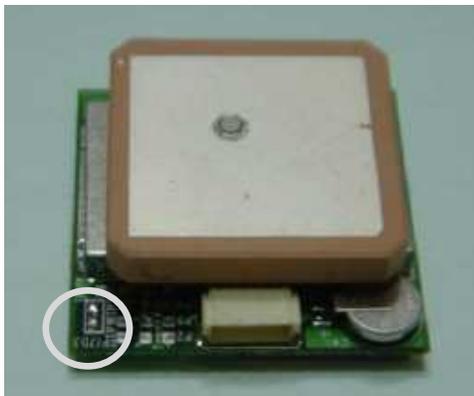


Figura 3.26. Receptor GPS.

3.2.8. Integración del hardware

Después de haber analizado y comprobado el funcionamiento básico de cada uno de los bloques y elementos que conforman al sistema, se procedió a montarlas en tarjetas *protoboard*, con la finalidad de corroborar el funcionamiento de todos los bloques en conjunto, además de ello, esto nos permitió realizar pruebas, cambios, o en su defecto integrar componentes adicionales conforme se avanzaba en el desarrollo del prototipo. Para mejor ejemplificación, en la figura 3.27 se muestra el prototipo del sistema.

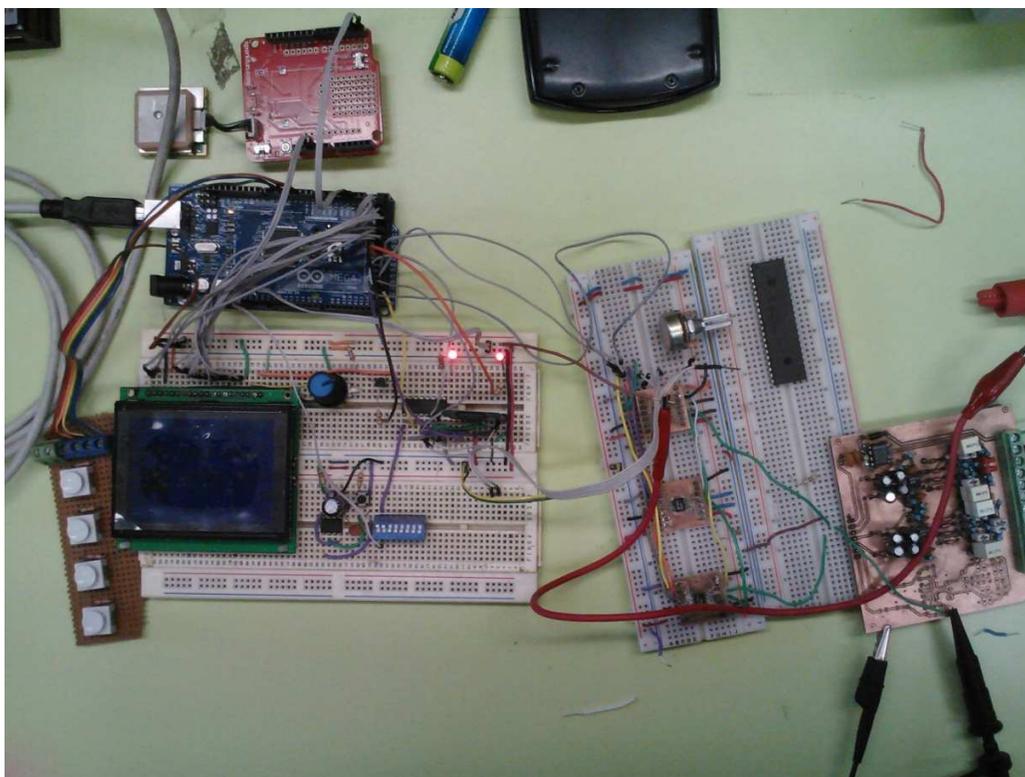


Figura 3.27. Hardware del sistema montado en tarjetas protoboard.

Diseño y Desarrollo del sistema "DIGI"

En las figuras 3.28 (a) y 3.28 (b) se muestran los diagramas eléctricos del sistema, en donde se observa la conexión de los distintos elementos mencionados anteriormente.

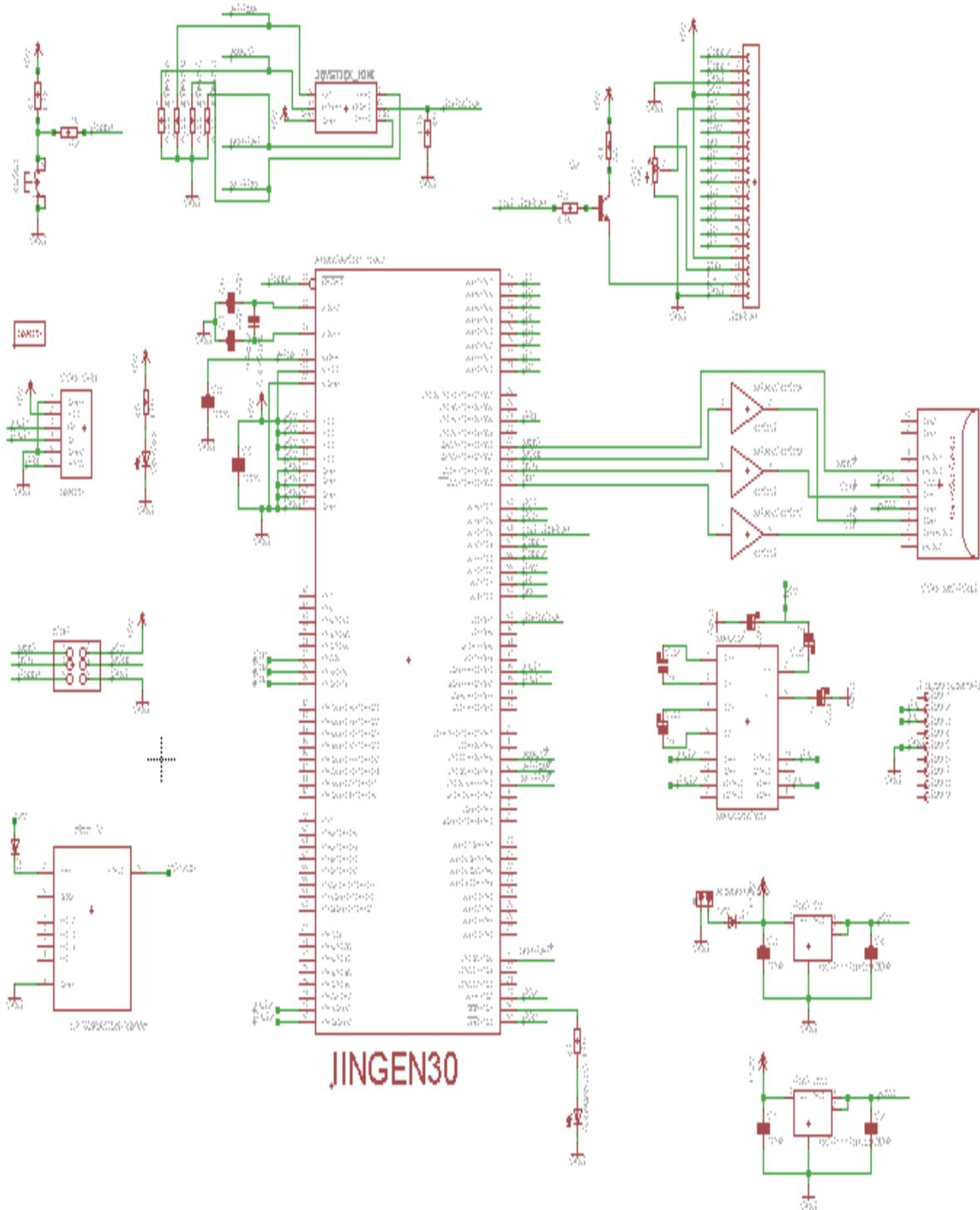


Figura 3.28 (a). Diagrama eléctrico del sistema "DIGI".

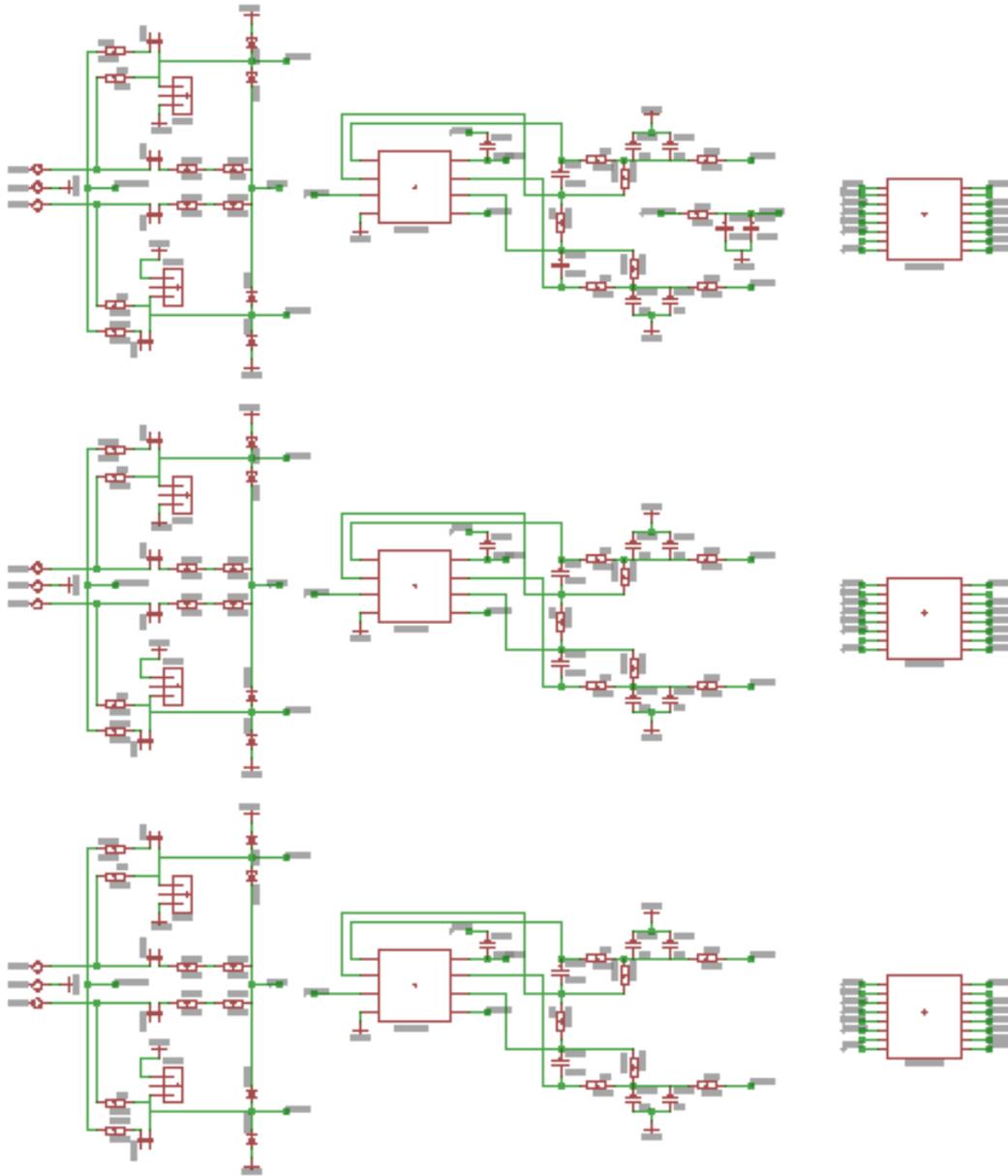


Figura 3.28 (b). Diagrama eléctrico del sistema "DIGI".

3.2.9. PCB del sistema

Al llegar a la etapa final del desarrollo del sistema, verificando que su funcionamiento es óptimo, y que no se requiere de ningún otro componente de hardware, se procedió a la integración en una tarjeta de circuito impreso (PCB: *Printed Circuit Board*), esto como primer prototipo funcional.

Algunas de las consideraciones que se tomaron para el diseño del circuito impreso son:

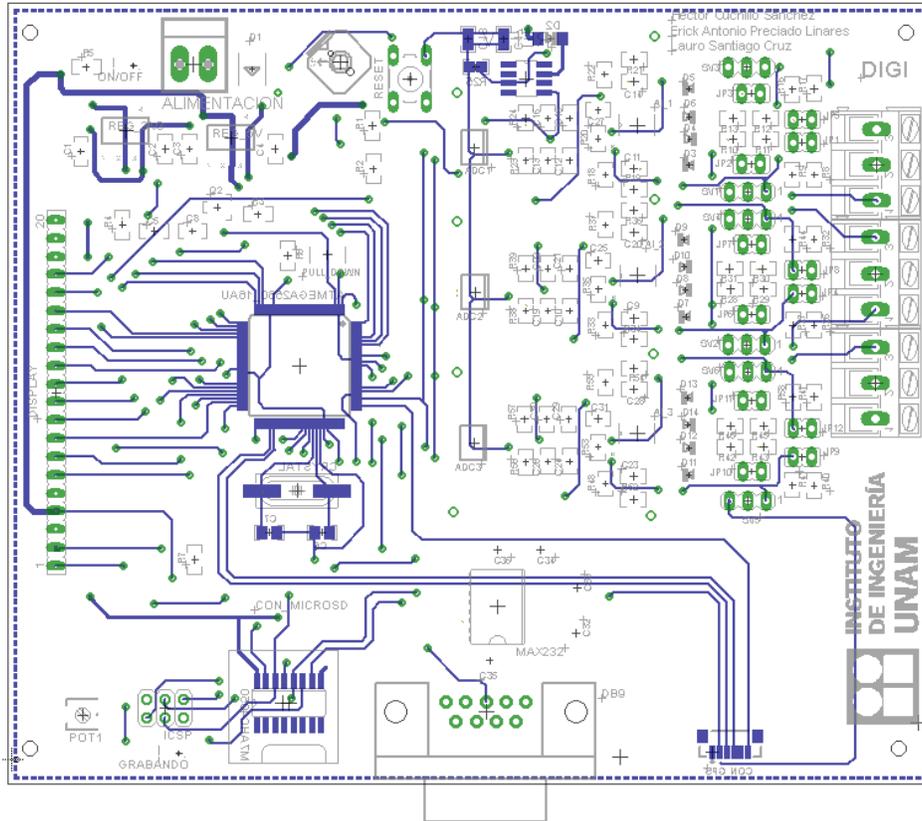


Figura 3.30. Capa inferior del circuito impreso.

Podemos observar que en el diseño del circuito impreso se añadió un conector DB9, por si en una implementación futura se desea establecer una comunicación serie RS-232. También se cuenta con las terminales para la programación del microcontrolador, a través de la interfaz ISP “In System Programming”.

3.3. Desarrollo del Software

Para el desarrollo del programa en el microcontrolador, Atmel pone a disposición de sus usuarios, una plataforma de desarrollo para sus dispositivos; esta herramienta es ATmel Studio, la cual permite programar, depurar, simular y grabar el código fuente, además, de que permite el uso de lenguajes de programación tales como, ensamblador, C y CC++. Sin embargo, para el desarrollo del programa, se optó por la plataforma de desarrollo de Arduino, debido a que tiene implementada una librería que permite el almacenamiento de información en memorias flash, específicamente en las memorias SD estándar y micro SD. Las funciones implementadas en esta librería facilitan enormemente las tareas necesarias para la creación de archivos y directorios, y el almacenamiento de información, cumpliendo con las especificaciones de los sistemas de archivos FAT16 y FAT32. Asimismo, existen otras librerías para el manejo de la UART y SPI útiles en el

sistema DIGI. A continuación se realiza una breve descripción de la interfaz de desarrollo de Arduino (IDE: *Integrated Development Environment*).

Descripción del entorno de desarrollo de Arduino

La IDE de Arduino está diseñada de tal forma que su uso sea lo más simple para el programador, ya que ésta cuenta con un editor de texto, en donde el usuario escribirá el código que desee ejecutar con el Arduino; una barra de herramientas con botones para funciones comunes (abrir o guardar un código, compilar o cargar un código a la tarjeta Arduino); una consola de texto en donde se muestra información correspondiente a la compilación del código, así como también, cuenta con una barra de menú en donde el usuario puede configurar algunas opciones del editor de texto para la elaboración del código. Con base en lo anterior en la figura 3.31 se muestra el entorno de desarrollo de Arduino.



Figura 3.31. Entorno de desarrollo de Arduino.

Como ya se mencionó, los botones que aparecen en la barra de herramientas permiten realizar funciones tales como: crear, compilar y cargar un programa, de manera general se describirán estos botones.



Verificar: Comprueba que el código esté libre de errores.



Cargar: Compila el código y lo carga a la placa Arduino.



Nuevo: Crea un nuevo *sketch*.



Abrir: Presenta un menú con todos los programas previamente diseñados, por lo que al elegir alguno de los programas diseñados, se abrirá en el editor de texto.



Guardar: Guarda el programa que se esté editando.



Monitor serial: Abre el monitor serial con la finalidad de tener comunicación con el Arduino y ver los datos transferidos por el puerto UART.

El código implementado en el entorno de desarrollo se denomina “*sketch*” y se guarda bajo la extensión `.ino`. Para poder trabajar con el programa, éste debe ser compilado, proceso mediante el cual se genera un archivo `.hex`. Este último es el que se usa para programar al microcontrolador.

Una vez que se ha creado el código y que ha resultado libre de error, tras ser compilado, es necesario cargarlo a la memoria *flash* de la tarjeta de desarrollo de Arduino, por lo que para esto es necesario conectar a la tarjeta con la PC mediante un cable USB A-B. Una vez que se ha conectado físicamente la tarjeta Arduino con la PC, es necesario verificar en el entorno de desarrollo el puerto en que se ha conectado la tarjeta, así como también se debe verificar que el hardware seleccionado corresponda a la placa que se ha conectado físicamente. Lo anterior es fácil de verificar, ya que en la esquina inferior derecha del entorno de desarrollo de Arduino se muestran dichas características tal y como se indica en la figura 3.32.

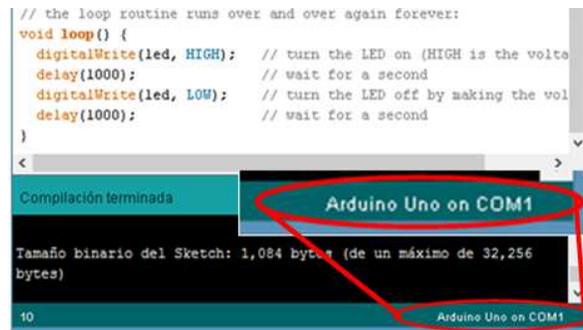


Figura 3.32. Verificación de puerto y placa en la IDE de Arduino.

3.3.1. Estructura del programa del microcontrolador

En las secciones anteriores se describió de manera general el diseño del hardware que se integra en el sistema DIGI, sin embargo, para que éste pueda funcionar adecuadamente se desarrolló un programa en la IDE de Arduino, para poder ser cargado al

microcontrolador del sistema, el cual gestionará dichas tareas. Es importante mencionar que la estructura del programa que se desarrolló está en función de las necesidades del usuario. A continuación se enlista de manera general las tareas a programarse en el microcontrolador.

- Adquisición y conversión de datos
 - Obtención de los 24 *bits* de interés dentro de las muestras proporcionadas por los convertidores analógicos digitales

- Adquisición de fecha y hora del GPS
 - Obtención de la hora UTC
 - Obtención de la fecha

- Almacenamiento en memoria micro SD
 - Creación de archivos
 - Estructura del encabezado de archivos
 - Procesamiento de información para su almacenamiento en forma binaria
 - Codificación de información

- Manejo del *joystick*
 - Detectar la tecla pulsada
 - Asignar una tarea de acuerdo a la tecla pulsada
 - Mostrar un mensaje en el *display* de acuerdo a la tecla pulsada

- Manejo del GLCD
 - Configuración del sistema
 - Mostrar menú de opciones
 - Generar y mostrar gráficas a partir de los datos obtenidos
 - Indicar el estado de algún proceso en curso
 - Mostrar resultados de los distintos procesos

- Módulo de comparación y captura
 - Configuración de los registros del módulo de comparación y captura
 - Cálculo del número de cuentas para desbordamiento por comparación, de acuerdo a la tasa de muestreo seleccionada

Cabe mencionar que, para el desarrollo del programa, se utilizó una tarjeta de desarrollo de Arduino.

El programa del microcontrolador está desarrollado de manera que cada una de las tareas o procesos, mencionados anteriormente, puedan ser llamados como una subrutina.

Inicialmente, al usuario se le presenta un menú de opciones, donde cada una de ellas lo lleva a un proceso o a un submenú, con lo cual puede configurar y poner a funcionar al sistema.

3.3.2. Rutina para la adquisición y conversión de datos

La comunicación entre los convertidores analógico digitales y el microcontrolador se lleva a cabo a través del protocolo de comunicación SPI (*Serial Peripheral Interface*). Para digitalizar la señal proveniente de los sensores sísmicos fue necesario crear un algoritmo que permite habilitar los convertidores analógicos digitales de cada canal secuencialmente, es decir, habilitar uno a la vez. Considerando que el bus del protocolo SPI tiene la característica de compartir la línea MISO (*Master Input Slave Output*) con todos los dispositivos esclavos, habilitamos un convertidor a la vez, poniendo en alta impedancia los demás se garantiza la digitalización de la señal de cada canal.

Para programar la operación del protocolo SPI se utilizó la librería SPI.h del entorno de desarrollo de Arduino, en esta librería se pueden utilizar las funciones para modificar la velocidad de transmisión de datos, el orden de salida de los datos (si se desea obtener primero el *bit* más significativo o el menos significativo) y el modo de transmisión de los datos.

Para el diseño del algoritmo que permite la digitalización de las señales provenientes de los sensores, es necesario establecer un nivel lógico bajo en la terminal CS del convertidor A/D correspondiente. Dando esto como resultado la adquisición de una trama de 32 *bits*, de los cuales el *bit* más significativo, *bit* 31, indica el fin de la conversión mediante un estado lógico bajo; el *bit* 30 es un *bit* que siempre está en estado bajo; del *bit* 29 al *bit* 5 se tiene el valor de la conversión, y los últimos 5 *bits* de esta trama indican el nivel de ruido de la conversión. En la figura 3.33 se ilustra la trama de *bits* mencionada.

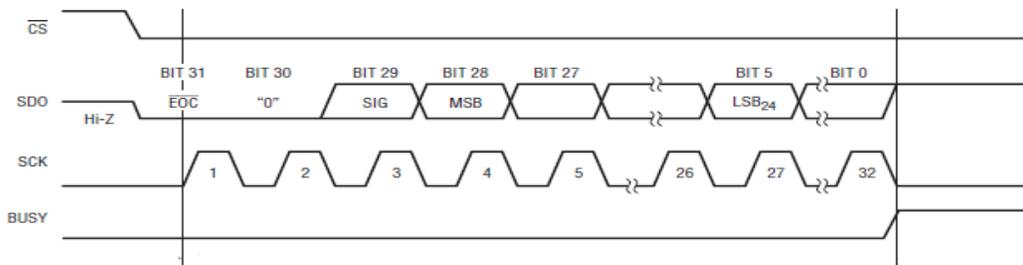


Figura 3.33. Trama de bits de salida del convertidor A/D.

La trama de *bits* mencionada es transmitida por medio de la línea MISO del protocolo SPI, por lo que para poder obtener el dato de la conversión es necesario crear en el algoritmo una variable del tipo “*long*”. Esta variable tiene una longitud de palabra de 32 *bits*, necesarios para almacenar los *bits* de salida del convertidor, esto para posteriormente hacer un corrimiento a la derecha y eliminar los 5 *bits* que se encuentran a la derecha del *bit*

3.3.3. Rutina para la adquisición de fecha y hora del GPS

Como se mencionó en la sección 3.2.7, el sistema DIGI contiene un receptor GPS para obtener los datos correspondientes a la hora UTC y fecha. En particular, el receptor GPS utilizado para lograr esta tarea, tiene la capacidad de transmitir algunos de los mensajes de salida del protocolo NMEA 0183, los cuales se describen brevemente en la tabla 3.9.

Cabecera de protocolo	Descripción
GGA (<i>Global Positioning System Fixed Data</i>)	Contiene información referente al tiempo, posición y datos fijos.
GSA (<i>GNSS DOP and Active Satellites</i>)	Contiene información referente al modo de operación del receptor GPS. (2D o 3D dependiendo del modo que se habilite)
GSV (<i>GNSS Satellites in View</i>)	Contiene información referente al número de satélites GPS que el receptor está percibiendo, el número de identificación de los satélites, así como la elevación, y la relación señal a ruido.
RMC (<i>Recommended Minimum Specific GNSS Data</i>)	Contiene información referente a la hora UTC, fecha, posición y velocidad.
VTG (<i>Course Over Ground and Ground Speed</i>)	Contiene información referente a la velocidad con referencia al suelo.
GLL (<i>Geographic Position – Latitude Longitude</i>)	Contiene información referente a la longitud, latitud, hora UTC de fijación de la posición y el estado de los datos.

Tabla 3.9. Descripción de los mensajes de salida del receptor GPS utilizado.

Con la finalidad de obtener los datos referentes a la fecha y hora UTC, se utilizó el mensaje de salida **RMC**, ya que es el único mensaje que contiene los datos necesarios para sincronizar al sistema “DIGI”.

En la figura 3.35 se ilustra el mensaje de salida RMC, del cual se indican los datos de fecha y hora UTC.

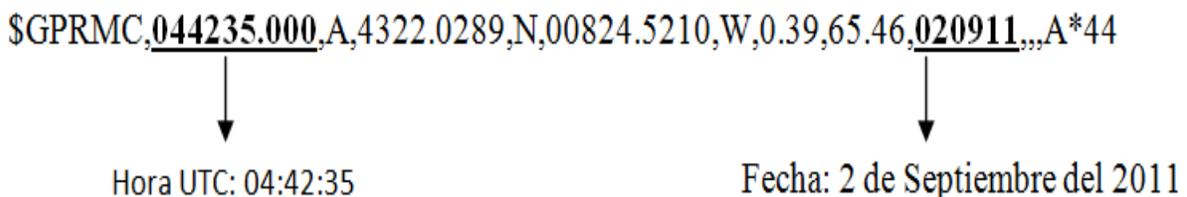


Figura 3.35. Mensaje RMC del protocolo NMEA 0183.

El número de caracteres del mensaje de salida RMC no es constante, por lo que en el desarrollo del algoritmo, para obtener los datos de fecha y hora UTC, fue necesario definir una variable del tipo *string* con una longitud de 150 caracteres. Esto para almacenar los caracteres provenientes de dicho mensaje, que es transmitido del receptor GPS al microcontrolador, por medio de las terminales Rx1 y Tx1 del puerto de comunicación serial (UART1) del microcontrolador. Una vez que se definió la variable para almacenar dicho mensaje, fue necesario fijar la velocidad de transmisión de datos de la UART1, que debe ser igual a 4800 bauds, este valor es el que recomienda el fabricante del receptor GPS utilizado. Teniendo hecha esta configuración, el programa tendrá que leer de forma continua los datos de dicho puerto, sin embargo, para almacenar el mensaje RMC se hace un recorrido de la cadena almacenada, buscando que el identificador de la cadena (los primeros seis caracteres contando el signo \$) tenga el comando “\$GPRMC”, con esta acción se garantizará que se ha almacenado el mensaje RMC.

Ya que se ha verificado que se tiene el mensaje de interés, se procede a recorrer nuevamente la cadena, para contar el número de separadores “,” que contiene esta misma, por lo que cuando se encuentre el primer separador, los caracteres que siguen corresponden a los datos de la hora UTC, por lo que se deben almacenar los siguientes seis caracteres en una variable del tipo cadena. Siguiendo con el recorrido de la cadena, y al encontrar el noveno separador, se obtendrán los caracteres correspondientes a la fecha, por lo que del mismo modo que con la hora, se deberán almacenar los siguientes seis caracteres en otra variable del mismo tipo.

El algoritmo mencionado para la adquisición de los datos de hora UTC y fecha se ejecuta cada vez que se abre un archivo nuevo para el almacenamiento de los datos, ya que en un principio la idea era que se leyeran cada segundo los datos del receptor GPS, sin embargo, al momento de reconstruir los datos almacenados de una señal conocida (señal senoidal), se observó que cada que se ejecutaba la rutina para la adquisición de datos del GPS se perdía información, trayendo como consecuencia que la señal no se reconstruyera adecuadamente. Por esta razón el algoritmo para la adquisición de los datos se ejecuta una sola vez, y con la finalidad de tener el registro de tiempo de los datos almacenados, se diseñó una aplicación en LAbVIEW que permite agregar la referencia de fecha y hora UTC, partiendo de los datos del GPS que se almacenan al iniciar cada archivo.

En la figura 3.36 se ilustra, por medio de un diagrama de flujo, el proceso descrito para obtener los datos de fecha y hora UTC.

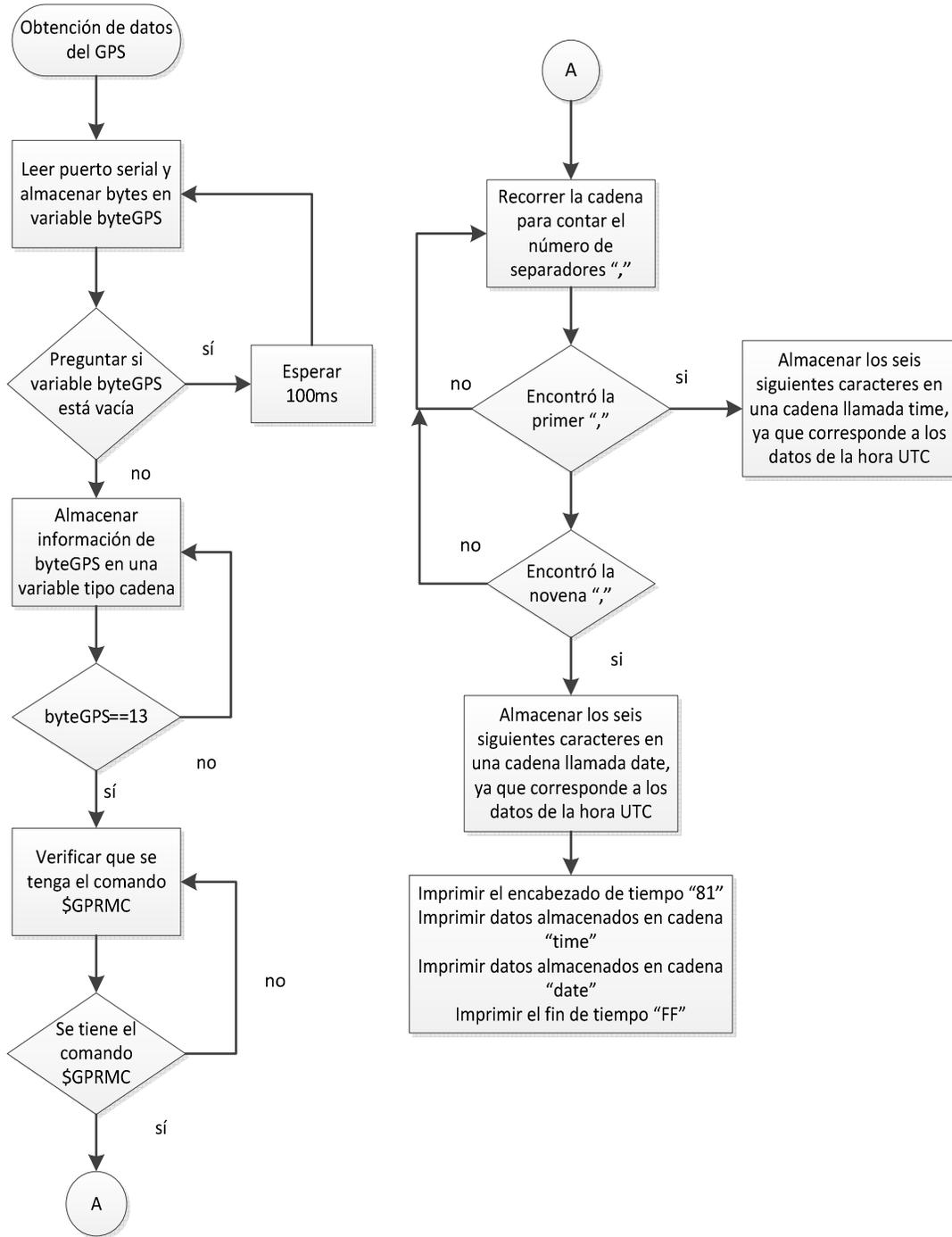


Figura 3.36. Diagrama de flujo para obtener los datos de hora y fecha del GPS.

3.3.4. Almacenamiento de datos

El almacenamiento de datos del sistema DIGI será implementado en una memoria micro SD, para ello, se hace uso de una librería (SD de Arduino), que permite realizar

operaciones de lectura y escritura en memorias flash SD. Esta librería soporta los sistemas de archivos FAT16 y FAT32 en tarjetas SD de capacidad estándar (SDSC) y de alta capacidad (SDHC). Además, el formato de los nombre de los archivos es 8.3, es decir, ocho caracteres para el nombre de los archivos y tres caracteres para la extensión. Algunas funciones utilizadas de la librería son:

- Crear, abrir y cerrar un archivo
- Escritura en codificación ASCII (*American Standard Code for Information Interchange*)
- Escritura de forma binaria

El contenido de cada archivo, figura 3.37, comienza por un encabezado, formado por ocho caracteres, que pueden ser elegidos de entre el alfabeto, y los números del 0 al 9; a continuación se encuentra la tasa de muestreo del archivo, después encontramos el identificador de inicio de marca de tiempo "81 decimal", seguido de la fecha (día, mes, año) y hora UTC (hora, minuto, segundo); a continuación encontramos el identificador de fin de marca de tiempo "FF" y finalmente se encuentran los datos de los tres canales de adquisición del sistema con sus respectivos *header* de inicio y fin.

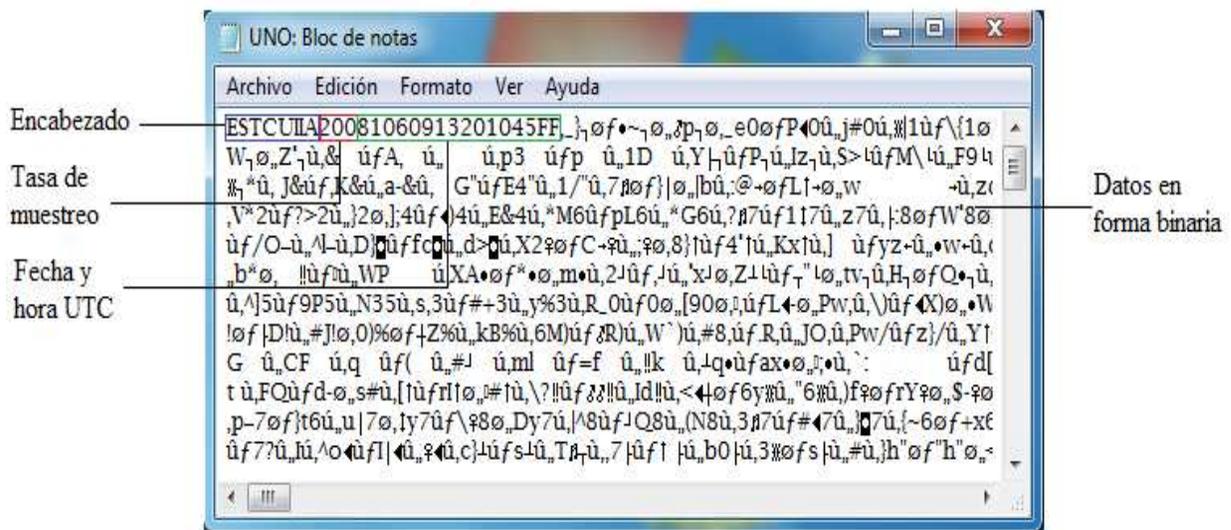


Figura 3.37. Estructura de datos almacenados en un archivo.

Los datos serán almacenados en la memoria en forma binaria, excepto el encabezado, la tasa de muestreo del archivo, la fecha y la hora UTC, los cuales serán almacenados con codificación ASCII.

El encabezado de los archivos se encuentra almacenado en una cadena, el contenido de fecha y hora en otras dos variables. Para poder almacenar esta información se hace uso de la función *print*, la cual nos permite almacenar información en codificación ASCII.

Para almacenar los datos de los sensores en la memoria, se hace uso de la instrucción *write*, la cual escribe un *byte* al archivo, tal y como se encuentra almacenada en la variable, sin realizar alguna codificación. Este tipo de escritura reduce el tamaño de los archivos, lo cual nos permite almacenar una mayor cantidad de datos, en comparación con la escritura con codificación ASCII.

Anteriormente se mencionó que los datos entregados por los convertidores analógicos digitales, tienen una extensión de 32 *bits* (4 *bytes*), sin embargo, los *bits* en los que se encuentra la información (valor de la conversión) son a partir del *bit 29* al *bit 5*.

Para almacenar el valor de la conversión en forma binaria, se tiene que realizar un procesamiento de los 32 *bits* que se obtienen del convertidor de un canal, para tomar de ellos solo los 24 *bits* que nos interesan. Para realizar esto se tienen cuatro variables (*w*, *x*, *y*, *z*) con los 4 *bytes* que se obtienen del convertidor, en donde *w* es el *byte* más significativo y *z* es el menos significativo. Adicional a las variables (*x*, *w*, *y*, *z*) se trabaja con las variables *aux1*, *aux2*, *aux3* y *aux4*, que son variables auxiliares y que nos ayudan a realizar el procesamiento de los datos y también para almacenar el resultado de todas las operaciones realizadas, estas últimas variables tienen el valor de la conversión.

El proceso en cuestión sólo obtiene y procesa los datos de un convertidor analógico digital. Para obtener los primeros 8 *bits* (más significativos) del valor de la conversión se realizan los siguientes pasos:

- Se realiza una operación lógica AND entre el *byte* 'w' y el valor 0x1F, con la finalidad de poner a cero los 3 *bit* más significativos del *byte* 'w'. El resultado de la operación se almacena en la variable 'aux1'.
- Se realiza un corrimiento a la izquierda de 3 *bits* a la variable 'aux1'.
- Se realiza una operación lógica AND entre el *byte* 'x' y el valor 0xE0, con la finalidad de poner a cero los 5 *bit* menos significativos del *byte* 'x'. El resultado de la operación se almacena en la variable 'aux2'.
- Se realiza un corrimiento a la derecha de 5 *bits* a la variable 'aux2'.
- Se realiza una operación OR entre *aux1* y *aux2*, el resultado se almacena en *aux1*.

Para obtener los siguientes 8 *bits* del valor de la conversión, se realiza el mismo proceso pero ahora se usan los *bytes* 'x', 'y' y las variables *aux2* y *aux3*. Finalmente para obtener los últimos 8 *bits* (menos significativos) del valor de la conversión se usa el mismo proceso pero con los *bytes* 'y', 'z' y las variables auxiliares *aux3* y *aux4*.

Con base en los requerimientos del usuario, los datos almacenados en la memoria micro SD tienen que cumplir con un formato en la estructura de la información, figura 3.38. Para los diferentes canales, los *bytes* de identificación tienen valores 0x82, 0x83 y 0x84 perteneciendo a los canales 1, 2 y 3, respectivamente. En la trama de cada canal, los tres

bytes siguientes (bajo, medio y alto) de cada paquete, contienen el valor de la muestra correspondiente.

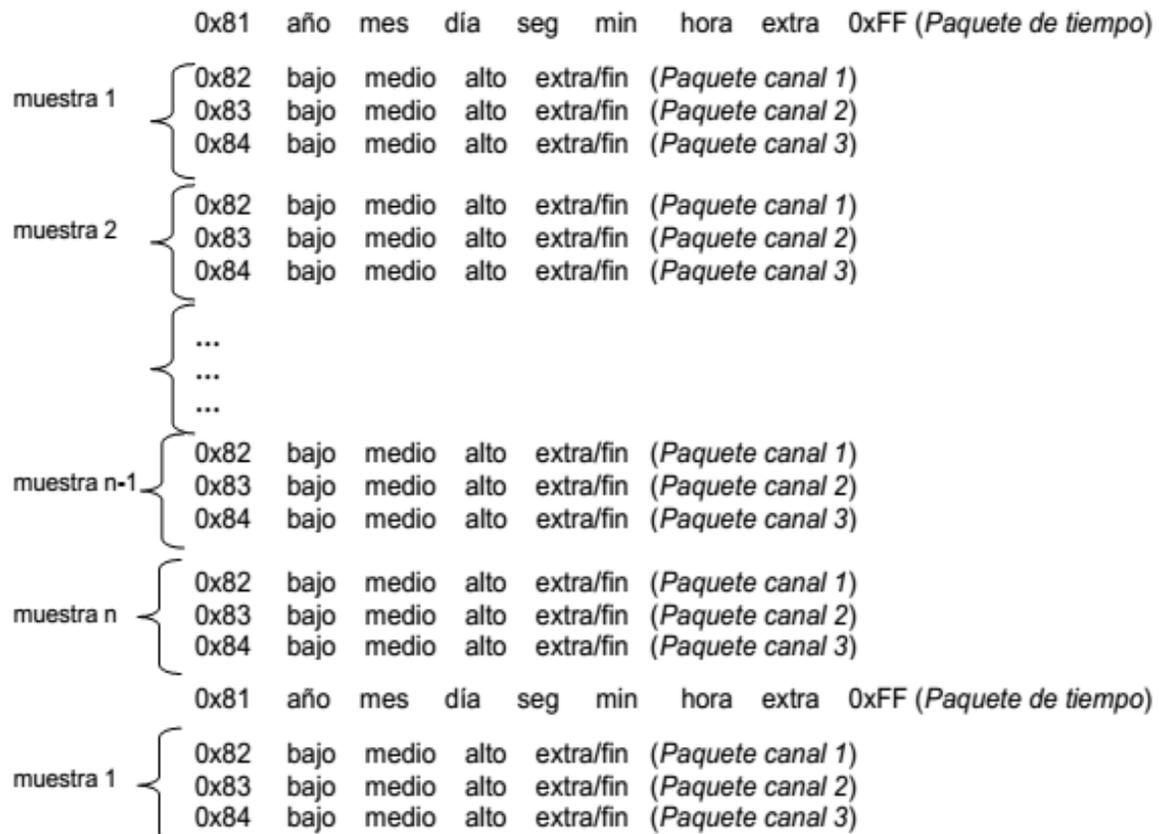


Figura 3.38. Formato de los datos requeridos por el usuario.

Otro requerimiento del usuario es que, los *bytes* de cada muestra se almacenen sin el *bit* número 7. El *byte* extra/fin contiene la información para completar el bit siete de los *bytes* bajo, medio y alto. El *byte* extra/fin tiene el siguiente formato:

<i>Bit0</i>	<i>bit</i> número 7 del <i>byte</i> bajo
<i>Bit1</i>	<i>bit</i> número 7 del <i>byte</i> medio
<i>Bit2</i>	<i>bit</i> número 7 del <i>byte</i> alto
<i>Bit3</i>	siempre 1
<i>Bit4</i>	siempre 1
<i>Bit5</i>	siempre 1
<i>Bit6</i>	siempre 1
<i>Bit7</i>	siempre 1

Para realizar el proceso que da el formato a los datos del sistema DIGI se hace uso de las variables "*bit_high*, *bit_middle*, *bit_low*, *extra*, *extra_fin*" y de las variables que contienen el valor de la muestra "*aux1*, *aux2*, *aux3*". Los pasos a seguir para obtener el valor del *bit_high* son:

- Se realiza una operación lógica AND entre la variable `aux1` y el valor `0x80`, el resultado se almacena en la variable `bit_high`.
- Se realiza un corrimiento a la derecha del contenido de la variable `bit_high`.
- Se realiza una operación lógica AND entre la variable `aux1` y el valor `0x7F`, el resultado se almacena en la variable `aux1`.

Para obtener el `bit_middle` se realiza un proceso similar al del `bit_high`, pero ahora ese `bit` se extrae de la variable `aux2`, además el corrimiento en este caso es de 6 `bits`. Para obtener el valor de `bit_low` el proceso es similar, con la diferencia de que el valor de ese `bit` se extrae de la variable `aux3`, además de que el corrimiento es de 7 `bits`. Finalmente, el `byte extra_fin`, está formado por el resultado de una operación lógica OR entre las variables `extra` (con valor `0xF8`), `big_high`, `bit_middle` y `bit_low`. En la figura 3.39 se muestra el código que realiza las tareas para la obtención del valor de la muestra (24 `bits`) y dar el formato requerido a los datos que contienen dicho valor.

```
void convertirl1()
{
    aux1 = w & 0x1F;
    aux1 = aux1 << 3;
    aux2 = x & 0xE0;
    aux2 = aux2 >> 5;
    aux1 = aux1 | aux2;
    bit_high = aux1 & 0x80;
    bit_high = bit_high >> 5;
    aux1 = aux1 & 0x7F;
    .....
    aux2 = 0;
    aux2 = x & 0x1F;
    aux2 = aux2 << 3;
    aux3 = y & 0xE0;
    aux3 = aux3 >> 5;
    aux2 = aux2 | aux3;
    bit_middle = aux2 & 0x80;
    bit_middle = bit_middle >> 6;
    aux2 = aux2 & 0x7F;
    .....
    aux3 = 0;
    aux3 = y & 0x1F;
    aux3 = aux3 << 3;
    aux4 = z & 0xE0;
    aux4 = aux4 >> 5;
    aux3 = aux3 | aux4;
    bit_low = aux3 & 0x80;
    bit_low = bit_low >> 7;
    aux3 = aux3 & 0x7F;
    .....
    extra_fin = ((extra | bit_high) | bit_middle) | bit_low;
}
```

Figura 3.39. Código para dar formato a los datos de un canal.

Una vez ejecutadas las tareas anteriores, las variables “aux1, aux2 y aux3”, contienen el valor de la conversión, con el formato requerido, por lo tanto, estas variables son las que se escribirán en la memoria micro SD. Cabe mencionar que hasta este momento la fecha y hora se encuentra almacenado en codificación ASCII, y aún no se le ha dado el formato requerido por el usuario.

Para tener control del tamaño de los archivos y que éstos sean homogéneos, se optó por la opción de monitorear el número de escrituras que se ha realizado en cada archivo. Para ello, se ha establecido como tamaño máximo del archivo 248 MB, al configurar la prueba el usuario establece el periodo de la misma, con esta información se calcula el número de escrituras totales que se efectuarán, de acuerdo a la ecuación 3.12.

$$\text{numero_escrituras} = m.p.s * (\text{dias} * 86400 + \text{horas} * 3600) \quad (3.12)$$

En la expresión anterior:

- **m.p.s.**= Frecuencia de muestreo (muestras por segundo)
- **86,400** = Cantidad de segundos en un día
- **3,600** = Cantidad de segundos en una hora

El tamaño del archivo será determinado por la ecuación 3.13, la cual toma en cuenta el número de escrituras y la cantidad de *bytes* que le corresponde a cada una de ellas.

$$\text{Tamaño}_{\text{archivo}} = (\text{numero}_{\text{escrituras}} * 15) + 16 + 3 \quad (3.13)$$

Dónde:

- **15** = (3 canales) * (5 *bytes* por muestra)
- **16** = Número de *bytes* para almacenar la fecha y hora UTC
- **3** = 3 *bytes* para indicar la tasa de muestreo de la prueba

Si el número de escrituras máximo que le corresponde a cada archivo se ha cumplido, y aún no ha terminado el periodo de la prueba, el programa cierra el archivo actual y abre un nuevo archivo para seguir almacenando. Este proceso se repite hasta que el periodo de la prueba termine y con ello se cierre el archivo actual para así guardar los datos almacenados.

En la figura 3.40 se muestra el diagrama de flujo para almacenar el encabezado, la hora y fecha proporcionada por el GPS y los datos de los sensores.

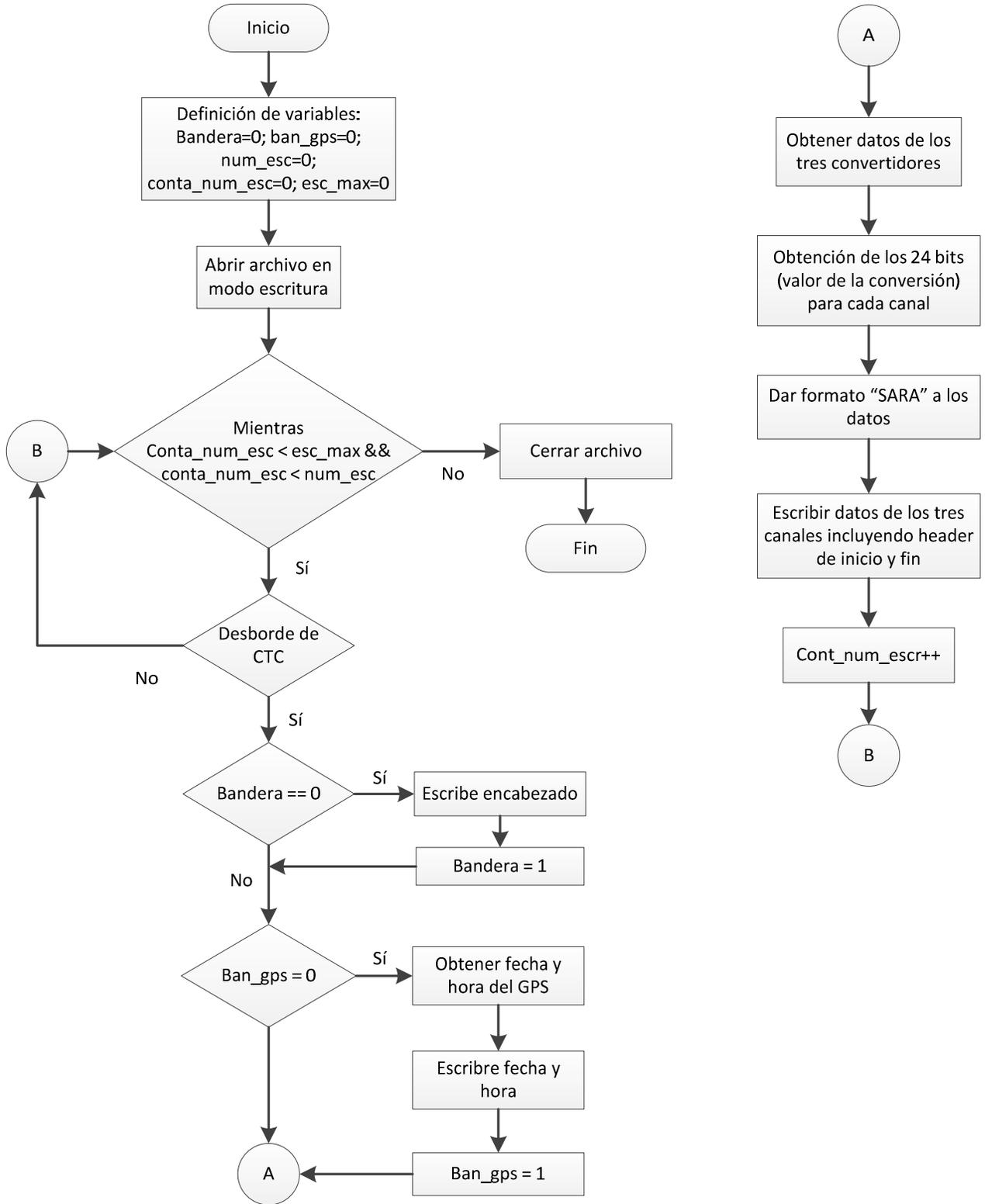


Figura 3.40. Diagrama de flujo del bloque de escritura de datos al archivo.

3.3.5. Manejo del *joystick*

Como se mencionó anteriormente, el *joystick* nos permitirá desplazarnos a través de los menús implementados en el microcontrolador y que serán mostrados en el *display* gráfico. El *joystick* consta de 5 interruptores (teclas), que han sido conectadas al microcontrolador para detectar la función que se les ha asignado:

- Terminal PE4: Tecla arriba
- Terminal PE5: Tecla abajo
- Terminal PG5: Tecla centro
- Terminal PE3: Menú principal
- Terminal PD7: Derecha (Reservada para futuras actualizaciones)

La detección de algún evento en las teclas se realiza a través de una función que monitorea un cambio de estado lógico en algunas de las terminales descritas anteriormente, y en las cuales están conectadas los interruptores del *joystick*.

El efecto de rebote mecánico, que se presenta en la mayoría de los interruptores, tiene una duración mínima de unos cuantos microsegundos y una máxima de algunas decenas de milisegundos.

Para evitar el rebote mecánico en nuestro sistema, se ejecutarán algunos retardos a través de una función implementada en la librería *delay.h*, esto con el objetivo de detectar eficazmente si la tecla ha sido presionada por el usuario. El periodo del retardo que será utilizado para descartar dicho rebote será de 20 ms, esto a partir de pruebas que se le hicieron al *joystick* utilizado, y considerando las condiciones de operación del sistema que se pueden presentar al operar a la intemperie.

Las distintas opciones de selección se presentarán en el *display*, las cuales serán elegidas a través del *joystick*. Esta selección se realizará desplazando una flecha o cursor que se posicionará en la opción seleccionada. Cada opción tendrá un valor único que será asignada a la variable *opc*, que servirá como apuntador para desplegar el menú seleccionado; adicionalmente se tienen otras dos variables, cuyo valor estará determinado por la cantidad de opciones presentadas en pantalla y el por valor de *opc*. Lo anterior se implementará en una función llamada *teclado (opc, lo, hi)*, que tiene como argumentos las variables *opc*, *lo* y *hi*. Por ejemplo, para el caso del menú principal, *opc=1* y si se tienen tres opciones a elegir, tendremos que *lo=1* y *hi=3*, las cuales nos permitirán desplazarnos entre esas opciones al presionar las teclas arriba o abajo y si el usuario lo desea seleccionar una de ellas.

También se implementará una función mensaje, la cual nos permitirá mostrar alguna opción o mensaje en pantalla a la cual apunta el cursor, esto una vez que se ha presionado la tecla de entrar. Al presionar la tecla arriba se disminuirá en 1 el valor de *opc*, y si se

pulsa la tecla abajo se incrementara en 1 el valor de la misma variable. El valor de la variable *opc* tendrá como límite superior e inferior a las variables *hi* y *lo* respectivamente, de manera que la selección siempre estará dentro de las opciones admitidas para el menú que actualmente se muestra en el *display*. En la figura 3.41 se presenta el diagrama de flujo para el manejo del *joystick*.

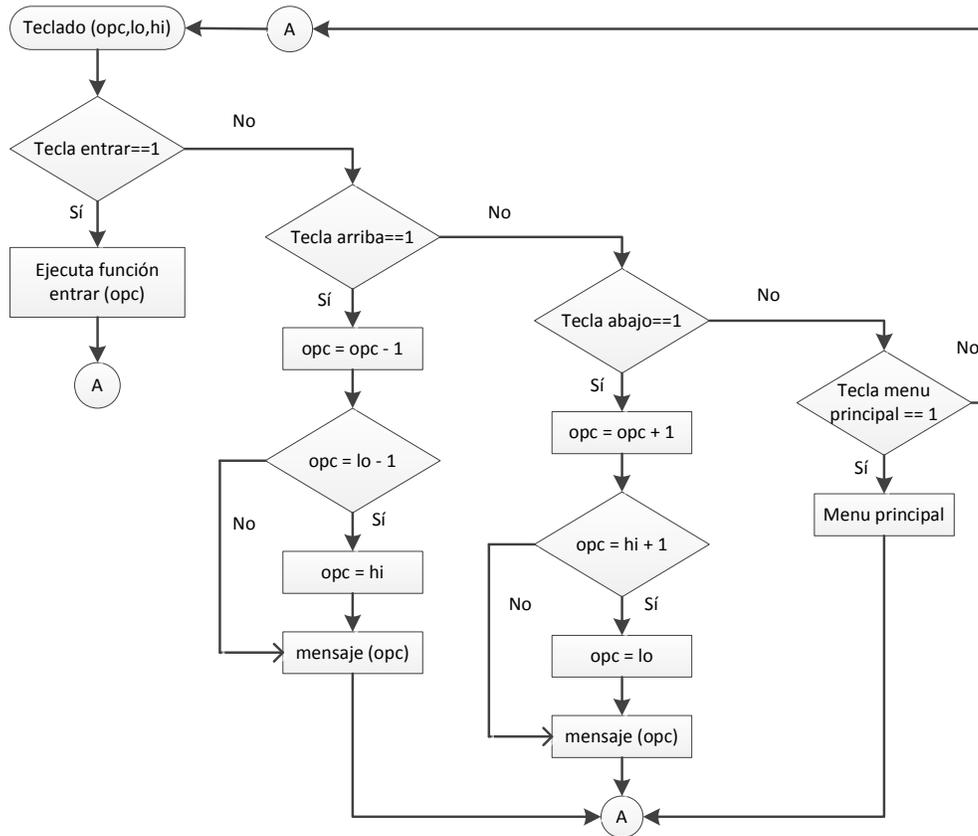


Figura 3.41. Diagrama de flujo para el manejo del joystick.

3.3.6. Manejo del GLCD

Para el manejo del *display* gráfico se hará uso de una librería. Esta librería cuenta con funciones que son compatibles con algunos de los microcontroladores de Atmel, entre ellos se encuentra el usado en el desarrollo del sistema. Algunas de las funciones que se usaron de esta librería son:

- Inicializar el *display*
- Modificar el color de los puntos entre blanco o negro
- Limpiar la ventana del *display*
- Dibujar líneas verticales, horizontales o líneas entre dos coordenadas
- Escribir datos en la posición actual
- Colocarse en una posición específica

- Dibujar un solo punto o pixel
- Dibujar rectángulos y círculos, permite rellenarlos al encender o apagar los puntos o pixeles
- Usar tipos de letras y tamaños incluidos en la librería
- Dibujar un mapa de *bit* en el *display*

A partir de estas funciones se pueden implementar otras que ayuden a la creación de gráficas y menús, que permitirán al usuario interactuar con el sistema, además de la visualización de tareas. En la figura 3.42 se muestra el diagrama de flujo de los diferentes menús y opciones que serán desplegados en el *display*.

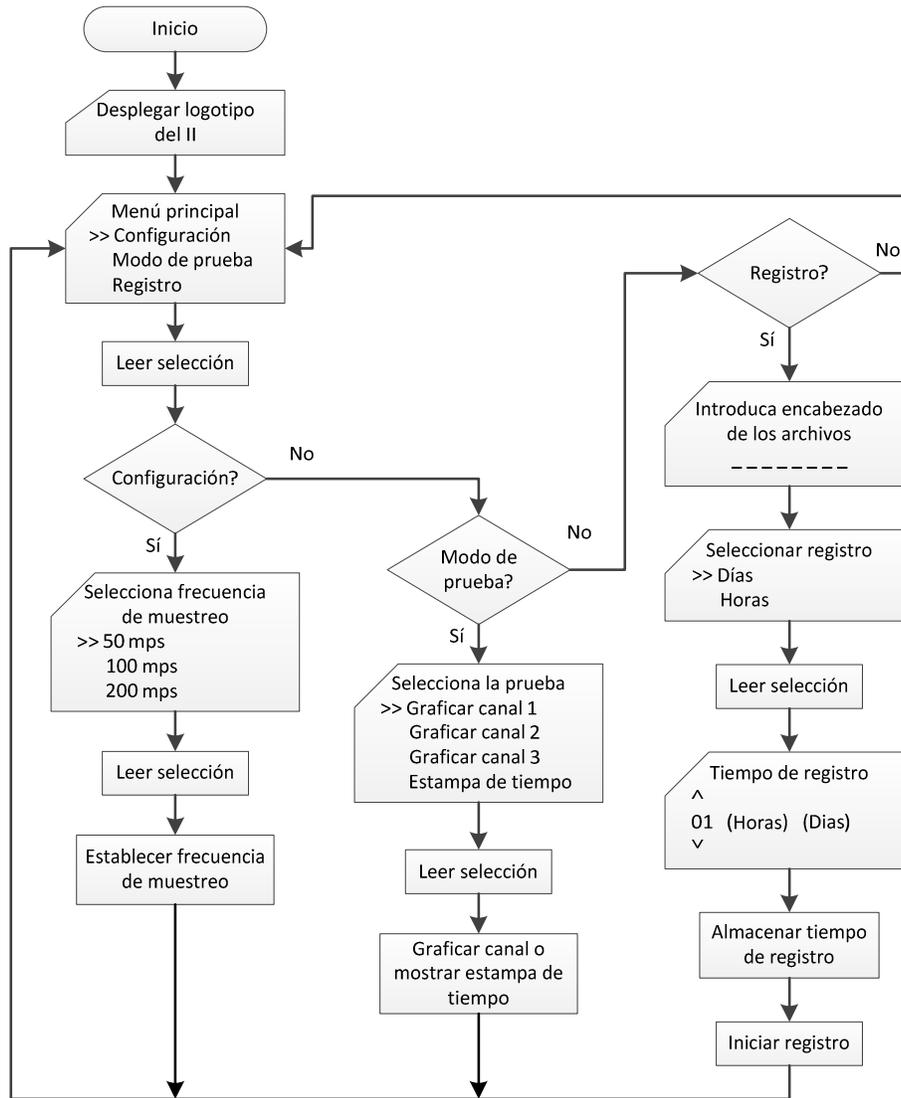


Figura 3.42. Diagrama de flujo de los menús mostrados en el *display* gráfico.

Selección de opciones a través de menús

Un menú constituye una serie de líneas de texto, en el que se pueden seleccionar las diferentes opciones disponibles. De acuerdo a las necesidades planteadas al principio del capítulo, se diseñarán una serie de pantallas que conformarán un menú para la interacción con el sistema; a través de la función teclado descrita anteriormente, que detecta las acciones en el *joystick*, con lo cual se podrá seleccionar entre las distintas opciones mostradas.

Para distinguir que opción ha sido seleccionada a través de las teclas arriba y abajo se moverá un cursor que indicará la opción seleccionada. Para lograr este efecto se escribió una función mensaje (opc), que dibujara el cursor (>>) en una posición asociada a la opción seleccionada. El resto de los menús siguen la misma lógica.

Al encender el sistema, se desplegará durante unos segundos el logotipo del Instituto de Ingeniería de la UNAM, figura 3.43.



Figura 3.43. Logotipo del Instituto de Ingeniería.

A continuación se mostrará un menú principal, figura 3.44, que permanecerá en el *display*, si no se ejecuta alguna otra acción en el *joystick*. Este menú cuenta con las siguientes opciones: configuración, modo de prueba y registros, que al ser seleccionadas nos abrirán un nuevo menú con más opciones.



Figura 3.44. Menú principal.

Configuración

En la opción de configuración se mostrará otro menú, en el cual se podrá seleccionar la frecuencia de muestreo, de entre los valores de 50, 100 y 200 muestras por segundo (mps). En la figura 3.45 se muestra el menú que se genera al seleccionar la opción de configuración.



Figura 3.45. Menú de selección de frecuencia de muestreo.

Una vez seleccionada la opción deseada, se deberá presionar la tecla entrar, con lo cual quedara establecida la frecuencia de muestreo del sistema, y el programa regresará al menú principal.

Modos de prueba

En la opción de modo de prueba, el usuario podrá seleccionar entre los canales 1, 2 y 3, para ver su comportamiento, uno a la vez, o ver la estampa de tiempo obtenida del GPS. En la figura 3.46 se muestra el menú de opciones al seleccionar modo de prueba.



Figura 3.46. Menú opciones en el modo de prueba.

El comportamiento de los canales se observará en la gráfica de los datos obtenidos de cada sensor. Para realizar la gráfica mencionada se programó una función que consiste en los siguientes pasos:

- Obtener 384 muestras del canal seleccionado
- Almacenar las muestras obtenidas en un arreglo de datos

- Realizar un corrimiento a la izquierda a los datos
- Graficar cada dato en la posición correspondiente

Se obtienen 384 muestras con la finalidad de mostrar todos los datos, en una secuencia de 3 gráficas continuas, ya que el ancho del *display* está formado por 128 puntos, por ello, sólo nos permite graficar en la pantalla completa 128 muestra.

El corrimiento de los datos se realiza con la finalidad de poder graficar la amplitud de las muestras, dado que la muestra está compuesta de 24 *bits*, así con el corrimiento se puede trazar graficas claras con la parte más significativa de cada muestra. Como el *display* usado tiene 64 puntos a lo alto, 32 puntos serán usados para graficar la parte negativa de una señal, mientras que los otros 32 puntos para la parte positiva. Graficar las muestras obtenidas tiene como propósito verificar el buen funcionamiento del sistema, es decir, verificar si la conexión de los geófonos es correcta. En la figura 3.47 se muestra la gráfica de 128 muestra de un canal.



Figura 3.47. Gráfica de un canal.

En el caso de la estampa de tiempo, el modo de prueba nos ayuda a saber si la recepción de datos del GPS es correcta. La fecha se mostrará en el formato de día, mes y año, mientras que el formato de la hora UTC es hora, minuto y segundo, como lo podemos observar en la figura 3.48.



Figura 3.48. Estampa de tiempo obtenida del GPS.

Modo Registro

Al seleccionar la opción registro dentro del menú principal, la pantalla del *display* mostrará un mensaje en el que se indica que se debe ingresar el encabezado de los archivos, figura 3.49. Este encabezado, como ya se mencionó, está formado por 8 caracteres, que pueden ser seleccionados de entre el alfabeto y los número del 0 al 9.

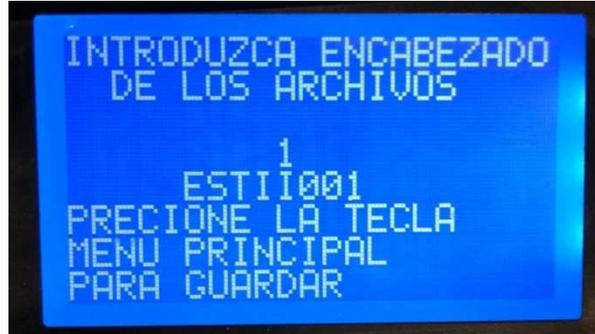


Figura 3.49. Encabezado de archivos.

Los caracteres y números se encuentran almacenados en un arreglo; para desplazarse a través de ellos se usarán las teclas arriba y abajo. Para seleccionar algún carácter o número se hará uso de la tecla de entrar, con lo cual se guarda el carácter seleccionado dentro de otro arreglo designado a almacenar los caracteres que formarán el encabezado de los archivos.

Una vez que se ha introducido el encabezado de los archivos, se procede a establecer el periodo de la prueba, el cual puede ser en días o en horas, como se observa en la figura 3.50.



Figura 3.50. Selección entre días y horas.

El rango para el periodo de pruebas en días está entre 1 y 60, mientras que el rango para horas se encuentra entre 1 y 99. La selección se podrá realizar a través de las teclas de arriba y abajo, con las que se puede aumentar o disminuir hasta llegar al periodo de registro deseado, figura 3.51, después se deberá presionar la tecla entrar para hacer efectiva la selección e iniciar el registro de datos.



Figura 3.51. Selección del periodo de prueba.

3.3.7. Módulo de comparación y captura

En nuestro sistema el módulo de comparación y captura (*CTC: Clear Time on Compare Match Mode*), es usado como disparo para obtener las muestras de los tres canales a través de los convertidores analógicos digitales. En la figura 3.52 se muestra el diagrama de bloques del módulo de comparación y captura empleado.

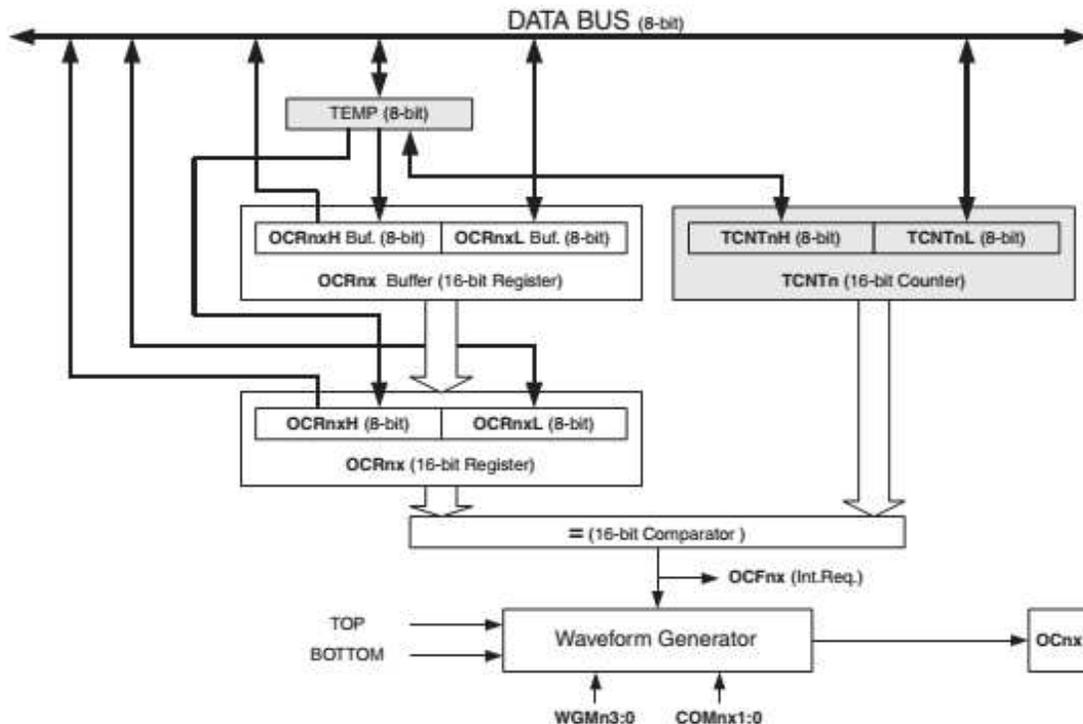


Figura 3.52. Diagrama de bloques del módulo de comparación y captura.

Para implementar la rutina de comparación y captura se utilizó el timer3, que es de 16 bits, algunos de los registros más importantes para la manipulación del timer3 en módulo de comparación y captura son:

- TCNT3. Es un contador (16 bits)
- OCR3B. Registro de comparación de salida (16 bits)
- TCCR3B. Registro de control del contador (8 bits)
- TIFR3. Registro de banderas de interrupción (8 bits)
- TIMSK3. Registro para enmascarar interrupciones (8 bits)

En la figura 3.53 se muestra la estructura del registro TCCR3A (*Timer/Counter3 Control Register A*), TCCR3B (*Timer/Counter3 Control Register B*), TIMSK3 (*Timer/Counter3 Interrupt Mask Register*) y TIFR3 (*Timer/Counter3 Interrupt Flag Register*).

Bit	7	6	5	4	3	2	1	0									
(0x91)	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 12.5%; border: 1px solid black;">ICNC3</td> <td style="width: 12.5%; border: 1px solid black;">ICES3</td> <td style="width: 12.5%; border: 1px solid black;">-</td> <td style="width: 12.5%; border: 1px solid black;">WGM33</td> <td style="width: 12.5%; border: 1px solid black;">WGM32</td> <td style="width: 12.5%; border: 1px solid black;">CS32</td> <td style="width: 12.5%; border: 1px solid black;">CS31</td> <td style="width: 12.5%; border: 1px solid black;">CS30</td> </tr> </table>								ICNC3	ICES3	-	WGM33	WGM32	CS32	CS31	CS30	TCCR3B
ICNC3	ICES3	-	WGM33	WGM32	CS32	CS31	CS30										
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W									
Initial Value	0	0	0	0	0	0	0	0									
Bit	7	6	5	4	3	2	1	0									
(0x90)	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 12.5%; border: 1px solid black;">COM3A1</td> <td style="width: 12.5%; border: 1px solid black;">COM3A0</td> <td style="width: 12.5%; border: 1px solid black;">COM3B1</td> <td style="width: 12.5%; border: 1px solid black;">COM3B0</td> <td style="width: 12.5%; border: 1px solid black;">COM3C1</td> <td style="width: 12.5%; border: 1px solid black;">COM3C0</td> <td style="width: 12.5%; border: 1px solid black;">WGM31</td> <td style="width: 12.5%; border: 1px solid black;">WGM30</td> </tr> </table>								COM3A1	COM3A0	COM3B1	COM3B0	COM3C1	COM3C0	WGM31	WGM30	TCCR3A
COM3A1	COM3A0	COM3B1	COM3B0	COM3C1	COM3C0	WGM31	WGM30										
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W									
Initial Value	0	0	0	0	0	0	0	0									
Bit	7	6	5	4	3	2	1	0									
(0x71)	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 12.5%; border: 1px solid black;">-</td> <td style="width: 12.5%; border: 1px solid black;">-</td> <td style="width: 12.5%; border: 1px solid black;">ICIE3</td> <td style="width: 12.5%; border: 1px solid black;">-</td> <td style="width: 12.5%; border: 1px solid black;">OCIE3C</td> <td style="width: 12.5%; border: 1px solid black;">OCIE3B</td> <td style="width: 12.5%; border: 1px solid black;">OCIE3A</td> <td style="width: 12.5%; border: 1px solid black;">TOIE3</td> </tr> </table>								-	-	ICIE3	-	OCIE3C	OCIE3B	OCIE3A	TOIE3	TIMSK3
-	-	ICIE3	-	OCIE3C	OCIE3B	OCIE3A	TOIE3										
Read/Write	R	R	R/W	R	R/W	R/W	R/W	R/W									
Initial Value	0	0	0	0	0	0	0	0									
Bit	7	6	5	4	3	2	1	0									
0x18 (0x38)	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 12.5%; border: 1px solid black;">-</td> <td style="width: 12.5%; border: 1px solid black;">-</td> <td style="width: 12.5%; border: 1px solid black;">ICF3</td> <td style="width: 12.5%; border: 1px solid black;">-</td> <td style="width: 12.5%; border: 1px solid black;">OCF3C</td> <td style="width: 12.5%; border: 1px solid black;">OCF3B</td> <td style="width: 12.5%; border: 1px solid black;">OCF3A</td> <td style="width: 12.5%; border: 1px solid black;">TOV3</td> </tr> </table>								-	-	ICF3	-	OCF3C	OCF3B	OCF3A	TOV3	TIFR3
-	-	ICF3	-	OCF3C	OCF3B	OCF3A	TOV3										
Read/Write	R	R	R/W	R	R/W	R/W	R/W	R/W									
Initial Value	0	0	0	0	0	0	0	0									

Figura 3.53. Registros para configurar el módulo de comparación y captura.

El registro de comparación de salida (OCR3A) es comparado con el valor del contador (TCNT3) todo el tiempo. En el modo de comparación y captura, cuando el valor del contador es igual al valor del registro de comparación de salida, el contador es puesto a cero y se levanta una bandera de igualdad de comparación (OCF3A) en el siguiente ciclo de reloj dentro del registro de bandera de interrupción. Si habilitamos el bit (OCIE3B) del registro TIMSK3, la bandera de igualdad de comparación generará una interrupción. La bandera (OCF3A) es limpiada automáticamente cuando la interrupción es ejecutada.

Alternativamente puede ser limpiada por software al escribir un 1 lógico en esa localidad del bit.

Diseño y Desarrollo del sistema “DIGI”

En la implementación del módulo de comparación y captura de nuestro sistema, se usó el método de enmascarar a la bandera de comparación, es decir, se pregunta continuamente por ella, y una vez que se presenta el evento, esta bandera es limpiada por software.

Para operar en “CTC”, los *bits* WGMn3:0 deben tener un valor de 4, como lo podemos ver en la figura 3.54.

Mode	WGMn3	WGMn2 (CTCn)	WGMn1 (PWMn1)	WGMn0 (PWMn0)	Timer/Counter Mode of Operation	TOP	Update of OCRnX at	TOVn Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCRnA	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP

Figura 3.54. Configuración de los bits WGMn3:0 para entrar en modo CTC.

Además, la frecuencia de reloj se preescaló a un valor de 64 ($f_{osc}/64$), para ellos el valor de los *bits* CSn2:0 tienen que tener un valor de 3, como lo podemos observar en la figura 3.55.

CSn2	CSn1	CSn0	Description
0	0	0	No clock source. (Timer/Counter stopped)
0	0	1	$clk_{IC}/1$ (No prescaling)
0	1	0	$clk_{IC}/8$ (From prescaler)
0	1	1	$clk_{IC}/64$ (From prescaler)
1	0	0	$clk_{IC}/256$ (From prescaler)
1	0	1	$clk_{IC}/1024$ (From prescaler)
1	1	0	External clock source on Tn pin. Clock on falling edge
1	1	1	External clock source on Tn pin. Clock on rising edge

Figura 3.55. Configuración de los bits CSn2:0 para preescalar la $f_{osc}/64$.

Una vez que se tiene configurado el modo de operación del módulo CTC y la frecuencia del contador ($f_{osc}/64$), se procede a calcular los valores que se introducirán en el registro OCR3A. Estos valores se pueden calcular a través de la expresión 3.14. Los valores calculados se muestran en la tabla 3.10.

$$OCR3A = \frac{\text{Perido de la tasa de muestreo}}{\text{Periodo de la frecuencia del cristal} * 64} \quad (3.14)$$

Frecuencia de muestreo	Valor del registro OCR3A
50 mps	5000
100 mps	2500
200 mps	1250

Tabla 3.10. Valores del registro OCR3A para diferentes tasas de muestreo.

En la figura 3.56 se muestra el diagrama de flujo del módulo CTC.

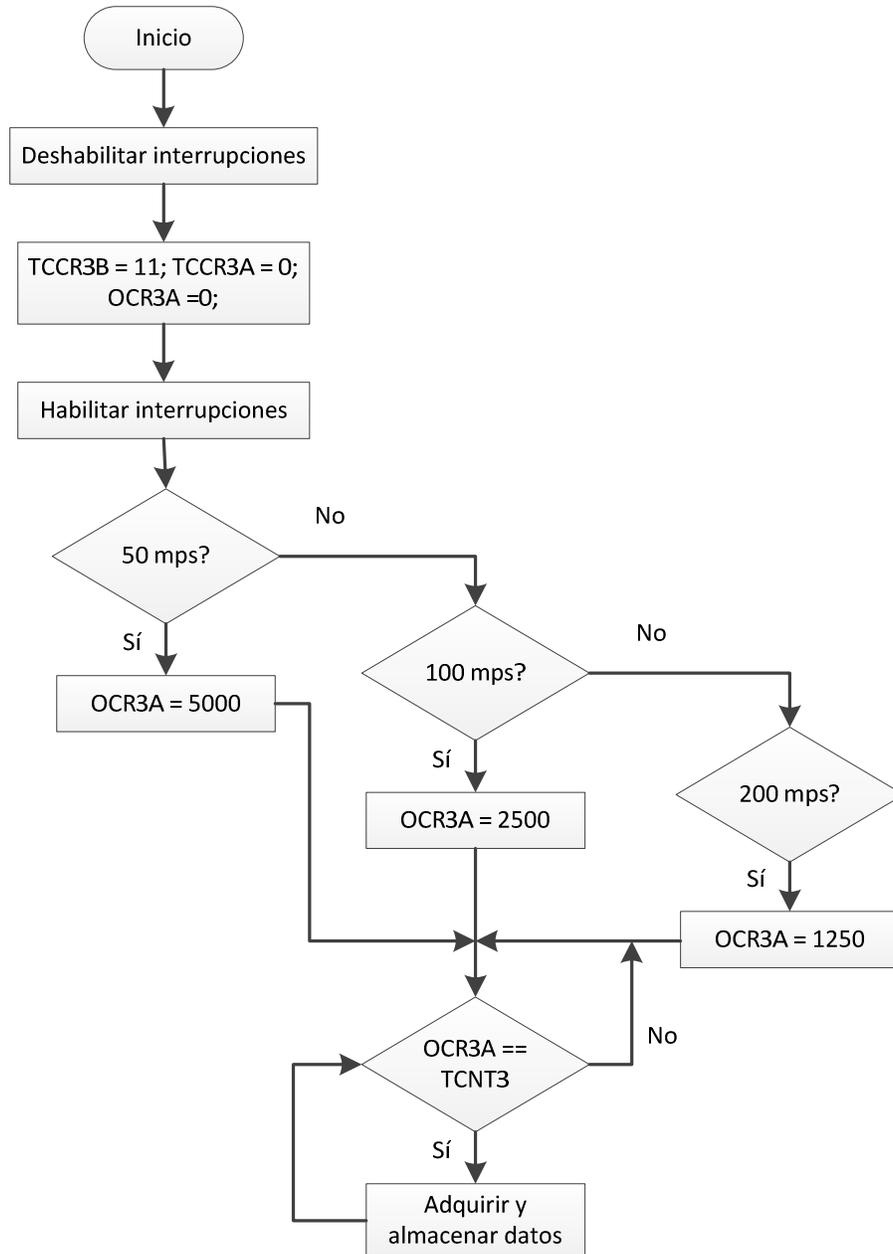


Figura 3.56. Diagrama de flujo del módulo de comparación y captura "CTC".

3.4. Procesamiento de datos

Como ya se mencionó, la referencia de tiempo dentro del archivo sólo se obtiene una vez, debido a que para obtenerla a través del receptor GPS requiere de un cierto periodo de tiempo, el cual es considerable en comparación al intervalo de adquisición de datos entre muestra y muestra, como consecuencia se presentaba pérdida de información.

Sin embargo, en las necesidades del usuario se requiere que cada muestra obtenida cuente con una referencia de tiempo. Esta referencia será introducida en el archivo a través de la ayuda de un programa desarrollado en LabVIEW.

3.4.1. Entorno de desarrollo de LabVIEW

LabVIEW (*Laboratory Virtual Instrument Engineering Workbench*) es un entorno de programación en el cual se pueden crear programas usando una notación gráfica, conectando nodos funcionales a través de alambres, sobre los cuales fluyen los datos. LabVIEW difiere de los lenguajes tradicionales de programación semejantes a C, C++ o Java, ya que en estos últimos, la forma de programación es a través de texto. LabView también es un sistema de desarrollo y ejecución de programas interactivos, diseñado para científicos e ingenieros.

El entorno de desarrollo de LabVIEW trabaja en computadoras con plataforma Windows, Mac OS X o Linux, además puede crear y correr programas dentro de una gran variedad de plataformas embebidas, incluyendo *FPGAs (Field Programmable Gate Arrays)*, *DSPs (Digital Signal Processors)* y microprocesadores.

LabVIEW tiene una extensa librería de funciones y subrutinas para ayudar en las tareas de programación, también contiene librerías de aplicaciones específicas como las de adquisición de datos (DAQ), Bus de Interfaz de Propósito General (GPIB), control de instrumento serial, análisis, representación y almacenamiento de datos y comunicación vía Internet.

Lenguaje de programación gráfico

El lenguaje de programación gráfico, mejor conocido como lenguaje G, está basado en diagramas de bloques. Este lenguaje nos permite realizar programas para tomar mediciones, analizar datos y presentarlos al usuario a través de una interfaz gráfica.

La programación gráfica reduce los detalles sintácticos asociados con los lenguajes de programación basados en texto, además permite concentrar el flujo de datos con la aplicación, ya que el uso simple de sintaxis no oscurece lo que el programa está realizando.

Como trabaja LabVIEW

Un programa en LabVIEW consiste de uno o más instrumentos virtuales (VIs, por sus siglas en inglés), son llamados así porque su apariencia y operación imitan en algunas ocasiones al instrumento físico. Sin embargo, son análogos a funciones, subrutinas o al programa principal de lenguajes de programación populares como C.

Un VI tiene tres partes importantes: un panel frontal, un diagrama de bloques y un icono.

El panel frontal es la interfaz de usuario interactiva de un VI, llamada así porque simula el panel frontal de un instrumento físico. Puede contener controles rotatorios, *push buttons* y muchos otros controles e indicadores.

El diagrama de bloques es el código fuente del VI, construido en el lenguaje de programación gráfico de LabVIEW. Además, es el programa ejecutable y está compuesto de VIs de bajo nivel, funciones, constantes y estructura de control para el programa en ejecución. En el diagrama de bloques se conectan los objetos apropiados a través de alambres para definir el flujo de datos entre ellos.

Los VIs son jerárquicos y modulares, se pueden usar como programas o subprogramas *top-level*. Con esta arquitectura, LabVIEW promueve el concepto de programación modular. Primero se divide una aplicación en una serie de tareas simples, después se construye un VI que cumpla con cada subtarea y por último estos VIs se combinan en un diagrama de bloques *top-level* para completar la aplicación.

En la tabla 3.11 se enlistan algunos términos comunes en LabVIEW, con su equivalencia en lenguajes de programación convencionales.

LabVIEW	Lenguaje Convencional
VI	Programa
Función	Función o método
subVI	Subrutina, subprograma, objeto
Panel frontal	Interfaz de usuario
Diagrama de bloques	Código programa
G	C, C++, Java, Pascal, BASIC, etc.

Tabla 3.11. Términos comunes de LabVIEW y sus equivalencias.

3.4.2. Generación de la estampa de tiempo

La estampa de tiempo (fecha y hora) nos ayudará a etiquetar cada conjunto de muestras, con el propósito de que cada dato obtenido de los sensores tenga una referencia de tiempo. Para generarla se requiere de una referencia de tiempo base (fecha y hora obtenida del receptor GPS en la cual se dio inicio la adquisición de datos). Esta base de tiempo se obtendrá de cada uno de los archivos creados por el sistema DIGI.

El contenido del archivo generado por el sistema DIGI, figura 3.57, comienza con un encabezado formado por ocho caracteres (primer recuadro), a continuación tres caracteres que indican la tasa de muestreo (segundo recuadro), después un identificador de inicio de marca de tiempo “81 decimal”, seguido de la fecha “día, mes, año” y la hora UTC “hora, minuto, segundo”, finalmente encontramos un identificador de fin de marca de tiempo “FF” decimal. Los dos últimos identificadores, junto con la fecha y hora, nos indican en qué momento se inició la prueba, (tercer recuadro) y finalmente se encuentran los datos de los tres canales del sistema con sus respectivos *headers* de inicio, para el canal 1 tenemos 0x82, para el canal 2 0x83 y finalmente para el canal 3 0x84.

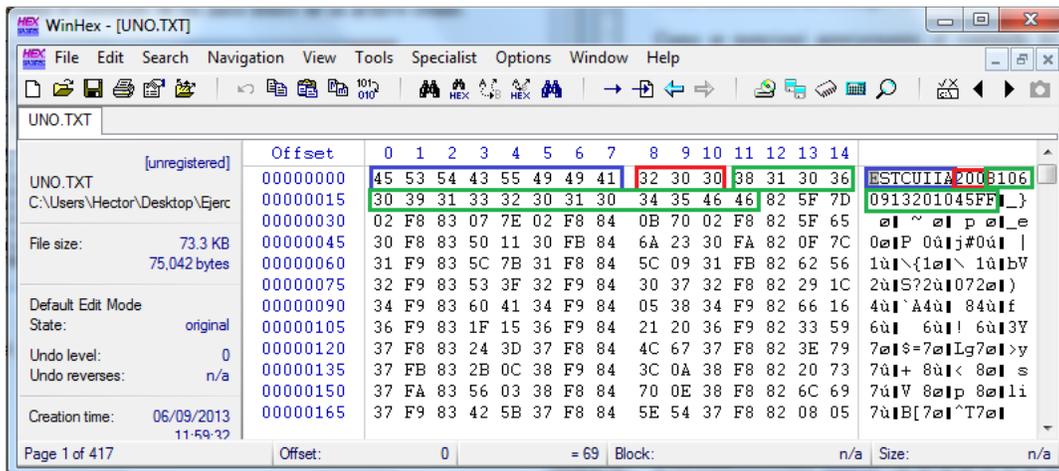


Figura 3.57. Distribución del contenido del archivo.

En la tabla 3.12 se muestra el número de *bytes* que le corresponden al encabezado de tiempo, tasa de muestreo, fecha y hora obtenida del receptor GPS y los datos adquiridos de los sensores.

<i>byte 1 – byte 8</i>	<i>Byte 9-byte 11</i>	<i>byte 12 – byte 27</i>	<i>byte 28 – último byte</i>
Encabezado de archivo	Tasa de muestreo	Fecha y hora del GPS	Datos

Tabla 3.12. Número de bytes de cada bloque de información.

En la figura 3.58 se muestra el diagrama de flujo para generar la estampa de tiempo.

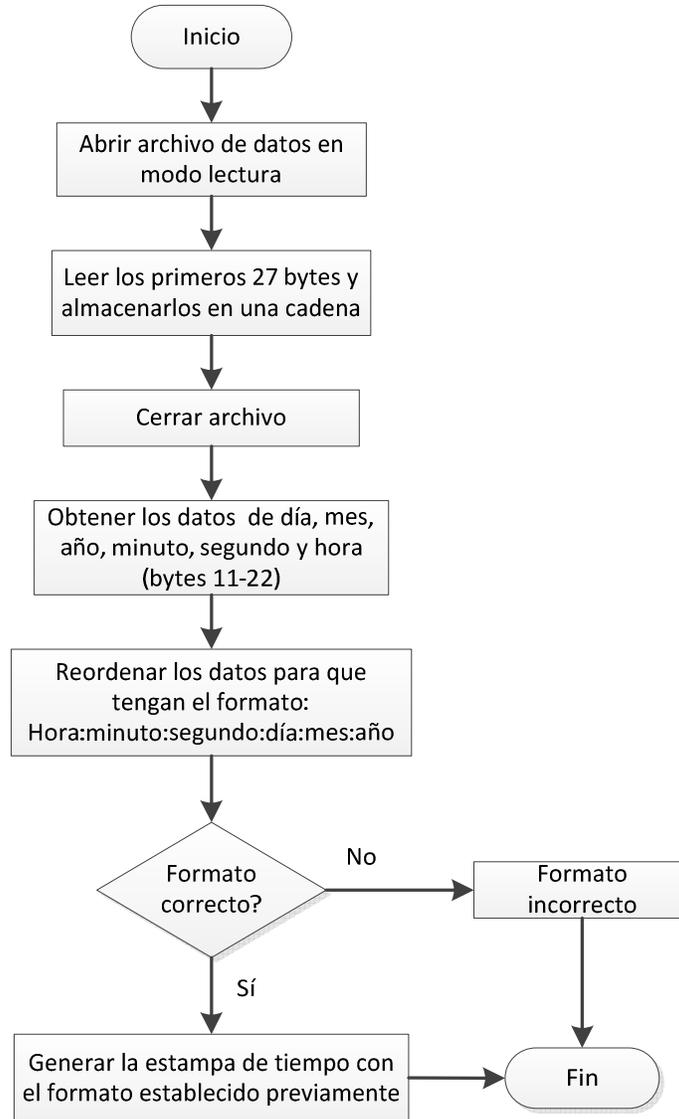


Figura 3.58. Generación de la estampa de tiempo.

La programación para generar la estampa de tiempo se realizó en LabVIEW. Algunas de las funciones o VI's más importantes para realizar esta programación son:

- *Read from text file.* Lee un número especificado de caracteres o líneas de un archivo de texto.
- *String Subset.* Regresa una sub cadena de la cadena de entrada, comenzado en el *offset* previamente establecido y contiene el número de caracteres especificados.
- *Concatenate Strings.* Concatena cadenas de entrada y arreglos de cadenas de una dimensión, dentro de una cadena simple.
- *Scan From String.* Escanea la cadena de entrada y la convierte al formato de cadena especificado.

Diseño y Desarrollo del sistema "DIGI"

- *Format Date/Time String*. Muestra un valor de estampa de tiempo o un valor numérico con el formato que se especifique, usando códigos de formato de tiempo.

Las funciones usadas para implementar la estampa de tiempo contempla el caso en el que se presente un año bisiesto. El formato que se está usando para generar la estampa de tiempo es el siguiente %H:%M:%S %d/%m/%y , donde:

- %H. Representación del número de horas (00-23)
- %M. Representación del número de minutos (00-59)
- %S. Representación del número de segundos (00-59)
- %d. Representación del día del mes (01-31)
- %m. Representación del mes del año (00-12)
- %y. Representación del año dentro del siglo (00-99)

En la figura 3.59 se muestra el diagrama de bloques (programación) para generar la estampa de tiempo. En el podemos observar la forma de como conectar los VI's para cumplir con esta tarea, además del flujo de datos que se tienen, desde la obtención de la hora y la fecha almacenados en el archivo de datos generado por el sistema DIGI, hasta la generación de la estampa de tiempo, con el formato especificado.

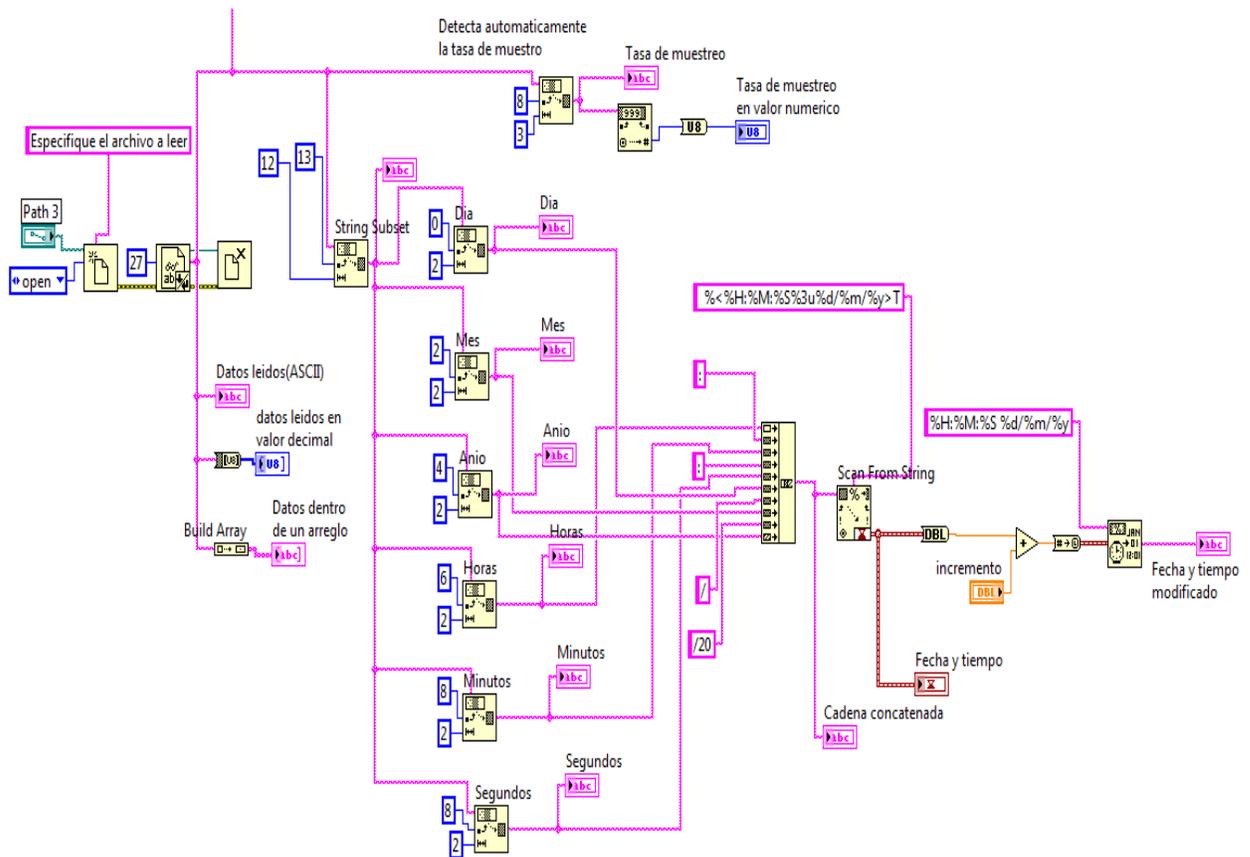


Figura 3.59. VI para generar la estampa de tiempo.

En la figura 3.60 se muestra el panel frontal, en donde podemos observar la base de tiempo (día, mes, año, hora, minuto y segundo) que fue obtenida del receptor GPS, la cual se encuentra dentro del archivo de datos generado por el sistema DIGI. Dicha base de tiempo es usada para generar la estampa de tiempo.

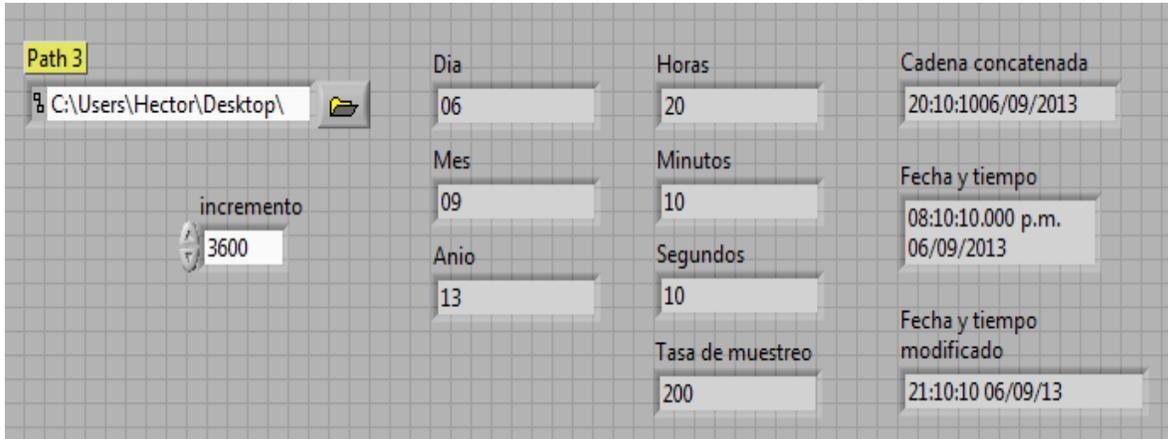


Figura 3.60. Interfaz para mostrar la estampa de tiempo.

Como podemos observar en la figura anterior, la estampa de tiempo tiene la siguiente estructura: 21:10:10 06/09/13, en donde se tiene ‘:’ como separador entre datos de hora, y ‘/’ como separador de datos de fecha. Todo el contenido forma una cadena de 15 caracteres, de los cuales para representar la hora, minuto, segundo, día, mes o año se necesitan dos caracteres para cada parámetro o dato.

La estampa de tiempo generada puede ser modificada, a través de una variable “incremento”, en la cual se introduce un incremento o decremento expresado en segundos, esto es de gran utilidad, ya que nos permitirá incrementar el valor de la estampa de tiempo, esto con la finalidad de etiquetar cada bloque de datos, tomando como base la referencia de tiempo en la cual se inició la prueba.

3.4.3. Colocación de la estampa en el archivo de datos

Para colocar una referencia de tiempo a cada bloque de datos concerniente a un segundo, se implementó un programa en LabVIEW, en el cual se creará un nuevo archivo en donde se copiara el encabezado y tasa de muestreo del archivo original, enseguida se colocará la estampa de tiempo generada con su respectivo *header* de inicio y fin, a continuación un bloque de datos de un segundo, el cual será copiado del archivo original, después se incrementará el valor de la estampa de tiempo en un segundo y se colocará en el nuevo archivo, a continuación se copiara el siguiente bloque de datos del archivo original y

se colocara en el nuevo archivo, este proceso se repitiera hasta etiquetar con una referencia de tiempo todos los bloques de datos del archivo original.

La estampa de tiempo al igual que los datos serán almacenados en forma binaria, para ello primero necesitamos convertir la estampa de tiempo a este formato. Esto se hace con la finalidad de que la información de fecha y hora, se encuentre en el mismo formato que los datos. Además, almacenar los datos en formato binario reduce el espacio que ocuparían si se realiza con codificación ASCII.

En la figura 3.61 se muestra el diagrama de flujo para pasar a formato binario el dato de año. El proceso es exactamente el mismo para los datos de mes, día, segundo, minuto y hora, los cuales serán almacenados en el mismo arreglo, y escribiendo al final el valor de 0xFF, con esto se obtiene un arreglo de datos, que contiene la estampa de tiempo con sus respectivos *headers* de inicio (0x81) y fin (0xFF) en formato binario.

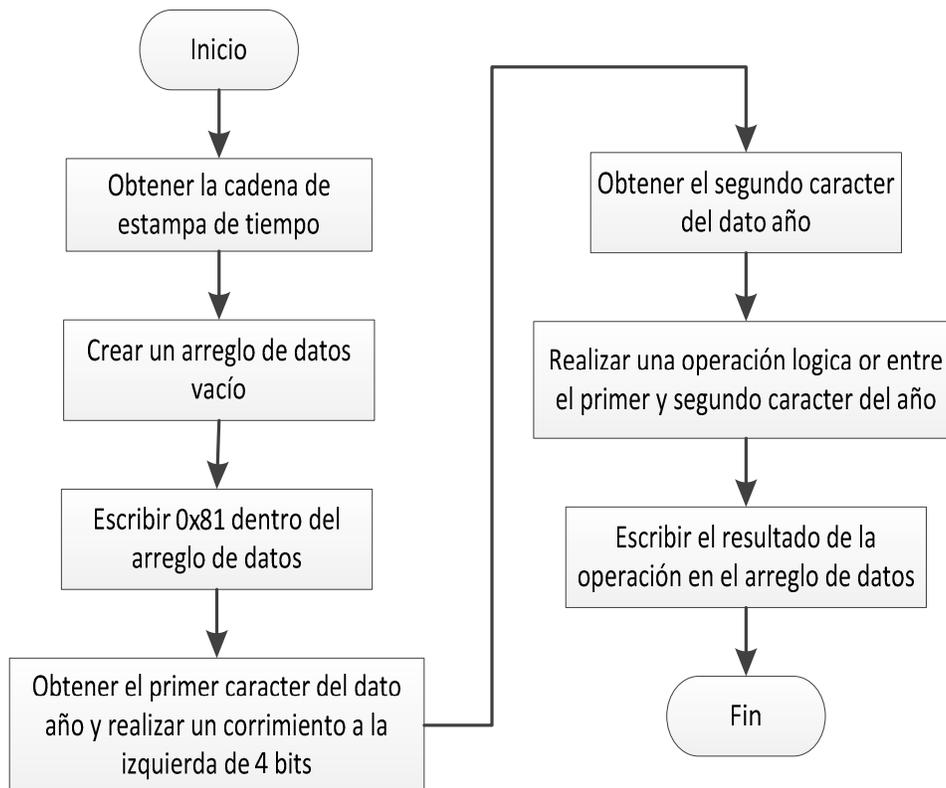


Figura 3.61. Algoritmo para convertir la estampa de tiempo a formato binario.

En la figura 3.62 se muestra el diagrama de bloques (programación) para convertir la estampa de tiempo a formato binario.

Una vez que se tiene la estampa de tiempo en formato binario, se procede a etiquetar cada bloque de datos de un segundo en un nuevo archivo, en la figura 3.63 se muestra el diagrama de flujo para realizar esta tarea.

Diseño y Desarrollo del sistema "DIGI"

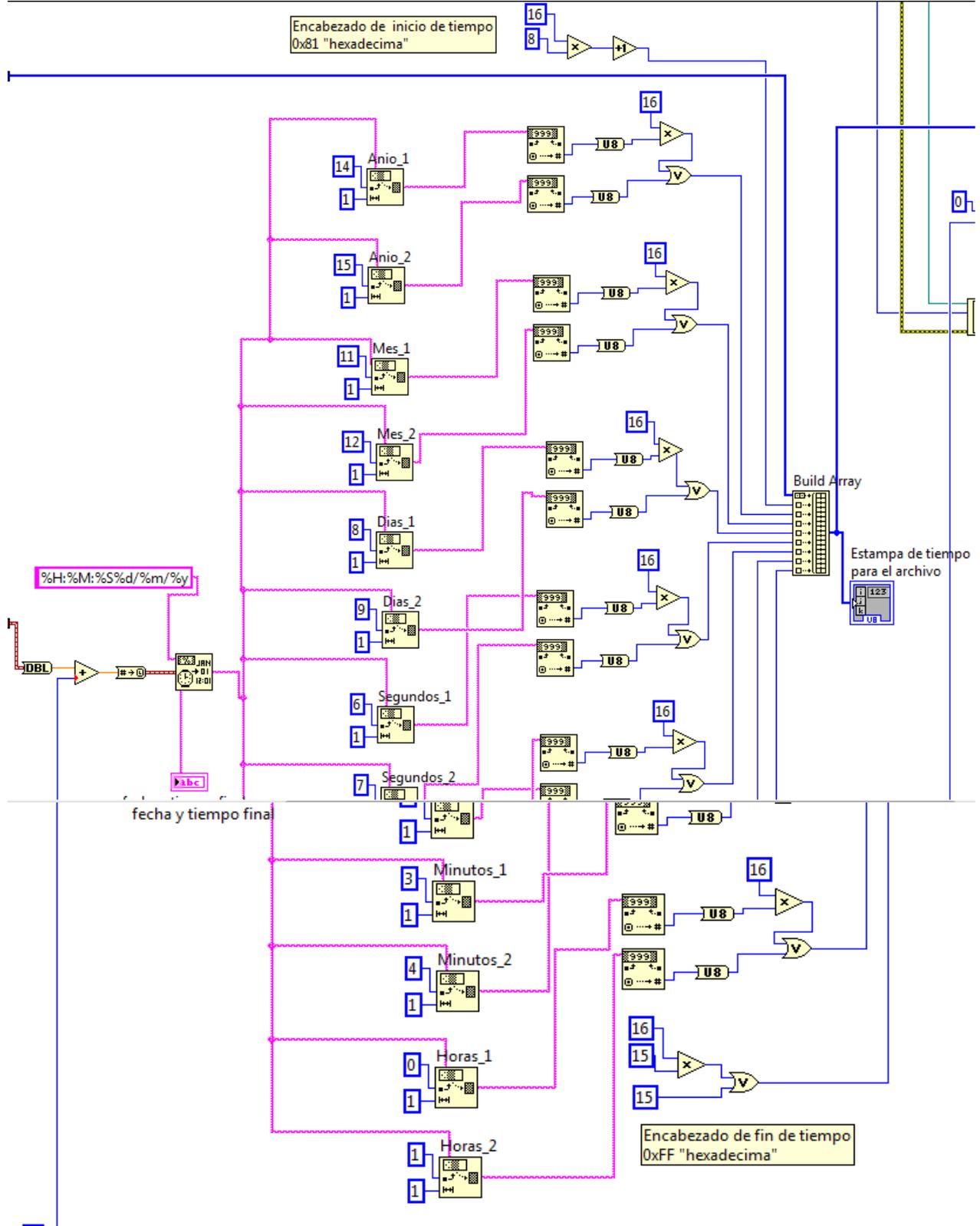


Figura 3.62. VI para convertir la estampa de tiempo a formato binario.

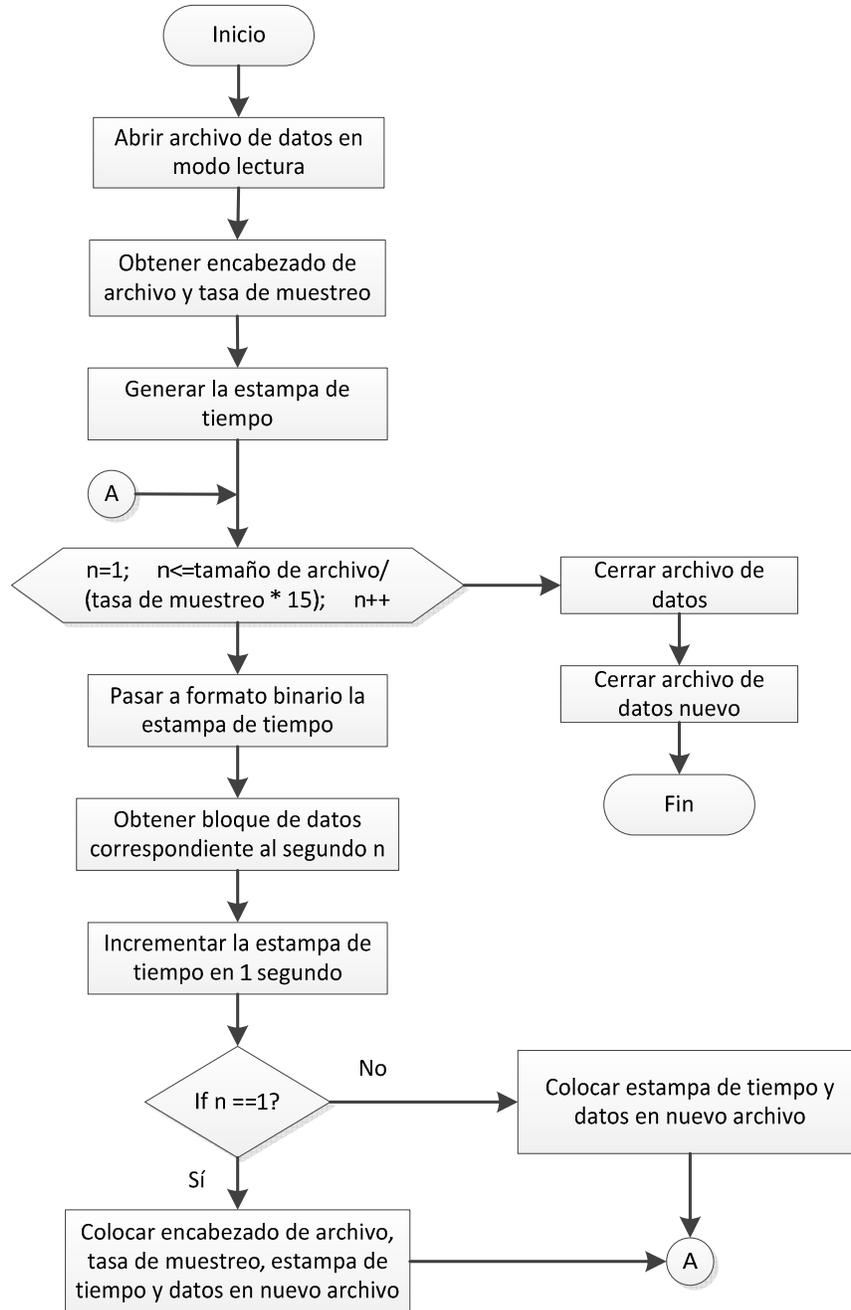


Figura 3.63. Algoritmo para etiquetar cada bloque de datos en un nuevo archivo.

En la figura 3.64 se muestra el diagrama de bloques mediante el cual se realiza la tarea de copiar los datos de un archivo y escribirlos dentro de otro, con su respectiva etiqueta de tiempo.

El número de iteraciones del ciclo *for*, es un control que nos permite copiar todos los datos del archivo original y escribirlos a un nuevo archivo, sin perder dato alguno. Este número está definido por el tamaño del archivo, la tasa de muestreo y el número de *bytes*

que le corresponde a una adquisición de datos de los tres canales (15 bytes), así obtenemos la ecuación 3.15, que nos permite calcular el número de iteraciones del ciclo *for*.

$$N = \frac{\text{Tamaño del archivo}}{\text{Frecuencia de muestreo} * 15} \quad (3.15)$$

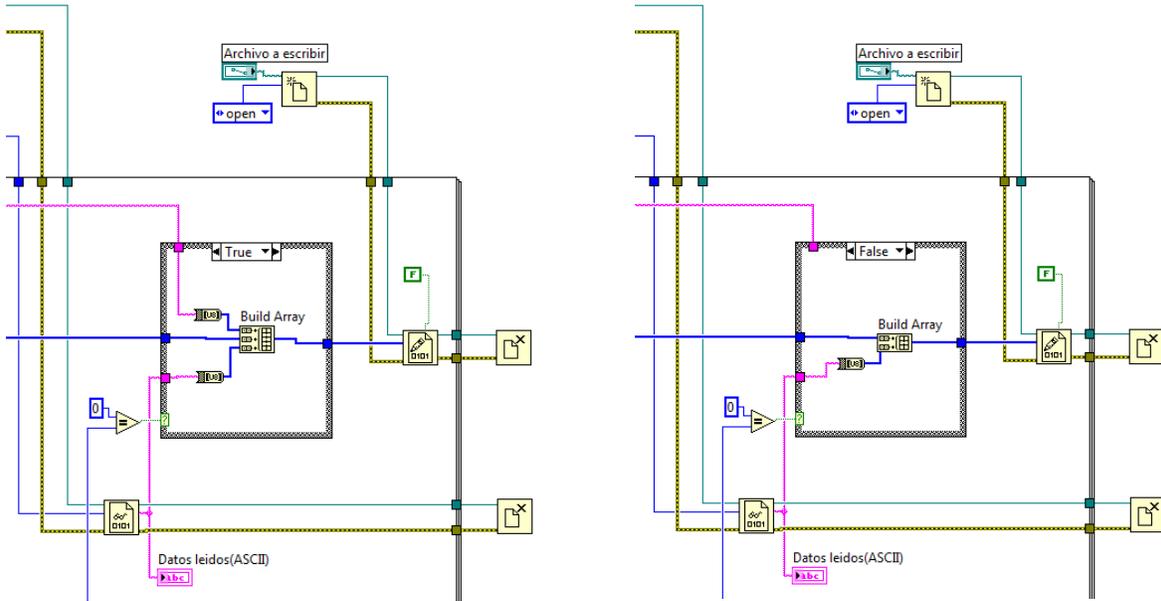


Figura 3.64. VI para copiar un bloque de datos de un archivo a otro.

En la figura 3.65 se muestra el panel frontal del programa que engloba todas las tareas: convertir la estampa de tiempo a formato binario, de copiar datos de un archivo y escribirlos a otro con su respectiva etiqueta de tiempo.

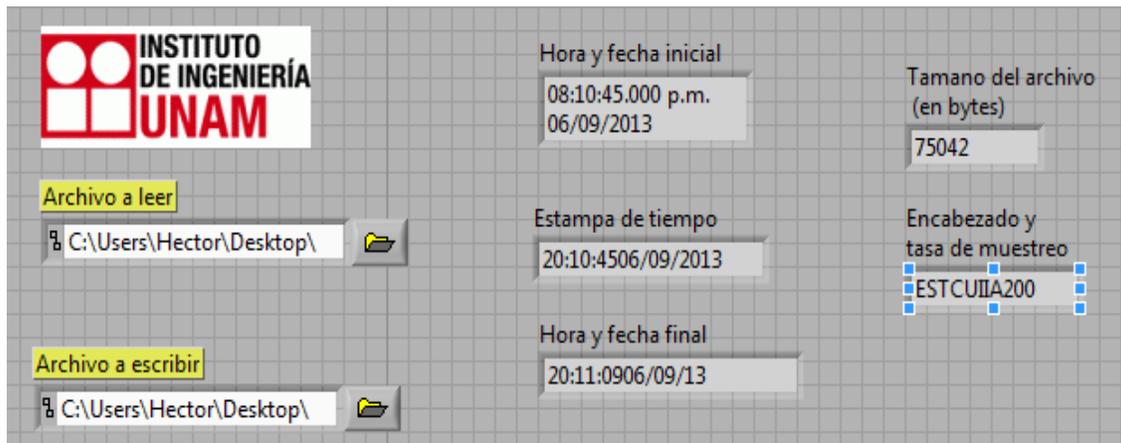


Figura 3.65. Interfaz para copiar los datos de un archivo a otro.

En la figura anterior podemos observar los datos que se extrajeron de un archivo al ejecutar el programa, entre ellos podemos mencionar el tamaño del archivo, encabezado y tasa de muestreo, hora y fecha de inicio de adquisición, estampa de tiempo y finalmente la hora y fecha final, que tendrá el último bloque de datos.

La figura 3.66 muestra la estructura del nuevo archivo, en el que los datos ya cuentan con una etiqueta de tiempo cada segundo, podemos observar que toda la información se encuentra almacenada en formato binario, excepto el encabezado de archivo y la tasa de muestreo como se había descrito anteriormente.

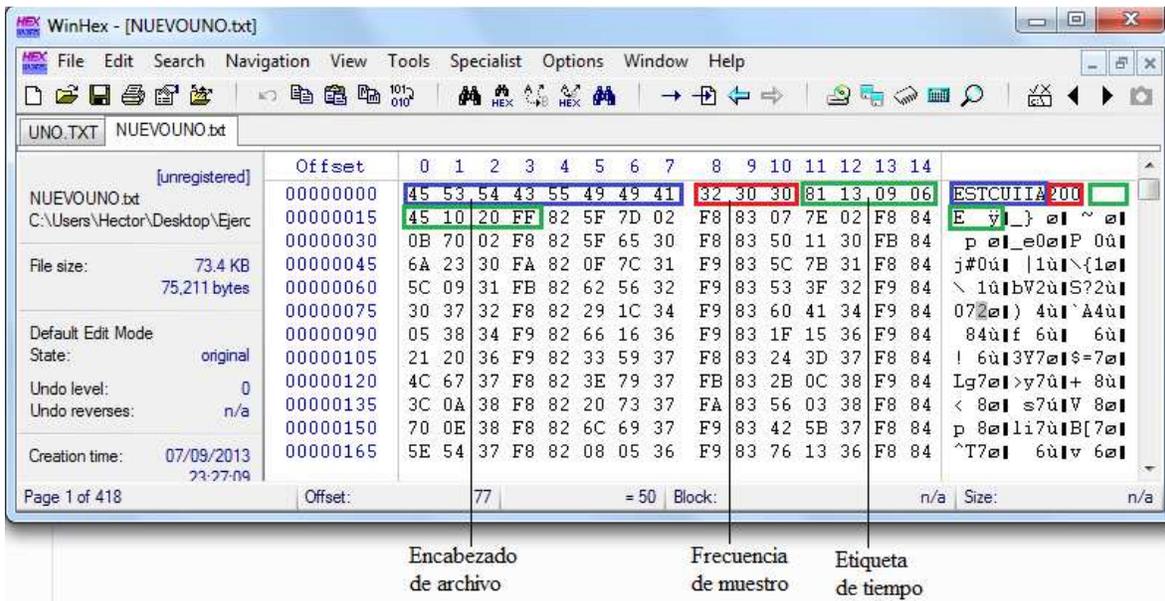


Figura 3.66. Estructura del nuevo archivo de datos.

En este capítulo se ha descrito de manera general el diseño y desarrollo del sistema DIGI. Abarcando el hardware que lo compone, así como también, el software que permite su funcionamiento. En el siguiente capítulo se presentarán las pruebas que se le realizaron al sistema desarrollado, con la finalidad de garantizar su funcionamiento.

Capítulo IV

Pruebas realizadas al sistema “DIGI”

En este capítulo se presentarán las pruebas que fueron realizadas a cada uno de los módulos que integran al sistema DIGI, posteriormente se presentarán las pruebas realizadas al sistema en general.

4.1. Integración del sistema en PCB

Como ya se mencionó en la sección 3.2.9, al realizar pruebas del sistema en tarjetas *protoboard*, y verificar que el sistema no requería más componentes para su funcionamiento, se diseñó la PCB del sistema. Una vez que fue diseñada la tarjeta del sistema DIGI, ésta se revisó cuidadosamente con la finalidad de garantizar que el ruteado del circuito no tuviera ningún error, así como también, se verificó que el tipo de encapsulado de los componentes correspondiera con los componentes a comprar.

En las figuras 4.1 y 4.2 se muestran las capas superior e inferior, respectivamente, de la PCB fabricada, antes de proceder al ensamblado de la misma.

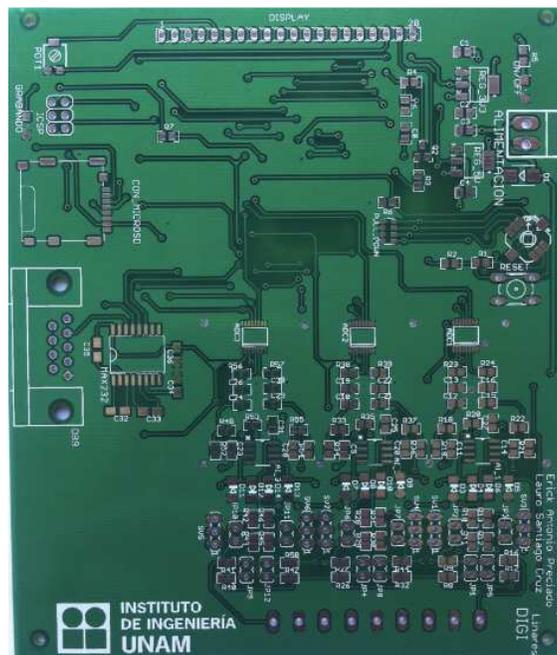


Figura 4.1. Capa superior de la PCB del sistema.

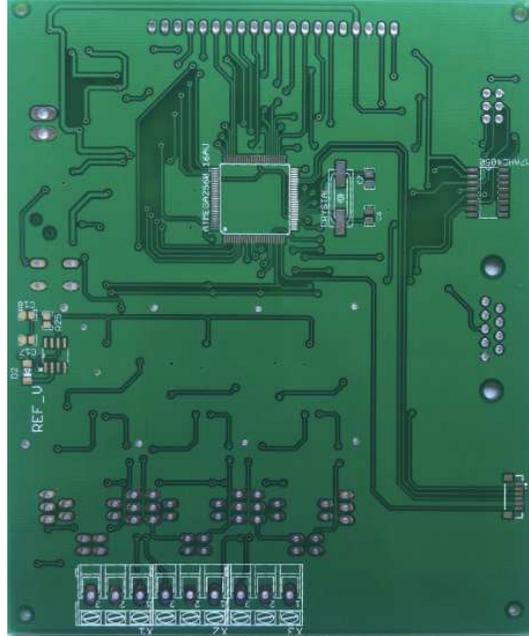


Figura 4.2. Capa inferior de la PCB del sistema.

El ensamblado del sistema se llevó a cabo en las instalaciones del Instituto de Ingeniería, resaltando que todos y cada uno de los componentes fueron soldados con las herramientas disponibles en ese momento. En la figura 4.3 se muestra la PCB con algunos de los componentes soldados. Así mismo, en la figura 4.4 se muestra la PCB ensamblada en su totalidad.

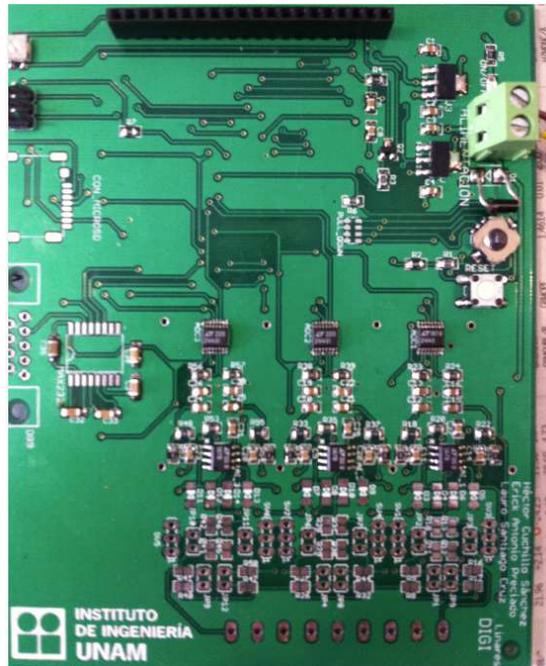


Figura 4.3. Ensamblado de componentes en la PCB.

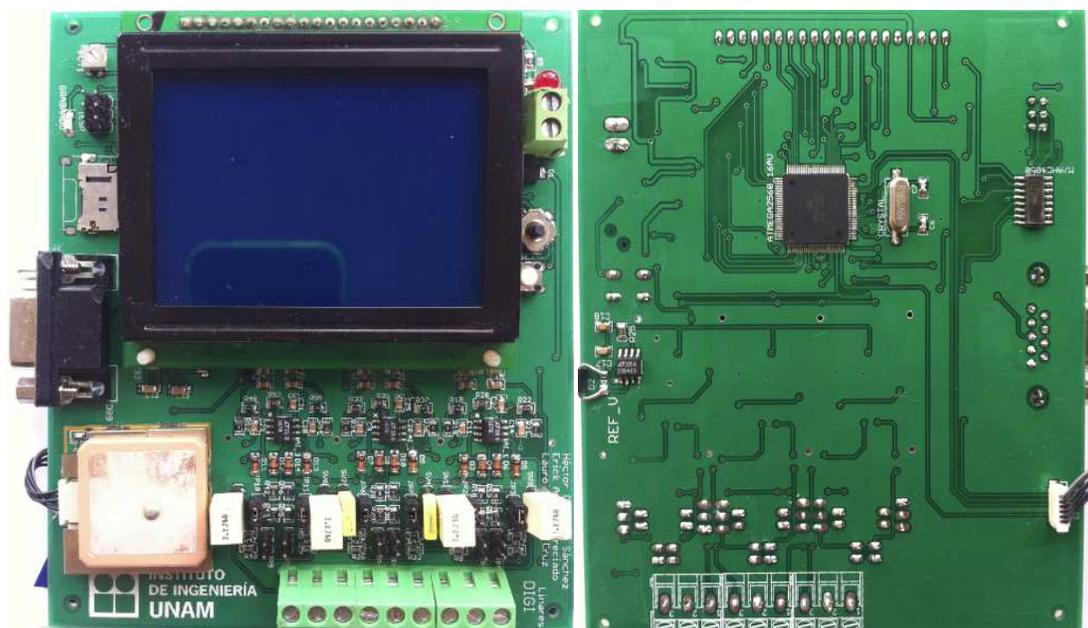


Figura 4.4. Prototipo final del sistema DIGI.

Una vez que se terminaron de soldar todos los elementos que componen al DIGI, se procedió a verificar que los componentes estuvieran bien soldados y que no hubiera cortos circuitos entre sus terminales. Después de realizar dicha verificación, se procedió a energizar la tarjeta, mediante una fuente con un valor de 12V, con la finalidad de medir la tensión en las terminales de alimentación de cada uno de los elementos que requieren ser energizados para su funcionamiento. También se midió la corriente que el sistema DIGI demanda a la fuente de poder, la cual fue de 115 mA, aproximadamente. En la figura 4.5 se muestra dicha medición.



Figura 4.5. Corriente demanda por el sistema DIGI.

Ya que se verificó que el sistema no tenía cortos circuitos y que los componentes del circuito estaban siendo energizados con 5V, se procedió a programar en la memoria flash del microcontrolador el *firmware* diseñado para la operación del sistema, esto con la

Pruebas realizadas al sistema “DIGI”

finalidad de poder realizar pruebas a cada uno de los módulos que integran al sistema y de esta manera garantizar su funcionamiento.

Para cargar al microcontrolador el *firmware* fue necesario utilizar el entorno de desarrollo “*Atmel Studio*”, debido a que el programador utilizado para este fin es el AVRISP mkII, figura 4.6.



Figura 4.6. Programador AVRISP mkII.

El proceso para cargar el *firmware* al microcontrolador es muy simple, ya que en primera instancia se tiene que energizar el DIGI con 12V, posteriormente se debe conectar el programador a una PC a través del puerto USB; así mismo, el programador se deberá conectar al DIGI mediante las terminales asignadas para este fin (terminales de programación ISP). Una vez que se ha realizado la conexión es importante mencionar que el programador contiene un LED, el cual se encenderá en color verde si la conexión se ha realizado adecuadamente, en caso contrario éste se encenderá en color rojo. En la figura 4.7 se muestra la conexión entre el programador, la PC y el sistema DIGI.



Figura 4.7. Conexión entre el programador, la PC y el DIGI.

Pruebas realizadas al sistema "DIGI"

Una vez que la conexión entre el programador y el DIGI es adecuada, en el entorno de desarrollo de *Atmel studio*, se tienen que comprobar cinco datos importantes en el menú *Interface settings*, los cuales son: el programador a utilizar, la matrícula del microcontrolador a programar, la interfaz para programar (en nuestro caso ISP), el identificador del microcontrolador (éste es un número en hexadecimal único para cada dispositivo), y el voltaje con el cual está siendo energizado el microcontrolador. Además, en este mismo menú se tiene que fijar la velocidad del reloj de la interfaz ISP.

En la figura 4.8 se muestra la pantalla del entorno de desarrollo de *Atmel studio* del menú *Interface settings*, en donde se muestran los datos a verificar antes de programar al microcontrolador.

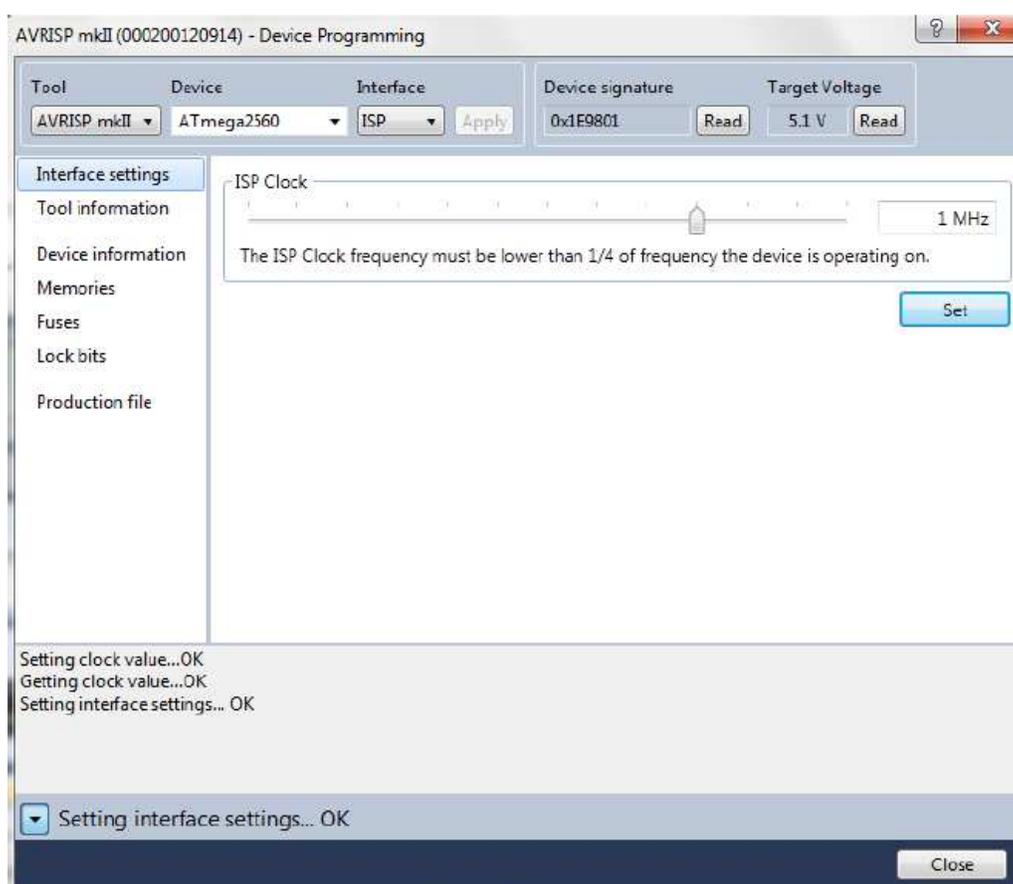


Figura 4.8. Características a verificar antes de programar al microcontrolador.

De la figura 4.8 se puede observar que en el lado izquierdo existen otros menús, tales como: *Tool information*, *Device information*, *Memories*, *Fuses*, *Look bits* y *Production file*. Por lo que para cargar el *firmware* al microcontrolador, se tiene que acceder a la opción *Memories*, la cual desplegará las opciones que se muestran en la figura 4.9.

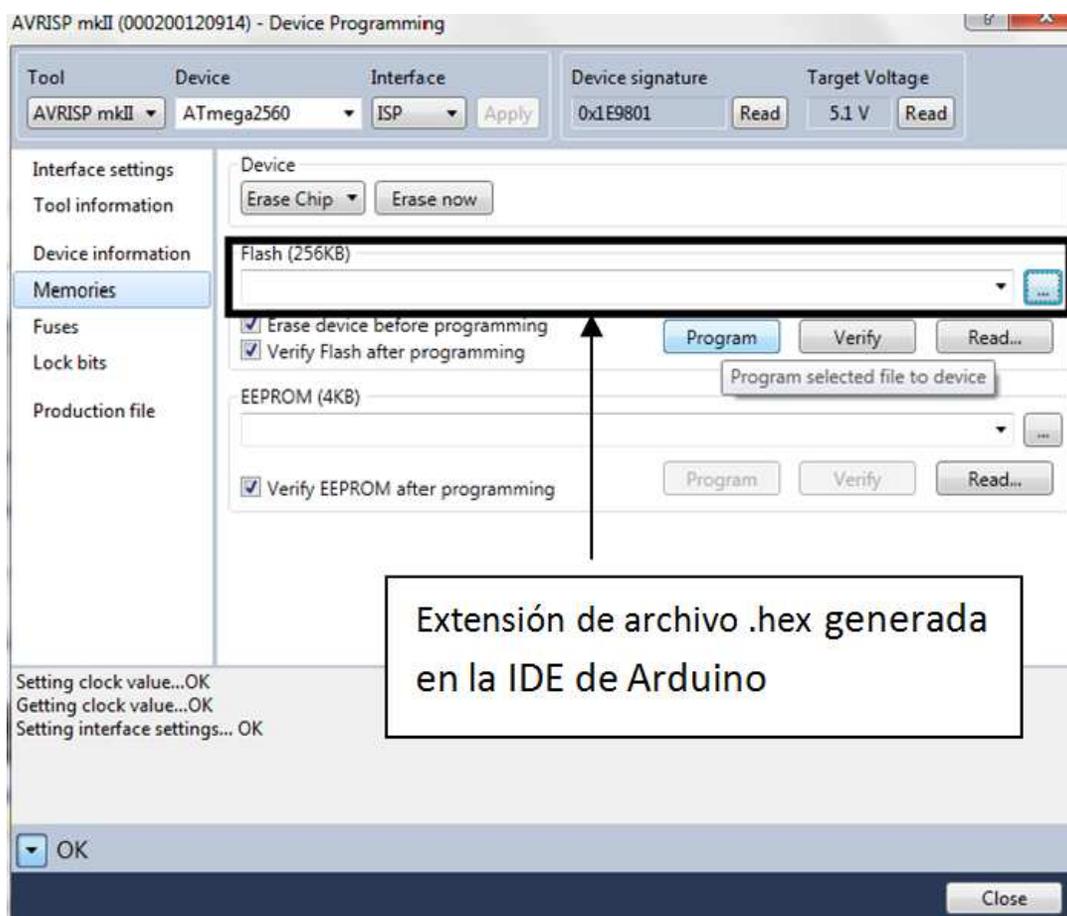


Figura 4.9. Opciones del menú “memories” para programar al microcontrolador.

De la figura 4.9 se puede observar que se ha indicado en un rectángulo, el espacio en donde se debe ingresar la extensión en donde la PC ha almacenado el archivo .hex. El cual es generado al compilar el código desarrollado en la IDE de Arduino, como se ha mencionado en la sección 3.3. Una vez que se ha escrito la extensión del archivo .hex, bastará con seleccionar la opción *Program* para concluir la programación de la memoria *flash* del microcontrolador.

4.2. Pruebas realizadas al módulo de acondicionamiento de señal

El módulo de acondicionamiento de señal quedó integrado por tres canales en donde, cada canal está compuesto por un filtro anti *aliasing*, un limitador de nivel a la entrada del amplificador, y un amplificador de instrumentación. En la figura 4.10 se ilustra el módulo de acondicionamiento de señal en la PCB.

Con base en lo anterior y para verificar el funcionamiento del circuito de acondicionamiento de señal, en la entrada analógica de cada canal se introdujo una señal

Pruebas realizadas al sistema "DIGI"

sinusoidal, proveniente de un generador de funciones. La amplitud y la frecuencia de dicha señal fueron fijadas con un valor de 500mVpp y 2 Hz respectivamente.

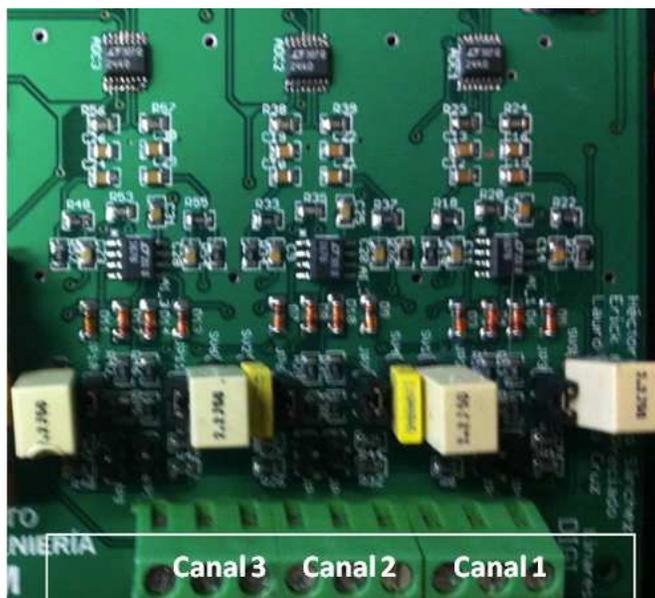


Figura 4.10. Módulo de acondicionamiento de señal en PCB.

Con base en las características eléctricas de la señal sinusoidal de prueba, y recordando que el amplificador fue diseñado con una ganancia de 12dB, la señal a la salida del amplificador fue medida con una amplitud del orden de 2Vpp. En la figura 4.11 se muestra la prueba realizada.

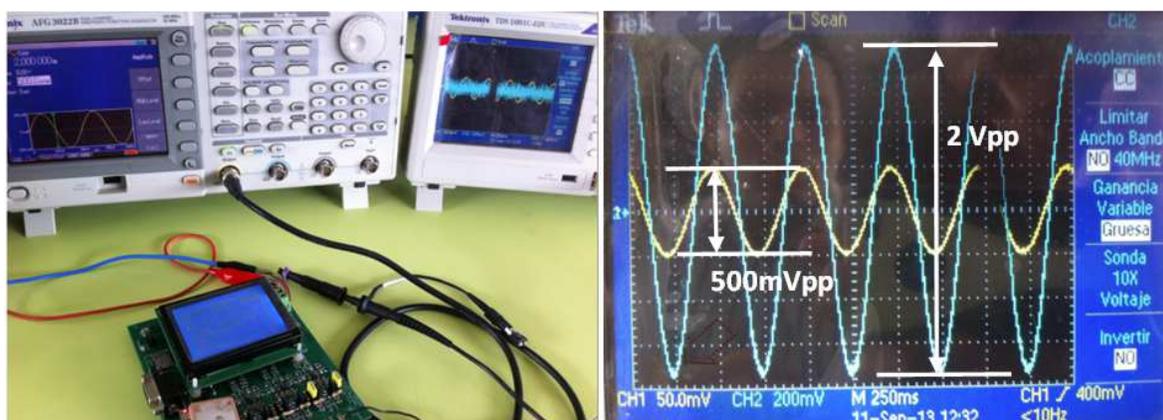


Figura 4.11. Prueba realizada al amplificador.

Cabe comentar que el amplificador del sistema DIGI está diseñado para cubrir el rango de frecuencias en el que se encuentra el fenómeno sísmico 0.001Hz – 1000Hz, por lo que para comprobar que el amplificador responde a ese ancho de banda, se realizó un barrido de frecuencias, con la finalidad de observar que la ganancia del amplificador se

Pruebas realizadas al sistema "DIGI"

atenuara -3dB para altas frecuencias. Se obtuvo como resultado que el amplificador mantiene la ganancia de 12dB hasta una frecuencia medida con el osciloscopio igual a 1.25 kHz, en donde, teóricamente fue diseñado para una frecuencia de 1 kHz. Sin embargo, para frecuencias menores a 1 Hz sólo se pudo probar hasta 0.01 Hz, debido a que el osciloscopio utilizado no tenía capacidad para realizar mediciones a frecuencias más bajas que 0.1Hz. Además de que al bajar la frecuencia en el generador de funciones ya se introducía mucho ruido, trayendo como consecuencia que la señal sinusoidal se perdiera.

Con base en lo anterior se probó y verificó que el amplificador funcionará de acuerdo los parámetros con que fue diseñado.

Para verificar el funcionamiento del circuito limitador de la amplitud de voltaje de la señal de entrada en los amplificadores, se varió la amplitud de la señal de prueba y con ayuda del osciloscopio se midió la señal a la entrada del amplificador. En la figura 4.12 se muestran el oscilograma resultado de la prueba.

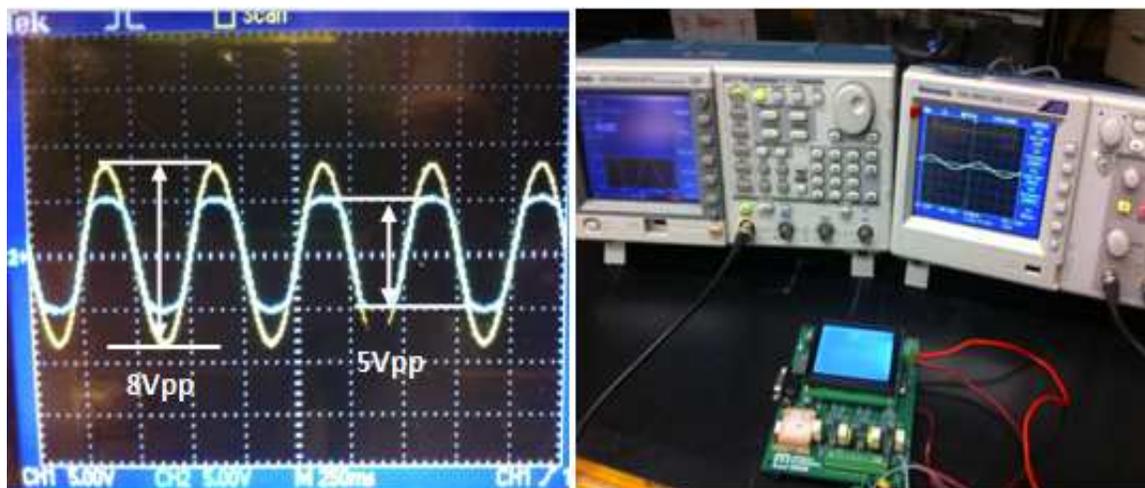


Figura 4.12. Prueba realizada a los limitadores de nivel.

Como se observa en la figura 4.12, la señal medida en la entrada del amplificador es recortada siempre y cuando tenga una tensión mayor a 5Vpp. Por lo que con lo anterior se estará protegiendo la entrada del convertidor A/D, ya que la magnitud de la señal de entrada en el amplificador siempre será una señal de 5Vpp como máximo.

Finalmente se probó y verificó la respuesta de los filtros pasa bajas que se encuentran a la entrada del sistema. Por lo que para probar el filtro pasa bajas para sensores pasivos, fue necesario cerrar los *jumpers* J1 y J3 y abrir J2 y J4 como se mencionó en la tabla 3.4 del capítulo 3. Aunado a esto se introdujo una señal sinusoidal de 1Vpp con la finalidad de realizar un barrido de frecuencias con el generador de funciones, y cuando se midiera con el osciloscopio 700mVpp a la salida del filtro, se tendría de esta manera la frecuencia de corte del filtro.

Pruebas realizadas al sistema "DIGI"

Cabe destacar que el filtro pasa bajas para sensores pasivos, se diseñó para una frecuencia de corte igual a 9Hz, por lo que al realizar la prueba se obtuvo que el filtro respondió a una frecuencia de 9.67 Hz. En la figura 4.13 se muestra el oscilograma de la respuesta del filtro pasa bajas para sensores pasivos.

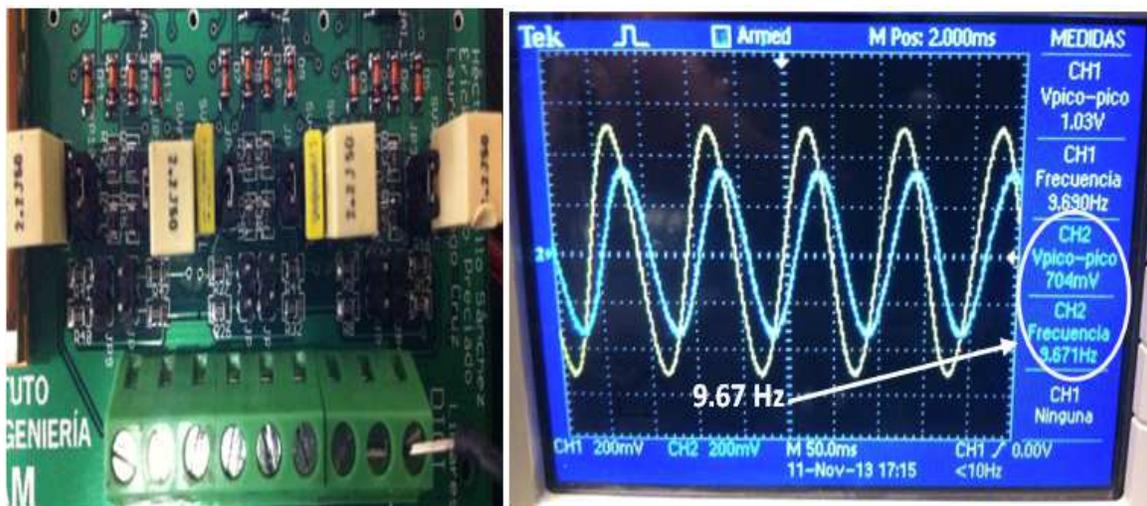


Figura 4.13. Respuesta del filtro pasa bajas para sensores pasivos.

Esta misma prueba se realizó para probar la respuesta en frecuencia del filtro diseñado para sensores activos. Por lo que para lograr esta acción, se abrieron los *jumpers* J1 y J3 y se cerraron los *jumpers* J2 y J4. El filtro fue diseñado para una frecuencia de corte de 18Hz, por lo que después de realizar la prueba se obtuvo que el filtro respondió a una frecuencia de 19.23 Hz. En la figura 4.14 se muestra el oscilograma de la respuesta del filtro pasa bajas para sensores activos.

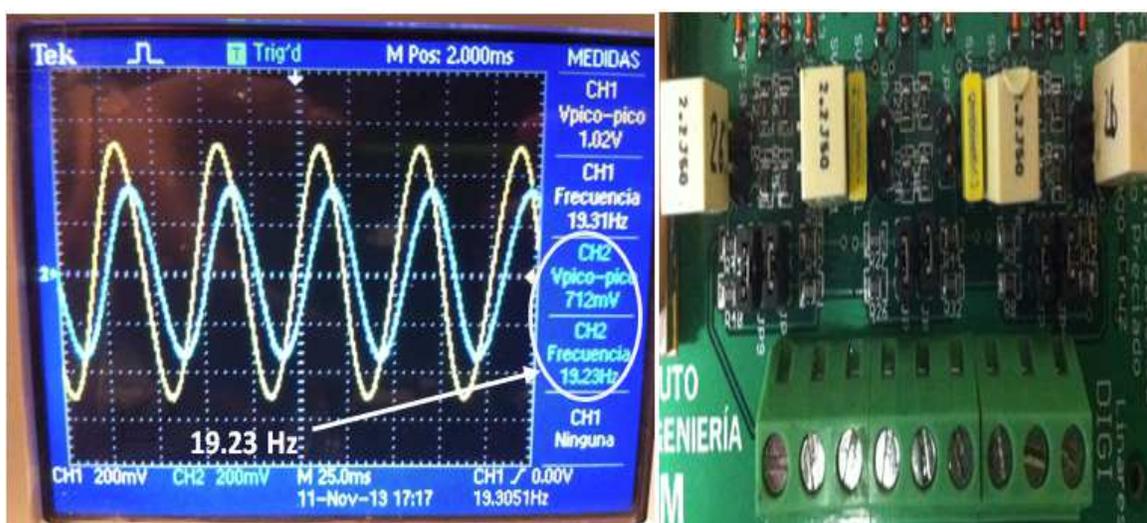


Figura 4.14. Respuesta del filtro pasa bajas para sensores activos.

Pruebas realizadas al sistema “DIGI”

Con base en lo anterior se garantiza que el valor del capacitor empleado para el filtro pasa bajas para los sensores pasivos y activos brinda una respuesta del filtro muy cercana al valor teórico diseñado para estos filtros.

Es importante mencionar que el filtro pasa bajas, sirve para limitar la banda de la señal de entrada, ya que como se mencionó en la sección 2.3.3 el convertidor A/D con arquitectura sigma – delta, que es el utilizado en el sistema DIGI, requiere a la entrada de la señal analógica un filtro anti *aliasing* pasivo de primer orden. Por lo que dependiendo de las necesidades del usuario, éste puede modificar la respuesta del filtro para evitar el fenómeno de *aliasing* en la digitalización de sus datos, esto se debe a que existen otros geófonos con otras frecuencias de operación diferentes a 4.5 Hz, que fue la frecuencia de los sensores utilizados para el diseño del sistema.

En la tabla 4.1 se muestra la relación entre diferentes valores de capacitores y la respuesta del filtro anti *aliasing* a considerar para evitar el traslape de las señales cuando se utilicen sensores diferentes a los ya establecidos.

Valor del Capacitor	Frecuencia de corte de filtro “anti <i>aliasing</i> ”	Tasa de muestro mínima
0.22 μF	88 Hz	200 muestras por segundo
0.33 μF	58 Hz	200 muestras por segundo
0.47 μF	41 Hz	100 muestras por segundo
0.68 μF	28 Hz	100 muestras por segundo
1 μF	19 Hz	50 muestras por segundo
2.2 μF	8.8 Hz	20 muestras por segundo

Tabla 4.1. Valores de capacitores para filtro antialiasing.

De esta manera se garantiza el funcionamiento de todo el circuito de acondicionamiento de acuerdo a las características con las que fue diseñado.

4.3. Pruebas realizadas al módulo de adquisición de datos del GPS

Para realizar las pruebas al módulo de adquisición de datos del GPS, fue necesario verificar que el receptor GPS no estuviera dañado. Esto se logró por medio de un programa que se encuentra disponible en el sitio *web* del fabricante del receptor GPS. Para ello se conectó el GPS a una PC por medio de un cable USB – RS-232, sin embargo, para realizar dicha conexión se utilizó un circuito integrado para hacer compatibles los niveles de voltaje entre el puerto serial RS-232 y los niveles TTL que maneja el receptor GPS. Una vez que se realizó dicha conexión entre el receptor GPS y la PC, se energizó el receptor con una fuente de 5V. De esta manera, el proceso para verificar el funcionamiento del receptor, se redujo

Pruebas realizadas al sistema "DIGI"

en abrir desde la PC el programa ya mencionado y seleccionar tanto el puerto en que se conectó el dispositivo a la PC, como el baudaje con el que opera el dispositivo (4800 bps).

El programa utilizado para la verificación del funcionamiento del receptor GPS, de manera general, despliega los datos referentes a la posición, el tiempo UTC, muestra un esquema en donde se ilustra gráficamente el número de satélites que en ese momento detecta el receptor GPS, además de que muestra las cadenas del protocolo NMEA 0183 soportadas por el dispositivo.

En la figura 4.15 (a) se muestra una captura de pantalla del programa utilizado cuando se conecta un receptor GPS que no está dañado. Por otra parte, en la figura 4.15 (b) se muestra una captura de pantalla del mismo programa pero con un receptor dañado. De las figuras mencionadas se puede concluir que el receptor a utilizar se encuentra libre de algún problema. Por lo que una vez verificado su funcionamiento, se procedió a realizar las pruebas en cuanto a la sincronización de datos para el sistema DIGI.

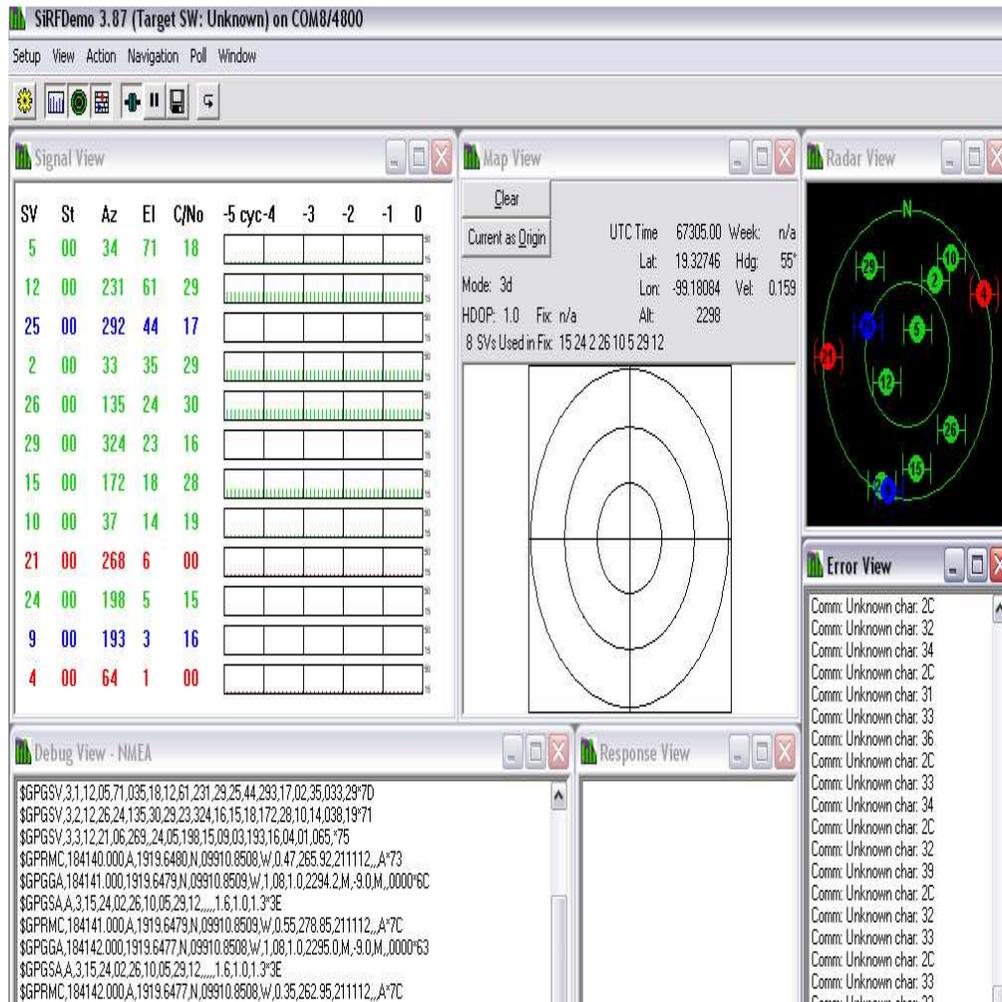


Figura 4.15 (a). Verificación de funcionamiento de GPS. (Continúa)

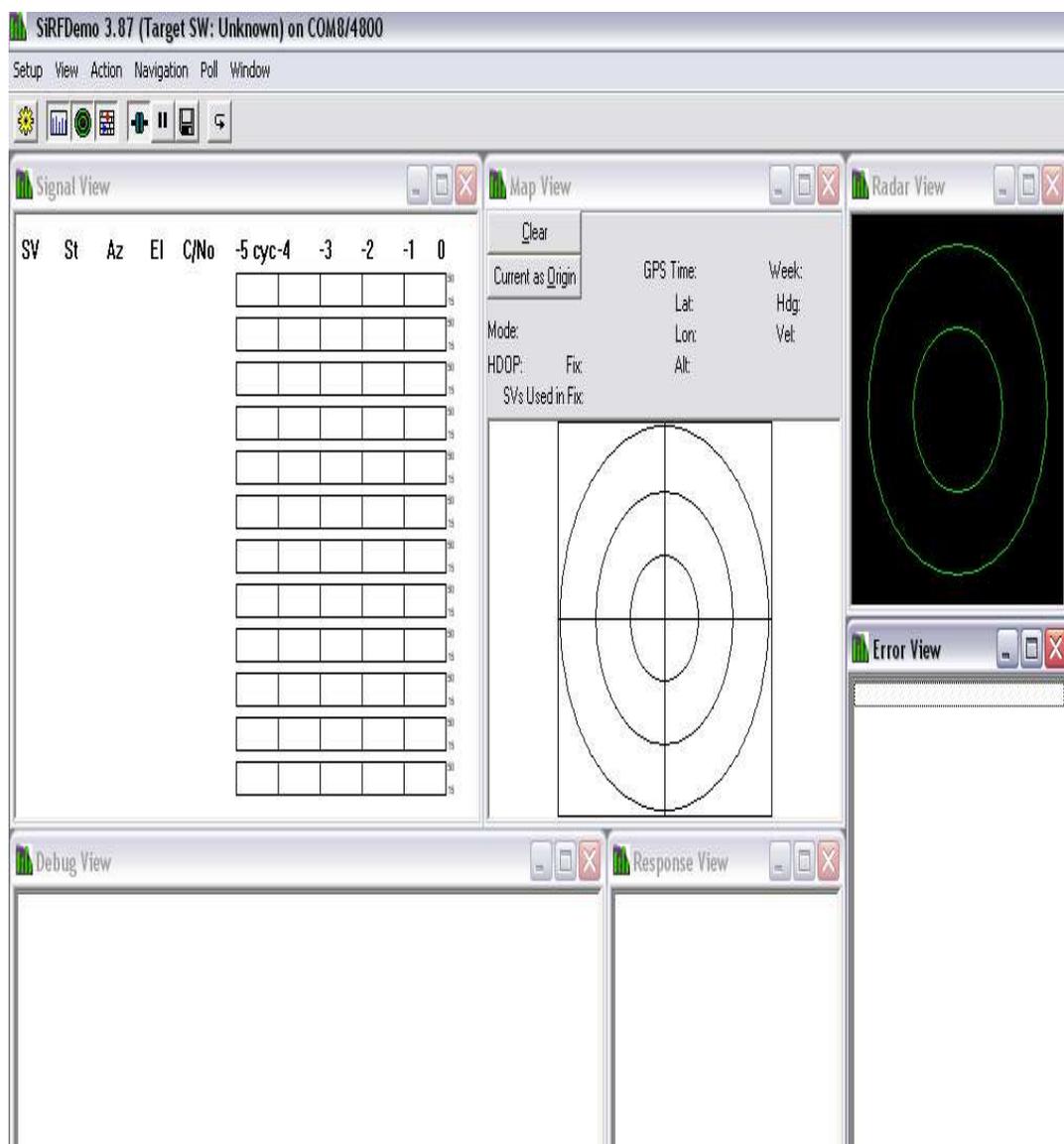


Figura 4.15 (b). Verificación de funcionamiento de GPS.

Dicho lo anterior y para verificar que los datos del paquete de tiempo obtenidos por el sistema DIGI son correctos, se compararon los datos obtenidos por el sistema DIGI con la fecha y hora proporcionados en el sitio *web* <http://www.worldtimeserver.com>, que proporciona la hora UTC exacta.

En la figura 4.16 se muestra una captura de pantalla del sitio *web* mencionado, con la hora UTC, en el mismo instante en que se le tomó una foto al sistema DIGI al adquirir los datos de hora y fecha.

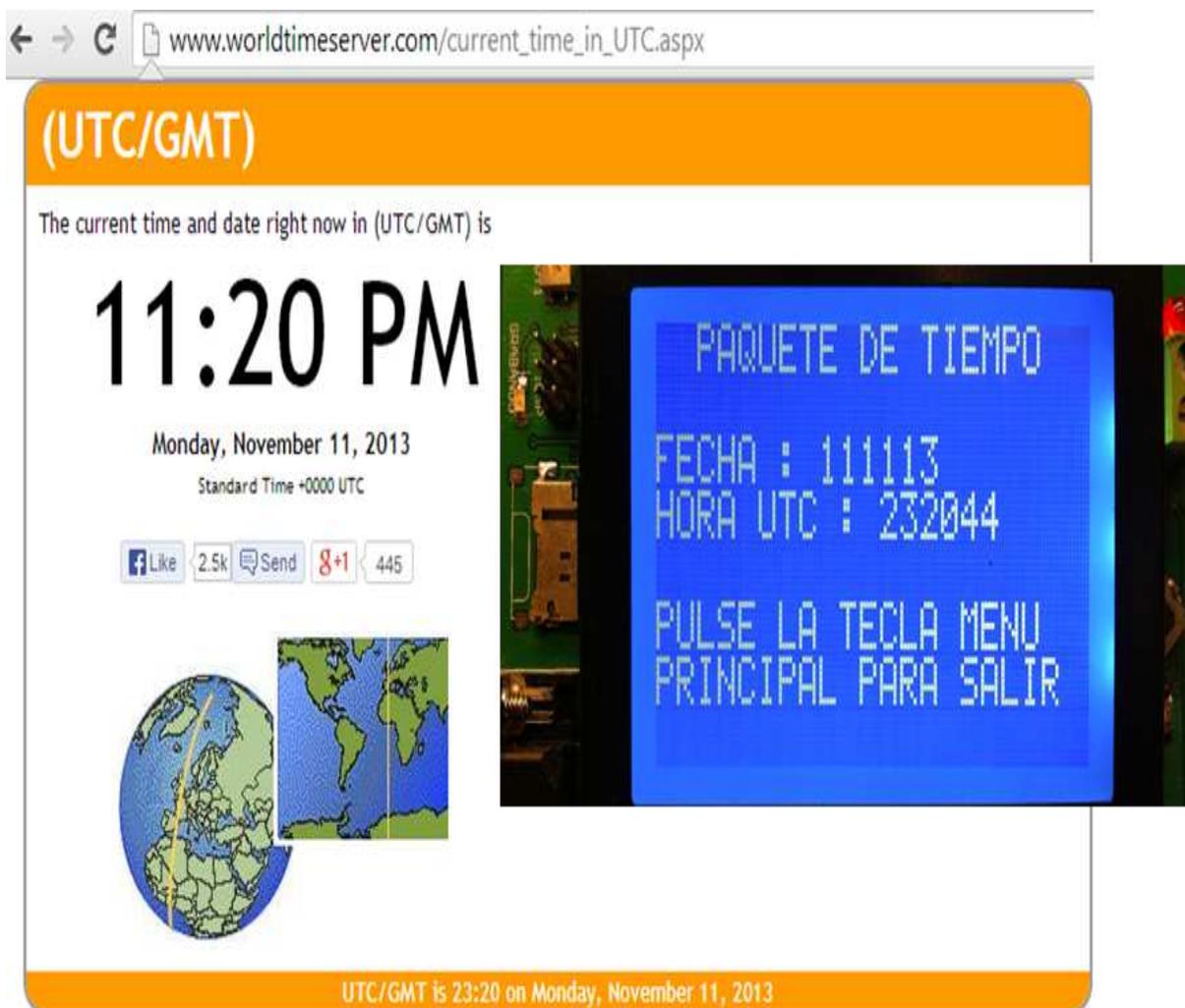


Figura 4.16. Comprobación de datos de fecha y hora.

4.4. Pruebas de adquisición y almacenamiento de datos

La prueba de adquisición y almacenamiento de datos tiene como objetivo comprobar que la información adquirida sea almacenada correctamente en los archivos dentro de la memoria micro SD. Se verificará que el tamaño de los archivos corresponda al periodo de registro previamente establecido en la prueba.

Para llevar a cabo la prueba mencionada, se configuró el sistema DIGI para realizar el registro durante 1 hora, a una tasa de muestreo de 200 muestras por segundo, la entrada a los tres canales fue una señal senoidal de 450 mVpp a 5 Hertz con un offset de 260 mV. Una vez concluido el periodo de registro de datos, se procedió a verificar el resultado de la prueba, comprobando la existencia y el tamaño del archivo. En la figura 4.17 observamos que el archivo registrado, llamado UNO, contiene 10,800,027 bytes.

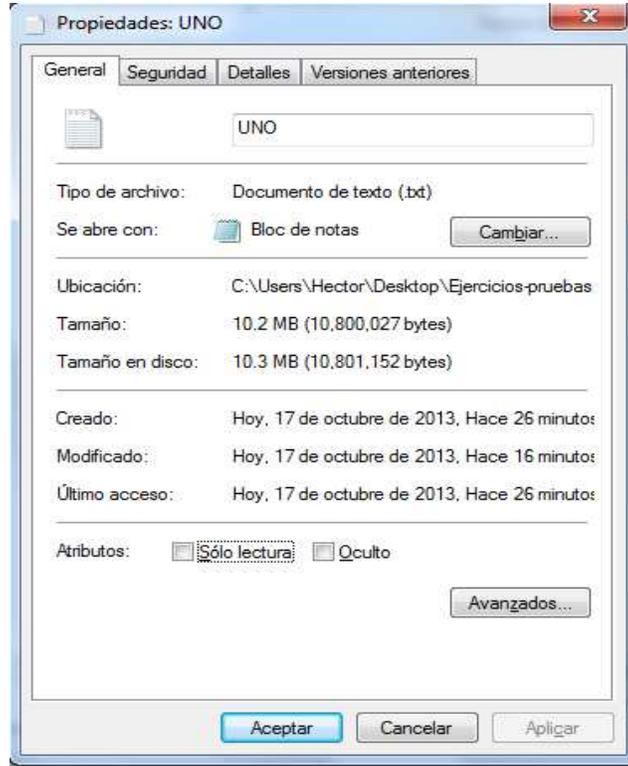


Figura 4.17. Archivo generado por el sistema DIGI para pruebas cortas.

La figura 4.18 muestra un extracto de datos del archivo UNO, con ayuda de un software llamado "WinHex", en ella podemos observar el tamaño del archivo en *bytes* (10,800,027), la base de tiempo obtenida del GPS (datos sombreados), y en general el contenido y propiedades del archivo.

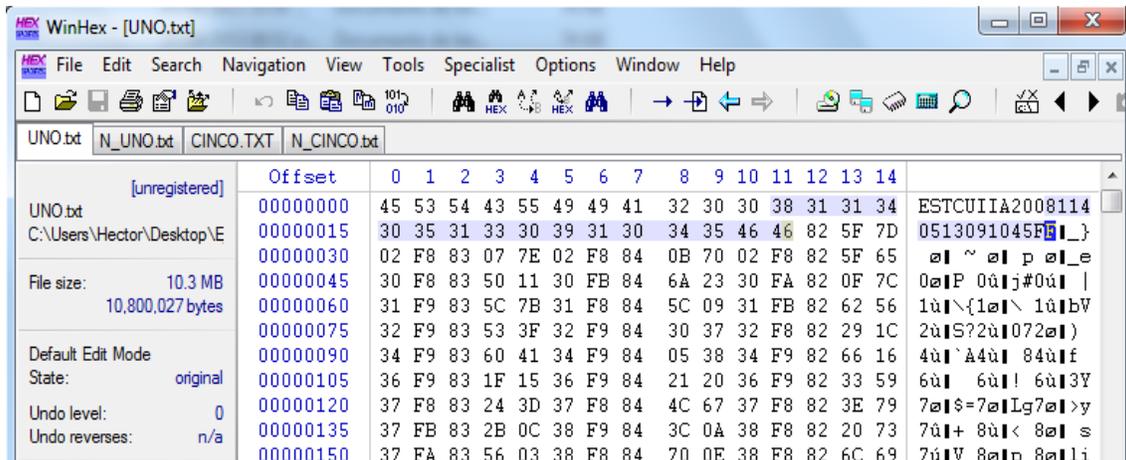


Figura 4.18. Extracto de datos del archivo generado por el sistema DIGI.

De acuerdo a las expresiones 3.13 y 3.14, el número de *bytes* que debe tener el archivo se puede calcular como:

$$\text{Tamaño} = ((200 * 1 * 3600) * 15) + 8 + 3 + 16 = 10,800,027 \text{ bytes}$$

Con lo cual se comprueba que la cantidad de *bytes* almacenados en el archivo (10,800,027) corresponde con el tamaño de *bytes* calculados, de acuerdo a los parámetros de la prueba.

4.5. Pruebas para verificar la estampa de tiempo

El propósito de esta prueba es verificar que la etiqueta o estampa de tiempo se introduzca correctamente en cada bloque de datos correspondiente a un segundo, dentro del nuevo archivo que alojará los datos almacenados por el sistema y los datos de la etiqueta de tiempo. También verificar que se respete la continuidad de la estampa de tiempo dentro del archivo conforme se van etiquetando más datos. Finalmente, comprobar que el tamaño del archivo corresponda con la cantidad de datos que se copiarán del archivo generado por el sistema DIGI, más los *bytes* necesarios para insertar la estampa de tiempo.

Como ya se mencionó anteriormente, la estampa o etiqueta de tiempo que se le introducirá a cada bloque de datos correspondiente a un segundo, será almacenada en formato binario, con la finalidad de reducir al mínimo el espacio de almacenamiento requerido, estas tareas se realizarán con ayuda de un programa desarrollado en LabVIEW.

La etiqueta de tiempo se encuentra en formato binario y tiene una longitud de 8 *bytes* (2 *bytes* para los *header* de inicio y fin, 3 *bytes* para la fecha y otros 3 *bytes* para la hora), con esto se puede obtener el número de *bytes* necesarios para etiquetar todos los datos del archivo, ecuación 4.1. Esto sin considerar en el tamaño de archivo los *bytes* empleados para almacenar el encabezado, la tasa de muestreo y el paquete de tiempo.

$$\# \text{ de bytes para la estampa de tiempo} = \frac{\text{Tamaño del archivo} - 27}{15 * \text{tasa de muestreo}} * 8 \quad (4.1)$$

En la expresión anterior:

- Tamaño del archivo se refiere al generado por el sistema DIGI
- **27** = Número de *bytes* usados para almacenar el encabezado de archivo, tasa de muestreo y paquete de tiempo.
- **15** = Número de *bytes* en cada escritura (3 canales * 5 *bytes* por muestra)
- **8** = Extensión en *bytes* de la estampa de tiempo en formato binario

Con base en el periodo de almacenamiento del archivo (1 hora) para 200 mps, se obtiene la cantidad de *bytes* necesarios para etiquetar todos los datos del archivo:

$$\text{Bytes estampa de tiempo} = \frac{10,800,027 - 27}{15 * 200} * 8 = 28,800 \text{ bytes}$$

Pruebas realizadas al sistema "DIGI"

Para conocer el tamaño del nuevo archivo, ecuación 4.2, se realiza la suma de la cantidad de *bytes* del archivo creado por el sistema DIGI, y la cantidad de *bytes* necesarios para colocar todas las etiquetas de tiempo, y así tener cada bloque de datos con su respectiva referencia de fecha y hora. A este resultado se le restan los 16 *bytes* que representan la base de tiempo original, obtenida del receptor GPS. Estos 16 *bytes* no serán copiados al nuevo archivo, ya que sólo fueron de utilidad para generar la etiqueta de tiempo.

$$\text{Archivo nuevo} = \text{Archivo original} + \text{Bytes estampa de tiempo} \quad (4.2)$$

En la expresión anterior:

- Archivo nuevo, se refiere a la cantidad de *bytes* que contendrá el nuevo archivo.
- Archivo original, se refiere a la cantidad de *bytes* almacenados en el archivo generado por el sistema DIGI.

A continuación se muestra el cálculo para obtener el tamaño del archivo final.

$$\text{Tamañ_archivo_nuevo} = 10,800,027 + 28,800 - 16 = 10,828,811 \text{ bytes}$$

La figura 4.19 muestra las propiedades del archivo final, el cual contiene los datos de la prueba con sus respectivas etiquetas de tiempo.

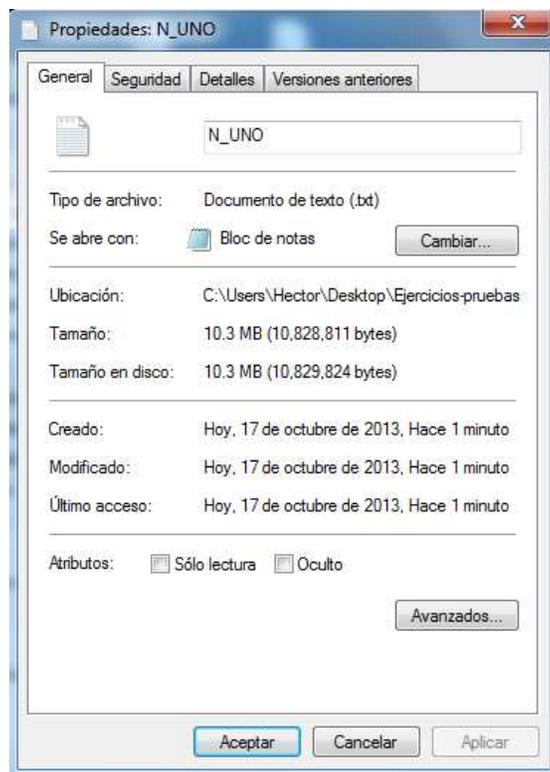


Figura 4.19. Tamaño del archivo final que ya contiene las etiquetas de tiempo.

Pruebas realizadas al sistema "DIGI"

La figura 4.20 muestra unos extractos de datos y las propiedades del archivo final para la prueba de un día. Los datos sombreados corresponden a la primera y segunda etiqueta de tiempo del archivo, en donde podemos observar que las etiquetas de tiempo difieren de 1 segundo, lo cual es correcto, ya que entre ellas existe un bloque de datos correspondiente a un segundo. Además, la primera etiqueta de tiempo corresponde con la base de tiempo obtenida del receptor GPS (figura 4.17), sólo difieren en el orden en el cual aparecen los datos de fecha y hora.

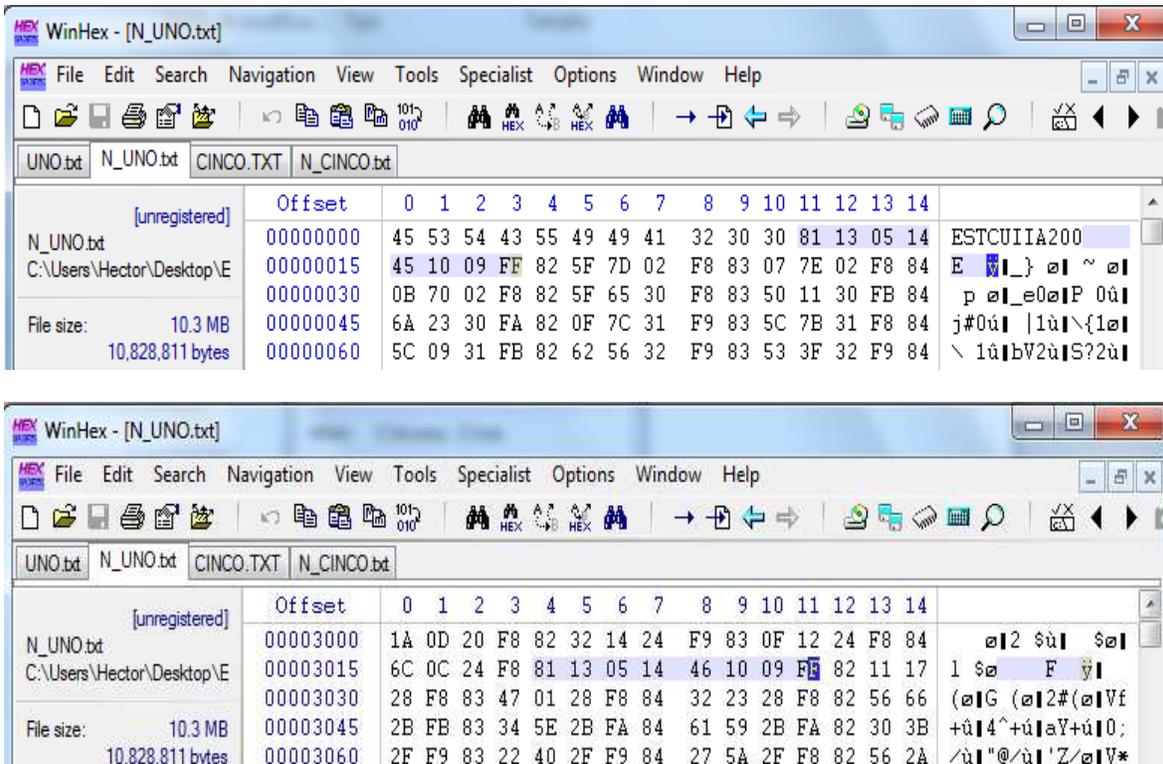


Figura 4.20. Archivos de prueba de un día.

Podemos observar que el tamaño del archivo final es exactamente igual a la cantidad obtenida a través del cálculo teórico, tomando en cuenta el tiempo de almacenamiento del archivo generado por el sistema DIGI. Con esto se comprueba que todos los datos de la prueba, así como sus etiquetas de tiempo, han sido almacenados satisfactoriamente dentro del archivo final.

4.6. Pruebas de reconstrucción de datos y señales

Esta prueba tiene la finalidad de verificar la veracidad de los datos almacenados por el sistema DIGI, con sus respectivas etiquetas de tiempo, insertadas con ayuda de un programa desarrollado en LabVIEW (apartados 4.4 y 4.5 respectivamente). Es decir,

Pruebas realizadas al sistema "DIGI"

graficar los datos registrados y comparar los parámetros de la señal reconstruida con los parámetros de la señal de entrada al sistema.

La figura 4.21 muestra las gráficas del archivo final que ya cuenta con la etiqueta de tiempo. Se eligió este archivo para comprobar que los datos fueron copiados satisfactoriamente del archivo originado por el sistema DIGI, con ayuda del programa de LabVIEW mencionado anteriormente.

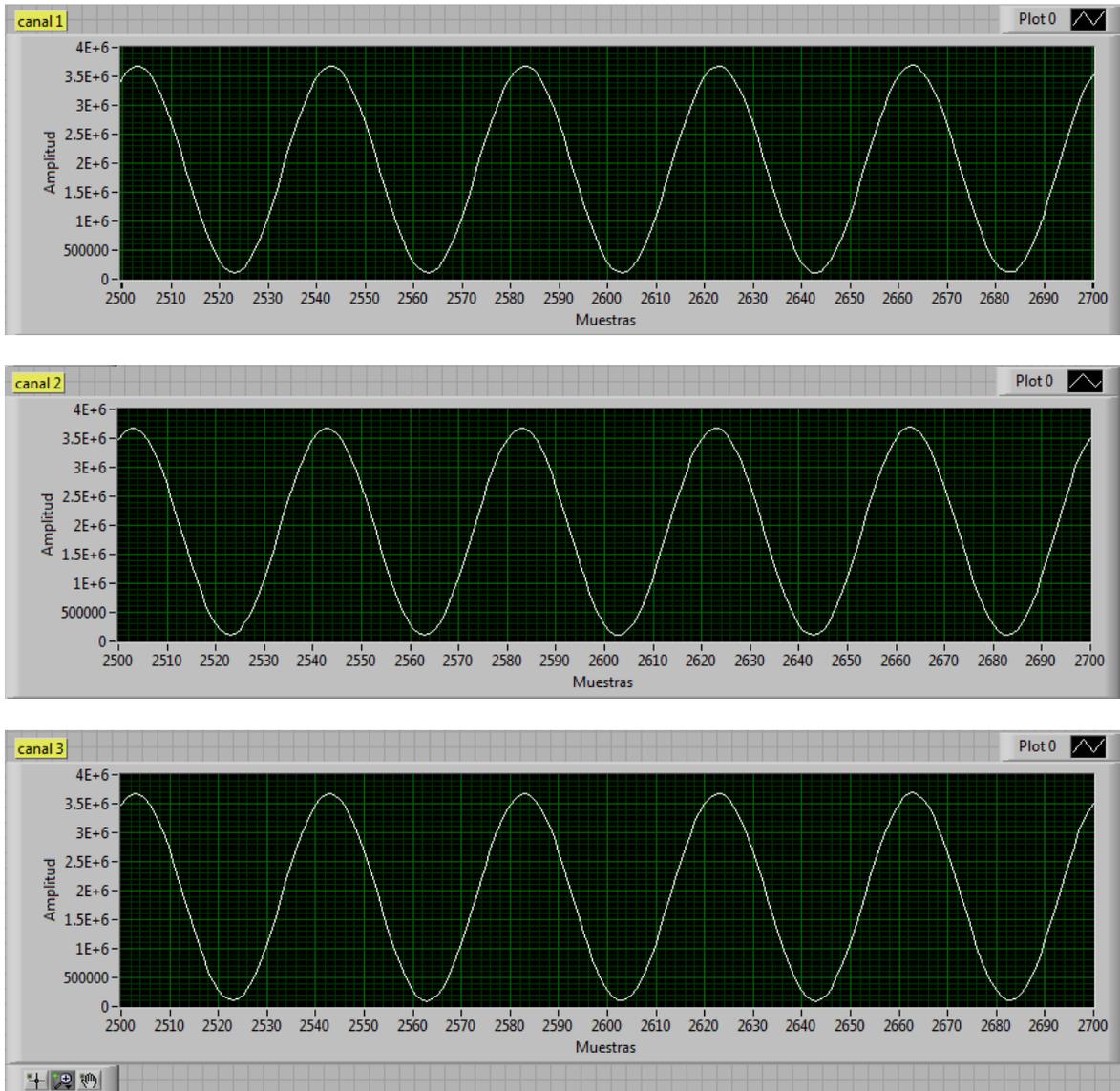


Figura 4.21. Gráficas del archivo de datos de prueba de un día.

En la figura anterior se observa que las gráficas generadas por los tres canales son prácticamente idénticas, además, se verifica que para esta prueba se tienen 5 ciclos de una señal sinusoidal en 200 muestras, lo cual es correcto, ya que la señal de entrada fue de 5 Hertz.

Pruebas realizadas al sistema "DIGI"

En cuanto a la amplitud, se observa que las gráficas registradas tienen 3.7×10^6 cuentas aproximadamente. Considerando que cada cuenta equivale a 119 nV, la amplitud de la señal registrada tiene un valor de 440 mVpp, que es congruente con el valor de la señal introducida al sistema DIGI, adicionalmente presenta un offset de 260 mVolts.

Para graficar la señal se hace uso de una aplicación desarrollada en LabVIEW, en la cual el usuario sólo tiene que especificar la ruta del archivo, la cantidad de bytes a leer y el *offset*, este último indica la posición del primer byte en el cual el programa comenzará a leer la cantidad de bytes especificados. Esta aplicación separa por canales la información obtenida del archivo, cada canal es codificado y finalmente graficado, como se observa en la figura anterior.

A partir del mismo archivo con el que se han estado desarrollando las pruebas, se obtiene los datos del paquete de tiempo (fecha y hora UTC), que se insertaron en el archivo de datos anteriormente, figura 4.22, esto con la finalidad de verificar que corresponden con la fecha de adquisición, y además, que sean continuos.



Encabezados de Tiempo en decimal: _							
	Año	Mes	Día	Segundo	Minuto	Hora	
0	81	13	5	14	45	10	9
0	81	13	5	14	46	10	9
	81	13	5	14	47	10	9
	81	13	5	14	48	10	9
	81	13	5	14	49	10	9
	81	13	5	14	50	10	9
	81	13	5	14	51	10	9
	81	13	5	14	52	10	9
	81	13	5	14	53	10	9
	81	13	5	14	54	10	9
	81	13	5	14	55	10	9

Figura 4.22. Encabezados de tiempo de archivo de un día.

De la figura 4.22 podemos observar que la primera etiqueta de tiempo, corresponde con la base de tiempo obtenida del receptor GPS (figura 4.17), sin embargo, no coincide con las propiedades de creación del archivo generado por el sistema DIGI (figura 4.17). Esto se debe a que la librería usada en el sistema "DIGI" para el almacenamiento de datos, tiene una declaración de variables que forman una fecha constante, la cual se establece como fecha de creación del archivo, y una vez que se accede al archivo por una PC, esta actualiza los atributos de fecha de creación y modificación del archivo.

También se realizaron pruebas de larga duración, en este caso se configuró el sistema DIGI para realizar el registro durante 7 días, a una tasa de muestreo de 200

Pruebas realizadas al sistema "DIGI"

muestras por segundo. En esta ocasión se conectaron unos sensores pasivos (geófonos) a los tres canales de adquisición. Una vez concluido el periodo de registro de datos, se procedió a verificar la existencia y tamaño de los archivos figura 4.23, en donde podemos observar que se crearon siete archivos con el mismo tamaño. Cada archivo contiene 259, 200, 019 *bytes* como lo muestra la ventana de propiedades del archivo.

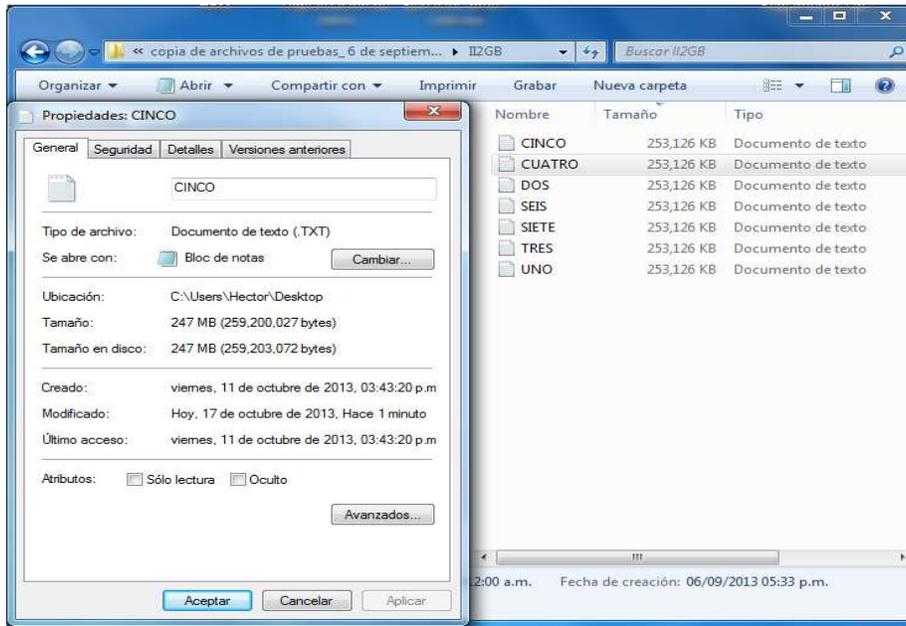


Figura 4.23. Archivos generados por el sistema DIGI para pruebas largas.

La figura 4.24 muestra un extracto de datos del archivo CINCO, en ella podemos observar el tamaño del archivo en *bytes* (259,200,027), la base de tiempo obtenida del GPS (datos sombreados), y en general el contenido y propiedades del archivo.

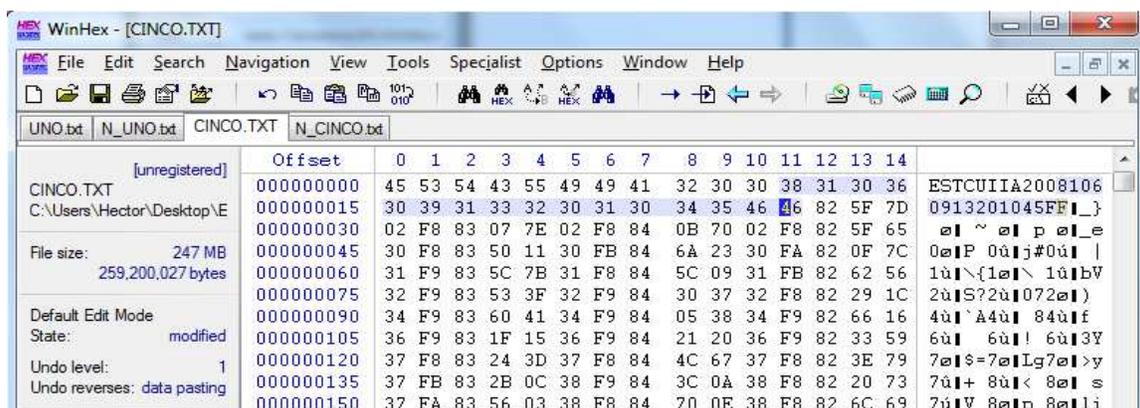


Figura 4.24. Datos del archivo generado por el DIGI para pruebas largas.

De acuerdo a las expresiones 3.13 y 3.14, el número de *bytes* que debe de contener cada archivo se puede calcular como:

Pruebas realizadas al sistema "DIGI"

$$\text{Tamaño} = ((200 * 1 * 86,400) * 15) + 8 + 3 + 16 = 259,200,027 \text{ bytes}$$

Con lo cual se comprueba que la cantidad de *bytes* almacenados en cada archivo (259, 200, 027) corresponde con el tamaño de *bytes* calculados, de acuerdo a los parámetros de la prueba.

Posteriormente se procedió a introducir la etiqueta de tiempo a los datos registrados por el sistema DIGI. De acuerdo a la ecuación 3.16 el número de *bytes* necesarios para insertar la etiqueta de tiempo es:

$$\text{Bytes estampa de tiempo} = \frac{259,200,000}{15 * 200} * 8 = 691,200 \text{ bytes}$$

Para conocer el tamaño del nuevo archivo, se realiza la suma de la cantidad bytes del archivo creado por el sistema DIGI, y la cantidad de bytes necesarios para colocar todas las etiquetas de tiempo. Con base en la ecuación 3.17, el tamaño del nuevo archivo es:

$$\text{Tamañ_archivo_nuevo} = 259,200,027 + 691,200 - 16 = 259,891,211 \text{ bytes}$$

La figura 4.25 muestra las propiedades del nuevo archivo, el cual contiene los datos de la prueba con sus respectivas etiquetas de tiempo.

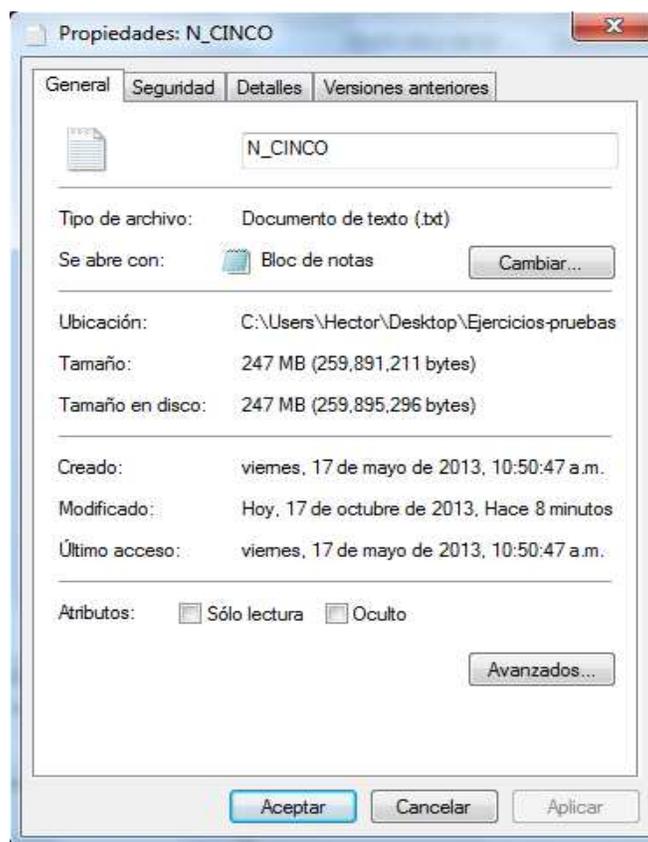


Figura 4.25. Tamaño del nuevo archivo.

Pruebas realizadas al sistema “DIGI”

La figura 4.26 muestra unos extractos de datos y las propiedades del archivo final para la prueba de siete días. Los datos sombreados corresponden a la primera y segunda etiqueta de tiempo del archivo, en donde podemos observar que las etiquetas de tiempo difieren de 1 segundo, lo cual es correcto, ya que entre ellas existe un bloque de datos correspondiente a un segundo. Además, la primera etiqueta de tiempo corresponde con la base de tiempo obtenida del receptor GPS (figura 4.24), sólo difieren en el orden en el cual aparecen los datos de fecha y hora.

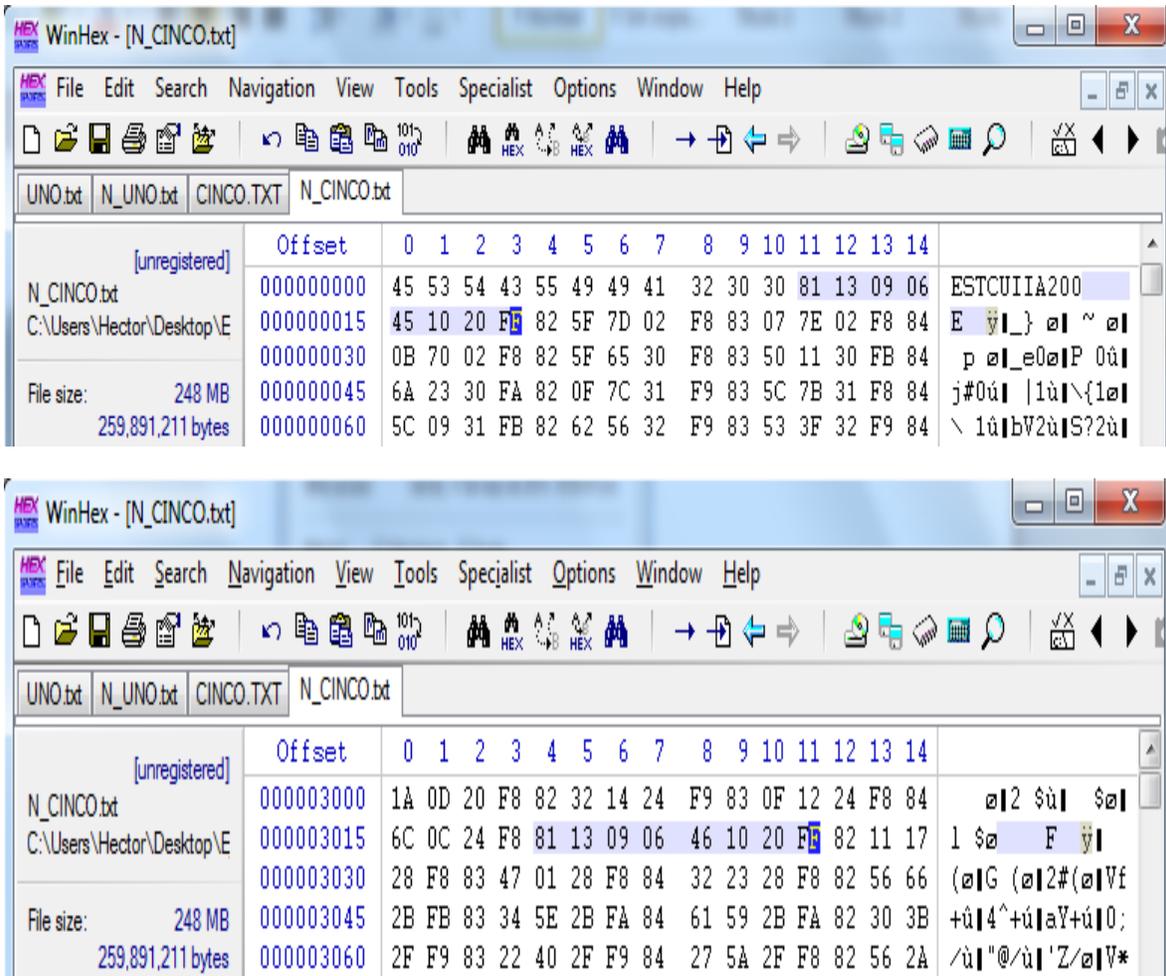


Figura 4.26. Archivos de prueba de siete días.

Podemos observar que el tamaño del archivo nuevo es exactamente igual a la cantidad obtenida a través del cálculo teórico. Con esto se comprueba que todos los datos de la prueba, así como sus etiquetas de tiempo, han sido almacenados satisfactoriamente dentro del archivo final.

Posteriormente se procedió a reconstruir los datos almacenados del archivo final. La figura 4.27 muestra las gráficas de uno de los archivos nuevos, que ya cuenta con la etiqueta de tiempo.

Pruebas realizadas al sistema “DIGI”

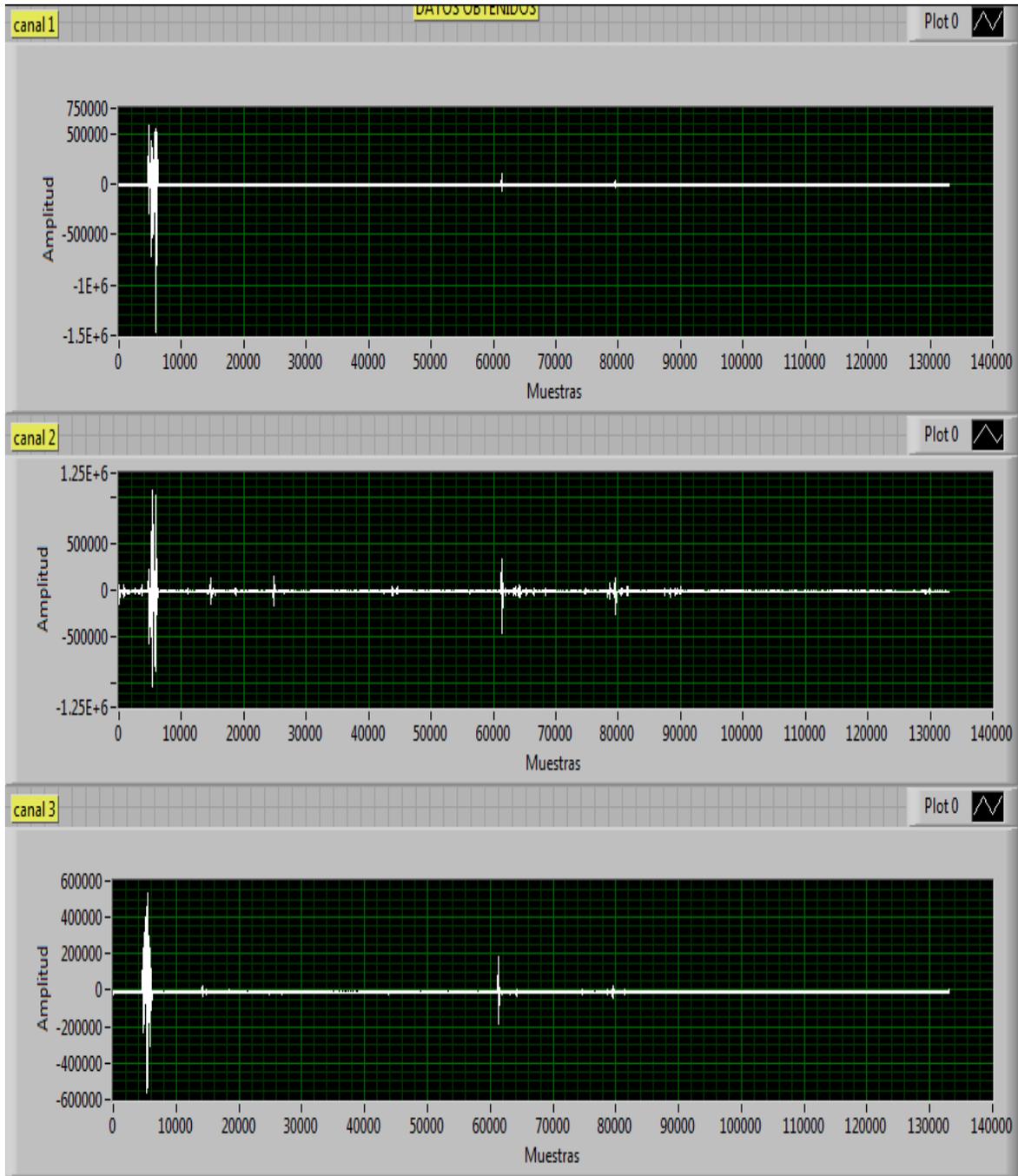


Figura 4.27. Gráficas a partir del archivo de datos de prueba de siete días.

En la figura anterior se observan las gráficas de los tres canales del sistema “DIGI”, en donde en cada canal se conectó un geófono.

Finalmente se obtienen los datos del paquete de tiempo (fecha y hora UTC), que se insertaron en el archivo de datos final, figura 4.28, esto con la finalidad de verificar que corresponden con la fecha de adquisición, y además, que sean continuos.

Encabezados de Tiempo en decimal: _							
	Año	Mes	Dia	Segundo	Minuto	Hora	
0	81	13	9	6	45	10	20
0	81	13	9	6	46	10	20
	81	13	9	6	47	10	20
	81	13	9	6	48	10	20
	81	13	9	6	49	10	20
	81	13	9	6	50	10	20
	81	13	9	6	51	10	20
	81	13	9	6	52	10	20
	81	13	9	6	53	10	20
	81	13	9	6	54	10	20
	81	13	9	6	55	10	20

Figura 4.28. Encabezados de tiempo de archivo de siete días.

De la figura 4.28 podemos observar que la primera etiqueta de tiempo, corresponde con la base de tiempo obtenida del receptor GPS (figura 4.23), sin embargo, no coincide con las propiedades de creación del archivo generado por el sistema DIGI (figura 4.22). Esto se debe a las mismas condiciones que se presentaron entre las propiedades del archivo y las etiquetas de tiempo, en las pruebas de 1 día.

Con el desarrollo de las pruebas mostradas, se observa que el sistema DIGI cumple con las especificaciones del usuario, es decir, adquiere satisfactoriamente la señal de entrada en los canales analógicos, también almacena toda la información sin perder de alguna muestra, y al reconstruir la datos digitalizados, podemos observar que las gráficas corresponden con la señal de entrada al sistema DIGI. Con lo anterior, podemos decir que el sistema DIGI funciona adecuadamente.

Capítulo V

Resultados y conclusiones

En este capítulo se presentarán los resultados finales a los que se llegaron con el desarrollo y diseño del sistema DIGI. Así como también se presentará la conclusión de este trabajo.

5.1. Resultados

El objetivo principal de este trabajo fue el diseño e implementación de un sistema de bajo costo, que permitiera la adquisición y el almacenamiento de señales provenientes de sensores sísmicos. Para lograr este objetivo se procedió a diseñar, por separado, cada uno de los bloques que integran al sistema DIGI y posteriormente se integraron cada uno de estos bloques en una tarjeta de prototipos *protoboard*, con la finalidad de realizar pruebas y mejoras al sistema.

Una vez que el sistema funcionó adecuadamente en la *protoboard*, se realizó el diseño y fabricación de la PCB que contendría todo el sistema, para que después de ensamblada dicha tarjeta, se realizaran diferentes pruebas que garantizaran el funcionamiento del sistema DIGI en la PCB.

Los resultados más importantes que se obtuvieron con el desarrollo del sistema DIGI son los siguientes:

- El sistema desarrollado cuenta con una interfaz usuario – máquina, la cual permite navegar, de una manera muy intuitiva, por las distintas funciones con las que cuenta el sistema.
- Las principales funciones con las que cuenta el sistema DIGI son las siguientes:
 - Configuración
 - Modo de prueba
 - Registro
- Por medio de la función “*Configuración*”, el usuario seleccionará la frecuencia de muestreo (50,100 o 200 mps).

- Con la función “*Modo de prueba*”, el usuario podrá verificar que cada uno de los canales se encuentren en operación, ya que esta opción permite graficar en el *display* la señal digitalizada. Así mismo, en esta misma función se muestran los datos de fecha y hora adquiridos mediante el receptor GPS.
- En la función “*Registro*”, el usuario configurará el tiempo en el que el sistema DIGI adquirirá y almacenará datos provenientes de los sensores sísmicos.
- El sistema DIGI es capaz de almacenar información hasta por 30 días ininterrumpidos, con una memoria de 8GB a una frecuencia de muestreo de 200 mps.

5.2. Conclusiones

Con base en lo descrito en el apartado de resultados, se puede concluir que a lo largo de todo el tiempo invertido en el diseño e implementación del sistema DIGI, se cumplió cabalmente con el objetivo principal que motivó la realización de este proyecto. El sistema DIGI permite al usuario realizar una configuración del equipo muy intuitiva, por medio de una interfaz usuario – máquina, implementada con un *joystick* y un *display* gráfico. De esta manera no será necesario que el usuario lleve una *laptop* cuando se pretendan realizar pruebas en campo.

Es importante mencionar que, con el desarrollo de este proyecto, se obtuvieron resultados que no fueron considerados al inicio del mismo. Ya que el sistema DIGI cuenta con su propio *software* para el procesamiento de los datos almacenados en la memoria micro SD, en donde se pueden observar cada una de las muestras adquiridas etiquetadas con su respectiva hora y fecha, así como obtener una gráfica del comportamiento del sistema.

Finalmente y de manera personal, el desarrollo de este sistema nos dio la oportunidad de aplicar nuestros conocimientos adquiridos a lo largo de la carrera, para diseñar y desarrollar un sistema embebido de bajo costo y alta confiabilidad, dando solución a un problema de ingeniería, enfocado en el área sísmica, que compete a la ingeniería geológica.

5.3. Recomendaciones

Durante el desarrollo del sistema DIGI se hicieron recomendaciones, sugerencias y observaciones que permitieron una versión final del sistema DIGI. Sin embargo, como en

todo prototipo electrónico, el sistema DIGI no está exento de adecuaciones que se puedan realizar para obtener un mejor funcionamiento. A continuación se presentan algunas de las recomendaciones que pueden mejorar esta primera versión del sistema.

Con lo que respecta al *Hardware*, algunas propuestas son:

- El sistema al estar alojado en un gabinete dificulta la operación del sistema DIGI con el usuario, ya que el usuario tendría que abrir el gabinete y manipular el *joystick* para su configuración. Una solución a este problema sería colocar el *joystick* en el gabinete y por medio de unos cables conectarlo a la PCB, de esta manera el usuario no tendría que abrir el gabinete en el que se aloja al DIGI.
- En el acondicionamiento de la señal, el usuario tiene que seleccionar manualmente el tipo de filtro a utilizar, dependiendo del sensor que se desee conectar. Para evitar esto, el filtro se podría seleccionar de manera digital.
- Podría utilizarse otro tipo de *display*, para tener una mejor resolución de los datos graficados provenientes de los sensores sísmicos.

En cuanto al *software*:

- Convendría implementar una rutina para detectar si la memoria micro SD se encuentra dañada.
- Hacer más eficiente el código, ya que existen algunas rutinas que se tuvieron que repetir en el programa, trayendo como consecuencia que sea bastante pesado.
- Desarrollar una rutina que permita censar la tensión del sistema a través del microcontrolador, con la intención de que si hay una variación crítica en la tensión del sistema, se cierre el archivo que se encuentra en escritura al momento de realizar una prueba, con la finalidad de no perder datos.
- Desarrollar una rutina que permita que el usuario pueda parar la prueba y que los datos sean almacenados. Ya que el sistema está diseñado para funcionar por un periodo de tiempo que se programa en la configuración del sistema, y hasta que concluya ese tiempo el usuario puede nuevamente interactuar con el sistema.
- Se podría implementar una rutina para que el sistema registre información en un modo de *trigger*.

- Desarrollar una rutina que utilicé el *watchdog* del microcontrolador, ya que si por alguna razón se presentara un error en la ejecución del programa, el *watchdog* reiniciará la ejecución del programa y el sistema continuará con la prueba que el usuario ha programado.

Apéndice A: Glosario de términos

Aliasing. Es el fenómeno que se presenta al digitalizar una señal analógica, siempre y cuando la frecuencia de muestreo sea menor a la frecuencia de la señal analógica.

ENOB. *Effective Number of Bits.* Es una forma de cuantificar la calidad de un convertidor analógico – digital. Por lo que un número elevado de este parámetro significa que los datos digitalizados son más precisos.

Firmware. Es un bloque de instrucciones que contiene tareas programadas para el funcionamiento de un sistema electrónico.

Geófono. Transductor de desplazamiento que convierte el movimiento del suelo a una señal eléctrica.

GLCD. *Graphic Liquid Crystal Display.* Es una pantalla plana formada por una matriz de pixeles monocromos colocados de una fuente de luz reflectora.

GPS. *Global Positioning System.* Sistema utilizado en la navegación del cual se pueden obtener datos como fecha, hora, posición entre otros.

IDE. *Integrated Development Environment.* Es una aplicación de software que proporciona servicios integrales para los programadores para el desarrollo de algún software.

Joystick. Dispositivo de control utilizado para la navegación en un sistema electrónico.

NMEA. Es un protocolo de comunicación que permite comunicar dispositivos de navegación tales como el GPS.

OCR. *Operation Condition Register.* Registro de la memoria micro SD, que nos permite conocer las condiciones de operación de la misma.

SD. *Secure Digital.* Son un tipo de memoria flash.

SDHC. *SD High Capacity.* Nomenclatura que hace referencia a las memorias SD de alta capacidad.

SDSC. *SD Standard Capacity.* Nomenclatura que hace referencia a las memorias SD de capacidad estándar.

SDXC. *SD Xtended Capacity.* Nomenclatura que hace referencia a las memorias SD de capacidad extendida.

SNR. *Signal to Noise Ratio.* Es una medida que compara la potencia de la señal con la potencia de ruido, comúnmente se expresa en dB.

SPI. *Serial Peripheral Interface.* Es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos.

UHS. *Ultra High Speed.* Hace referencia a la gran velocidad de transmisión de información en memorias.

USART. *Universal Synchronous Asynchronous Receiver Transmitter.* Módulo que proporciona al microcontrolador la capacidad de comunicarse por interfaces seriales RS-232 y RS-485.

UTC. *Universal Time Coordinated.* Es un estándar de la referencia de tiempo internacional mediante la cual se calculan todos los usos horarios.

Bibliografía

Libros

- Boylestand R., Nashelsky L., *Electrónica: Teoría de circuitos y dispositivos electrónicos*, octava edición, Pearson Prentice Hall, México, 2003.
- Ceballos J., *Lenguaje de programación en C.*, tercera edición., Alfaomega., España., 2002.
- Dorf R., Svoboda J., *Circuitos eléctricos: Introducción al análisis y diseño*, tercera edición, Alfaomega, México, 2003.
- H.M. Deitel, P.J. Deitel, *Como programar en C/C++*, segunda edición, Pearson Prentice Hall, México, 1995.
- Huerta E., Mangiaterra A., *GPS: posicionamiento satelital*, segunda edición, UNR editora, Argentina, 2005.
- Lázaro A., Del Rio J, *LabVIEW 7.1 Programación Gráfica para el control de instrumentación*, Thomson, 2005
- Manolakis D., Proakis J., *Digital Signal Processing: Principles, algorithms, and applications*, tercera edición, Pearson Prentice Hall, USA, 2005.
- Pallás R., *Sensores y acondicionadores de señal*, cuarta edición, Marcombo, España, 2003.

Artículos y Tesis

- Alguacil G., Havskov J., *Instrumentation in earthquake seismology*, Universidad de Granada, España, 2002.
- Flores O., *Batalla de microcontroladores ¿AVR o PIC?*, Universidad Tecnológica de El Salvador, San Salvador, 2009.
- Guinchard M., Janssens S., *Review of sensors for low frequency seismic vibration measurement*, CERN, Ginebra, 2011.

- Hernández H., *Diseño e implementación de una interfaz de registro de datos para la unidad SR04.*, Tesis de Licenciatura., UNAM, 2010.
- Reiss J., *Understanding sigma – delta modulation: The solved and unsolved issues*, AES member, 2008.
- Tapia D., *Adquisición y procesamiento de datos provenientes de la unidad sísmica SR04*, Tesis de Licenciatura, UNAM, 2009.

Manuales y hojas de especificaciones

- *Auto-zero amplifiers ease the design of high-precision circuits*. TechnicalPaper, June 2005, Texas InstrumensIncorporated.
- *AVR microcontroller datasheet*
- *Delta Sigma ADC Bridge Measurement Techniques*. Application note 96, January 2005, Linear Technology.
- *NMEA reference manual*. Technical paper, Document revision 1.3 January 2005, SiRF Technology Inc.
- *Microsoft Extensible Firmware Initiative FAT 32 File System Specification FAT: General Overview of On-Disk Format*. Version 1.03, December 6, 2000 Microsoft Corporation
- *Principles of Sigma-Delta modulator for analog to digital converters*. Technical Paper, April 2008, Motorola.
- *SD Specifications, Part 1 Physical Layer Simplification*. Version 4.10, January 22, 2013, SD Group.
- *The effective number of bits (ENOB) of my R&S DigitalOscilloscope*. Technical Paper, April 2011, Rohde & Schwarz.

Páginas de internet

- <http://www.geovirtual.cl/>
- <http://www2.ssn.unam.mx:8080/website/jsp/Epicentro-magnitud>

- <http://mexico.cnn.com/nacional/2012/11/15/mexico-pais-con-alta-sismicidad>
- <http://electronicdesign.com/analog/enob-isn-t-always-best-figure-merit>
- <http://synnick.blogspot.mx/2012/05/manejo-de-interrupciones.html>
- <http://arduino.cc/>
- <http://www.greenwichmeantime.com>
- <http://www.engblaze.com/microcontroller-tutorial-avr-and-arduino-timer-interrupts/>
- <http://www.cursomicros.com/avr/programadores/interfaces-de-programacion.html#tpi>
- <http://www.atmel.com/images/doc2549.pdf>